\chapter{Evaluation}

\section {Customer Requirements}

These are the original objectives specified by the client at the analysis stage of making this system.

\textbf{General Objectives}

\begin{itemize}

\item To have an interactive and easily navigable graphical user interface, applying a suitable colour scheme and layout

\item To make the database concise and adjustable

\item To create various lessons, with a wide range of challenges, which effectively teach students how to do trigonometry and Pythagoras

\item To create tasks which are relevant to the lessons to be completed by the user in order to test their progress

\item To allow this progress to be recorded in an easily accessible and readable database

\item To incorporate algorithms which find and/or check the solution given by the user accurately and give clear and easy to read outputs to correspond with said inputs

\item To have some access restrictions to certain levels of user

\item To make the program accessible only from various computers with permissions

\end{itemize}

\textbf{Specific Objectives}

\begin{itemize}

\item To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\item To include the following topics: Trigonometry, Pythagoras, 3D Trigonometry, 3D Pythagoras

\item To include a range of difficulty levels, which can challenge every user's level of ability

\item Use drag and drop, text boxes and drop down menus for inputs

\item To include interactive 2D graphics which give a clearer idea of the method being shown to the user

\item To have a database which can be accessed by different computers online

\item Use a specific, continuous and attractive colour scheme in every window

\item To have medium sized, highly visible icons

\item To have all input buttons randomised to avoid double clicking and guessing from memory

\item To have small error message windows which pop up and disappear on a timer

\item To include images and shapes which contrast the colour scheme so they are visible and readable

\end{itemize}

\textbf{Core Objectives}

\begin{itemize}

\item To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\item To make the database easy to access and easy to read

\item To include primarily trigonometry based topics, such as how to use the sine, cosine and tan rules

\item To include an initial, moderate difficulty in order to cater for a majority of students

\item To make the database functional and able to store the requested details

\end{itemize}

\textbf{Other Objectives}

\begin{itemize}

\item To position buttons, text boxes and drag and drop boxes in within the layout of the graphical user interface in such a way that cheating and lucky guessing can be minimised

\item To make the database adjustable if necessary

\item Use a more interesting range of input types like drawing boxes rather than just clicking and typing

\item To include a wider range of difficulties to challenge every student on the right level for them

\item To include a wider range of topics such as pythagoras, then 3D trigonometry and 3D pythagoras

\end{itemize}

\subsection{General Objective 1: }

To have an interactive and easily navigable graphical user interface, applying a suitable colour scheme and layout

\subsubsection{Objective Met?}

This objective has been met. My windows all use a consistent layout which includes large buttons, for easy clicking, and a highly visible colour scheme. Labels make it clear how to navigate to certain parts of the system, and a colour code is used to imply to the user what the purpose of each button is. For example, blue to proceed, red to go back, green to submit/complete something, and yellow to mark an input. Each window has a large title so the user always knows where they are.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_1}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\caption{This window uses the generic layout and colour scheme....}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_2}

\caption{...as does this one, and all of the others, making it consistent throughout the system}

\end{figure}

\subsection{General Objective 2: }

To make the database concise and adjustable

\subsubsection{Objective Met?}

This objective has been met. The database is small, as it only stores five different entities, which technically is a short-coming but for a separate objective. The database is easily visible and the table widget which displays it is designe to fit itself in any sized screen, and scroll bars are available if necessary. The number of rows is hard-coded as there is a maximum number of records that can be recorded in version 1 of the system, and before these records are saved there are just blank spaces in a preset table.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_2}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The database only has five entities, and is displayed in a single table widget}

\end{figure}

\subsection{General Objective 3: }

To create various lessons, with a wide range of challenges, which effectively teach students how to do trigonometry and Pythagoras

\subsubsection{Objective Met?}

This objective has been met. There are twelve lessons which all cover trigonometry, Pythagoras, or an aspect of either one, as well as some vector lessons.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_3}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\caption{The menu which leads to any of the twelve lessons, all relevant to trigonometry in some way}

\end{figure}

\subsection{General Objective 4: }

To create tasks which are relevant to the lessons to be completed by the user in order to test their progress

\subsubsection{Objective Met?}

This objective has been met. Each lesson has three homework tasks which are based directly on the content of said lesson, and upon completing each task the user will have a better idea of how good their knowledge of the topic is.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_4}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_4}

\caption{Example of a menu showing all of the homework relevant to some lessons}

\end{figure}

\subsection{General Objective 5: }

To allow this progress to be recorded in an easily accessible and readable database

\subsubsection{Objective Met?}

This objective has been met. There is a table widget accessible from one click on the home screen which, when its window is opened, immediately displays up-to-date information from the database file. The size of the table widget has been altered to make it readable and fit well with most monitor sizes.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_5}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The database information is easy to access from the home screen, displayed in a readable table}

\end{figure}

\subsection{General Objective 6: }

To incorporate algorithms which find and/or check the solution given by the user accurately and give clear and easy to read outputs to correspond with said inputs

\subsubsection{Objective Met?}

This objective has been met. There are various check methods in the parent homework classes which are used to check the value of the user's input, which is passed into the method from the subclass, against a hard-coded answer. The user will be told if they are right or wrong, and error messages prevent them from missing questions.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_6}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_6}

\caption{a piece of code/check method which checks the users input and tells them if they are right or not}

\end{figure}

\subsection{General Objective 7: }

To have some access restrictions to certain levels of user

\subsubsection{Objective Met?}

This objective has not been met. This system ended up being a single user, offline program which can only be used by one person per machine, which it has to be installed on. There are no administrator only aspects to the system; every possible user will have the same access to the whole program.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_7}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_7}

\caption{The welcome screen, absent of any log in requests or identity}

\end{figure}

\subsection{General Objective 8: }

To make the program accessible only from various computers with permissions

\subsubsection{Objective Met?}

This objective has been met in one sense, but has not been met in another. To use this program, the installer has to be obtained from me directly, and only the computer which you install it on can use it. There are no restrictions on which computers can use it, as long as it runs on Windows. However, there is no logging in function in the system, so there is no way to know for sure who is using the system of the people who might have access to a computer it is installed on, so effectively this objective has not been met.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_8}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_8}

\caption{The application in the task bar - anyone on the computer can use it}

\end{figure}

\subsection{Specific Objective 1: }

To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\subsubsection{Objective Met?}

This objective has not been met. I have included tasks which require the application of skills rather than problem solving tasks, like the current and soon to be out dated curriculum.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_9}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_9}

\caption{An example of the subject material used, still based on application of skills rather than problem solving}

\end{figure}

\subsection{Specific Objective 2: }

To include the following topics: Trigonometry, Pythagoras, 3D Trigonometry, 3D Pythagoras

\subsubsection{Objective Met?}

This objective has been met. All of the topics listed in the objective have lessons and homework tasks in the system, and other sub topics have been included as well.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_10}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\caption{A menu which leads to lessons of all of the listed topics}

\end{figure}

\subsection{Specific Objective 3: }

To include a range of difficulty levels, which can challenge every user's level of ability

\subsubsection{Objective Met?}

This objective has been met. Each lesson has three homework tasks which accompany it. One easy one, one medium one and one hard one, so every level of ability can be tested appropriately.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_11}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{A menu leading to tasks of different difficulty levels}

\end{figure}

\subsection{Specific Objective 4: }

Use drag and drop, text boxes and drop down menus for inputs

\subsubsection{Objective Met?}

This objective has been partially met. Drag and drop functionality was not included, but replaced with multiple choice buttons. Text boxes and drop down menus were included.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_12}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_1}

\caption{An example of line edits in use for text inputs from a keyboard}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_2}

\caption{An example of the drop down combo boxes}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_3}

\caption{An example of the mulitple choice button question which replaced the drag and drop input type}

\end{figure}

\subsection{Specific Objective 5: }

To include interactive 2D graphics which give a clearer idea of the method being shown to the user

\subsubsection{Objective Met?}

This objective has been met. Many images have been created (or occasionally taken from Google Images) to provide the user with a better idea of how a method works. However they are not technically interactive, although the user uses some of them to be able to find a solution.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_13}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_13}

\caption{An example of the mathematical diagrams used to show a user how to solve a problem}

\end{figure}

\subsection{Specific Objective 6: }

To have a database which can be accessed by different computers online

\subsubsection{Objective Met?}

This objective has not been met. There is no online functionality at all in this system. Each database is created independently on each individual machine the system is installed on, and can only record the progress of the user who owns the machine.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_14}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The simple database with no online capabilities whatsoever, just a single user progression recorder}

\end{figure}

\subsection{Specific Objective 7: }

Use a specific, continuous and attractive colour scheme in every window

\subsubsection{Objective Met?}

This objective has been met. The colour scheme is consistent, attractive and has a code, making it specific, and therefore has high usability.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_15}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\caption{Again, the colour scheme is consistent in each window}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_2}

\caption{It is also kind on the eyes and the colour code is memorable once you figure out what it is}

\end{figure}

\subsection{Specific Objective 8: }

To have medium sized, highly visible icons

\subsubsection{Objective Met?}

This objective has been met. All of the widgets in the system are just the right size, not too big or small. They generally fit well on the screen, as long as the screen is at least 20 inches wide.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_16}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_16}

\caption{An example of the large widgets used to make it easier for all users to see them and use them quicker}

\end{figure}

\subsection{Specific Objective 9: }

To have all input buttons randomised to avoid double clicking and guessing from memory

\subsubsection{Objective Met?}

This objective has not been met. I could not find a suitable randomisation solution, so the location of each answer will be hard-coded. On the other hand, no two homework task windows which have a similar layout will ever appear together, so there is no chance of accidentally clicking a button on a second screen after a double click.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_17}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_17}

\caption{This multiple choice question has buttons with answers displayed on them}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_17}

\caption{Same question, same answers, same order}

\end{figure}

\subsection{Specific Objective 10: }

To have small error message windows which pop up and disappear on a timer

\subsubsection{Objective Met?}

This objective has been met. Error messages appear at appropriate times to help the user input a correct answer or remind them that they have missed a question.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_18}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_18}

\caption{One of the error messages preventing the user from missing a question}

\end{figure}

\subsection{Specific Objective 11: }

To include images and shapes which contrast the colour scheme so they are visible and readable

\subsubsection{Objective Met?}

this objective has been met. The background colour of every window in the system is white, so any colour contrasts it well. Generally the images are of shapes which all have borders, so any white filling will not be mistaken for background.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_19}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_19}

\caption{An example of some of the many images included, all of which have good or reasonable resolution}

\end{figure}

\subsection{Core Objective 1: }

To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\subsubsection{Objective Met?}

This objective has not been met. I have included in the system problems which require the application of skills rather than problem solving skills which the new curriculum includes. The subject material is still useful however, even for the nuew curriculum to an extent.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_20}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_20}

\caption{An example of problem solving methods}

\end{figure}

\subsection{Core Objective 2: }

To make the database easy to access and easy to read

\subsubsection{Objective Met?}

This objective has been met. All information in the database is displayed in a table widget which is very easy to access internally, and the text is large and fits well on the screen.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_21}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The information from the database being displayed in an easy to read format}

\end{figure}

\subsection{Core Objective 3: }

To include primarily trigonometry based topics, such as how to use the sine, cosine and tan rules

\subsubsection{Objective Met?}

This objective has been met. Subject material covering the majority of GCSE trigonometry has been included in the lessons in the system, as well as the homework which goes with them.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_22}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_22}

\caption{An example of the subject material covered in the system's lessons}

\end{figure}

\subsection{Core Objective 4: }

To include an initial, moderate difficulty in order to cater for a majority of students

\subsubsection{Objective Met?}

This objective has been met. Each lesson has three homework tasks based on the subject material in said lesson. One is easy, one medium and one hard, so every level of ability can be tested.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_23}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{A menu showing tasks of varying difficulties}

\end{figure}

\subsection{Core Objective 5: }

To make the database functional and able to store the requested details

\subsubsection{Objective Met?}

This objective has partially been met. The database is functional and stores enough data for a user to gain some idea of how good they are at the maths topics included in the system. However, not all of the entities originally requested by the client are stored in the database.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_24}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{Only five of the entities originally requested by the user}

\end{figure}

\subsection{Other Objective 1: }

To position buttons, text boxes and drag and drop boxes in within the layout of the graphical user interface in such a way that cheating and lucky guessing can be minimised

\subsubsection{Objective Met?}

This objective has been met. At no point do two similar homework screens appear one after the other, so it is highly unlikely that a double click would accidentally cause a problem by clicking a button on the second screen.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_25}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_1}

\caption{The generic first screen of a homework task...}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_2}

\caption{...always followed by a generic second screen with a completely different layout}

\end{figure}

\subsection{Other Objective 2: }

To make the database adjustable if necessary

\subsubsection{Objective Met?}

This objective has not been met. The user cannot change the database manually at all.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_26}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_26}

\caption{There are no options to amend the database}

\end{figure}

\subsection{Other Objective 3: }

Use a more interesting range of input types like drawing boxes rather than just clicking and typing

\subsubsection{Objective Met?}

This objective has not been met. I could not get a drag and drop functionality to work, so all inputs involve either typing or single clicks.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_27}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_1}

\caption{An example of line edits in use for text inputs from a keyboard}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_2}

\caption{An example of the drop down combo boxes, only using a single click}

\end{figure}

\subsection{Other Objective 4: }

To include a wider range of difficulties to challenge every student on the right level for them

\subsubsection{Objective Met?}

This objective has been met. Each student can choose to begin on an easy or medium task, then if they are comfortable they can try a hard task

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_28}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{A menu showing the range of difficulties of homework tasks}

\end{figure}

\subsection{Other Objective 5: }

To include a wider range of topics such as pythagoras, then 3D trigonometry and 3D pythagoras

\subsubsection{Objective Met?}

This objective has been met. All of the topics listed in the objective as well as another topic, vectors, have been included in the system.

\subsubsection{Evidence: }

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/objective\_29}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_29}

\caption{A menu showing the range of topics}

\end{figure}

\section{Effectiveness}

\subsection{General Objective 1: }

To have an interactive and easily navigable graphical user interface, applying a suitable colour scheme and layout

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the interface readable?

\item How quickly can the user navigate?

\item Is the colour scheme suitable?

\item Is the layout consistent?

\item Is the interface robust?

\end{itemize}

The graphical user interface I have used in my system is consistent with every window in that it uses the same colour code for widgets, the background, and text. The sizing is the same for pretty much all widgets and images. The widgets are big enough but not too big, so the user can quickly find a button or input box they are looking for, and clearly read the labels which tell the user what the purpose of each widget is. It is quite obvious throughout the system what each widget does due to the large text labelling. The widgets all fit on the screen, and the layout is logical in that the buttons are all level and evenly spaced. Each window is the system can be accessed quickly because there aren't too many windows to go through to find them. The interface is mostly robust; occasionally you have to click twice for a button to work after just having opened a new window, as it has not registered it quickly enough. This occurs for two seconds maximum. Otherwise all of the navigtion inputs work fine. There are no bugs or errors, except for those which have error messages if a user misses a question on a homework. Overall, the graphical user interface is effective.

\subsubsection{Evidence}

These images all show different windows using the same colour scheme, widget size and layout.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_2}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_4}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_1}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_2}

\end{figure}

\subsection{General Objective 2: }

To make the database concise and adjustable

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the database concise?

\item Is the database adjustable?

\item Is the database easy to use?

\end{itemize}

The table widget where the information from the database can be viewed is concise as it only has five entities in it, and the table fits on the screen well. However, this is technically a short coming as the database does not store all of the entities the client originally requested. Furthermore, the database is not at all adjustable. The user has to delete the entire database file manually from the system files to be able to make a fresh database. They cannot reset any individual task scores. Therefore this objective is not effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The database is concise, but that wasn't a requirement. If it has more entities it would be effective}

\end{figure}

\subsection{General Objective 3: }

To create various lessons, with a wide range of challenges, which effectively teach students how to do trigonometry and Pythagoras

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Do the lessons include the trigonometry and Pythagoras topics?

\item Do the lessons contain variation and challenges?

\item Is the lesson content accurate and effective at teaching students the topic?

\end{itemize}

There are twelve lessons, most of which cover a trigonometry or Pythagoras related topic. These lessons are mostly varied (sometimes two separate lessons have content which can be linked), although there aren't many challenges because the lessons are designed to help the user become able to overcome a challenge which is more likely to be in a homework task. The content in the lessons is often detailed and is formatted in a way which makes it easier for the user to read and understand, e.g. spaces between steps in methods. Therefore this objective is moderately effective; this kind of thing can sometimes depend on the user's ability and motivation, although it is possible that some lessons are harder to understand than others. Therefore this objective is mostly effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\caption{This menu shows the range of topics}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_20}

\caption{This is an example of the content used in the lessons}

\end{figure}

\subsection{General Objective 4: }

To create tasks which are relevant to the lessons to be completed by the user in order to test their progress

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the homework relevant to the lessons?

\item Is the homework useful in helping the user understand the topics better?

\item Do the homework results give the user an accurate idea of how good they are at the topic?

\item Is it easy for the user to record and view their progress using the automatic internal database saving functionality?

\end{itemize}

Each lesson has three homework tasks directly related to the lesson content, designed to test the user's new knowledge of the topic. Each is of a different difficulty level so a range of abilities can be tested. The homework results are measured in marks, which generally tend to be single or double. This information is recorded effectively in a database, so the user can see their scores at any time once they complete a task, however due to the lack of the originally requested entities there is not really much to work with to accurately measure your own ability. Furthermore, despite attempts to reduce the opportunities to cheat, it is inevitably possible for the user to guess answers. Therefore this objective is mostly effective, but it does have its problems.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{This menu shows an example of the homework directly linked to the lessons, with varying difficulties}

\end{figure}

\subsection{General Objective 5: }

To allow this progress to be recorded in an easily accessible and readable database

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the progress recorded right?

\item Is the database easily accessible?

\item Is the database readable?

\end{itemize}

The progress is always recorded as soon as the user finishes the task, and the answer checking algorithms are effective, so the user can always immediately view the new data in a table by simply clicking a single button once they have finished a task which will open the window with the database table. The text is large and the table itself is designed to fit in most monitor sizes, and there are scroll bars for if the table doesn't fit anyway, so all the data will always be visible. Therefore the database information is always easy to access and easy to read, so this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/effective\_5\_1}

\caption{All of these questions were correct...}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/effective\_5\_2}

\caption{...the scores are immediately viewable in the database table}

\end{figure}

\subsection{General Objective 6: }

To incorporate algorithms which find and/or check the solution given by the user accurately and give clear and easy to read outputs to correspond with said inputs

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Do the algorithms check the user's input accurately?

\item Do the algorithms always give the appropriate output?

\item Are the algorithms robust?

\item Do the algorithms work with each subclass?

\end{itemize}

The algorithms always take the text of the input box they have used, so they will always check exactly what the user has input, and the hard-coded answers take into account the special characters like measurements or the {$^0$} symbol. Therefore the algorithms will always be able to tell the user whether or not they are correct, and using the appropriate error messages. The algorithms have been tested with each subclass, as they are all written in a parent class, so it wil always work, and each task will have data saved in the database once completed. Due to error messages the user will never be able to skip a question, so the values saved in the database will always be accurate. Therefore this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/effective\_5\_1}

\caption{All of these questions were correct and the user was informed they were correct}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/effective\_6\_2}

\caption{This question was wrong and an error message popped up to tell the user. The arrow shows were the attempts remaining has decremented}

\end{figure}

\subsection{General Objective 7: }

To have some access restrictions to certain levels of user

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are there access restrictions in place?

\item Do the access restrictions work effectively in preventing some users accessing areas of the system?

\end{itemize}

This objective was never implemented, so it is completely ineffective.

\subsubsection{Evidence}

There is no evidence - this objective was never implemented.

\subsection{General Objective 8: }

To make the program accessible only from various computers with permissions

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are there permission requirements in place?

\item Is the system only accessible from certain computers?

\end{itemize}

This objective was never implemented, so it is completely ineffective.

\subsubsection{Evidence}

There is no evidence - this objective was never implemented.

\subsection{Specific Objective 1: }

To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Does the system include material from the up-to-date curriculum?

\item Does the system include material that is relevant to GCSE maths?

\end{itemize}

The lessons in the system use material which focuses on the application of skills, like the older GCSE maths curriculum, rather than the newer problem solving version of the curriculum, so some of the material in the system may soon go out of date. However, there isn't too much difference between the new and the old versions; this system will still teach users the methods and concepts needed at GCSE level trigonometry, some of which will still be applicable to the new curriculum. Therefore this objective is partially effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_9}

\caption{An example of the subject material used in the lessons - corresponds with the current and soon to be replaced curriculum}

\end{figure}

\subsection{Specific Objective 2: }

To include the following topics: Trigonometry, Pythagoras, 3D Trigonometry, 3D Pythagoras

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the Trigonometry topic included in the system lessons?

\item Is the Pythagoras topic included in the system lessons?

\item Is the 3D Trigonometry topic included in the system lessons?

\item Is the 3D Pythagoras topic included in the system lessons?

\item Do the topics included in the system collectively cover the general trigonometry topic thoroughly enough?

\end{itemize}

All of the listed topics have been included in the system lessons. Some are touched on in more than one lesson, and there are other topics covered as well, such as vectors. Collectively, these topics cover everything necessary at GCSE level trigonometry, Pythagoras, and vectors, which is a whole unit in the curriculum. Therefore this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_29}

\caption{This menu shows the range of topics included in the system lessons}

\end{figure}

\subsection{Specific Objective 3: }

To include a range of difficulty levels, which can challenge every user's level of ability

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is there a range of difficulty levels?

\item Do these difficulty levels have a wide enough difference to distinguish levels of user skill and allow users of all ability to challenge themselves?

\end{itemize}

Every lesson has three corresponding homework tasks based on the content of the lesson; one is easy, one is medium, and one is hard, so each level of ability can choose to be comfortable or challenge themselves. The difference between difficulties is quite large, so that sometimes a user might not be able to answer a hard question, but will be able to answer a medium question with little difficulty. Therefore this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{This menu shows the range of difficulty levels of the trigonometry based homework tasks}

\end{figure}

\subsection{Specific Objective 4: }

Use drag and drop, text boxes and drop down menus for inputs

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Has drag and drop functionality been included in the system?

\item Have text boxes been included in the system?

\item Have drop down menus been included in the system?

\item Is there a wide range of input types to make the system more varied?

\item Can inputs be used quickly and efficiently?

\end{itemize}

Text boxes and drop down menus have been included in the system, but drag and drop functionality has not. Instead multiple choice buttons were included. As a result, the only two hardware inputs are typing and single clicks, so the range of input types is small and potentially boring. None the less, these inputs are always quick and effective; there is no lag or delay, and the buttons to mark the answer are quick too. Overall, due to the absence of drag and drop functionality, this objective is only partially effective, but it is sufficient for the user to use the system easily.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/effective\_5\_1}

\caption{An example of the drop down inputs}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_3}

\caption{An example of the multiple choice buttons which replaced the drag and drop functionality}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_1}

\caption{An example of the text box inputs}

\end{figure}

\subsection{Specific Objective 5: }

To include interactive 2D graphics which give a clearer idea of the method being shown to the user

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are 2D graphics included in the system?

\item How high is the resolution of these graphics?

\item Are the graphics used relevant to the screen they are on?

\item Are the graphics interactive?

\end{itemize}

2D images from both Google Images and manually drawn in Paint have been included in the system. The resolution of most of the images is high as they have been screen-shotted using Snipping tool. All of the images are entirely relevant to the screen they are on, for example a picture of a vector problem next to the vector button on a menu, or a picture showing how a method is done using the same measurements as the text in the example. Unfortunately none of the pictures are interactive, although they are useful for the user off screen to use for sketches on paper, if that is how they learn new methods. Therefore this objective is partially effective as the images are there, and relevant, but not interactive.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_20}

\caption{An example of a picture of a method using the same measurements as the text explanation}

\end{figure}

\subsection{Specific Objective 6: }

To have a database which can be accessed by different computers online

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is there a functional database?

\item Does the system support online functionality for the database?

\end{itemize}

There is a functional database, however no online functionality was implemented, so this objective is completely ineffective.

\subsubsection{Evidence}

There is no evidence for this objective - it was never implemented.

\subsection{Specific Objective 7: }

Use a specific, continuous and attractive colour scheme in every window

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is there a colour scheme specifically designed for this system?

\item Is the colour scheme used throughout the system continuous and appropriate?

\item Is the colour scheme used throughout the system attractive?

\end{itemize}

There is a colour scheme designed specifically for this system; blue means to continue, red means to return, yellow means to check, purple means to choose, and green means to submit or complete. The background of each window is white, so any colour contrasts it well. The colour scheme is used in every window, and it is attractive and appropriate; pleasant solid block colours used for a system with younger users. This objective is effective.

\subsubsection{Evidence}

These pictures show a few of the windows, all using the same colour scheme:

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_1}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_1\_2}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_4}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_2}

\end{figure}

\subsection{Specific Objective 8: }

To have medium sized, highly visible icons

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are the icons of an appropriate size?

\item Do the icons fit in a screen of any size?

\item are the icons visible?

\end{itemize}

The icons/widgets used throughout this system are all sized the same. They are a nice size to be clearly visible and look nice without being too big and excessive. They all fit on a screen of twenty inches wide or bigger, which is the average size of a school standard monitor. The icons are very visible; the colour scheme makes it clear what the purpose of each widget is, and they are all labelled to some degree, so the user can see what they do exactly. Therefore this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_9}

\caption{An example of the icons/widgets used, all labelled appropriately}

\end{figure}

\subsection{Specific Objective 9: }

To have all input buttons randomised to avoid double clicking and guessing from memory

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are the multiple choice buttons in a randomised order?

\item Do the buttons appear in a different order to prevent position memorisation?

\end{itemize}

A randomisation method could not be implemented into the classes, so the buttons will appear in the same place every time. Therefore this objective is not effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_3}

\caption{The buttons when a window is opened}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_12\_3}

\caption{The buttons when the same window is opened again - in the same place}

\end{figure}

\subsection{Specific Objective 10: }

To have small error message windows which pop up and disappear on a timer

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are there error messages in place to assist the user?

\item Do the error messages appear at the appropriate time?

\item Do the error messages disappear on a timer?

\end{itemize}

There are many error messages in place to assist the user. They tell the user if they are wrong, or have missed a question. They always appear at the appropriate time because they each have a separate class which is called in the appropriate method. They are not on a timer to disappear, however, which goes against the objective, although could be more useful should the user need more time to read them. Therefore this objective is generally effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_18}

\caption{An error message appearing to prevent the user from missing a question, dismissable at the user's own pace}

\end{figure}

\subsection{Specific Objective 11: }

To include images and shapes which contrast the colour scheme so they are visible and readable

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Have images and/or shapes been included?

\item Do these images and/or shapes contrast the colour scheme?

\item Are these images and/or shapes generally visible?

\end{itemize}

Shapes in the form of images have been included which are relevant to another widget or help to explain a mathematical method. The background is always white so as long as the shapes and images have borders of colour, which they all do, they are visible. Therefore this objective is effective as all images can be seen and are relevant to a task or widget.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_13}

\caption{Example of a shape image against the background, clearly visible}

\end{figure}

\subsection{Core Objective 1: }

To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Does the system include material from the up-to-date curriculum?

\item Does the system include material that is relevant to GCSE maths?

\end{itemize}

The lessons in the system use material which focuses on the application of skills, like the older GCSE maths curriculum, rather than the newer problem solving version of the curriculum, so some of the material in the system may soon go out of date. However, there isn't too much difference between the new and the old versions; this system will still teach users the methods and concepts needed at GCSE level trigonometry, some of which will still be applicable to the new curriculum. Therefore this objective is partially effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_9}

\caption{An example of the subject material used in the lessons - corresponds with the current and soon to be replaced curriculum}

\end{figure}

\subsection{Core Objective 2: }

To make the database easy to access and easy to read

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item How quickly can the database be accessed?

\item How easy to read is the information in the database?

\end{itemize}

The database can be accessed very quickly by just clicking on the progress button from the home screen. A window will immediately appear with a table displaying the contents of the database in large text, so it is readable too. The information displayed in the database is always refreshed whenever the screen is opened, so it will always be up to date. Therefore this objective has been met.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The database - large readable text and accessible from the home screen}

\end{figure}

\subsection{Core Objective 3: }

To include primarily trigonometry based topics, such as how to use the sine, cosine and tan rules

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Are trigonometry based topics the primary topic of the system?

\item Have the example sub topics been included?

\end{itemize}

There are two sections on trigonometry, and only one on Pythagoras and one on vectors, with a summary section on all of them together, so trigonometry is the main topic included in the system. Furthermore, Pythagoras and vectors also involve trigonometry to some degree. The topics of how to use the sine, cosine and tan rules have been included, as well as all of the other aspects of GCSE level trigonometry. Therefore, this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_29}

\caption{This menu shows the range of topics including primarily trigonometry}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_22}

\caption{This is part of the lesson which explains the sine, cosine and tan rules}

\end{figure}

\subsection{Core Objective 4: }

To include an initial, moderate difficulty in order to cater for a majority of students

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is there a default difficulty level?

\item Can all users find a task at their level of ability?

\end{itemize}

The lessons themselves are of a moderate difficulty; they are not designed to challenge the users, but help prepare them for the challenges in the homework tasks of easy to harder levels. Therefore this objective is effective, as all users should start by using the lessons.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_9}

\caption{The subject material here explains a method to the user and is formatted in an easy to read and understand way}

\end{figure}

\subsection{Core Objective 5: }

To make the database functional and able to store the requested details

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the database functional?

\item Does the database store the requested details?

\end{itemize}

The database is functional, but it does not store all of the requested details. Only enough for the system to serve a basic purpose of allowing the users to broadly keep track of their progress. Therefore this objective is not effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_2}

\caption{The database only stores five entities, enough for the user to have a rough idea of their ability level}

\end{figure}

\subsection{Other Objective 1: }

To position buttons, text boxes and drag and drop boxes in within the layout of the graphical user interface in such a way that cheating and lucky guessing can be minimised

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Have the widgets been positioned differently in each subclass?

\item Are the layouts of the two homework windows in a stack different?

\item Have the layouts been designed in a way which ultimately reduces or cancels lucking guessing?

\end{itemize}

Each subclass is the same, so it might be easier to remember the answers to some questions, or it might be harder for some people because they all look similar. The first homework screen layout is completely different to the second homework screen layout, so it is unlikely that a double click will get a user an accidental mark, an if they accidentally click finish then an error message will prevent them from cloding the stack window. Generally, it is unlikely that the user will accidentally click a button to an irreversible end, however, with all systems like this, it is inevitably possible for the user to make a lucky guess. Therefore this objective is only partially effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_1}

\caption{This is the generic first homework screen}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_25\_2}

\caption{The second homework screen uses a different layout}

\end{figure}

\subsection{Other Objective 2: }

To make the database adjustable if necessary

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is the database adjustable?

\end{itemize}

The database is not adjustable at all. The user just has to stick with their first progress record or delete it entirely manually. This objective is not effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_26}

\caption{There are no options to adjust the database}

\end{figure}

\subsection{Other Objective 3: }

Use a more interesting range of input types like drawing boxes rather than just clicking and typing

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Have alternative input types been used other than clicking and typing?

\item Do these input types provide a wider range of input overall?

\end{itemize}

Only clicking and typing inputs were used. Drag and drop functionality could not be implemented and drawing boxes were not used. This limits the range of input types severely, so this objective is not effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_16}

\caption{An example of a drop down menu, which only requires clicking from the mouse}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_17}

\caption{Again, the buttons only require clicking from the mouse, there is no drag and drop functionality for variation}

\end{figure}

\subsection{Other Objective 4: }

To include a wider range of difficulties to challenge every student on the right level for them

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Is there a range of difficulty levels?

\item Do these difficulty levels have a wide enough difference to distinguish levels of user skill and allow users of all ability to challenge themselves?

\end{itemize}

Every lesson has three corresponding homework tasks based on the content of the lesson; one is easy, one is medium, and one is hard, so each level of ability can choose to be comfortable or challenge themselves. The difference between difficulties is quite large, so that sometimes a user might not be able to answer a hard question, but will be able to answer a medium question with little difficulty. Therefore this objective is effective.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_11}

\caption{This menu shows the range of difficulty levels of the trigonometry based homework tasks}

\end{figure}

\subsection{Other Objective 5: }

To include a wider range of topics such as pythagoras, then 3D trigonometry and 3D pythagoras

\subsubsection{Effective?}

\textbf{Criteria: }

\begin{itemize}

\item Has the Pythagoras topic been included?

\item Has the 3D Trigonometry topic been included?

\item Has the 3D Pythagoras topic been included?

\item Have any other topics been included?

\end{itemize}

All of the topics listed, as well as vectors, have been included in the lessons in the system. Therefore this objective is effective, as it provides users with a wider range of subject material.

\subsubsection{Evidence}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/evidence\_29}

\caption{This menu shows the range of topics included in the system}

\end{figure}

\section{Learnability}

When I first consulted my client I took into account how much experience they have already had using software such as my system. They are comfortable using the internet, so with the assistance of the user manual they should have no problem installing Python 3.4 and PyQt4. They, along with the other people the client intends to use this system with, have all had experience using similar educational maths programs, such as MyMaths, which have essentially the same purpose as my system. Therefore I tried to make some aspects of the system somewhat similar to the ones conventionally used in educational maths programs, such as the layout, including the navigation of the system, the topics, and the rules of saving and popping error messages, such as making sure all questions have been attempted. None the less, this system would probably be easy enough for less experienced users anyway, as generally the buttons are clearly labelled, appropriately sized, and organised in a convenient way (branch menu). Although some users may have issues finding the exact right topic in the sub-menus, being a reason for the big red return buttons which make it easy to go back and try another menu.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/learnability\_1}

\caption{An example of the return buttons used to make navigating the sub-menus easier}

\end{figure}

When designing the system I kept in mind the fact that saving data can be a more complex function if it is done manually, especially by an inexperienced user, so I made all of the database functions automatic; they occur when the user simply clicks a button to finish a homework, or opens the progress screen. This cancels out the need for users to learn new skills which they might not have already learned from using similar systems.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/learnability\_2}

\caption{The finish button being clicked to automatically save progress for the user}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/learnability\_3}

\caption{The record which was just saved immediately being updated to the database}

\end{figure}

I also endeavoured to make the database itself very easy to access and understand internally in the system; the system accesses the information from a separate file and displays it in a window in the program, which can be accessed by the user very quickly from the home screen, and even queried for specific details should the user be searching for a specific record. the options in hte combo boxes are as clear as I can make them to make it easier for the user to determine how to use the query function when they try it for the first time.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/learnability\_4}

\caption{An example of the database being queried for easy access to specific records}

\end{figure}

Error messages were also incorporated to help the user understand why the sytem isn't working as they expected, should they fail to answer a question properly and try to proceed to the next page and be unsure why they cannot.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/learnability\_5}

\caption{An example of an error message telling the user why they cannot proceed}

\end{figure}

Generally, this system is very easy to use, even for people with little experience with such software, and care has been taken to ensure that the interface is very clear and the error messages are sufficient to help a user fix a progression related problem should they need the help. The only concern is that the user has no way to reset the database internally, so once they start using the system they have to either keep their progress or delete the database file manually.

\section{Usability}

I shall evaluate how easy to use each different aspect of the system is in order to gain an idea of whether or not the overall system has a good level of usability. The criteria I will use to measure the usability of each aspect include readability, convenience, time spent looking across the screen for things and common logic which the user needs to be able to understand.

The graphical user interface has large buttons with clear, blunt text which states the purpose of the button. The buttons also have a user friendly colour scheme which helps to make it clear what might happen when a button is clicked. For example, yellow means to mark an input, blue means to continue to a new screen, and red means to return to a previous screen. This assists the user in distinguishing the purposes of buttons which are sometimes positioned quite close to each other. The sub menus have six buttons, five which open a new menu and one which returns to a past screen, so it helps the user to immediately see which of the six buttons is going to take them back. The user expressed a high level of satisfaction with the user interface in general, suggesting that already it is a very usable interface. All of the input boxes have been sized to match the buttons, and the pictures have been sized to fit in the left over spaces, and to be relevant to the buttons they are placed next to, such as a trigonometry picture next to the trigonometry menu button. The readability of the graphical user interface is good, as all of the text has been enlarged to fill the screen as appropriate, so a user should have no trouble figuring out which buttons to click to find things in the system. The database screen and report widgets are accessible after three mouse clicks, and each lesson or homework after five, so it is convenient to get to any screen if you know where it is. The downside is perhaps having to search through each sub menu to find a topic. All of the widgets are pretty much equally sized, so target acquisition for the users eyes should be fast all round. Finally, each button is labelled appropriately, each picture is relevant and each title is clear, so there is common logic for the user to understand easily when navigating the system. Overall, the graphical user interface has a high usability level.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/usability\_1}

\caption{An example of the highly readable and easy to navigate menus}

\end{figure}

The storage of data has, in some ways, a high level of usability, but a low level in others. Firstly, it has a high level of usability because the user literally does not have to do any manual saving, loading, or database management, as the system saves and reads everything internally and automatically at certain points. All they need to do is complete the homeworks using easy inputs and maths skills, the learning of which is their responsibility, in order to record progress. They do not need to learn any new skills to be able to maintain the system's records, which can be considered convenient, should the user wish to stick to one 'attempt' at getting the best scores they can from the sytem. However, if the user wanted to delete their current progress and start again, they would have to manually remove the database file in order to reset their progress, which they might not know how to do. the lack of an internal 'drop table' function could be seen as inconvenient, despite it not being the purpose of the system or a client specified objective. Furthermore, the system does not save all of the data which the client wanted to be saved; only about half of the objective information is recorded throughout the system. Therefore the entire database itself has limited usability as the client will struggle to keep track of student's progress as originally intended. The information itself is all displayed in a table widget on the progress screen two clicks away from the welcome screen, so is quick to access and find as it is displayed in a huge table in the top right corner of the screen, one of the places where the user is likely to look first. The text itself is a nice size and is on a well-contraasted white background. In order to improve convenience, time spent and common logic I implemented a report widget where the user can quickly and easily query the database using large and easily usable combo boxes to select query criteria, the results of which will then appear in a similar table widget for the user to view immediately. Overall, the level of usability of the database is moderate, as information can be easily and quickly viewed, and recorded without any extra skills from the user, however once they start using the system they have to stick with the database they have unless they manually delete it.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/usability\_2}

\caption{Shows the clearly displayed information from the database}

\end{figure}

The subject material used in this system's lessons and homework sections is built up using large, clear text in a readable font against a nicely contrasting background color, accompanied by relevant pictures which can also show the user a mathematical method graphically, and large, clearly purposed buttons, line edits and combo boxes for a variation of input types. The user should have no trouble understanding what the questions are asking of them, and the lessons are supported by sufficient graphical images and text to give the user a clear example of a mathematical technique used to solve a problem. All of the images have either a good or a reasonable resolution. The buttons make it clear how to proceed or return to a previous screen. The colours used are all kind on the eyes. At no point should the user spend more than twenty seconds navigating from screen A to screen B (e.g. welcome screen to a homework screen) as the menus are all easy to use and understand. Therefore, generally the physical appearance of the system gives it quite a high usability.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/usability\_3}

\caption{The home screen showing the good colour scheme and the high resolution pictures}

\end{figure}

The error messages used in the system have a sole purpose of making it easier for hte user to use, therefore they naturally have a high level of usability. I have ensured that they are easy to understand and dismiss, and only appear when absolutely necessary to minimise disturbance for the user. The only foreseeable potential issue with the error messages is that they are quite small, so the user might have to squint to properly read the text, depending on the size of their monitor and the distance between their eyes and the screen. Otherwise, the text is simple and make it obvious what the problem is (all of the problems which might trigger errors in the system at all are simple ones), and they are dismissable by simply clicking the 'ok' button in the middle. Readibility is limited, convenience is high, as they only appear to help the user, time spent looking is low as they appear right in the middle of the screen making them impossible to miss, and common logic is high as they give clear instructions to the user. Therefore the only thing limiting the error messages' usability slightly is their small size, which is a result of using default QErrorMessage widgets. Otherwise error message usability is overall a high level.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/usability\_4}

\caption{An example of the small but clear error messages}

\end{figure}

Conclusively, the usability level of the overall system is reasonably high, as three of the four aspects of the system have been measured to have a high level of usability, and only one has a low-level of usability.

\section{Maintainability}

My system should be very easy to maintain. As of the completion of version 1 of the system, there are no bugs or errors which flag up in the IDLE, and there are no infinite loops or dead ends which the user wouldn't be able to escape without closing the entire application and restarting it. So unless more entire modules were added in a future version, there are currently no bugs to be maintained or fixed. I have used clear variable names throughout the modules which make it somewhat obvious what their purposes are, such as \textbf{self.layout} consistently for the PyQt4 layout of each window. Generic names have been used where required for each type of variable, such as \textbf{count} for stepper variables. Because I have so many sub classes within my system, I was able to use similar names for each sub class, such as \textbf{Trig1StackSidesEasy} and \textbf{Trig1StackSidesMedium} where classes inherit from the same parent class. For every parent class, all of its subclasses are created and altered in the same file; all of the first lesson page classes are in the same file, with almost exactly the same layout, and all of the second lesson pages are in a different file together, also with almost the same layout. This way, a programmer can determine what type of window the error is occurring on, look at the title of the window, and find the subclass easily in the file with the subclasses for that type of window. All of my code has been formatted in such a way that makes it easier to find bits of code. For example, the class begins at the top, followed by a constructor and the super(), then all of the PyQt variables are assigned, followed by algorithm variables, then PyQt window structuring, then connections, and lastly the methods in the order of the buttons they were connected to (this system is almost entirely event-driven). Furthermore, all of the database controller code is in one clearly named file, so if there are any problems involving the modification or accessing of the database the programmer will know where to look. Finally, every different line of code has been commented on at least once (some code is used consistently in many files, so an explanation will be in at least one of them). These comments give a clear explanation of the code's purpose and how it works or what it connects to. Therefore, a programmer will be able to find which section of code is responsible for causing a problem by reading the comments.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_1}

\caption{An example of subclasses which share a file import being created in the same file with similar names}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_2}

\caption{An example of subclasses which share a parent class being created in the same file with similar names}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_3\_1}

\caption{This shows the structure of the code in a class, consistent in all modules}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_3\_2}

\caption{This shows the structure of the code in a class, consistent in all modules}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_3\_3}

\caption{This shows the structure of the code in a class, consistent in all modules}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_4\_1}

\caption{This shows the database controller code all in the same class}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_4\_2}

\caption{This shows the database controller code all in the same class}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_4\_3}

\caption{This shows the database controller code all in the same class}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_5}

\caption{Examples of comments used to explain the purpose of code segments}

\end{figure}

In terms of system parameters, there are none which could or should be manually changed or ever changed, as usually the parameters are passed into the database controller for database changes to be made. If the structure of the entire database were changed in a future version, then new parameters might need to be placed in the methods in the database controller class and in the places where these methods are called. Otherwise, for the current version of the system, the database can only be modified if the appropriate parameters have been obtained, such as a task score, as error messages are in place to make it impossible for a user to skip a parameter. All of the possible parameter values are hard-coded, and that need not ever be changed as it would potentially involve writing pointlessly complex code for the same purpose as the current code.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_6}

\caption{The parameters accepted by the database controller methods}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_7}

\caption{Example of an error message used to prevent the user skipping a parameter}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_8}

\caption{Example of hard-coded parameters which must not be changed}

\end{figure}

On the other hand, one possible short coming relating to the system's maintainability is that, with the subclasses, they are all crammed into one file, so the programmer will have to scroll through the file to find the right subclass. Another issue is that it is not immediately clear where the file with the destination of a connection is, for example, in one file a method might be designed to open another window with a lesson, but if there was an error in that method, the programmer would have to know where the class which is being opened is in the files (i.e. the lesson page class) if the class name was wrong or something like that.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/maintainability\_9}

\caption{Examples of a method where it can be unclear where the class which is being assigned to the variable is located}

\end{figure}

Overall, my code has been organised in a way which is intended to maximise maintainability by making it easier to know where a piece of code is likely to be. The classes are organised under clear file names, and relevant variable, method and class names have been used to make it clear what the purpose of a class might be, or the function of a method. It might be difficult to locate specific subclasses, but there are comments in place to make it easier in some cases, such as which file contains a parent class which subclasses are inheriting from. Finally, the system parameters all work solidly and need never be changed. Therefore, this system has a reasonably high level of maintainability.

\section{Suggestions for Improvement}

\textbf{Include Administrator Capabilities}

By far the biggest short coming from the original client objective specification, the lack of administrator capabilities prevents the client from having an effective way to ensure that students are making the required progress, resorting them to trust alone. Responsible students will be more likely to achieve more by using this system despite not being monitored by the client, but less responsible students might miss out on the boost that this system could provide them.

\textbf{Include a local area network system which can connect multiple computers with multiple user accounts}

It would be useful to have different accounts so that students can access their progress from any computer with the system installed, or at least any computer in a LAN. This way they would be able to use the system in class.

\textbf{Include more entities in the database}

The current database is useful for showing the user which aspects of the maths in the system they need to improve on, but it would be far better to have names and ratings recorded so the client could monitor the students' progress more easily.

\textbf{Include an input type that isn't just clicking or typing, such as drag and drop functionality}

There is a range of input types, but they are not necessarily that far off from one another. The system includes typing in text boxes, selecting from drop down combo boxes (multiple choice) and clicking multiple choice buttons, which all involve clicking or typing. Drag and drop functionality would provide a much wider range of input types and make the system somewhat less boring to use after a while.

\section{End User Evidence}

The following images are of a feedback form which I provided to the client so they could convey to me the extent to which they were satisfied with the system.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_feedback\_1}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_feedback\_2}

\end{figure}

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_feedback\_3}

\end{figure}

\subsection{Questionnaires}

This was a brief questionnaire given to the client to give a broad idea of how satisfied they are with the system.

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_questionnaire}

\end{figure}

\subsection{Graphs}

This graph shows the balance of how satisfied the client is with the system:

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_graph}

\caption{According to this graph, the customer was satisfied with 68\% of the system.}

\end{figure}

This graph shows each of the client's 'yes' marks against the number of 'no' marks related to whether or not each objective was achieved (out of 32 objectives/sub-objectives, some of which were only partially achieved):

\begin{figure}[H]

\includegraphics{C:/Users/Jordan/git/COMP4Coursework2/Evaluation/client\_graph\_2}

\caption{According to this graph, the customer believed I had achieved 71.9\% of their specified objectives.}

\end{figure}

\chapter{User Manual}

\section{Introduction}

\textbf{Purpose: }

The purpose of this system is to give users a friendly experience when learning the trigonometry, pythagoras and vector aspects of maths, most likely as part of their GCSE preparation. The user can view lessons which provide sufficient knowledge to be able to gain some understanding of the topics, which they can then test using the homework provided. They can view their scores which are a reflection of their ability and progress in the subject. The end goal is that they do well in their maths GCSE, and this program is designed to help them do that.

\textbf{Intended Audience: }

The audience my program is targeting is GCSE students, in years 10 and 11 in the British education system (because this system is tailored more specifically to the syllabus of British exam boards) who have begun to prepare for their GCSE maths exam. Mainly year 10's are targeted as they will still be learning the basics of trigonometry, whereas year 11's should already know most of the subject, yet this system could still help them revise. Teachers could also be considered a client for this system as they might wish to use it in the school at which they are employed. Even parents of GCSE students or younger could be a client if they wished to provide their children with access to this system to boost their maths skills.

\section{Installation}

\subsection{Prerequisite Installation}

This system has been compiled to a windows executable (.exe) so no alternative software prerequisites are required to make use of this system on a windows operating system. Other operating systems have not been tested or developed for, so a windows operating system is required, preferably any of Windows 7, 8, or 10.

The following hardware list represents the minimum amount of hardware needed to be able to access and use every feature in the system:

\begin{itemize}

\item A keyboard and a mouse for input

\item A visual display unit, preferably at least 20 inches wide, for all of the widgets to be properly visible and not over-lapping

\item A hard disk drive (HDD) for storage

\item A minimum of 512 megabytes of main memory (RAM) to perform processing

\item An internet connection to download the software packages required to run the system (Python 3.4, PyQt4)

\end{itemize}

\subsubsection{Installing Python}

1. To install the appropriate version of Python (in order to be sure that it will be compatible), firstly go to \url{https://www.python.org/downloads/}.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_1}

\end{figure}

2. Click on the first big yellow button titled \textbf{Download Python 3.5.1} (or whatever updated version) if you want the latest version of Python. \textbf{Be aware, you will have to find a compatible version of PyQt elsewhere, as not all versions are on the same website as the one intended for this version of Python.}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_2}

\end{figure}

3. Alternatively, scroll down to the list of past Python releases and click the one that says \textbf{Python 3.4.3 - 2015-02-25}, to use the version which the system was created with.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_3}

\end{figure}

4. If you chose the most recent version, the download will begin at the bottom of your window, and should take no longer than 1 minute depending on your internet connection speed.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_4}

\end{figure}

5. Click the button with the download on it - A security warning will pop up. This file, presuming you selected the right one using the instructions above, is safe, so click run. Then a new window will open. \textbf{Please skip to step 8.}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_5}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_6}

\end{figure}

6. If you selected a past version, a different window will open in your internet browser. Click on the \textbf{Windows x86 MSI Installer} for Windows. The download will begin at the bottom of your window, and should take no longer than 1 minute depending on your internet connection speed. Click on the button, and a similar window to the other version will open.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_7}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_8}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_9}

\end{figure}

7. A security warning will pop up. This file, presuming you selected the right one using the instructions above, is safe, so click run.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_10}

\end{figure}

8. A setup wizard window will open (for whichever version of Python you chose). It is recommended that you leave install for all users ticked and click next.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_11}

\end{figure}

9. A directory should already be selected, most likely in your C: drive. This is an ideal place to save the Python files, but if you would prefer that they were saved somewhere else then click the 'Up' button to search through alternative locations. Once your directory has been selected, click next to continue.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_12}

\end{figure}

10. On the next window, most of the settings should be left alone. It might be useful however to add python to your path by selecting the bottom option, \textbf{Add python.exe to Path}, and choosing the first option, \textbf{Will be installed on local hard drive}. However this is not essential for using the system. Once you have decided on these settings click next.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_13}

\end{figure}

11. When you are asked if you would like to allow the system to make changes to your computer, click yes. Then wait a minute for Python to install. Once it has finished, click finish. Now Python is on your computer and ready to use.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_python\_14}

\end{figure}

\subsubsection{Installing PyQt4}

1. To install PyQt4 (not PyQt5 as it won't be compatible with the system), firstly go to \url{https://riverbankcomputing.com/software/pyqt/download}.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_1}

\end{figure}

2. This system is designed to run on a 32 bit version of Python, so choose the \textbf{PyQt4-4.11.4-gpl-Py3.4-Qt4.8.7-x32.exe} file for the Windows 32-bit installer.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_2}

\end{figure}

3. When you click on the link a new window will open in your browser telling you your download will begin shortly. Give it a few seconds and you should see the download button appear in the bottom left corner of the window.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_3}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_4}

\end{figure}

4. Click the button and an install wizard window will open. Click the next button to begin setting up PyQt4.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_5}

\end{figure}

5. Click the 'I Agree' button to accept the terms and conditions of the PyQt4 software package - assuming you've found the right package using the instructions, it is perfectly safe.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_6}

\end{figure}

6. Everything that comes with the software package is ticked for you by default - do not untick anything, as it may be something required to run the system. Leave the check boxes as they are and click next.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_7}

\end{figure}

7. The install wizard will ask you to select a folder for the PyQt4 files. It should have already selected your Python folder if you have installed Python already, in which case leave it and click install to ensure that the files are accessible by Python. If it does not already have a Python folder selected, it is recommended that you install Python first, or, if you have, search for Python in your file explorer and select Python 3 manually by clicking the browse button. Once you have Python3 selected, click install.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_8}

\end{figure}

8. Wait one minute for the installation to finish, then click the finish button. PyQt4 will now be installed on your computer and ready to be used for the system.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_pyqt\_9}

\end{figure}

\subsection{System Installation}

1. Navigate to the directory where you placed the installer file.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_system\_1}

\end{figure}

2. Double click with the left mouse button on the installer to start the installation process. You will be asked to choose a location to install the system. It is recommended that you go with the default recommendation, but otherwise you can use the arrows to navigate to a different directory. Once you have selected a directory, click next.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_system\_2}

\end{figure}

3. A security window will ask whether you want to allow the system to make changes to your computer. Click yes, then wait a few seconds for the system to install. Finally click finish, and the program will be installed on your computer ready for use.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/install\_system\_3}

\end{figure}

\subsection{Running the System}

\textbf{Make sure you have installed the system first by using the instructions in section 5.2.2}

1. Navigate to the directory where you installed the system files previously. It will run the same way wherever it has been saved, as long as the file \textbf{student\\_database.db} is in the same folder as the application file.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/run\_system\_1}

\end{figure}

2. Double click on the application file title \textbf{Trigonometry Education Program} to start running the system.

3. The system will load and appear on the screen. It is now running and can be used as desired.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/run\_system\_2}

\end{figure}

\section{Tutorial}

\subsection{Introduction}

In this section I will explain how to use each aspect of the system in order for the user to gain the full benefits of its purpose; using this guide the user should have no problems with the system which they will not be able to easily deal with. Each aspect will be explained using detailed instructions and annotated images to assist the user in understanding each aspect of the system.

\subsection{Assumptions}

The assumption that the user has basic computer capabilities and can use the mouse and keyboard has been made, and that the system is already running and ready for the user to use. Otherwise, things like the navigation of the system and the functions available to use in the system have not been assumed as known by the user.

\subsection{Tutorial Questions}

\subsubsection{How do I access the lessons?}

To access the lessons there is a branch system of menus which can be followed down to a lesson. From the home screen (pictured below) click the lessons button, then use the names on the buttons in the next menu to find the lesson which you want to use. The names of the sub menus are relevant to the lesson topics accessible from that sub menu. For example, to find the Pythagoras lessons, click the Pythagoras button and the Pythagoras Theorem lesson will be accessible from that sub-menu.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q1\_a}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q1\_b}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q1\_c}

\end{figure}

Upon clicking a lesson button the lesson window will open.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q1\_d}

\end{figure}

\subsubsection{How do I access the homework?}

Accessing the homework uses essentially the same system as accessing the lessons, only there are more options to choose from on the sub-menus. For each lesson, there are three homework tasks of varying difficulty. Starting from the home screen, click on the homework button (pictured below).

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q2\_a}

\end{figure}

From the homework topic menu, choose a topic related to the homework you wish to complete, and the sub-menu with the button to open that homework will open. If it is the wrong sub-menu, simply press return and try the other menus. Clicking the button with the title of the homework you want to complete will open that homework.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q2\_b}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q2\_c}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q2\_d}

\end{figure}

\subsubsection{How do I view my progress?}

To view all of the information stored in the database, from the home screen (pictured below) click the progress button.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q3\_a}

\end{figure}

All of the information will immediately be displayed in the progress window which will open upon clicking the progress button.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q3\_b}

\end{figure}

\subsubsection{How can I look at specific details of my progress?}

From the progress screen, accessed by clicking the progress button on the home screen, click on the report button to open a new window.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q4\_a}

\end{figure}

The first combo box contains all the names of the tasks. Click on the arrow, and select the name of the task which you want to query the database for. Then do the same with the next combo box, only this time you are choosing the score you want to query. You can choose to query one or the other by leaving the input blank. Once you have chosen your query criteria, click on the submit button and the relevant information will appear in the table to the left.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q4\_b}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q4\_c}

\end{figure}

\subsubsection{How do I know I won't lose progress by exiting the program?}

The only way you could lose progress is by closing a homework window using the cross in the top right hand corner of the window instead of using the next or finish button - only the inputs on those screens will be lost, and you will have to put them in again if you want the saved results. As long as the next or finish button is clicked, the progress from the respective page will be saved automatically by the system. Progress already stored in the database can only be lost if it is over-written with better results, or if you manually delete the \textbf{student\\_database.db} file. Otherwise, information in the database will not delete itself at any point.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q5\_a}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q5\_b}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q5\_c}

\end{figure}

\subsubsection{How do I save my progress?}

You do not need to manually save anything. The only way to save results is by clicking the next button on a homework first page, to save the first half of a record, then by clicking the finish button on a homework second page, to save the second half of a record. All saves will be made internally and automatically, and all new information will immediately be available for viewing in the progress window or the report window.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q6\_a}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q6\_b}

\end{figure}

\subsubsection{What do I do if an error message pops up?}

Details on how to deal with specific error messages are in section 5.4, but in order to remove an error message window from the centre of the screen you must simply click the OK button and then proceed to do what the error message advised you do.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/q7\_a}

\end{figure}

\subsection{Saving}

There is only one file saved and accessed by the system and that is the \textbf{student\\_database.db} file. All saving is done internally and automatically by the system; when the user clicks next on a homework first page window, the task name and first question score is saved, then when they click finish on the second homework screen, the other question scores are updated on the record just saved prior to opening this screen. The methods run when the buttons are clicked will automatically execute a database query, so the user does not have to manually save anything, just complete all of the questions and close the stack window using the finish button, not the cross in the top right corner of the window.

\subsection{Limitations}

The database cannot be reset by the user in-system. This was a secondary objective considered which did not make it into the system, so unless it is updated in the future the only way a user would be able to reset all of their progress would be to manually delete the \textbf{student\\_database.db} file from their system files. Upon running the program a new database would be created with the same name, and it would be initially empty.

The system does not have any of the administrative capabilities originally proposed in the analysis, essentially meaning that the client on this case will not be able to monitor students' progress, although they will still have the same level of opportunity to gain knowledge of the relevant aspects of maths in preparation for their GCSE's.

\section{Error Recovery}

\subsection{ErrorMessage2}

This error message appears if the user inputs a wrong data type for an answer - it is essentially a friendly way of giving them another go because they made quite a big mistake, not just a wrong answer of the right data type, without removing an attempt.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/error\_pic\_1}

\end{figure}

To solve this error, simply go back to the same input box which the wrong answer was typed into, and try again using a decimal value - a number with a point before and after at least 1 digit. For example, 4.5 or 7.13.

\subsection{ErrorMessage3}

This error message appears if the user inputs a wrong data type for an answer - it is essentially a friendly way of giving them another go because they made quite a big mistake, not just a wrong answer of the right data type, without removing an attempt.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/error\_pic\_2}

\end{figure}

To solve this error, simply go back to the same input box which the wrong answer was typed into, and try again using an integer - a whole number such as 5 or 120.

\subsection{ErrorMessage4}

This error message appears if the user inputs a wrong data type for an answer - it is essentially a friendly way of giving them another go because they made quite a big mistake, not just a wrong answer of the right data type, without removing an attempt.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/error\_pic\_3}

\end{figure}

To solve this error, simply go back to the same input box which the wrong answer was typed into, and try again using a string - a word or answer consisting of letters like 'triangle'.

\subsection{ErrorMessage5}

This error message just informs the user that they got an answer incorrect so they know to have another go, if they have attempts remaining. To deal with this error simply try again until either you get it right or run out of attempts.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/error\_pic\_4}

\end{figure}

\subsection{ErrorMessage8}

This error message appears if the user has not answered and checked all of the questions on a homework. If this message appears, find the input box which you have not yet attempted, check your input until you get it right or run out of attempts, then you will be able to proceed to the next screen. The reason for this error is that a score value is required for the system to be able to execute the SQL query which saves the score to the database.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/error\_pic\_5}

\end{figure}

\section{System Recovery}

\subsection{Backing-up Data}

1. It is quite simple to back up the data for this system. There is only one file with data, and that will be called \textbf{student\\_database.db} and - if you used the default file directory when installing the system - is located in your local \textbf{C:} drive. Follow this path: \textbf{This PC, Local Disk (C:), Program Files (x86), Trigonometry Education Program}.

The file \textbf{student\\_database.db} will be near the bottom of this folder.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/backup\_1}

\end{figure}

2. Left click on the \textbf{student\\_database.db} file ONCE and then right click on it when it is highlighted.

In the menu box, select 'Copy'. Alternatively, when it is highlighted, hold Ctrl and press C.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/backup\_2}

\end{figure}

3. Now navigate to a folder separate from the folder the file is currently in, to avoid the backup also being lost should there be a fault with the folder. For example, \textbf{This PC, Local Disk(C:), Python34}.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/backup\_3}

\end{figure}

4. Now right click on a white space in the folder, and when the menu box appears, select 'Paste'. Alternatively, hold Ctrl and press P.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/backup\_4}

\end{figure}

Now you have a copy of the file with all of the database details on it, which can be accessed to replace the other copy should anything happen to it, like corruption or deletion. If you want to be even more secure you can upload a copy to Dropbox or a similar website.

\subsection{Restoring Data}

1. To restore the \textbf{student\\_database.db} file, firstly navigate to where you pasted the copy of it from the backing-up data section. If you have not already done this, chances are you will not be able to get the file back at all if the original was corrupted.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/restore\_1}

\end{figure}

2. Left click on the file to highlight it, then right click on it to make the menu box appear. Select the 'Copy' option. Alternatively, when it is highlighted, hold Ctrl and press C.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/restore\_2}

\end{figure}

3. Now navigate to the folder where all of the system files are kept, where either the file you want to replace is or the deleted file was. If you followed the installation guide's recommendation, it should be in \textbf{Local Disk (C:), Program Files (x86), Trigonometry Education Program}.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/restore\_3}

\end{figure}

4. If a corrupt or unwanted file is there, left click on it to highlight it, then right click anywhere and select 'Paste'. Alternatively, just hold Ctrl and press P when in the folder.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/restore\_4}

\end{figure}

5. A window will appear asking you if you want to replace or duplicate the file. Click replace.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Manual/restore\_5}

\end{figure}

6. If a file is not already there, just hold Ctrl and press P, and the copy will immediately be placed in the folder.

Your restored \textbf{student\\_database.db} file is now in the folder ready to be used.

\chapter{System Maintenance}

\section{Environment}

\subsection{Software}

I used the following software to produce my system:

\begin{itemize}

\item Python 3.4

\item Python IDLE

\item PyQt4 (Contains some HTML elements)

\item SQLite3

\item Internet Explorer

\end{itemize}

\subsection{Usage Explanation}

\textbf{Python 3.4: }

Python 3.4 was the most up to date release of Python when I began the implementation of the system, and I continued to use it throughout, despite 3.5 being released, to avoid any incompatibilities with the other software used. I use this language because it is the language I used to learn the fundamentals of prorgamming and therefore the one I am most familiar with.

\textbf{Python IDLE: }

Python IDLE is the environment which Python uses, and which I am most familiar with.

\textbf{PyQt4: }

PyQt4 is Qt, a separate GUI language, altered to work with Python in order to create a clear, smoothly operating graphical interface, and was also used to change the colours and sizes of widgets in the system. Some elements of HTML are in Qt which allows for colour changing and size adjusting.

\textbf{SQLite3: }

SQLite3 is included in the Python 3.4 software download package by default, and is useful for creating a very effective database structure which can be easily accessed and modified uisng Python and PyQt code.

\textbf{Internet Explorer: }

I used IE to look online to find out how to use code which I had not used before and which wasn't fundamental to learn for my A-Level course, such as how to use PyQt4 to change the sizes and colours of widgets.

\subsection{Features Used}

\textbf{Python 3.4: }

Python 3.4 allowed me to write my code and be able to test it, even in a GUI form when that point was reached.

\textbf{Python IDLE: }

Python 3.4 comes with an IDLE environment which can be used to easily and frequently test my system as it is created; it allows you to view the system as it would look following distribution, with the IDLE window being used for inputs and outputting errors, which is very useful for fixing said errors. It also has a very clear colour-coded scheme for the code, making it easier to find segments of code.

\textbf{PyQt4: }

PyQt4 comes with many pre-coded tools which I could use to implement a graphical use interface; I was able to create classes which were derived from pre-coded files in the software package, such as buttons and windows. It also gave me limited HTML capabilities with which I could change the look of the GUI.

\textbf{SQLite3: }

SQLite3 was already part of the Python 3.4 package and provided effective SQL capabilities which I used to write data to a structured database and retrieve data ready to be output to the user.

\textbf{Internet Explorer: }

Wasn't a part of the implementation, was only used to research code which I could use. Provided useful access to StackOverflow.

\section{System Overview}

\subsection{Graphical User Interface}

The GUI provides a navigation tool for the user to use to navigate the different parts of the system in a friendly and easy to use way. Buttons are used to connect the screens and make it easy to access them by simply clicking them. Line edits, combo boxes and other buttons are used for a variation of input methods, which have been enlarged to give the system a more commercial look. Text boxes and images are used to provide an output which makes it clear that the system is intended for educational purposes. The interface also has clear titles so the user will always know where they are in the system.

\subsection{Navigation of Windows}

The windows have been structured in such a way that it is necessary to understand how each of them are connected; The welcome screen and home screen are in a stack so that the welcome screen will not be accessible once the user is already in the system. The menus are all connected individually using subclassed buttons and methods, to ensure that each window can be accessed from at least one of the previous menus, but only from one. The lessons and homeworks are in stacks so that the user cannot have only one screen of a homework open at one time, and the user's inputs on each screen will be kept there until the entire stack is closed, should they decide to go back to something. There is a home screen, which connects to two topic menus, which connect to five specific menus each, which each connect to two, three or six lessons or homeworks, like a branch system. The connections which share a menu are all in the same file to make them easier to find.

\subsection{Viewing a Lesson}

Each lesson is accessed from a derived lesson menu, all of which are in the same file (so all lesson connections are also in the same file), and each lesson consists of two pages in a stack together, both of which are subclassed from separate 'page 1' and 'page 2' parent classes. There are buttons which can allow the user to easily cancel the lesson and return to the menu, continue to the next screen, and close the window when they are done. There is one line edit with a simple test question in each 'page 2', but this isn't recorded in the database. The line edit answer can be checked by clicking the check answer button, and an algorithm is run which will tell the user whether or not they are correct by checking the user's input against the hard-coded answer in the sub-class. This algorithm is in the lesson 'page 2' parent class.

\subsection{Completing a Homework}

The homeworks are accessed in exactly the same way as the lessons, except they are branched from the homework button on the home screen, and use different connections in the following menus. Again, there are buttons for easy navigation, although the homework widgets also have more line edits, combo boxes and multiple choice buttons for inputting answers, the scores from which are saved to the database. Each answer is checked using individual algorithms which are in the parent homework 'page 1' and 'page 2' classes. These algorithms essentially do the same thing, just working with different input types. They check the user's input against the hard-coded answers in the sub-classes, and give error messages if the user is wrong, until they run out of attempts, which decrement with every wrong answer as part of the algorithm. Once a question is either correct, or the user has run out of attempts, the algorithm will disable the input widegts as appropriate to prevent the user having too many goes or saving to the database twice and entering a loop or other error.

\subsection{Storing in the Database}

The database methods are all ina separate class, which can be accessed by all of the other files when needed. For example, when a homework score needs to be saved the file with the homework in it will be able to access the database class and the appropriate method, and pass through the variables from the homework into the SQL insert statements. The only times the database is written to is when the user completes a homework; the task name and first question score is saved after clicking next to the second page, and the scores for the second, third and fourth questions are saved after clicking finish. If the task has been done before and the new scores are better, they will over-write the old ones. The update statements are separate for each question score so that they don't all have to be better for one to over-write. The database is accessed for output when the user loads the progress viewer or the report widget; all information based on the corresponding query is fetched and displayed in the QTableWidgets.

\subsection{Viewing the Database Information}

There are two ways for the user to view information stored in the database in the system: The progress window, accessible from the home screen, and the report window, accessible from the progress screen, both of which use QTableWidgets to display the information. The progress screen just displays all data in the database in the format you would expect; each piece of data is under the right column and in the right row. The report screen begins blank and is filled with all information relevant to the query which can be made by the user to search for specific task names or scores, using combo boxes for input. SQL statements are used to search for the relevant data, which is fetched, and organised in the table widget using iteration.

\subsection{The Task Data}

The task data is obtained by clicking the next button on any first homework page, and the hard-coded task name variable in the sub-class will be recorded to the database. Once it is saved once, it will never change or disappear, as there is no need. All task names will appear under the Task Name column in the qtablewidgets.

\subsection{The Score Data}

The score for the first question of any homework will be calculated using the algorithms and then saved to the database with the task name when next is clicked. Clicking next will also save the values 0 to the other 3 question's spaces in the database. Update statements are used so that whenever finish is clicked on the second page, the scores will be over-written whether they are 0's, or if the task has been done before, values less than the new score.

\section{Code Structure}

\subsection{Database Controller Class}

\begin{python}

class Database:

def \_\_init\_\_(self, db\_name):

self.\_db\_name = db\_name

self.table\_name = "Student"

self.create\_table(self.table\_name)

\end{python}

This class contains all of the database manipulating code for the system. It can be accessed by any file using the global database variable which can be called from methods which need it, making it easier to keep all SQL code and PyQt code separate. db\\_name is the name of the variable which represents the database which is called at the bottom of the file; each method can be called to change this database variable and the variables which change it are passed through from the subclasses in other files, which is efficient as most of the files are subclassed so there will be no collisions between data. The table name is hard-coded so it will never change and the system will always be able to search for the same name to check if the database exists when it is run. The create\\_table method is then run, which either creates a new table, leaves the old one or replaces the old one.

\subsubsection{execute\\_sql method}

\begin{python}

def execute\_sql(self, sql):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute(sql)

\end{python}

This method is called every time an SQL statement is executed in the other methods in this class. It connects to the database using the name which is passed through from the global variable, which will always be the same as it is hard-coded; this connection is only needed to be written once as every SQL statement will be executed following this connection being made in this method. Then the cursor, which is the control structure, is also always in this method so is not need in the others. Then the SQL statement, which is passed through from the method which is calling the execute\\_sql method, is executed from using the cursor, which will make the appropriate change to the database.

\subsubsection{create\\_table method}

\begin{python}

def create\_table(self, table\_name):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select name from sqlite\_master where name=?",(table\_name,))

result = cursor.fetchall()

keep\_table = True

if len(result) == 1:

response = input("The table {0} already exists, do you wish to recreate it (y/n): ".format(table\_name))

if response == "y":

keep\_table = False

print("The {0} table will be recreated - all existing data will be lost".format(table\_name))

cursor.execute("drop table if exists {0}".format(table\_name))

db.commit()

else:

print("The existing table was kept")

else:

keep\_table = False

if not keep\_table:

sql = """create table Student

(TaskID text,

Qone integer,

Qtwo integer,

Qthree integer,

Qfour integer,

primary key(TaskID))"""

cursor.execute(sql)

db.commit()

\end{python}

This method is run as soon as the program is run; the first SQL statement searches all sqlite3 files to check if a database called student already exists, and returns all of the values, in this case either 1 or 0. If it does exist, it will ask the user whether or not they want to over-write the existing database. At the moment this is useful for testing but it will not be in the final version; the user will not be able to over-write the database unless they do it manually with the settings I might put in as a secondary objective. The variables used to check the name of the table are passed in so that they can be different, but they won't be as for now the database name is hard-coded. The SQL statement which creates the actual table uses the execute\\_sql function (previous section) to connect to the database and make the changes. db.commit() makes sure the changes stay and are not forgotten.

\subsubsection{insert\\_data\\_first method}

\begin{python}

def insert\_data\_first(self, task, correct\_count):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select TaskID from Student where TaskID = '{0}'".format(task))

info = cursor.fetchall()

if len(info) != 0:

sql = "UPDATE Student SET Qone = '{0}' WHERE TaskID = '{1}' AND Qone < '{2}'".format(correct\_count, task, correct\_count)

else:

sql = "insert into Student(TaskID, Qone, Qtwo, Qthree, Qfour) values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(task, correct\_count, str(0), str(0), str(0))

self.execute\_sql(sql)

\end{python}

This method is called in the homework page 1 subclasses. It can be called from any of the 24 subclasses, and the polymorphed variables from each one are passed through, so that, for example, there are no cross-overs of data like the Sides Easy task name being recorded as the parent default. The SQL here firstly checks to see if a task with the same task name already exists, and if it does, each value will only be over-written if it is greater than the exisitng one. Otherwise it will create a new record and save the values from the second pages as 0 so the same SQL statements can be used, update statements, whether the record existed or not before. the execute\\_sql method is run in order to make the changes to the table.

\subsubsection{insert\\_data\\_second method}

\begin{python}

def insert\_data\_second(self, task, count\_2, count\_3, count\_4):

with sqlite3.connect(self.\_db\_name) as db:

sql = "UPDATE Student SET Qtwo = '{0}' WHERE TaskID = '{1}' AND Qtwo < '{2}'".format(count\_2, task, count\_2)

self.execute\_sql(sql)

sql\_2 = "UPDATE Student SET Qthree = '{0}' WHERE TaskID = '{1}' AND Qthree < '{2}'".format(count\_3, task, count\_3)

self.execute\_sql(sql\_2)

sql\_3 = "UPDATE Student SET Qfour = '{0}' WHERE TaskID = '{1}' AND Qfour < '{2}'".format(count\_4, task, count\_4)

self.execute\_sql(sql\_3)

\end{python}

The SQL statements in this method are individual so that they don't all have to be greater values than the existing value in each column to be able to over-write the previous value. These statements work regardless of whether or not a record for the task already existed because the previous method (which is always run before this one) will write 0 values to each attribute in the table. Like the insert\\_data\\_first method, this can be called from each homework page 2 subclass to avoid data collisions.

\subsubsection{get\\_query}

\begin{python}

def get\_query(self, data, score\_data):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select \* from Student WHERE TaskID = '{0}' or Qone = '{1}'".format(data, score\_data))

report = cursor.fetchall()

return report

\end{python}

This method is called in the report widget, and the SQL statement searches for data in the database which is equal to the value of the variables passed through, which come from the combo boxes where the user selects the information they want to search for. data and score\\_data come from the combo boxes so that each time the button is clicked the variables can be changed and the table which displays the data is immediately updated based on the new query.

\subsubsection{GetAllNames method}

\begin{python}

def GetAllNames(self):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select \* from Student")

students = cursor.fetchall()

return students

\end{python}

This method fetches all data currently in the database and returns it so that in the progress window, where this method is called automatically, all data is displayed in the table.

\subsection{DatabaseWidget Class}

\begin{python}

class DatabaseWidget(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.title = QLabel("Progress")

self.title.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumWidth(60)

self.back.setMinimumHeight(100)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.report = QPushButton("Report")

self.report.setMinimumWidth(60)

self.report.setMinimumHeight(100)

self.report.setFont(QFont("Courier", 40))

self.database = QTableWidget()

self.database.setRowCount(24)

self.database.setColumnCount(5)

self.database\_header = ("Task Name", "Question 1", "Question 2", "Question 3", "Question 4")

self.database.setHorizontalHeaderLabels(self.database\_header)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

self.database.setStyleSheet("QTableView {selection-background-color: #A3C1DA;}")

students = g\_database.GetAllNames()

count = 0

for student in students:

self.database.setItem(count, 0, QTableWidgetItem(student[0]))

self.database.setItem(count, 1, QTableWidgetItem(str(student[1])))

self.database.setItem(count, 2, QTableWidgetItem(str(student[2])))

self.database.setItem(count, 3, QTableWidgetItem(str(student[3])))

self.database.setItem(count, 4, QTableWidgetItem(str(student[4])))

count += 1

self.layout = QGridLayout()

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.database, 0, 1)

self.layout.addWidget(self.back, 4, 0)

self.layout.addWidget(self.report, 4, 1)

self.setLayout(self.layout)

self.back.clicked.connect(self.selected\_back)

self.report.clicked.connect(self.selected\_report)

\end{python}

This class is the class which contains the QTableWidget which displays all data from the database. This window can be accessed by clicking the progress button on the home screen. When this window is loaded, the all information is automatically fetched from the database controller and immediately displayed in the QTableWidget so the user can easily access all information. The report widget is for querying and viewing specific imformation, and can be accessed from this window. The for loop is used to display each item in the database in the right place; the list value represents the position of the piece of data in the table. For example each student[0] is a task name, and is displayed in the first column. A count increments so that the next row is accessed each time and the first record isn't constantly over-written up until the last record. This window is accessible from the home page so the user can view all progress without having to go through any other menus.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This method simply closes the window, so the previous window is displayed. It is called when the previous button is clicked.

\subsubsection{selected\\_report method}

\begin{python}

def selected\_report(self):

report\_widget = ReportWidget()

report\_widget.show()

report\_widget.\_raise()

\end{python}

This method opens the report window when the report button is clicked.

\subsection{ReportWidget Class}

\begin{python}

class ReportWidget(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.header = QLabel("Report")

self.header.setFont(QFont("Courier", 30))

self.task\_box\_label = QLabel("Please select a task\nto query: ")

self.task\_box\_label.setFont(QFont("Courier", 25))

self.task\_box = QComboBox()

self.task\_box.setMinimumWidth(60)

self.task\_box.setMinimumHeight(100)

self.task\_box.setFont(QFont("Courier", 30))

self.task\_box.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.task\_box.addItem("")

self.task\_box.addItem("Sides Easy")

self.task\_box.addItem("Sides Medium")

self.task\_box.addItem("Sides Hard")

self.task\_box.addItem("SOHCAHTOA Easy")

self.task\_box.addItem("SOHCAHTOA Medium")

self.task\_box.addItem("SOHCAHTOA Hard")

self.task\_box.addItem("Finding Angles Easy")

self.task\_box.addItem("Finding Angles Medium")

self.task\_box.addItem("Finding Angles Hard")

self.task\_box.addItem("3D Trigonometry Easy")

self.task\_box.addItem("3D Trigonometry Medium")

self.task\_box.addItem("3D Trigonometry Hard")

self.task\_box.addItem("Pythagoras' Theorem Easy")

self.task\_box.addItem("Pythagoras' Theorem Medium")

self.task\_box.addItem("Pythagoras' Theorem Hard")

self.task\_box.addItem("3D Pythagoras Easy")

self.task\_box.addItem("3D Pythagoras Medium")

self.task\_box.addItem("3D Pythagoras Hard")

self.task\_box.addItem("Vectors Easy")

self.task\_box.addItem("Vectors Medium")

self.task\_box.addItem("Vectors Hard")

self.task\_box.addItem("Easy Summary")

self.task\_box.addItem("Medium Summary")

self.task\_box.addItem("Hard Summary")

self.score\_box\_label = QLabel("Please input the maximum\nscore you would like\nto query: ")

self.score\_box\_label.setFont(QFont("Courier", 25))

self.score\_box = QComboBox()

self.score\_box.setMinimumWidth(60)

self.score\_box.setMinimumHeight(100)

self.score\_box.setFont(QFont("Courier", 30))

self.score\_box.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.score\_box.addItem(None)

self.score\_box.addItem("6")

self.score\_box.addItem("5")

self.score\_box.addItem("4")

self.score\_box.addItem("3")

self.score\_box.addItem("2")

self.score\_box.addItem("1")

self.score\_box.addItem("0")

self.back = QPushButton("Return")

self.back.setMinimumWidth(60)

self.back.setMinimumHeight(100)

self.back.setFont(QFont("Courier", 30))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.submit = QPushButton("Query")

self.submit.setMinimumWidth(60)

self.submit.setMinimumHeight(100)

self.submit.setFont(QFont("Courier", 30))

self.submit.setStyleSheet("QPushButton {background-color: green; color: white;}")

self.db = QTableWidget()

self.db.setRowCount(24)

self.db.setColumnCount(5)

self.db\_header = ("TaskName", "Question 1", "Question 2", "Question 3", "Question 4")

self.db.setHorizontalHeaderLabels(self.db\_header)

self.db.setStyleSheet("QTableWidget {selection-background-color: #A3C1DA;}")

self.layout = QGridLayout()

self.setLayout(self.layout)

self.layout.addWidget(self.db, 0, 0)

self.layout.addWidget(self.task\_box\_label, 0, 1)

self.layout.addWidget(self.task\_box, 1, 1)

self.layout.addWidget(self.score\_box\_label, 2, 1)

self.layout.addWidget(self.score\_box, 3, 1)

self.layout.addWidget(self.back, 4, 0)

self.layout.addWidget(self.submit, 4, 1)

self.back.clicked.connect(self.selected\_back)

self.submit.clicked.connect(self.selected\_submit)

\end{python}

This widget is similar to the progress window except it is used to search for specific data in the database. A QTableWidget is used to display all the relevant data which is fetched using the selected\\_submit method. The window uses combo boxes to select inputs; all of the possible task names and scores are placed in the combo box for the user to choose from. This window is accessible from the progress window because it would not be convenient for it to be anywhere else; here, the user is already checking database information.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This closes the window and the progress screen is displayed - if the user opens the report widget again the table widget will be blank, it won't save the previous query.

\subsubsection{selected\\_submit method}

\begin{python}

def selected\_submit(self):

\_count = 0

data = self.task\_box.currentText()

score\_data = self.score\_box.currentText()

report = g\_database.get\_query(data, score\_data)

for count in range(24):

self.db.setItem(count, 0, QTableWidgetItem(None))

self.db.setItem(count, 1, QTableWidgetItem(None))

self.db.setItem(count, 2, QTableWidgetItem(None))

self.db.setItem(count, 3, QTableWidgetItem(None))

self.db.setItem(count, 4, QTableWidgetItem(None))

for record in report:

self.db.setItem(\_count, 0, QTableWidgetItem(record[0]))

self.db.setItem(\_count, 1, QTableWidgetItem(str(record[1])))

self.db.setItem(\_count, 2, QTableWidgetItem(str(record[2])))

self.db.setItem(\_count, 3, QTableWidgetItem(str(record[3])))

self.db.setItem(\_count, 4, QTableWidgetItem(str(record[4])))

\_count += 1

\end{python}

This is the method which accesses the database using the get\\_query method (DatabaseWidget Class). The information is displayed in a QTableWidget like in the progress window, only it will usually need fewer rows to display all of the fetched information. This method passes in the variables which are taken from the text in the combo boxes which are selected by the user, and fetches all information from the database which matches these variables. This method sets all of the values in the table widget to blank before it displays the new queried data so that every time a query is made the previous query's data is gone; only one query at a time is necessary.

\subsection{FirstScreen Class}

\begin{python}

class FirstScreen(QWidget):

NameEntered = pyqtSignal()

def \_\_init\_\_(self):

super().\_\_init\_\_()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.message = QLabel("Welcome to the Triangle Geometry Education Program")

self.message.setFont(QFont("Courier", 40))

self.message.setAlignment(Qt.AlignCenter)

self.cont = QPushButton("Continue")

self.cont.setMinimumHeight(110)

self.cont.setMinimumWidth(60)

self.cont.setFont(QFont("Courier", 40))

self.pic = QLabel()

self.pic.setPixmap(QPixmap("powered\_by\_python"))

self.pic.setAlignment(Qt.AlignCenter)

self.layout = QGridLayout()

self.setLayout(self.layout)

self.layout.addWidget(self.pic, 0, 0)

self.layout.addWidget(self.message, 1, 0)

self.layout.addWidget(self.cont, 2, 0)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

self.cont.clicked.connect(self.enter)

\end{python}

This class contains the template for the first screen which is displayed when the system is run; this screen is only displayed at the start, once the user selects continue it cannot be accessed again until the next session. Its main purpose is to be user friendly and give the user a comprehensible start to the system. HTML is included in the PyQt, which sets the positioning of the widgets in the window and changes the size and colour of the button and text. This is part of a stack widget so that it does not remain open behind the home screen; the home screen replaces it. A pyqtSignal is used to send the signal for the connection when the continue button is clicked to change the current screen.

\subsubsection{enter method}

\begin{python}

def enter(self):

self.nameEntered.emit()

\end{python}

This method contains the signal which tells python to switch to the next screen when the button is clicked - it essentially changes nameEntered to true, which is the condition to display the home screen in place of the first screen.

\subsection{UserAccountWidget Class}

\begin{python}

class UserAccountWidget(QWidget):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent\_window = parent

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.lessons = QPushButton("Lessons")

self.lessons.setMinimumWidth(90)

self.lessons.setMinimumHeight(110)

self.lessons.setFont(QFont("Courier", 40))

self.homework = QPushButton("Homework")

self.homework.setMinimumWidth(90)

self.homework.setMinimumHeight(110)

self.homework.setFont(QFont("Courier", 40))

self.progress = QPushButton("Progress")

self.progress.setMinimumWidth(90)

self.progress.setMinimumHeight(110)

self.progress.setFont(QFont("Courier", 40))

self.lessons\_label = QLabel("To view lessons\nand learn more,\nclick here! ")

self.lessons\_label.setFont(QFont("Courier", 25))

self.homework\_label = QLabel("To access the\nhomework set for\nyou to complete,\nclick here! ")

self.homework\_label.setFont(QFont("Courier", 25))

self.database\_label = QLabel("To view your\nprogress so far,\nclick here! ")

self.database\_label.setFont(QFont("Courier", 25))

self.log\_out = QPushButton("Exit Program")

self.log\_out.setMinimumWidth(90)

self.log\_out.setMinimumHeight(110)

self.log\_out.setFont(QFont("Courier", 40))

self.log\_out.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

self.picture = QLabel()

self.picture.setPixmap(QPixmap("student\_account\_home\_pic"))

self.picture.setAlignment(Qt.AlignCenter)

self.homework\_pic = QLabel()

self.homework\_pic.setPixmap(QPixmap("student\_home\_homework"))

self.homework\_pic.setAlignment(Qt.AlignCenter)

self.smiler = QLabel()

self.smiler.setPixmap(QPixmap("smile"))

self.smiler.setAlignment(Qt.AlignCenter)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

self.layout = QGridLayout()

self.layout.addWidget(self.lessons, 0, 1)

self.layout.addWidget(self.picture, 0, 2)

self.layout.addWidget(self.homework, 1, 1)

self.layout.addWidget(self.progress, 2, 1)

self.layout.addWidget(self.lessons\_label, 0, 0)

self.layout.addWidget(self.homework\_label, 1, 0)

self.layout.addWidget(self.database\_label, 2, 0)

self.layout.addWidget(self.picture, 1, 3)

self.layout.addWidget(self.log\_out, 2, 3)

self.layout.addWidget(self.smiler, 2, 2)

self.layout.addWidget(self.homework\_pic, 1, 2)

self.setLayout(self.layout)

self.lessons.clicked.connect(self.selected\_lessons)

self.homework.clicked.connect(self.selected\_homework)

self.progress.clicked.connect(self.selected\_progress)

self.log\_out.clicked.connect(self.log\_out\_selected)

\end{python}

This class is the template for the second window in the stack with the first screen (FirstScreen Class) and is accessed by clicking continue from the first screen. This window has the buttons with connections to every other screen in the window; in other words, the top of the branch. The lessons button takes the user to menus where they can find lessons, the homework button takes them to homework menus, and the progress buttons opens the database viewer. This window is always open so the user can finish a task, close it, and be returned here to select their next task. It also contains the exit program button which closes the entire system immediately, with no messages asking if they're sure, and no windows are left open.

\subsubsection{log\\_out\\_selected method}

\begin{python}

def log\_out\_selected(self):

sys.exit()

\end{python}

This method is connected to the exit program button and closes down the entire system. All data in the database is saved in a separate file so there is no need to try and keep anything open or remember any inputs which haven't been saved yet; they just have to go back and do it again.

\subsubsection{selected\\_lessons method}

\begin{python}

def selected\_lessons(self):

lessonmenuwidget = LessonMenuWidget()

lessonmenuwidget.show()

lessonmenuwidget.\_raise()

lessonmenuwidget.showMaximized()

\end{python}

This simply opens the lesson menu which is in a separate window, so when they press return or close that window, the home screen will still be open ready to access any other part of the system or close it down.

\subsubsection{selected\\_homework method}

\begin{python}

def selected\_homework(self):

homeworkmenuwidget = HomeworkMenuWidget()

homeworkmenuwidget.show()

homeworkmenuwidget.\_raise()

homeworkmenuwidget.showMaximized()

\end{python}

This simply opens the homework menu which is in a separate window, so when they press return or close that window, the home screen will still be open ready to access any other part of the system or close it down.

\subsubsection{selected\\_progress method}

\begin{python}

def selected\_progress(self):

databasewidget = DatabaseWidget()

databasewidget.show()

databasewidget.\_raise()

databasewidget.showMaximized()

\end{python}

This simply opens the progress menu which is in a separate window, so when they press return or close that window, the home screen will still be open ready to access any other part of the system or close it down.

\subsection{LessonMenuWidget Class}

\begin{python}

class LessonMenuWidget(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.t1 = QPushButton("Trigonometry 1")

self.t1.setMinimumWidth(90)

self.t1.setMinimumHeight(110)

self.t1.setFont(QFont("Courier", 40))

self.t1\_pic = QLabel()

self.t1\_pic.setPixmap(QPixmap("t1\_pic"))

self.t1\_pic.setAlignment(Qt.AlignCenter)

self.t2 = QPushButton("Trigonometry 2")

self.t2.setMinimumWidth(90)

self.t2.setMinimumHeight(110)

self.t2.setFont(QFont("Courier", 40))

self.t2\_pic = QLabel()

self.t2\_pic.setPixmap(QPixmap("t2\_pic"))

self.t2\_pic.setAlignment(Qt.AlignCenter)

self.pyt = QPushButton("Pythagoras")

self.pyt.setMinimumWidth(90)

self.pyt.setMinimumHeight(110)

self.pyt.setFont(QFont("Courier", 40))

self.pyt\_pic = QLabel()

self.pyt\_pic.setPixmap(QPixmap("pyt\_pic"))

self.pyt\_pic.setAlignment(Qt.AlignCenter)

self.pytrig = QPushButton("Vectors")

self.pytrig.setMinimumWidth(90)

self.pytrig.setMinimumHeight(110)

self.pytrig.setFont(QFont("Courier", 40))

self.pytrig\_pic = QLabel()

self.pytrig\_pic.setPixmap(QPixmap("pytrig\_pic"))

self.pytrig\_pic.setAlignment(Qt.AlignCenter)

self.sum = QPushButton("Summary")

self.sum.setMinimumWidth(90)

self.sum.setMinimumHeight(110)

self.sum.setFont(QFont("Courier", 40))

self.sum\_pic = QLabel()

self.sum\_pic.setPixmap(QPixmap("sum\_pic"))

self.sum\_pic.setAlignment(Qt.AlignCenter)

self.back = QPushButton("Return")

self.back.setMinimumWidth(90)

self.back.setMinimumHeight(110)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.lesson\_label = QLabel("Lessons")

self.lesson\_label.setFont(QFont("Courier", 40))

self.select = QLabel("Please select a topic: ")

self.select.setFont(QFont("Courier", 25))

self.title\_pic = QLabel()

self.title\_pic.setPixmap(QPixmap("title\_lessons"))

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

self.layout = QGridLayout()

self.layout.addWidget(self.title\_pic, 0, 0)

self.layout.addWidget(self.t1\_pic, 1, 0)

self.layout.addWidget(self.t1, 1, 1)

self.layout.addWidget(self.t2, 2, 0)

self.layout.addWidget(self.t2\_pic, 2, 1)

self.layout.addWidget(self.pyt\_pic, 3, 0)

self.layout.addWidget(self.pyt, 3, 1)

self.layout.addWidget(self.pytrig, 4, 0)

self.layout.addWidget(self.pytrig\_pic, 4, 1)

self.layout.addWidget(self.sum\_pic, 5, 0)

self.layout.addWidget(self.sum, 5, 1)

self.layout.addWidget(self.back, 6, 0)

self.\_centralwidget = QWidget()

self.\_centralwidget.setLayout(self.layout)

self.setCentralWidget(self.\_centralwidget)

self.t1.clicked.connect(self.selected\_t1)

self.t2.clicked.connect(self.selected\_t2)

self.pyt.clicked.connect(self.selected\_pyt)

self.pytrig.clicked.connect(self.selected\_pytrig)

self.sum.clicked.connect(self.selected\_sum)

self.back.clicked.connect(self.selected\_back)

\end{python}

This class is the template for the menu screen which connects to the five individual subclassed menus which contain the buttons for the specific lessons; it is the middle screen in a branch menu of three screens. It is accessed by clicking the lessons button on the home screen (UserAccountHome Class), and it connects to five different final menus from which the user can access the lessons. This structure is used to make it easier to navigate to specific lesson topics, and to fit clear and visible buttons on the page, which are all relevant to each other.

\subsubsection{selected\\_t1 method}

\begin{python}

def selected\_t1(self):

trig\_1\_widget = Trigonometry1()

trig\_1\_widget.show()

trig\_1\_widget.\_raise()

trig\_1\_widget.showMaximized()

\end{python}

This method is connected to the trigonometry 1 button and is run when said button is clicked; one of five of the derived lesson menus opens with buttons to connect to the lessons which are all relevant to the title and each other.

\subsubsection{selected\\_t2 method}

\begin{python}

def selected\_t2(self):

trig\_2\_widget = Trigonometry2()

trig\_2\_widget.show()

trig\_2\_widget.\_raise()

trig\_2\_widget.showMaximized()

\end{python}

This method is connected to the trigonometry 2 button and is run when said button is clicked; one of five of the derived lesson menus opens with buttons to connect to the lessons which are all relevant to the title and each other.

\subsubsection{selected\\_pyt method}

\begin{python}

def selected\_pyt(self):

pythagoras\_widget = Pythagoras()

pythagoras\_widget.show()

pythagoras\_widget.\_raise()

pythagoras\_widget.showMaximized()

\end{python}

This method is connected to the pythagoras button and is run when said button is clicked; one of five of the derived lesson menus opens with buttons to connect to the lessons which are all relevant to the title and each other.

\subsubsection{selected\\_pytrig method}

\begin{python}

def selected\_pytrig(self):

pyth\_trig\_widget = PythagTrig()

pyth\_trig\_widget.show()

pyth\_trig\_widget.\_raise()

pyth\_trig\_widget.showMaximized()

\end{python}

This method is connected to the vectors button and is run when said button is clicked; one of five of the derived lesson menus opens with buttons to connect to the lessons which are all relevant to the title and each other (the topic of this section has changed since the connections were coded, hence the unusual variable name).

\subsubsection{selected\\_sum method}

\begin{python}

def selected\_sum(self):

summary\_widget = Summary()

summary\_widget.show()

summary\_widget.\_raise()

summary\_widget.showMaximized()

\end{python}

This method is connected to the summary button and is run when said button is clicked; one of five of the derived lesson menus opens with buttons to connect to the lessons which are all relevant to the title and each other.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This method closes the window and returns the user to the home screen, which is left open for quick accessibility.

\subsection{HomeworkMenuWidget Class}

\begin{python}

class HomeworkMenuWidget(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.title = QLabel()

self.title.setFont(QFont("Courier", 40))

self.ht1 = QPushButton("Trigonometry 1")

self.ht1.setMinimumWidth(90)

self.ht1.setMinimumHeight(110)

self.ht1.setFont(QFont("Courier", 40))

self.ht2 = QPushButton("Trigonometry 2")

self.ht2.setMinimumWidth(90)

self.ht2.setMinimumHeight(110)

self.ht2.setFont(QFont("Courier", 40))

self.hpyt = QPushButton("Pythagoras")

self.hpyt.setMinimumWidth(90)

self.hpyt.setMinimumHeight(110)

self.hpyt.setFont(QFont("Courier", 40))

self.hpytrig = QPushButton("Vectors")

self.hpytrig.setMinimumWidth(90)

self.hpytrig.setMinimumHeight(110)

self.hpytrig.setFont(QFont("Courier", 40))

self.hsum = QPushButton("Summary")

self.hsum.setMinimumWidth(90)

self.hsum.setMinimumHeight(110)

self.hsum.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumWidth(90)

self.back.setMinimumHeight(110)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.ht1\_pic = QLabel()

self.ht1\_pic.setPixmap(QPixmap("homework\_trig\_1\_pic"))

self.ht1\_pic.setAlignment(Qt.AlignCenter)

self.ht2\_pic = QLabel()

self.ht2\_pic.setPixmap(QPixmap("homework\_trig\_2\_pic"))

self.ht2\_pic.setAlignment(Qt.AlignCenter)

self.hpyt\_pic = QLabel()

self.hpyt\_pic.setPixmap(QPixmap("homework\_pythag\_pic"))

self.hpyt\_pic.setAlignment(Qt.AlignCenter)

self.hpytrig\_pic = QLabel()

self.hpytrig\_pic.setPixmap(QPixmap("homework\_vectors\_pic"))

self.hpytrig\_pic.setAlignment(Qt.AlignCenter)

self.hsum\_pic = QLabel()

self.hsum\_pic.setPixmap(QPixmap("homework\_summary\_pic"))

self.hsum\_pic.setAlignment(Qt.AlignCenter)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

self.layout = QGridLayout()

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.ht1\_pic, 2, 0)

self.layout.addWidget(self.ht1, 2, 1)

self.layout.addWidget(self.ht2, 3, 0)

self.layout.addWidget(self.ht2\_pic, 3, 1)

self.layout.addWidget(self.hpyt\_pic, 4, 0)

self.layout.addWidget(self.hpyt, 4, 1)

self.layout.addWidget(self.hpytrig, 5, 0)

self.layout.addWidget(self.hpytrig\_pic, 5, 1)

self.layout.addWidget(self.hsum\_pic, 6, 0)

self.layout.addWidget(self.hsum, 6, 1)

self.layout.addWidget(self.back, 7, 0)

self.\_centralwidget = QWidget()

self.\_centralwidget.setLayout(self.layout)

self.setCentralWidget(self.\_centralwidget)

self.ht1.clicked.connect(self.selected\_ht1)

self.ht2.clicked.connect(self.selected\_ht2)

self.hpyt.clicked.connect(self.selected\_hpyt)

self.hpytrig.clicked.connect(self.selected\_hpytrig)

self.hsum.clicked.connect(self.selected\_hsum)

self.back.clicked.connect(self.selected\_back)

\end{python}

Pretty much the same as the LessonMenuWidget Class (LessonMenuWidget Class)This class is the template for the menu screen which connects to the five individual subclassed menus which contain the buttons for the specific homeworks; it is the middle screen in a branch menu of three screens. It is accessed by clicking the homework button on the home screen (UserAccountHome Class), and it connects to five different final menus from which the user can access the homework. This structure is used to make it easier to navigate to specific homework topics, and to fit clear and visible buttons on the page, which are all relevant to each other.

\subsubsection{selected\\_ht1 method}

\begin{python}

def selected\_ht1(self):

trigonometry\_1\_homework = Trigonometry1HW()

trigonometry\_1\_homework.show()

trigonometry\_1\_homework.\_raise()

trigonometry\_1\_homework.showMaximized()

\end{python}

This method is connected to the trigonometry 1 button and is run when said button is clicked; one of five of the derived homework menus opens with buttons to connect to the homework which are all relevant to the title and each other.

\subsubsection{selected\\_ht2 method}

\begin{python}

def selected\_ht2(self):

trigonometry\_2\_homework = Trigonometry2HW()

trigonometry\_2\_homework.show()

trigonometry\_2\_homework.\_raise()

trigonometry\_2\_homework.showMaximized()

\end{python}

This method is connected to the trigonometry 2 button and is run when said button is clicked; one of five of the derived homework menus opens with buttons to connect to the homework which are all relevant to the title and each other.

\subsubsection{selected\\_hpyt method}

\begin{python}

def selected\_hpyt(self):

pythagoras\_homework = PythagorasHW()

pythagoras\_homework.show()

pythagoras\_homework.\_raise()

pythagoras\_homework.showMaximized()

\end{python}

This method is connected to the pythagoras button and is run when said button is clicked; one of five of the derived homework menus opens with buttons to connect to the homework which are all relevant to the title and each other.

\subsubsection{selected\\_hpytrig method}

\begin{python}

def selected\_hpytrig(self):

pythag\_trig\_homework = PythagTrigonometryHW()

pythag\_trig\_homework.show()

pythag\_trig\_homework.\_raise()

pythag\_trig\_homework.showMaximized()

\end{python}

This method is connected to the vectors button and is run when said button is clicked; one of five of the derived homework menus opens with buttons to connect to the homework which are all relevant to the title and each other.

\subsubsection{selected\\_hsum method}

\begin{python}

def selected\_hsum(self):

summary\_homework = SummaryHW()

summary\_homework.show()

summary\_homework.\_raise()

summary\_homework.showMaximized()

\end{python}

This method is connected to the summary button and is run when said button is clicked; one of five of the derived homework menus opens with buttons to connect to the homework which are all relevant to the title and each other.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This method closes the window and returns the user to the home screen, which is left open for quick accessibility.

\subsection{ParentLessonMenu Class}

\begin{python}

class ParentLessonMenu(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.title = QLabel()

self.button\_1 = QPushButton()

self.button\_1.setMinimumHeight(110)

self.button\_1.setMinimumWidth(60)

self.button\_1.setFont(QFont("Courier", 40))

self.button\_2 = QPushButton()

self.button\_2.setMinimumHeight(110)

self.button\_2.setMinimumWidth(60)

self.button\_2.setFont(QFont("Courier", 40))

self.button\_3 = QPushButton()

self.button\_3.setMinimumHeight(110)

self.button\_3.setMinimumWidth(60)

self.button\_3.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumHeight(100)

self.back.setMinimumWidth(60)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue}")

self.layout = QGridLayout()

self.setLayout(self.layout)

self.layout.addWidget(self.back, 3, 0)

self.back.clicked.connect(self.selected\_back)

\end{python}

This is the parent class which provides all of the default attributes for the derived lesson menus (e.g. Trigonometry1(ParentLessonMenu), Trigonometry1 Class). The derived lesson menu file imports from this file, and the five subclasses inherit these attributes, along with some polymorphism to make them different from one another. These subclasses are connected by buttons in the previous menu (LessonMenuWidget Class) and contain buttons to the individual lessons; there are three buttons made in the parent class here, then one, two or three of them can be added to the layout in each subclass depending on how many lessons there are. All of the connections, along with methods to open the corresponding window, are in the subclasses so that the three buttons can be over-ridden to connect to various lessons.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This will close the window and the user will be returned to the previous menu (LessonMenuWidget Class) if they want to change topic.

\subsection{ParentHomeworkMenuClass Class}

\begin{python}

class ParentHomeworkMenuClass(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.title = QLabel()

self.button\_1 = QPushButton()

self.button\_1.setMinimumHeight(110)

self.button\_1.setMinimumWidth(60)

self.button\_1.setFont(QFont("Courier", 40))

self.button\_2 = QPushButton()

self.button\_2.setMinimumHeight(110)

self.button\_2.setMinimumWidth(60)

self.button\_2.setFont(QFont("Courier", 40))

self.button\_3 = QPushButton()

self.button\_3.setMinimumHeight(110)

self.button\_3.setMinimumWidth(60)

self.button\_3.setFont(QFont("Courier", 40))

self.button\_4 = QPushButton()

self.button\_4.setMinimumHeight(110)

self.button\_4.setMinimumWidth(60)

self.button\_4.setFont(QFont("Courier", 40))

self.button\_5 = QPushButton()

self.button\_5.setMinimumHeight(110)

self.button\_5.setMinimumWidth(60)

self.button\_5.setFont(QFont("Courier", 40))

self.button\_6 = QPushButton()

self.button\_6.setMinimumHeight(110)

self.button\_6.setMinimumWidth(60)

self.button\_6.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumHeight(100)

self.back.setMinimumWidth(60)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white;}")

self.pic\_1 = QLabel()

self.pic\_1.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setAlignment(Qt.AlignCenter)

self.pic\_3 = QLabel()

self.pic\_3.setAlignment(Qt.AlignCenter)

self.pic\_4 = QLabel()

self.pic\_4.setAlignment(Qt.AlignCenter)

self.pic\_5 = QLabel()

self.pic\_5.setAlignment(Qt.AlignCenter)

self.pic\_6 = QLabel()

self.pic\_6.setAlignment(Qt.AlignCenter)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue}")

self.layout = QGridLayout()

self.setLayout(self.layout)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.back, 10, 0)

self.back.clicked.connect(self.selected\_back)

\end{python}

This is the parent class which provides all of the default attributes for the derived homework menus (e.g. Trigonometry1HW(ParentHomeworkMenuClass), Trigonometry1HW Class). The derived homework menu file imports from this file, and the five subclasses inherit these attributes, along with some polymorphism to make them different from one another. These subclasses are connected by buttons in the previous menu (HomeworkMenuWidget Class) and contain buttons to the individual lessons; there are six buttons made in the parent class here, then three or siz of them can be added to the layout in each subclass depending on how many lessons there are. All of the connections, along with methods to open the corresponding window, are in the subclasses so that the three buttons can be over-ridden to connect to various homework.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.close()

\end{python}

This will close the window and the user will be returned to the previous menu (HomeworkMenuWidget Class) if they want to change topic.

\subsection{Trigonometry1(ParentLessonMenu) Class}

\begin{python}

class Trigonometry1(ParentLessonMenu):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.button\_1.setText("Sides")

self.button\_2.setText("SOHCAHTOA")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("trig\_1\_pic"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("trig\_1\_pic\_2"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_2, 1, 1)

self.layout.addWidget(self.button\_1, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.button\_1.clicked.connect(self.SidesAHO)

self.button\_2.clicked.connect(self.SOHCAHTOA)

\end{python}

This class is one of the five subclasses which inherit from the ParentLessonMenu Class (ParentLessonMenu Class) and are the third in a branch consisting of three stages of screens; home screen, topic menu and derived menu. Most of the classes attributes are in the parent class, but the button names, connections and methods are coded here to over-ride and allow for difference between the five subclasses. Each button (except for the return button) connects to a stack widget which contains the first and second screen of each lesson in a stack together. Each of the five subclasses are essentially the same, and serve the same purpose; their only differences are the lessons they connect to, hence the branch system.

\subsubsection{SidesAHO method}

\begin{python}

def SidesAHO(self):

sides\_aho = Trig1StackSides()

sides\_aho.show()

sides\_aho.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the sides lesson.

\subsubsection{SOHCAHTOA method}

\begin{python}

def SOHCAHTOA(self):

sohcahtoa = Trig1StackSOHCAHTOA()

sohcahtoa.show()

sohcahtoa.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the SOHCAHTOA lesson.

\subsection{Trigonometry1HW(ParentHomeworkMenuClass) Class}

\begin{python}

class Trigonometry1HW(ParentHomeworkMenuClass):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title.setPixmap(QPixmap("trig\_1\_title"))

self.button\_1.setText("Sides Easy")

self.button\_2.setText("Sides Medium")

self.button\_3.setText("Sides Hard")

self.button\_4.setText("SOHCAHTOA Easy")

self.button\_5.setText("SOHCAHTOA Medium")

self.button\_6.setText("SOHCAHTOA Hard")

self.pic\_1.setPixmap(QPixmap("trig\_1\_pic\_1\_h"))

self.pic\_2.setPixmap(QPixmap("trig\_1\_pic\_2\_h"))

self.pic\_3.setPixmap(QPixmap("trig\_1\_pic\_3\_h"))

self.pic\_4.setPixmap(QPixmap("trig\_1\_pic\_4\_h"))

self.pic\_5.setPixmap(QPixmap("trig\_1\_pic\_5\_h"))

self.pic\_6.setPixmap(QPixmap("trig\_1\_pic\_6\_h"))

self.layout.addWidget(self.button\_1, 1, 0)

self.layout.addWidget(self.pic\_1, 1, 1)

self.layout.addWidget(self.pic\_2, 2, 0)

self.layout.addWidget(self.button\_2, 2, 1)

self.layout.addWidget(self.button\_3, 3, 0)

self.layout.addWidget(self.pic\_3, 3, 1)

self.layout.addWidget(self.pic\_4, 4, 0)

self.layout.addWidget(self.button\_4, 4, 1)

self.layout.addWidget(self.button\_5, 5, 0)

self.layout.addWidget(self.pic\_5, 5, 1)

self.layout.addWidget(self.pic\_6, 6, 0)

self.layout.addWidget(self.button\_6, 6, 1)

self.button\_1.clicked.connect(self.sides\_aho\_easy)

self.button\_2.clicked.connect(self.sides\_aho\_medium)

self.button\_3.clicked.connect(self.sides\_aho\_hard)

self.button\_4.clicked.connect(self.sohcahtoa\_easy)

self.button\_5.clicked.connect(self.sohcahtoa\_medium)

self.button\_6.clicked.connect(self.sohcahtoa\_hard)

\end{python}

Essentially serving the same purpise as the Trigonometry1(ParentLessonMenuClass), only it connects to the homeworks from the homework button on the home screen, this class is one of the five subclasses which inherit from the ParentHomeworkMenuClass Class (ParentHomeworkMenuClass Class) and are the third in a branch consisting of three stages of screens; home screen, topic menu and derived menu. Most of the classes attributes are in the parent class, but the button names, connections and methods are coded here to over-ride and allow for difference between the five subclasses. Each button (except for the return button) connects to a stack widget which contains the first and second screen of each homework in a stack together. Each of the five subclasses are essentially the same, and serve the same purpose; their only differences are the homework they connect to, hence the branch system.

\subsubsection{sides\\_aho\\_easy method}

\begin{python}

def sides\_aho\_easy(self):

sides\_aho\_1 = Trig1StackSidesEasy()

sides\_aho\_1.show()

sides\_aho\_1.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the sides easy homework.

\subsubsection{sides\\_aho\\_medium method}

\begin{python}

def sides\_aho\_medium(self):

sides\_aho\_2 = Trig1StackSidesMedium()

sides\_aho\_2.show()

sides\_aho\_2.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the sides medium homework.

\subsubsection{sides\\_aho\\_hard method}

\begin{python}

def sides\_aho\_hard(self):

sides\_aho\_3 = Trig1StackSidesHard()

sides\_aho\_3.show()

sides\_aho\_3.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the sides hard homework.

\subsubsection{sohcahtoa\\_easy method}

\begin{python}

def sohcahtoa\_easy(self):

sohcahtoa\_1 = Trig1StackSOHCAHTOAEasy()

sohcahtoa\_1.show()

sohcahtoa\_1.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the SOHCAHTOA easy homework.

\subsubsection{sohcahtoa\\_medium method}

\begin{python}

def sohcahtoa\_medium(self):

sohcahtoa\_2 = Trig1StackSOHCAHTOAMedium()

sohcahtoa\_2.show()

sohcahtoa\_2.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the SOHCAHTOA medium homework.

\subsubsection{sohcahtoa\\_hard method}

\begin{python}

def sohcahtoa\_hard(self):

sohcahtoa\_3 = Trig1StackSOHCAHTOAHard()

sohcahtoa\_3.show()

sohcahtoa\_3.\_raise()

\end{python}

This connects to the stack widget which contains both the first and second screen of the SOHCAHTOA hard homework.

\subsection{Trig1StackSides Class}

\begin{python}

class Trig1StackSides(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SidesAHOWidget(self)

self.second\_widget = SidesAHOWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

This is one of twelve stack classes which each contain the two screens for one lesson. This stack widget has the two screens added to the layout (imported from the files where each widget's class is coded), then when the stack is opened the first screen is displayed, and can switch to the second in the same window. This is accessed by clicking the corrersponding button in the first derived lesson menu (Trigonometry1(ParentLessonMenu) Class). The contents of both screens will be left how the user leaves them until the entire stack is closed, even if they are switching between screens.

\subsection{ParentLessonLayout Class}

\begin{python}

class ParentLessonLayout(QWidget):

def \_\_init\_\_(self, parent = None):

super().\_\_init\_\_()

self.parent = parent

self.title = QLabel()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.back = QPushButton("Return")

self.back.setMinimumHeight(50)

self.back.setMinimumWidth(60)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.next = QPushButton("Next")

self.next.setMinimumHeight(50)

self.next.setMinimumWidth(60)

self.next.setFont(QFont("Courier", 40))

self.lesson\_1 = QTextEdit()

self.lesson\_1.setMinimumHeight(400)

self.lesson\_1.setMinimumWidth(80)

self.lesson\_1.setFont(QFont("Courier", 20))

self.lesson\_1.setReadOnly(True)

self.lesson\_2 = QTextEdit()

self.lesson\_2.setMinimumHeight(400)

self.lesson\_2.setMinimumWidth(80)

self.lesson\_2.setFont(QFont("Courier", 20))

self.lesson\_2.setReadOnly(True)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

self.layout = QGridLayout()

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.lesson\_1, 1, 0)

self.layout.addWidget(self.lesson\_2, 1, 1)

self.layout.addWidget(self.back, 3, 0)

self.layout.addWidget(self.next, 3, 1)

self.setLayout(self.layout)

self.back.clicked.connect(self.selected\_back)

self.next.clicked.connect(self.selected\_next\_page)

\end{python}

This is the parent class which provides the default attributes to each of the twelve first lesson screens. The file with the subclasses imports from here, and this parent class is never seen by the user; only the subclasses are accessible. This contains the attributes which are shared by all subclasses, while the next page connection method is in each subclass, in order to switch to the relevant second page. The background colour and screen maximising is coded here so all subclasses will have the same background colour and will be maximised. Each subclass is placed in a stack with its corresponding second screen so that the two pages are always the right ones and relevant to each other.

\subsubsection{selected\\_back method}

\begin{python}

def selected\_back(self):

self.parent.close()

\end{python}

This method closes the stack window and returns the user to the home screen.

\subsection{ParentLessonPage2 Class}

\begin{python}

class ParentLessonPage2(QWidget):

def \_\_init\_\_(self, parent = None):

super().\_\_init\_\_()

self.parent = parent

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.answer = QLineEdit()

self.answer.setMinimumWidth(80)

self.answer.setMinimumHeight(110)

self.answer.setFont(QFont("Courier", 40))

self.previous = QPushButton("Previous")

self.previous.setMinimumHeight(110)

self.previous.setMinimumWidth(60)

self.previous.setFont(QFont("Courier", 40))

self.previous.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.check = QPushButton("Check Answer")

self.check.setMinimumHeight(110)

self.check.setMinimumWidth(60)

self.check.setFont(QFont("Courier", 40))

self.check.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.finish = QPushButton("Finish")

self.finish.setMinimumHeight(110)

self.finish.setMinimumWidth(60)

self.finish.setFont(QFont("Courier", 40))

self.finish.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

self.text\_1 = QTextEdit()

self.text\_1.setMinimumWidth(80)

self.text\_1.setMinimumHeight(110)

self.text\_1.setFont(QFont("Courier", 20))

self.text\_1.setReadOnly(True)

self.text\_2 = QTextEdit()

self.text\_2.setMinimumWidth(80)

self.text\_2.setMinimumHeight(110)

self.text\_2.setFont(QFont("Courier", 20))

self.text\_2.setReadOnly(True)

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

self.layout = QGridLayout()

self.setLayout(self.layout)

self.layout.addWidget(self.text\_1, 0, 0)

self.layout.addWidget(self.text\_2, 0, 1)

self.layout.addWidget(self.previous, 3, 0)

self.layout.addWidget(self.answer, 3, 1)

self.layout.addWidget(self.finish, 4, 0)

self.layout.addWidget(self.check, 4, 1)

self.previous.clicked.connect(self.previous\_selected)

self.check.clicked.connect(self.check\_selected)

self.finish.clicked.connect(self.finish\_selected)

\end{python}

This is the parent class which provides the default attributes to each of the twelve second lesson screens. The file with the subclasses imports from here, and this parent class is never seen by the user; only the subclasses are accessible. This contains the attributes which are shared by all subclasses, while the previous page connection method is in each subclass, in order to switch to the relevant first page. The background colour and screen maximising is coded here so all subclasses will have the same background colour and will be maximised. Each subclass is placed in a stack with its corresponding first screen so that the two pages are always the right ones and relevant to each other.

\subsubsection{previous\\_selected method}

\begin{python}

def previous\_selected(self):

self.parent.stack.setCurrentIndex(0)

\end{python}

This method connects to the first screen in the stack widget which this screen is also in (Trig1StackSides class) - this method does not need to be over-ridden because the variable which is the first screen in the stack index is declared in the stack itself, which was connected to from the derived lesson menu (Trigonometry1(ParentLessonMenu) Class).

\subsubsection{check\\_selected method}

\begin{python}

def check\_selected(self):

if self.answer.text() == self.answer\_lesson:

self.answer.setText("{0} Correct".format(self.answer\_lesson))

else:

self.answer.setText("Incorrect")

self.answer.setReadOnly(True)

self.check.setEnabled(False)

\end{python}

This method checks the input from the user up against the hard-coded variable which is set in each subclass - the variable is passed in from the subclass so the method can use entirely the same code for each subclass, hence why it is in the parent class.

\subsubsection{finish\\_selected method}

\begin{python}

def finish\_selected(self):

self.parent.close()

\end{python}

This just closes the stack and returns the user to the home screen, so it can be in the parent class too.

\subsection{SidesAHOWidget(ParentLessonLayout) Class}

\begin{python}

class SidesAHOWidget(ParentLessonLayout):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent = parent

self.title.setPixmap(QPixmap("sides\_lesson\_title"))

self.lesson\_1.setText("Every triangle has 3 sides, and each side has a name.\nThe HYPOTENUSE is the longest side, and is always oppposite the right-angle of a triangle.\nThe length can be found using Pythagoras' Theorem of a\u00b2 + b\u00b2 = c\u00b2.\nSine function: sin(x) = Opposite {0} Hypotenuse".format(chr(247)))

self.lesson\_2.setText("The OPPOSITE is the side opposite the angle being used.\nCosine function: cos(x) = Adjacent {0} Hypotenuse".format(chr(247)))

\end{python}

This is one of the twelve first lesson screens which inherits from the parent lesson page 1 class (ParentLessonLayout Class) and is placed in a stack widget (Trig1StackSides Class) with the corresponding second sides lesson screen. This way it is not possible for only one of the two lesson screens to be open at any time, as they are in the same stack which keeps changes until it is closed entirely.

\subsubsection{selected\\_next\\_page method}

\begin{python}

def selected\_next\_page(self):

self.parent.stack.setCurrentIndex(1)

\end{python}

This method switches to the second screen which shares the stack widget with this widget (Trig1StackSides Class).

\subsection{SidesAHOWidgetPage2(ParentLessonPage2) Class}

\begin{python}

class SOHCAHTOAWidgetPage2(ParentLessonPage2):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent = parent

self.answer.setText("m")

self.text\_1.setText("Example 1:\n1. Label O, A, H\n2. Write down SOHCAHTOA\n3. Two sides are involved: O,H\n4. So use O {0} S x H\n5. We want to find H so cover it up to leave H = (O {0} S(0))\n6. Translate: Press 15 {0} SIN(35) = 26.151702, so ans = 26.2m\n7. Check it's sensible: Yes, it's about twice as big as 15, as the diagram suggests.".format(chr(247)))

self.text\_1.setMinimumHeight(380)

self.text\_2.setText("You have to figure out yourself which formula to use to find this answer.\nHere's a hint: cut the triangle down the middle and it becomes a right-angled triangle.\n \n \n \n \n \n \nPut your answer in the box below:")

self.text\_2.setMinimumHeight(380)

self.pic = QLabel()

self.pic.setPixmap(QPixmap("sohcahtoa\_lesson\_pic\_2.png"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("sohcahtoa\_lesson\_pic\_3.png"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.pic, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.answer\_lesson = "26.5m"

\end{python}

This class is one of the twelve second lesson screens which inherits from the lesson page 2 parent class (ParentLessonPage2 Class) and is placed in a stack widget (Trig1StackSOHCAHTOA Class) with the corresponding first sides lesson screen. This way it is not possible for only one of the two lesson screens to be open at any time, as they are in the same stack which keeps changes until it is closed entirely.

\subsection{Trig1StackSidesEasy Class}

\begin{python}

class Trig1StackSidesEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SidesAHOEasyWidget(self)

self.second\_widget = SidesAHOEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

This is one of twenty-four stack classes which each contain the two screens for one homework. This stack widget has the two screens added to the layout (imported from the files where each widget's class is coded), then when the stack is opened the first screen is displayed, and can switch to the second in the same window. This is accessed by clicking the corrersponding button in the first derived homework menu (Trigonometry1HW(ParentHomeworkMenuClass) Class). It is useful to have the two homework screens in a stack together so the user cannot close one screen and skip to the next, or vice-versa and repeat a page, potentially crashing the program. The use can return to one page, but the second will still be there, and they cannot re-submit the score of the first page until the second page has been closed and the stack is closed entirely and they re-do the whole homework. The contents of both screens will be left how the user leaves them until the entire stack is closed, even if they are switching between screens.

\subsection{ParentHomeworkPage1Class Class}

\begin{python}

class ParentHomeworkPage1Class(QWidget):

def \_\_init\_\_(self, parent = None):

super().\_\_init\_\_()

self.parent = parent

self.showMaximized()

self.task = ""

self.allow\_cont = False

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.next = QPushButton("Next")

self.next.setMinimumHeight(110)

self.next.setMinimumWidth(60)

self.next.setFont(QFont("Courier", 40))

self.cancel = QPushButton("Cancel")

self.cancel.setMinimumHeight(110)

self.cancel.setMinimumWidth(60)

self.cancel.setFont(QFont("Courier", 40))

self.cancel.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.check = QPushButton("Check Answers")

self.check.setMinimumHeight(110)

self.check.setMinimumWidth(60)

self.check.setFont(QFont("Courier", 40))

self.check.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.reset = QPushButton("Reset Answers")

self.reset.setMinimumHeight(110)

self.reset.setMinimumWidth(60)

self.reset.setFont(QFont("Courier", 40))

self.reset.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.title = QLabel()

self.title.setFont(QFont("Courier", 40))

self.question\_1 = QLabel()

self.question\_1.setFont(QFont("Courier", 20))

self.question\_1\_shape = QLabel()

self.question\_1\_shape.setFont(QFont("Courier", 40))

self.answer\_a = QLineEdit()

self.answer\_a.setMinimumHeight(70)

self.answer\_a.setMinimumWidth(60)

self.answer\_a.setFont(QFont("Courier", 30))

self.answer\_b = QLineEdit()

self.answer\_b.setMinimumHeight(70)

self.answer\_b.setMinimumWidth(60)

self.answer\_b.setFont(QFont("Courier", 30))

self.answer\_c = QLineEdit()

self.answer\_c.setMinimumHeight(70)

self.answer\_c.setMinimumWidth(60)

self.answer\_c.setFont(QFont("Courier", 30))

self.answer\_d = QLineEdit()

self.answer\_d.setMinimumHeight(70)

self.answer\_d.setMinimumWidth(60)

self.answer\_d.setFont(QFont("Courier", 30))

self.answer\_e = QLineEdit()

self.answer\_e.setMinimumHeight(70)

self.answer\_e.setMinimumWidth(60)

self.answer\_e.setFont(QFont("Courier", 30))

self.answer\_f = QLineEdit()

self.answer\_f.setMinimumHeight(70)

self.answer\_f.setMinimumWidth(60)

self.answer\_f.setFont(QFont("Courier", 30))

self.q1a = QLabel("")

self.q1a.setFont(QFont("Courier", 20))

self.q1b = QLabel("")

self.q1b.setFont(QFont("Courier", 20))

self.q1c = QLabel("")

self.q1c.setFont(QFont("Courier", 20))

self.q1d = QLabel("")

self.q1d.setFont(QFont("Courier", 20))

self.q1e = QLabel("")

self.q1e.setFont(QFont("Courier", 20))

self.q1f = QLabel("")

self.q1f.setFont(QFont("Courier", 20))

self.score\_box = QLabel("Score: X/X")

self.score\_box.setFont(QFont("Courier", 30))

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

self.layout = QGridLayout()

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.question\_1, 1, 0)

self.layout.addWidget(self.q1a, 1, 1)

self.layout.addWidget(self.reset, 1, 2)

self.layout.addWidget(self.question\_1\_shape, 2, 0)

self.layout.addWidget(self.answer\_a, 2, 1)

self.layout.addWidget(self.q1b, 3, 1)

self.layout.addWidget(self.answer\_b, 4, 1)

self.layout.addWidget(self.q1c, 5, 1)

self.layout.addWidget(self.answer\_c, 6, 1)

self.layout.addWidget(self.q1d, 7, 1)

self.layout.addWidget(self.answer\_d, 8, 1)

self.layout.addWidget(self.q1e, 9, 1)

self.layout.addWidget(self.answer\_e, 10, 1)

self.layout.addWidget(self.q1f, 11, 1)

self.layout.addWidget(self.answer\_f, 12, 1)

self.layout.addWidget(self.cancel, 13, 0)

self.layout.addWidget(self.check, 13, 1)

self.layout.addWidget(self.next, 13, 2)

self.setLayout(self.layout)

self.check.clicked.connect(self.check\_selected)

self.reset.clicked.connect(self.reset\_selected)

self.cancel.clicked.connect(self.cancel\_selected)

self.next.clicked.connect(self.next\_selected)

self.answers = []

self.answers.append(self.answer\_a)

self.answers.append(self.answer\_b)

self.answers.append(self.answer\_c)

self.answers.append(self.answer\_d)

self.answers.append(self.answer\_e)

self.answers.append(self.answer\_f)

\end{python}

This is the parent class which provides the default attributes to each of the twenty-four first homework screens. The file with the subclasses imports from here, and this parent class is never seen by the user; only the subclasses are accessible. This contains the attributes which are shared by all subclasses, while the next page connection method is in each subclass, in order to switch to the relevant second page. The background colour and screen maximising is coded here so all subclasses will have the same background colour and will be maximised. Each subclass is placed in a stack with its corresponding second screen so that the two pages are always the right ones and relevant to each other.

\subsubsection{check\\_selected method}

\begin{python}

def check\_selected(self):

self.allow\_cont = False

self.correct\_count = 0

if self.answer\_a.text() == self.answer\_1\_a:

self.answer\_a.setText("{0} Correct".format(self.answer\_a.text()))

self.correct\_count += 1

else:

self.answer\_a.setText("Incorrect")

if self.answer\_b.text() == self.answer\_1\_b:

self.answer\_b.setText("{0} Correct".format(self.answer\_b.text()))

self.correct\_count += 1

else:

self.answer\_b.setText("Incorrect")

if self.answer\_c.text() == self.answer\_1\_c:

self.answer\_c.setText("{0} Correct".format(self.answer\_c.text()))

self.correct\_count += 1

else:

self.answer\_c.setText("Incorrect")

if self.answer\_d.text() == self.answer\_1\_d:

self.answer\_d.setText("{0} Correct".format(self.answer\_d.text()))

self.correct\_count += 1

else:

self.answer\_d.setText("Incorrect")

if self.answer\_e.text() == self.answer\_1\_e:

self.answer\_e.setText("{0} Correct".format(self.answer\_e.text()))

self.correct\_count += 1

else:

self.answer\_e.setText("Incorrect")

if self.answer\_f.text() == self.answer\_1\_f:

self.answer\_f.setText("{0} Correct".format(self.answer\_f.text()))

self.correct\_count += 1

else:

self.answer\_f.setText("Incorrect")

for a in self.answers:

a.setReadOnly(True)

self.check.setEnabled(False)

self.reset.setEnabled(False)

self.allow\_cont = True

\end{python}

This method uses variables which are passed in from each subclass, whichever one it is being called from, which are checked against hard-coded answers which are declared in the subclass too, so the variable name can be the same in the method allowing it to be written once in the parent class. The purpose of this method is to check the inputs of each of the six line edits in the subclass being used, then tell the user whether or not they are correct based on the hard-coded answers in the subclasses.

\subsubsection{next\\_selected method}

\begin{python}

def next\_selected(self):

cont = False

while not cont:

for a in self.answers:

if a.text() == "":

error\_message = ErrorMessage8()

error\_message.show()

error\_message.\_raise()

cont = False

cont = True

if self.allow\_cont:

g\_database.insert\_data\_first(self.task, self.correct\_count)

self.open\_page\_2()

self.hide()

else:

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

\end{python}

This checks to see if the question has been fully answered before allowing the user to continue to the next screen and saving the first stage of the record in the database. this can be in the parent class because, again, the variables are hard-coded and passed through from the subclass, for the database method.

\subsubsection{reset\\_selected method}

\begin{python}

def reset\_selected(self):

self.answer\_a.setText(None)

self.answer\_b.setText(None)

self.answer\_c.setText(None)

self.answer\_d.setText(None)

self.answer\_e.setText(None)

self.answer\_f.setText(None)

\end{python}

this can be in the parent class because it doesn't matter what the values in the line edits are, it serves the same purpose of removing them.

\subsubsection{cancel\\_selected method}

\begin{python}

def cancel\_selected(self):

self.parent.close()

\end{python}

This closes the entire stack regardless of which stack it is, hence why this method can be in the parent class.

\subsubsection{open\\_page\\_2 method}

\begin{python}

def open\_page\_2(self):

self.parent.stack.setCurrentIndex(1)

\end{python}

The screen represented by the index of the stack is deternmined by what is hard-coded in the stack widget (Trig1SidesEasyStack Class) so the variable name is always the same, switching to the screen hard-coded into the stack, so this method can also be in the parent class.

\subsection{HomeworkPage2ParentClass Class}

\begin{python}

class HomeworkPage2ParentClass(QWidget):

def \_\_init\_\_(self, parent = None):

super().\_\_init\_\_()

self.parent = parent

self.task = ""

self.showMaximized()

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.\_button\_1 = QPushButton()

self.\_button\_1.setMaximumWidth(200)

self.\_button\_1.setMinimumWidth(110)

self.\_button\_1.setMinimumHeight(110)

self.\_button\_1.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_2 = QPushButton()

self.\_button\_2.setMaximumWidth(200)

self.\_button\_2.setMinimumWidth(110)

self.\_button\_2.setMinimumHeight(110)

self.\_button\_2.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_3 = QPushButton()

self.\_button\_3.setMaximumWidth(200)

self.\_button\_3.setMinimumWidth(110)

self.\_button\_3.setMinimumHeight(110)

self.\_button\_3.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_4 = QPushButton()

self.\_button\_4.setMaximumWidth(200)

self.\_button\_4.setMinimumWidth(110)

self.\_button\_4.setMinimumHeight(110)

self.\_button\_4.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_5 = QPushButton()

self.\_button\_5.setMaximumWidth(200)

self.\_button\_5.setMinimumWidth(110)

self.\_button\_5.setMinimumHeight(110)

self.\_button\_5.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_6 = QPushButton()

self.\_button\_6.setMaximumWidth(200)

self.\_button\_6.setMinimumWidth(110)

self.\_button\_6.setMinimumHeight(110)

self.\_button\_6.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.question\_2 = QLabel()

self.question\_2.setFont(QFont("Courier", 30))

self.shape\_2 = QLabel()

self.shape\_2.setFont(QFont("Courier", 30))

self.question\_3 = QLabel()

self.question\_3.setFont(QFont("Courier", 30))

self.shape\_3 = QLabel()

self.shape\_3.setFont(QFont("Courier", 30))

self.question\_4 = QLabel()

self.question\_4.setFont(QFont("Courier", 30))

self.answer\_2 = QComboBox()

self.answer\_2.setMinimumWidth(60)

self.answer\_2.setMinimumHeight(110)

self.answer\_2.setFont(QFont("Courier", 40))

self.answer\_2.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.mark\_2 = QPushButton("Mark it | 2")

self.mark\_2.setMinimumWidth(60)

self.mark\_2.setMinimumHeight(110)

self.mark\_2.setFont(QFont("Courier", 40))

self.mark\_2.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.answer\_3 = QComboBox()

self.answer\_3.setMinimumWidth(60)

self.answer\_3.setMinimumHeight(110)

self.answer\_3.setFont(QFont("Courier", 40))

self.answer\_3.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.mark\_3 = QPushButton("Mark it | 2")

self.mark\_3.setMinimumWidth(60)

self.mark\_3.setMinimumHeight(110)

self.mark\_3.setFont(QFont("Courier", 40))

self.mark\_3.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.previous = QPushButton("Previous")

self.previous.setMinimumWidth(60)

self.previous.setMinimumHeight(110)

self.previous.setFont(QFont("Courier", 40))

self.previous.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.finish = QPushButton("Finish")

self.finish.setMinimumWidth(60)

self.finish.setMinimumHeight(110)

self.finish.setFont(QFont("Courier", 40))

self.finish.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

self.attempts\_button = QPushButton("3 Attempts left")

self.attempts\_button.setMinimumHeight(60)

self.attempts\_button.setMinimumWidth(90)

self.attempts\_button.setMaximumWidth(200)

self.attempts\_button.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.attempts\_button.setEnabled(False)

self.layout = QGridLayout()

self.layout.addWidget(self.question\_2, 0, 0)

self.layout.addWidget(self.shape\_2, 0, 1)

self.layout.addWidget(self.answer\_2, 1, 0)

self.layout.addWidget(self.mark\_2, 1, 1)

self.layout.addWidget(self.\_button\_1, 1, 2)

self.layout.addWidget(self.\_button\_2, 1, 3)

self.layout.addWidget(self.\_button\_3, 2, 2)

self.layout.addWidget(self.question\_3, 2, 0)

self.layout.addWidget(self.\_button\_4, 2, 3)

self.layout.addWidget(self.\_button\_5, 3, 2)

self.layout.addWidget(self.\_button\_6, 3, 3)

self.layout.addWidget(self.shape\_3, 2, 1)

self.layout.addWidget(self.attempts\_button, 4, 2)

self.layout.addWidget(self.answer\_3, 3, 0)

self.layout.addWidget(self.mark\_3, 3, 1)

self.layout.addWidget(self.question\_4, 0, 2)

self.layout.addWidget(self.previous, 5, 0)

self.layout.addWidget(self.finish, 5, 3)

self.setLayout(self.layout)

self.mark\_2.clicked.connect(self.selected\_mark\_2)

self.mark\_3.clicked.connect(self.selected\_mark\_3)

self.previous.clicked.connect(self.selected\_previous)

self.finish.clicked.connect(self.selected\_finish)

self.\_button\_1.clicked.connect(self.check\_button\_1)

self.\_button\_2.clicked.connect(self.check\_button\_2)

self.\_button\_3.clicked.connect(self.check\_button\_3)

self.\_button\_4.clicked.connect(self.check\_button\_4)

self.\_button\_5.clicked.connect(self.check\_button\_5)

self.\_button\_6.clicked.connect(self.check\_button\_6)

self.attempts\_remaining\_a = 2

self.attempts\_remaining\_b = 2

self.attempts\_remaining\_c = 3

self.correct\_count\_2 = 0

self.correct\_count\_3 = 0

self.correct\_count\_4 = 0

self.answer\_question\_4 = None

\end{python}

This is the parent class which provides the default attributes to each of the twenty-four second homework screens. The file with the subclasses imports from here, and this parent class is never seen by the user; only the subclasses are accessible. This contains the attributes which are shared by all subclasses, while the previous page connection method is in each subclass, in order to switch to the relevant first page. The background colour and screen maximising is coded here so all subclasses will have the same background colour and will be maximised. Each subclass is placed in a stack with its corresponding first screen so that the two pages are always the right ones and relevant to each other.

\subsubsection{check\\_button\\_1 method}

\begin{python}

def check\_button\_1(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_1.text() == self.answer\_question\_4:

self.\_button\_1.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_1.setText("Incorrect")

self.\_button\_1.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

\end{python}

This method checks the text on the button which was clicked to see if it matches the hard-coded answer in each subclass; the variable names of the buttons are the same, in the parent class, only the text is over-ridden, so these methods can be in the parent class as the variables being checked are the same, only their values which are passed through are different in each subclass.

\subsubsection{selected\\_mark\\_2 method}

\begin{python}

def selected\_mark\_2(self, attempts\_remaining\_a):

self.correct\_count\_2 = 0

if self.answer\_2.currentText() == "20":

self.correct\_count\_2 += 1

self.mark\_2.setText("Correct!")

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

else:

self.attempts\_remaining\_a -= 1

self.mark\_2.setText("Mark it|{0}".format(self.attempts\_remaining\_a))

if self.attempts\_remaining\_a == 0:

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

error\_message = ErrorMessage5()

error\_message.show()

error\_message.\_raise()

return self.attempts\_remaining\_a, self.correct\_count\_2

\end{python}

As with the check\\_button\\_1 method, the variable names of the combo boxes are the same, created in the parent class, and it is the values of the combo boxes whiuch are passed through, so the checking methods can be the same written only once in the parent class.

\subsubsection{selected\\_previous method}

\begin{python}

def selected\_previous(self):

self.parent.stack.setCurrentIndex(0)

\end{python}

The screen being represented by the index here is hard-coded in the stack which is currently being operated, so this method can use the same code for each stack.

\subsubsection{selected\\_finish method}

\begin{python}

def selected\_finish(self):

if self.attempts\_button.text() != "1 mark!" and self.attempts\_button.text() != "No more attempts":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_2.text() != "Correct!" and self.mark\_2.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_3.text() != "Correct!" and self.mark\_3.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

else:

g\_database.insert\_data\_second(self.task, self.correct\_count\_2, self.correct\_count\_3, self.correct\_count\_4)

self.parent.close()

\end{python}

This method's purpose is to check to see if the questions have been answered before saving the scores to the database and returning the user to the home screen; again, the variable names of the widgets being checked are the same and written in the parent class, only the contents of the inputs are being checked, making this method suitable for the parent class.

\subsection{SidesAHOEasyWidget(ParentHomeworkPage1Class) Class}

\begin{python}

class SidesAHOEasyWidget(ParentHomeworkPage1Class):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent = parent

self.task = "Sides Easy"

self.title.setText("Sides Easy")

self.question\_1.setText("Question 1: Look at the diagram below\nand answer the following questions: ")

self.q1a.setText("Which side is oppopsite angle A? ")

self.q1b.setText("Which side is adjacent to angle Q? ")

self.q1c.setText("Which side is the hypotenuse? ")

self.q1d.setText("Which formula would you use to find AB? ")

self.question\_1\_shape.setPixmap(QPixmap("sides\_easy\_q1"))

self.answer\_1\_a = "BC"

self.answer\_1\_b = "AC"

self.answer\_1\_c = "AB"

self.answer\_1\_d = "cosine"

self.answer\_1\_e = "5"

self.answer\_1\_f = "6"

\end{python}

This is one of the twenty-four first homework screens which inherits from the parent homework page 1 class (ParentHomeworkPage1 Class) and is placed in a stack widget (Trig1SidesEasyStack Class) with the corresponding second sides easy homework screen. This way it is not possible for only one of the two homework screens to be open at any time, as they are in the same stack which keeps changes until it is closed entirely.

\subsubsection{open\\_page\\_2 method}

\begin{python}

def open\_page\_2(self):

self.parent.stack.setCurrentIndex(1)

\end{python}

The screen being represented by the index here is hard-coded in the stack which is currently being operated, so this method can use the same code for each stack.

\subsection{SidesAHOEasyWidget2(HomeworkPage2ParentClass) Class}

\begin{python}

class SidesAHOEasyWidget2(HomeworkPage2ParentClass):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent = parent

self.task = "Sides Easy"

self.question\_2.setText("Question 2\nWhat is the length\nof b?")

self.question\_3.setText("Question 3\nWhat is the length\nof c?")

self.question\_4.setText("Question 4\nWhat is the\nlength of a?")

self.answer\_2.addItem("10")

self.answer\_2.addItem("20")

self.answer\_2.addItem("30")

self.answer\_3.addItem("10")

self.answer\_3.addItem("20")

self.answer\_3.addItem("30")

self.\_button\_1.setText("60")

self.\_button\_2.setText("50")

self.\_button\_3.setText("40")

self.\_button\_4.setText("30")

self.\_button\_5.setText("20")

self.\_button\_6.setText("10")

self.answer\_question\_4 = "40"

\end{python}

This class is one of the twenty-four second homework screens which inherits from the homework page 2 parent class (HomeworkPage2ParentClass Class) and is placed in a stack widget (Trig1SidesEasyStack Class) with the corresponding first sides easy homework screen. This way it is not possible for only one of the two homework screens to be open at any time, as they are in the same stack which keeps changes until it is closed entirely.

\subsection{ErrorMessage2(QErrorMessage) Class}

\begin{python}

class ErrorMessage2(QErrorMessage):

def \_\_init\_\_(self):

super().\_\_init\_\_()

message = "Invalid data type - please make sure you are inputting a decimal value"

QErrorMessage.showMessage(self, message)

\end{python}

This is one of the error message classes which inherits from the built in QErrorMessage class; these are displayed at certain points throughout the program, mainly in the homework if a user gets a wrong answer. these are useful becuase they can all be called at any point in another file simply by importing the file with the error messages into it , similar to the database controller, where all of the SQL code is separated.

\section{Variable Listing}

\begin{center}

\begin{longtable}{|p{3cm}|p{3cm}|p{4cm}|p{4cm}|} \hline

\textbf{Variable Name} & \textbf{Data Type} & \textbf{Purpose of Variable} & \textbf{Class reference} \\ \hline

self.\\_db\\_name & String & This is the variable which is passed through to the create\\_table method to check if this table already exists & Database Controller Class (4.3.1) \\ \hline

self.table\\_name & String & This is the name of the table which is created in the create\\_table method, and is passed through so the name will always be the same & Database Controller Class (4.3.1) \\ \hline

cursor & List & This variable is the structure controller for the database - it makes the changes when SQL code is executed & Database Controller Class (4.3.1) \\ \hline

result & List & This is assigned to the contents of the database which is fetched when the system checks to see if the table already exists & Database Controller Class (4.3.1) \\ \hline

info & List & This is assigned to the contents of the database which is fetched when a task is searched for to see if it exists already when saving a new record & Database Controller Class (4.3.1) \\ \hline

report & List & This is assigned to the contents of the database which is fetched when a user queries specific data in the database & Database Controller Class (4.3.1) \\ \hline

students & List & This is assigned to the contents of the database which is fetched when the progress screen is opened and all data is needed for display & Database Controller Class (4.3.1) \\ \hline

sql & String & This is the name of the variable which is always passed into the execute\\_sql method whenever another method need to access the database; it is the sql statement requiring execution & Database Controller Class (4.3.1) \\ \hline

response & String & This is an input which asks the user whether or not they would like to over-write the existing database - accepts y or n & Database Controller Class (4.3.1) \\ \hline

keep\\_table & Boolean & This is the boolean which checks whether the user wants to keep the existing table or not; input y for True, n for False & Database Controller Class (4.3.1) \\ \hline

g\\_database & Database Class & This variable represents the entire database and is called whenever a method wants to access the database & Database Controller Class (4.3.1) \\ \hline

self.cont & None & This variable is called to make changes to the continue button's appearance and to assign connections to methods for the button, which takes the user from the first screen to the home screen & All PyQt Windows \\ \hline

self.lessons & None & This variable is called to make changes to the lessons button's appearance and to assign connections to methods for the button, which takes the user from the home screen to the lesson topic menu & UserAccountClass (4.3.5) \\ \hline

self.homework & None & This variable is called to make changes to the homework button's appearance and to assign connections to methods for the button, which takes the user from the home screen to the homework topic menu & UserAccountClass (4.3.5) \\ \hline

self.progress & None & This variable is called to make changes to the progress button's appearance and to assign connections to methods for the button, which takes the user from the home screen to the progress screen & UserAccountClass (4.3.5) \\ \hline

self.log\\_out & None & This variable is called to make changes to the log out button's appearance and to assign connections to methods for the button, which closes the entire system when clicked & UserAccountClass (4.3.5) \\ \hline

self.t1 & None & This variable is called to make changes to the trigonometry 1 button's appearance and to assign connections to methods for the button, which takes the user from the lesson topic menu to the trigonometry 1 lesson sub menu & LessonMenuWidget (4.3.6)\\ \hline

self.ht1 & None & This variable is called to make changes to the homework trigonometry 1 button's appearance and to assign connections to methods for the button, which takes the user from the homework topic menu to the trigonometry 1 homework sub menu & HomeworkMenuWidget (4.3.7) \\ \hline

self.back & None & This variable is called to make changes to the return button's appearance and to assign connections to methods for the button, which returns the user from the screen they were on to the previous screen & All Menu Classes\\ \hline

self.report & None & This variable is called to make changes to the report button's appearance and to assign connections to methods for the button, which opens the report window from the progress window & DatabaseWidget (4.3.2) \\ \hline

self.submit & None & This variable is called to make changes to the submit button's appearance and to assign connections to methods for the button, which submits the contents of the combo boxes on the report widget to query the database & ReportWidget (4.3.3) \\ \hline

self.button\\_1 & None & This variable is called to make changes to the first menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.button\\_2 & None & This variable is called to make changes to the second menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.button\\_3 & None & This variable is called to make changes to the third menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.button\\_4 & None & This variable is called to make changes to the fourth menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.button\\_5 & None & This variable is called to make changes to the fifth menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.button\\_6 & None & This variable is called to make changes to the sixth menu button's appearance and to assign connections to methods for the button, which takes the user from the sub menu to the corresponding lesson & ParentLessonMenuClass (4.3.8), ParentHomeworkMenuClass (4.3.9) \\ \hline

self.next & None & This variable is called to make changes to the next button's appearance and to assign connections to methods for the button, which switches from the first window in a stack to the second, and saves the score and task name from the first screen to the database & ParentLessonLayout (4.3.13), ParentHomeworkPage1Class (4.3.18) \\ \hline

self.previous & None & This variable is called to make changes to the previous button's appearance and to assign connections to methods for the button, which switches from the second window in a stack to the first & ParentLessonPage2 (4.3.14), HomeworkPage2ParentClass (4.3.19) \\ \hline

self.check & None & This variable is called to make changes to the check button's appearance and to assign connections to methods for the button, which checks if the contents of the line edit in the lesson is correct & ParentLessonPage2 (4.3.14), ParentHomeworkPage1Class (4.3.18) \\ \hline

self.finish & None & This variable is called to make changes to the finish button's appearance and to assign connections to methods for the button, which closes the stack widget in use and saves the scores of the second screen to the database & ParentLessonPage2 (4.3.14), HomeworkPage2ParentClass (4.3.19) \\ \hline

self.cancel & None & This variable is called to make changes to the cancel button's appearance and to assign connections to methods for the button, which closes the stack widget in use without making any changes to the database & ParentLessonLayout (4.3.13), ParentHomeworkPage1Class (4.3.18) \\ \hline

self.reset & None & This variable is called to make changes to the reset button's appearance and to assign connections to methods for the button, which clears the contents of all 6 line edits on the first homework screen & ParentHomeworkPage1Class (4.3.18) \\ \hline

self.mark\\_2 & None & This variable is called to make changes to the first mark it button's appearance and to assign connections to methods for the button, which checks to see if the contents of the combo box is correct & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.mark\\_3 & None & This variable is called to make changes to the second mark it button's appearance and to assign connections to methods for the button, which checks to see if the contents of the combo box is correct & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.attempts\\_button & None & This variable is called to make changes to the atempts button's appearance and to assign connections to methods for the button, which simply displays the number of attempts the user has left on question 4 because it suits the design more than a QLabel & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_1 & None & This variable is called to make changes to the first mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_2 & None & This variable is called to make changes to the second mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_3 & None & This variable is called to make changes to the third mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_4 & None & This variable is called to make changes to the fourth mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_5 & None & This variable is called to make changes to the fifth mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.\\_button\\_6 & None & This variable is called to make changes to the sixth mulitple choice button's appearance and to assign connections to methods for the button, which is one of the 6 multiple choice buttons for question 4 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.title & String & This variable is called to make changes to the title QLabel's appearance and position. This label tells the user the title of the window which is currently open & All PyQt Classes\\ \hline

self.header & String & This variable contains a group of strings which are assigned to be the headers of the columns on the QTableWidgets which display the contents of the database & DatabaseWidget (4.3.2), ReportWidget (4.3.3) \\ \hline

self.task\\_box\\_label & String & This variable is used to make changes to the task box label's appearance and position. This label asks the user to select an option from the combo box & ReportWidget (4.3.3) \\ \hline

self.score\\_box\\_label & String & This variable is used to make changes to the score box label's appearance and position. This label asks the user to select an option from the combo box & ReportWidget (4.3.3) \\ \hline

self.lessons\\_label & String & This variable is used to make changes to the lessons label's appearance and position. This label identifies the purpose of the lessons button to the user & UserAccountwidget (4.3.5) \\ \hline

self.homework\\_label & String & This variable is used to make changes to the homework label's appearance and position. This label identifies the purpose of the homework button to the user & UserAccountwidget (4.3.5) \\ \hline

self.progress\\_label & String & This variable is used to make changes to the progress label's appearance and position. This label identifies the purpose of the progress button to the user & UserAccountwidget (4.3.5) \\ \hline

self.select & String & This variable is used to make changes to the select label's appearance and position. This label asks the user to select an option & LessonMenuWidget (4.3.6) \\ \hline

self.title\\_pic & Blob & This variable is used to make changes to the size and position of the image assigned to the label. This image is user friendly and identifies the topic for the user & All PyQt Classes \\ \hline

self.question\\_2 & String & This variable is used to make changes to the question 2 label's appearance and position. This parent label is a question which appears on each homework page 2 & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.task\\_box & String, Integer, Real & This variable is used to make changes to the task query combo box's appearance and position. This combo box contains the options of tasks to query, which are passed through to the SQL statement when selected & ReportWidget (4.3.3) \\ \hline

self.score\\_box & String, Integer, Real & This variable is used to make changes to the score query combo box's appearance and position. This combo box contains the options of scores to query, which are passed through to the SQL statement when selected & ReportWidget (4.3.3) \\ \hline

self.answer\\_2 & String, Integer, Real & This variable is used to make changes to the answer 2 combo box's appearance and position. This combo box contains the options of answers to the generic question 2 on the second homework page & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.answer\\_3 & String, Integer, Real & This variable is used to make changes to the answer 3 combo box's appearance and position. This combo box contains the options of answers to the generic question 3 on the second homework page & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.lesson\\_1 & String & This variable is used to make changes to the size and position of the lesson 1 QTextEdit. This contains the first part of each generic lesson on the first lesson pages & ParentLessonLayout (4.3.13) \\ \hline

self.lesson\\_2 & String & This variable is used to make changes to the size and position of the lesson 2 QTextEdit. This contains the second part of each generic lesson on the first lesson pages & ParentLessonLayout (4.3.13) \\ \hline

self.text\\_1 & String & This variable is used to make changes to the size and position of the text 1 QTextEdit. This contains the third part of each generic lesson on the second lesson pages & ParentLessonPage2 (4.3.14) \\ \hline

self.text\\_2 & String & This variable is used to make changes to the size and position of the text 2 QTextEdit. This contains the fourth part of each generic lesson on the second lesson pages & ParentLessonPage2 (4.3.14) \\ \hline

self.database & String, Integer & This variable is used to make changes to the size and formatting of the database QTableWidget, which contains all data fetched from the database when the progress screen is opened & DatabaseWidget (4.3.2) \\ \hline

self.db & String, Integer & This variable is used to make changes to the size and formatting of the db QTableWidget, which contains all data fetched from the database when a user queries for specific information & Reportwidget (4.3.3) \\ \hline

pal & Blob & This variable is used to set the background colour of the entire window which it is used in & All PyQt Classes \\ \hline

self.layout & None & This variable is used to assign the layout to be used as a QGridLayout, which is a useful layout for easily positioning widgets & All Classes \\ \hline

self.stack & None & This variable is used in the lesson and homework stacks to make the layout to be used a stack layout, so that one window can contain both parts of each lesson or homework & Trig1StackSides (4.3.12), Trig1SidesEasyStack (4.3.17) \\ \hline

self.widget & None & This variable is used where the stack widgets are used to set the widget type to be used as a QWidget, which contains the stack layout & All PyQt QGridLayout Classes \\ \hline

self.\\_centralwidget & None & This variable is used to set the class it is in as the priority window to be opened when an event is activated; the central window to be used first & ParentLessonMenu (4.3.6), HomeworkMenuWidget (4.3.7) \\ \hline

NameEntered & Boolean & This variable is a pyqtSignal which is used to tell the system that the continue button has been clicked and allow it to switch to the home screen, which is in the stack layout with the home screen & FirstScreen (4.3.4) \\ \hline

self.parent\\_window & None & This variable is used to pass down the parent values required to enable a window to be in a stack & UserAccountWidget(4.3.5) \\ \hline

self.parent & None & This variable is used to pass down the parent values required to enable a window to be in a stack & ParentLessonLayout (4.3.13), ParentLessonPage2 (4.3.14), ParentHomeworkPage1Class (4.3.18), HomeworkPage2ParentClass (4.3.19) \\ \hline

lessonmenuwidget & Class & This variable is assigned to the lesson menu widget so that it can be added to the layout and controlled for displaying and closing & UserAccountClass (4.3.5) \\ \hline

homeworkmenuwidget & Class & This variable is assigned to the homework menu widget so that it can be added to the layout and controlled for displaying and closing & UserAccountClass (4.3.5) \\ \hline

databasewidget & Class & This variable is assigned to the progress widget so that it can be added to the layout and controlled for displaying and closing & UserAccountClass (4.3.5) \\ \hline

trig\\_1\\_widget & Class & This variable is assigned to the trigonometry 1 lesson menu widget so that it can be added to the layout and controlled for displaying and closing & LessonMenuWidget (4.3.6) \\ \hline

trigonometry\\_1\\_homework & Class & This variable is assigned to the trigonometry 1 homework menu widget so that it can be added to the layout and controlled for displaying and closing & HomeworkMenuWidget (4.3.7) \\ \hline

sides\\_aho & Class & This variable is assigned to the first lesson screen widget so that it can be added to the layout and controlled for displaying and closing & Trigonometry1(ParentLessonMenu) (4.3.10) \\ \hline

sides\\_aho\\_1 & Class & This variable is assigned to the second lesson screen widget so that it can be added to the layout and controlled for displaying and closing & Trigonometry1HW(ParentHomeworkMenuClass) (4.3.11) \\ \hline

self.first\\_widget & Class & This variable is assigned to the screen at the top of the stack so that it can be added to the stack layout and controlled for displaying and closing & Trig1StackSides (4.3.12), Trig1SidesEasyStack (4.3.17) \\ \hline

self.second\\_widget & Class & This variable is assigned to the second screen in the stack so that it can be added to the stack layout and controlled for displaying and closing & Trig1StackSides (4.3.12), Trig1SidesEasyStack (4.3.17) \\ \hline

report\\_widget & Class & This variable is assigned to the lesson menu widget so that it can be controlled for displaying and closing & DatabaseWidget (4.3.2) \\ \hline

count & Integer & This variable is used to increment for each row that is filled when the database QTableWidget is filled so that the first row isn't constantly over-written until the last record & DatabaseWidget (4.3.2) \\ \hline

\\_count & Integer & This variable is used to increment for each row that is filled when the db QTableWidget is filled so that the first row isn't constantly over-written until the last record & ReportWidget (4.3.3) \\ \hline

self.correct\\_count & Integer & This variable is used in the check\\_selected method in the homework page 1 parent class to count the number of line edits which contain a correct input from the user & ParentHomeworkPage1Class (4.3.18) \\ \hline

self.correct\\_count\\_2 & Integer & This variable is used to count the correct answers in the generic second homework questions & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.correct\\_count\\_3 & Integer & This variable is used to count the correct answers in the generic third homework questions & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.correct\\_count\\_4 & Integer & This variable is used to count the correct answers in the generic fourth homework questions & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.task & String & This is the hard-coded variable in each subclass for the homeworks which is saved to the database - it is the task name which is intended to be saved with the scores & ParentHomeworkPage1Class (4.3.18) \\ \hline

self.answer\\_lesson & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the lesson answers & ParentLessonLayout (4.3.13) \\ \hline

self.answer\\_1\\_a & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the first of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_1\\_b & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the second of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_1\\_c & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the third of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_1\\_d & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the fourth of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_1\\_e & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the fifth of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_1\\_f & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the sixth of the six homework page 1 line edits & SidesAHOEasyWidget(ParentHomeworkPage1Class) (4.3.18) \\ \hline

self.answer\\_question\\_4 & Integer, Real, String & This variable is hard-coded in each subclass for the check method to compare the user's input to a question against; it is the correct answer expected for the multiple choice question for on each homework page 2 & HomeworkPage2ParentClass (4.3.19) \\ \hline

cont & Boolean & This boolean is used to decide whether or not the user can switch from the first to the second screen in a stack; if they have not answered all questions, this variable will stay false until they do & ParentHomeworkPage1Class (4.3.18) \\ \hline

self.allow\\_cont & Boolean & this boolean is used to decide whether or not the user can close a homework stack; if they have not answered all questions on a homework page 2 it will stay false until they do & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.attempts\\_remaining\\_a & Integer & This variable is used to count how many attempts the user has at question 2 before all input methods involved in said question are disabled; it will decrement with each wrong input checked by the checking methods & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.attempts\\_remaining\\_b & Integer & This variable is used to count how many attempts the user has at question 3 before all input methods involved in said question are disabled; it will decrement with each wrong input checked by the checking methods & HomeworkPage2ParentClass (4.3.19) \\ \hline

self.attempts\\_remaining\\_c & Integer & This variable is used to count how many attempts the user has at question 4 before all input methods involved in said question are disabled; it will decrement with each wrong input checked by the checking methods & HomeworkPage2ParentClass (4.3.19) \\ \hline

data & List & This variable is passed through to the method which queries the database; it is the task name value selected in the corresponding combo box & ReportWidget (4.3.3) \\ \hline

score\\_data & List & This variable is passed through to the method which queries the database; it is the score value selected in the corresponding combo box & ReportWidget (4.3.3) \\ \hline

self.answer & List & This variable is used to add the six self.answer\\_1\\_x variables to a list so that the same changes can be made to all of them in a for loop rather than typing the changes out separately six times & ParentHomeworkPage1Class (4.3.18) \\ \hline

error\\_message & Class & This variable is assigned to an error message class so that it can be displayed at any point in a method when necessary & ErrorMessage2(QErrorMessage) (4.3.22) \\ \hline

message & String & this is the message which is displayed in an error message class when it is called - used to tell the user what they have done wrong & ErrorMessage2(QErrorMessage) (4.3.22) \\ \hline

\end{longtable}

\end{center}

\section{System Evidence}

\subsection{User Interface}

\begin{figure}[H]

\label{fig: Second Screen}\caption{First Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_1}

\end{figure}

This screen is in the first window which opens whent he system is run; it contains the title of the system, a picture, and a welcome message for user friendliness, and a continue button is present for switching to the home screen.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Home Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

This screen is in the same stack window as the first screen and is displayed after the continue button is clicked. It contains four buttons: lessons, homework, progress and exit program, for accessing the lesson topic menu, the homework topic menu, the database viewer and exiting the program respectively. It also contains a picture for each of the three navigation buttons which suggest where te button will take them; they are relevant to the next screens and are user friendly.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Lesson Topic Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_3}

\end{figure}

This screen is displayed when the user selects the lessons button on the home screen. It contains a title image which tells the user which screen they are on, followed by five buttons, each of which navigate to a topic specific sub-menu (which all appear the same, except for text and images, as they share a parent class). Each button is accompanied by a user friendly picture relevant to the topic menu they lead to. There is a return button which closes the window and returns the user to the home screen.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Trigonometry 1 Lesson Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

This window opens when the user clicks the first button on the lesson topic menu. It is one of five subclasses from the ParentLessonMenu, so these five windows essentially contain the same layout, only with different words, pictures and connections. They each contain a title image which tells the user which screen they are on, and have either two or three buttons, each accompanied by a picture, depending on how many lessons are in the selected topic menu. There is a return button which closes the window and returns the user to the lesson topic menu.

\begin{figure}[H]

\label{fig: Second Screen}\caption{SOHCAHTOA 1st Lesson screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_5}

\end{figure}

This window is accessed by clicking the corresponding button on one of the five derived lesson menus. It is one of twelve first lessons screens which inherit from a parent class; they each contain a title image, two text boxes for lessons and explanations, and cancel button, and a next button, along with one or two images depending on the lesson.

\begin{figure}[H]

\label{fig: Second Screen}\caption{SOHCAHTOA 2nd Lesson screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_6}

\end{figure}

This window is in the same stack as the SOHCAHTOA 1st lesson screen, is also subclassed from a parent, and is accessed by clicking the next button. It contains a further two text boxes for lessons and examples, a check button with a line edit for a practice question which you can check the answer to, a previous button to switch back to the first lesson screen, and finish button to close the stack, and one or two images, again, depending on the lesson.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Homework Topic Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_7}

\end{figure}

This screen is displayed when the user selects the homework button on the home screen. It contains a title image which tells the user which screen they are on, followed by five buttons, each of which navigate to a topic specific sub-menu (which all appear the same, except for text and images, as they share a parent class). Each button is accompanied by a user friendly picture relevant to the topic menu they lead to. There is a return button which closes the window and returns the user to the home screen.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Trigonometry 1 Homework Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_8}

\end{figure}

This window opens when the user clicks the first button on the homework topic menu. It is one of five subclasses from the HomeworkMenuParentClass, so these five windows essentially contain the same layout, only with different words, pictures and connections. They each contain a title image which tells the user which screen they are on, and have either three or six buttons, each accompanied by a picture, depending on how many homework tasks are in the selected topic menu. There is a return button which closes the window and returns the user to the homework topic menu.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Sides Easy 1st Homework Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

This window is accessed by clicking the corresponding button on one of the five derived homework menus. It is one of twenty-four first homework screens which inherit from a parent class; they each contain a title image, a question, a reset button, a check answers button, a cancel button, a next button, and six line edits into which the user can input their answers to the question, which can be marked using the check button or all cleared using the reset button.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Sides Easy 2nd Lesson screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_10}

\end{figure}

This window is in the same stack as the Sides Easy 1st homework screen, is also subclassed from a parent, and is accessed by clicking the next button. It contains three questions as QLabels, two combo boxes, two mark it buttons, six multiple choice buttons, a button showing the remaining number of attempts, a previous button, and a finish button.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Progress Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_11}

\end{figure}

This window opens when the user selects progress from the home screen. It contains a title, a return button, a report button, which opens the report widget, and a QTableWidget which contains the data fetched from the database automatically everytime the screen is loaded.

\begin{figure}[H]

\label{fig: Second Screen}\caption{Report Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_12}

\end{figure}

This window opens when the user selects report from the progress screen. It contains a QTableWidget which displays the results of the query, a return button, a task combo box, a score combo box and a query button, which are used by the user to input a query for the database.

\subsection{ER Diagram}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/er\_diagram.png}

\label{fig:print\_function\_result}

\end{figure}

\subsection{Class Diagrams}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{MyWindow Class} \\ \hline

NameEntered : Boolean \\

self.login\\_widget : Class \\

self.student\\_home : Class \\

self.stack : Class \\

self.widget : Class \\ \hline

enter\\_program : Boolean \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{Database Class} \\ \hline

self.\\_db\\_name : String \\

self.table\\_name : String \\ \hline

execute\\_sql : String \\

create\\_table : String \\

insert\\_data\\_first : String, Integer \\

insert\\_data\\_second : String, Integer, Integer, Integer \\

get\\_query : String, Integer \\

GetAllNames : \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{DatabaseWidget Class} \\ \hline

students : List \\

count : Integer \\ \hline

selected\\_back : \\

selected\\_report : \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{ParentHomeworkPage1Class} \\ \hline

self.task : String \\

self.allow\\_cont : Boolean \\

self.answers : List \\ \hline

check\\_selected : \\

next\\_selected : \\

reset\\_selected : \\

cancel\\_selected : \\

open\\_page\\_2 : \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{HomeworkPage2ParentClass} \\ \hline

self.task : String \\

self.attempts\\_remaining\\_a : Integer, Real \\

self.attempts\\_remaining\\_b : Integer, Real \\

self.attempts\\_remaining\\_c : Integer, Real \\

self.correct\\_count\\_2 : Integer, Real \\

self.correct\\_count\\_3 : Integer, Real \\

self.correct\\_count\\_4 : Integer, Real \\

self.answer\\_question\\_4 : Real \\ \hline

check\\_button\\_1 : Integer, Real \\

check\\_button\\_2 : Integer, Real \\

check\\_button\\_3 : Integer, Real \\

check\\_button\\_4 : Integer, Real \\

check\\_button\\_5 : Integer, Real \\

check\\_button\\_6 : Integer, Real \\

selected\\_mark\\_2 : Integer, Real \\

selected\\_mark\\_3 : Integer, Real \\

selected\\_previous : \\

selected\\_finish : \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{7cm}|} \hline

\textbf{QErrorMessage} \\ \hline

message : String \\ \hline

x.showMessage : String \\ \hline

\end{tabular}

\end{center}

\subsection{Navigation Diagram}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/nav\_diagram.png}

\label{fig:print\_function\_result}

\end{figure}

\subsection{Database Table Views}

\textbf{Task Entity: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/task\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This is the task name entity where the name of the task hard-coded from the subclass homework has been recorded along with the scores.

\textbf{Qone Entity: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/q\_one\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This records the count of the number of correct line edit answers in the subclass for the task.

\textbf{Qtwo Entity: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/q\_two\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This records the score of the second homework question from each subclass.

\textbf{Qthree Entity: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/q\_three\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This records the score of the third homework question from each subclass.

\textbf{Qfour Entity: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/q\_four\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This records the score of the fourth homework question from each subclass.

\textbf{All Entities: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Maintenance/all\_db.png}

\label{fig:print\_function\_result}

\end{figure}

This shows a whole record for a task that has been saved along with the scores that were obtained on said task.

\subsection{Database SQL}

\textbf{create\\_table method: }

\begin{python}

def create\_table(self, table\_name):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select name from sqlite\_master where name=?",(table\_name,))

result = cursor.fetchall()

keep\_table = True

if len(result) == 1:

response = input("The table {0} already exists, do you wish to recreate it (y/n): ".format(table\_name))

if response == "y":

keep\_table = False

print("The {0} table will be recreated - all existing data will be lost".format(table\_name))

cursor.execute("drop table if exists {0}".format(table\_name))

db.commit()

else:

print("The existing table was kept")

else:

keep\_table = False

if not keep\_table:

sql = """create table Student

(TaskID text,

Qone integer,

Qtwo integer,

Qthree integer,

Qfour integer,

primary key(TaskID))"""

cursor.execute(sql)

db.commit()

\end{python}

This method is responsible for creating the table, and is run whenever the system is run, as it also checks to see if the database already exists before starting. It connects to the database which is identified by the \textit{g\\_database} variable which is passed in to the method containing the database file name. The cursor (structure controller) searches all existing database files to see if one named 'Student' (the table\\_name variable also passed into the method) already exists. If it does, it asks the user whether or not they want to over-write it; if they input yes, it will the drop the current table and create a new one using the SQL statement at the end of the class. Otherwise, if the boolean keep\\_table remains true, the program will just move on; likewise if the table doesn't already exist, in which the same sql statement will be used to create it. The SQL statement in this method creates the table called Student, which is the name used to check the database files at the start of the method, and gives the following entities: TaskID, which is the task name and a string, then four question entities which each contain an integer. The SQL here is executed in the same method, rather than the execute\\_sql method.

\textbf{Important Database variables: }

\begin{itemize}

\item self.\\_db\\_name - This is the name of the database file which is passed into the class when a variable is assigned to it (in this case it is "student\\_database.db") and used to identify which database file each SQL method needs to connect to.

\item self.table\\_name - This is the name of the table which is created and is used to check to see if this table already exists, and to identify which table to make changes to when SQL code is executed.

\item g\\_database - This variable is assigned to the database class and given a file name which is passed into the class to be used for identifying the file intended for use.

\end{itemize}

\subsection{SQL Queries}

\subsubsection{create\\_table method: }

\begin{python}

"select name from sqlite\_master where name=?",(table\_name,))

\end{python}

This SQL query takes the name of the table which is being used (Student) and uses it to check the database files to see if a table with that name already exists, so that it can be over-written if need be.

\begin{python}

"drop table if exists {0}".format(table\_name))

\end{python}

This SQL query takes the table name (Student) if an existing table with that name is found, and deletes it entirely, if the user chooses to over-write it.

\begin{python}

sql = """create table Student

(TaskID text,

Qone integer,

Qtwo integer,

Qthree integer,

Qfour integer,

primary key(TaskID))"""

\end{python}

This SQL query creates the Student table if it does not already exist; it adds the five entities and then makes TaskID the primary key.

\subsubsection{insert\\_data\\_first method: }

\begin{python}

"select TaskID from Student where TaskID = '{0}'".format(task))

\end{python}

The task variable, representing the name of the task being completed, is passed through from the first homework screen in use. The SQL statement uses this variable to search the database for any records already existing with the same task name; all records where the task name is the same are fetched and returned to the method in the form of a list variable. If one is found it knows to update rather than create a new record, otherwise it just creates a new record with that task name.

\begin{python}

"UPDATE Student SET Qone = '{0}' WHERE TaskID = '{1}' AND Qone < '{2}'".format(correct\_count, task, correct\_count)

\end{python}

If a record with the same task name already exists, the update statement is used to record the new correct\\_count (score of the first homework question) but only if the new score is better than the old one, checked using the less than symbol. The WHERE part of the statement ensures that the changes are only made to the one record with the same task name rather than all records.

\begin{python}

"insert into Student(TaskID, Qone, Qtwo, Qthree, Qfour) values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(task, correct\_count, str(0), str(0), str(0))

\end{python}

This is the insert statement used if a record with the same task name is not found; it saves the task name along with the score of the first homework question, then saves values of 0 in the other three questions entities. This is so that the same update statements can be used in the next method regardless of whether or not a record previously existed, as values are there to be over-written anyway because this method will always be run first due to the access limitations of the second screen (having to submit the first question score before proceeding).

\subsubsection{insert\\_data\\_second method: }

\begin{python}

"UPDATE Student SET Qtwo = '{0}' WHERE TaskID = '{1}' AND Qtwo < '{2}'".format(count\_2, task, count\_2)

\end{python}

This SQL statement overwrites the value of the Qtwo entity in the record with the same task name, which will have just been written by the previous method from the first homework screen. If the value already there is less than the new score, then it will be over-written, whether it is a fresh 0 or an existing 4. The count variable is passed through from the second homework screen which calls the method.

\begin{python}

"UPDATE Student SET Qthree = '{0}' WHERE TaskID = '{1}' AND Qthree < '{2}'".format(count\_3, task, count\_3)

\end{python}

This SQL statement overwrites the value of the Qthree entity in the record with the same task name, which will have just been written by the previous method from the first homework screen. If the value already there is less than the new score, then it will be over-written, whether it is a fresh 0 or an existing 4. The count variable is passed through from the second homework screen which calls the method.

\begin{python}

"UPDATE Student SET Qfour = '{0}' WHERE TaskID = '{1}' AND Qfour < '{2}'".format(count\_4, task, count\_4)

\end{python}

This SQL statement overwrites the value of the Qfour entity in the record with the same task name, which will have just been written by the previous method from the first homework screen. If the value already there is less than the new score, then it will be over-written, whether it is a fresh 0 or an existing 4. The count variable is passed through from the second homework screen which calls the method.

\subsubsection{get\\_query method: }

\begin{python}

"select \* from Student WHERE TaskID = '{0}' or Qone = '{1}'".format(data, score\_data)

\end{python}

This is used to query the database using variables passed through from the contents of the combo boxes in the report widget, where the user selects which details to query. The SQL statement searches the database for records which exist containing either one of the two variable values passed through, then returns them to a list variable which can be used to display the result.

\subsubsection{GetAllNames method: }

\begin{python}

"select \* from Student"

\end{python}

This SQL statement fetches all information currently stored in the database and returns it to be displayed in the progress window using a list variable to call functions.

\section{Testing}

\subsection{Summary of Results}

Throughout the testing stage I had a few manageable issues, and one quite major issue, but all of these I was able to solve without too much trouble. The biggest problem was having to cut out all administrator capabilities from the system, and turn it into a single user program, due to a lack of time and knowledge. However, referring to the parts of the system which I actually created, the system was proven to be both reliable and robust one these problems had been fixed, as there is now no point where an endless loop can be entered or a crash occurs, and all data is stored responsibly. Save the administrator aspects, the system generally meets the user requirements as it is user friendly, easy to use and contains the subject material needed to teach trigonometry using a range of input types.

The problems I encountered includes the following:

\begin{itemize}

\item The report widget having to be put on the same screen as the query inputs as I could not find a way to pass variables through to a separate pop-up window class

\item Not having the knowledge to implement a drag and drop system into the homework, and replacing it with a multiple choice button system

\item Implementing separate accounts with log-ins and access restrictions for an administrator

\item Being able to save a total score to the database due to not being able to pass the variables through to the same window to be transformed together

\item The database would not over-write an existing record with the same task name, it would just crash, until I implemented an update SQL statement

\item The QTableWidgets would only display one record because they kept over-writing in the first row, until I placed a for loop in to increment the row count

\end{itemize}

\textbf{Below is the results table from the testing stage: }

\begin{landscape}

\begin{center}

\begin{longtable}{|p{2.5cm}|p{4cm}|p{4cm}|p{4.5cm}|p{3cm}|} \hline

\textbf{Test Number} & \textbf{Expected Result} & \textbf{Test Data} & \textbf{Actual Result} & \textbf{Screenshot Numbers} \\ \hline

1.003 & The lessons menu should be displayed & Click the lessons button & The lessons menu was displayed, as expected & Figures 3.1, 3.2 \\ \hline

1.004 & The homework menu should be displayed & Click the homework button & The homework menu was displayed, as expected & Figures 3.3, 3.4 \\ \hline

1.005 & The progress window should be displayed & Click the progress button & The progress window was displayed, as expected & Figures 3.5, 3.6 \\ \hline

1.006 & The program should close entirely & Click the exit program button & The entire program closed, as expected & Figures 3.7, 3.8 \\ \hline

1.007 & The Trigonometry 1 lesson menu should be displayed & Click the Trigonometry 1 button & The Trigonometry 1 lesson menu was displayed, as expected & Figures 3.9, 3.10 \\ \hline

1.012 & The lesson menu should be closed and the home screen displayed & Click the return button & The lesson menu closed and the home screen was displayed, as expected & Figures 3.11, 3.12 \\ \hline

1.013 & The SOHCAHTOA first lesson screen should be displayed & Click the SOHCAHTOA button & The SOHCAHTOA first lesson screen was displayed, as expected & Figures 3.13, 3.14 \\ \hline

1.015 & The Trigonometry 1 lesson menu should be closed and the lesson menu displayed & Click the return button & The Trigonometry 1 lesson menu closed and the lesson menu was displayed, as expected & Figures 3.15, 3.16 \\ \hline

1.031 & The SOHCAHTOA first lesson screen should be closed and the Trigonometry 1 lesson menu should be displayed & Click the return button & The SOHCAHTOA first lesson screen was closed and the Trigonometry 1 lesson menu was displayed, as expected & Figures 3.17, 3.18 \\ \hline

1.032 & The SOHCAHTOA second lesson screen should replace the first SOHCAHTOA lesson screen in display & Click the next button & The second SOHCAHTOA lesson screen replaced the first SOHCAHTOA lesson screen, as expected & Figures 3.19, 3.20 \\ \hline

1.033 & The SOHCAHTOA first lesson screen should replace the second SOHCAHTOA lesson screen in display & Click the previous button & The first SOHCAHTOA lesson screen replaced the second SOHCAHTOA lesson screen, as expected & Figures 3.21, 3.22 \\ \hline

1.034 & The input typed in the line edit should be checked and the user told whether they were correct or not & Click the button & The correct answer was registered as correct and the wrong answer was registered as incorrect, as expected & Figures 3.23, 3.24 \\ \hline

1.035 & The stack window with the SOHCAHTOA lesson should close and the Trigonometry 1 lesson menu be displayed & Click the finish button & The SOHCAHTOA lesson stack was closed and the Trigonometry 1 lesson menu was displayed, as expected & Figures 3.25, 3.26 \\ \hline

1.097 & The Trigonometry 1 homework menu should be displayed & Click the trigonometry 1 button & The Trigonometry 1 homework menu was displayed, as expected & Figures 3.27, 3.28 \\ \hline

1.102 & The homework menu should close and the home screen should be displayed & Click the return button & The homework menu closed and the home screen was displayed, as expected & Figures 3.29, 3.30 \\ \hline

1.103 & The first sides easy homework screen should be displayed & Click the sides easy button & The first sides easy homework screen was displayed, as expected & Figures 3.31, 3.32 \\ \hline

1.135 & The sides easy homework stack should close and the trigonometry 1 homework menu displayed & Click the return button & The sides easy homework stack was closed and the trigonometry 1 homework menu was displayed, as expected & Figures 3.33, 3.34 \\ \hline

1.136 & The 6 line edits should be checked and the user told how many were right, and marks given & Click the check answers button & The line edits were checked, correct answers were recognised and incorrect answers were rejected, as expected & Figures 3.35, 3.36 \\ \hline

1.137 & The 6 line edits contents should all be reset to empty & Click the reset button & All 6 line edits were cleared, as expected & Figures 3.37, 3.38 \\ \hline

1.138 & The second sides easy homework screen should replace the first sides easy homework screen in the stack, and the score from the first question should be stored in the database & Click the next button & The second sides easy homework screen replaced the first sides easy homework screen in the stack, and the correct score count was stored in the database, as expected & Figures 3.39, 3.40 \\ \hline

1.139 & The input in the combo box should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click the mark it button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.41, 3.42 \\ \hline

1.140 & The input in the combo box should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click the mark it button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.43, 3.44 \\ \hline

1.141 & The correct button should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click each possible button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.45, 3.46, 3.47 \\ \hline

1.142 & The first sides easy homework screen should replace the second sides easy homework screen in the stack & Click the previous button & The first sides easy homework screen replaced the second sides easy homework screen in the stack, as expected & Figures 3.48, 3.49 \\ \hline

1.143 & the sides easy homework stack should be closed and the home screen should be displayed; The scores from the questions should be stored in the database & Click the finish button & The sides easy homework stack was closed, the home screen was displayed, and the scores were saved to the database, as expected & Figures 3.50, 3.51 \\ \hline

1.380 & The progress screen should be closed and the home screen displayed & Click the return button & The progress screen was closed and the home screen was displayed, as expected & Figures 3.52, 3.53 \\ \hline

1.431 & The report screen should be closed and the progress screen displayed & Click the return button & The report screen was closed and the home screen was displayed, as expected & Figures 3.54, 3.55\\ \hline

1.432 & The relevant information should be fetched from the database and displayed in the same window & Click the query button & Relevant information was found and displayed in the database in hte same window, as expected & Figures 3.56, 3.57 \\ \hline

1.441 & The welcome screen should close and the home screen should be displayed & Click the continue button & The welcome screen was closed and the home screen was displayed, as expected & Figures 3.58, 3.59 \\ \hline

1.442 & The report screen should open and the progress screen hidden & Click the report button & The progress screen closed and the report screen was displayed, as expected & Figures 3.60, 3.61 \\ \hline

2.003 & If the input is correct, the word correct should be displayed, otherwise incorrect shoud be displayed & 5 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.62, 3.63 \\ \hline

2.016 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 1 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.017 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 2 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.018 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 3 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.019 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 4 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.020 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 5 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.021 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 6 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.022 & If the contents of the combo box is correct, correct should appear in the button next to it, otherwise an attempt will be removed & 20; 10 & For the correct answer, correct was printed, an incorrect for the incorrect answer, as expected & Figures 3.66, 3.67 \\ \hline

2.023 & If the contents of the combo box is correct, correct should appear in the button next to it, otherwise an attempt will be removed & 40; 15 & For the correct answer, correct was printed, an incorrect for the incorrect answer, as expected & Figures 3.68, 3.69 \\ \hline

2.024 & If the right button is clicked, display correct, otherwise display incorrect & Each button in order & When the right button was clicked, a mark was added, and when the wrong buttons were clicked attempts were removed, as expected & Figures 3.70, 3.71 \\ \hline

2.295 & The information relevant to the input should be fetched from the database and displayed & Sides Easy; Pythagoras Theorem Hard & The relevant task name was fetched and displayed, as expected & Figure 3.72 \\ \hline

2.296 & The information relevant to the input should be fetched from the database and displayed & 70\%; 80\% & The relevant scores were fetched and displayed, as expected & Figure 3.73 \\ \hline

3.009 & The task names should be stored under the 'Task Names' header in the database & Complete a task & The task name was stored under 'Task Names', as expected & Figure 3.74 \\ \hline

3.011 & The IndividualPercentScores should be stored under the 'QOne', 'QTwo', 'QThree', and 'QFour' headers respective to the question number in the database & Complete a task & The scores were under their respective headings, as expected & Figure 3.75 \\ \hline

4.015 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.76, 3.77, 3.78 \\ \hline

4.016 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.79, 3.80, 3.81 \\ \hline

4.017 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.82, 3.83, 3.84 \\ \hline

4.018 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The appropriate outputs were given for every possile input tested & 3.85, 3.86 \\ \hline

5.001 & The client should be satisfied with the overall system & Show the client each aspect of the system & N/A & \\ \hline

5.003 & The task names and scores should be being saved to an sqlite database, 5 columns, as many rows as there are tasks & Complete a task & There are the right number of headings in the table all with the correct information being displayed in them, as expected & Figure 3.87 \\ \hline

5.004 & Only the task names and scores should be stored in the database & Complete a task & They are the only pieces of information being stored, as expected & Figure 3.88 \\ \hline

5.006 & No illegitimate information or personal information should be being stored & Complete a task (only source of information for the database) & No illegitimate information or personal information is being stored, only task names and scores, so the DPA cannot be breached anyway & Figure 3.89 \\ \hline

\end{longtable}

\end{center}

\end{landscape}

\subsection{Known Issues}

The only problem remaining is that of the administrator aspect of the system not being possible to create using the time and knowledge I have. In the Python 3.4 shell, every time a window is opened an error to do with the \\_raise function pops up, however this does not affect the running of the program in any way and the user will never see this error in the distributed version of the system. Without, this function, the windows will not be displayed, so I cannot simply remove it. Lastly, I will not be able to record total scores from tasks as I still am unable to find a way to pass the appropriate variables through to the appropriate classes.

Otherwise, all of the other problems mentioned in section 3.4.2 have either been solved or replaced with an alternative working solution.

\section{Code Explanations}

\subsection{Difficult Sections}

\textbf{selected\\_submit method (ReportWidget Class 4.3.3)}

\begin{python}

def selected\_submit(self):

\_count = 0

data = self.task\_box.currentText()

score\_data = self.score\_box.currentText()

report = g\_database.get\_query(data, score\_data)

for count in range(24):

self.db.setItem(count, 0, QTableWidgetItem(None))

self.db.setItem(count, 1, QTableWidgetItem(None))

self.db.setItem(count, 2, QTableWidgetItem(None))

self.db.setItem(count, 3, QTableWidgetItem(None))

self.db.setItem(count, 4, QTableWidgetItem(None))

for record in report:

self.db.setItem(\_count, 0, QTableWidgetItem(record[0]))

self.db.setItem(\_count, 1, QTableWidgetItem(str(record[1])))

self.db.setItem(\_count, 2, QTableWidgetItem(str(record[2])))

self.db.setItem(\_count, 3, QTableWidgetItem(str(record[3])))

self.db.setItem(\_count, 4, QTableWidgetItem(str(record[4])))

\_count += 1

\end{python}

The part where the database method fetches from the database worked fine here, the problem was displaying the data in the QTableWidget in the report window. Only one record would be displayed, and at first I thought it was because the SQL kept over-writing the previous record, and that the fault was with the insert statements. However it turned out to be to do with the formatting of the fetched information in the QTableWidget; I had to fiddle with counts and tried putting them in different places before all of the data would be displayed, then had to do it again to clear the table with each new query. Eventually the data was all displayed on each row, fixed by an incrementing row count.

\textbf{Makes the background white (all PyQt classes)}

\begin{python}

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

\end{python}

This section of simple PyQt code was very hard to come across. I had to research how to change the background colour of a PyQt window on the internet as I had no experience beforehand in this area of PyQt. A very long time was spent trying out similar code which actually only changed the colour of the widgets on the layout rather than the background of the entire window itself. In fact, often everything but the part I wanted to change colour would actually change colour. Eventually, I tried this section of code, and it finally worked, and is useful because I can stick it in parent classes a few times and it will be in all of the subclasses, making the background colour consistent.

\textbf{next\\_selected method (ParentHomeworkPage1Class 4.3.18)}

\begin{python}

def next\_selected(self):

cont = False

while not cont:

for a in self.answers:

if a.text() == "":

error\_message = ErrorMessage8()

error\_message.show()

error\_message.\_raise()

cont = False

cont = True

if self.allow\_cont:

g\_database.insert\_data\_first(self.task, self.correct\_count)

self.open\_page\_2()

self.hide()

else:

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

\end{python}

With this method, it was difficult to find the right indentation for the error messages to work with the intended logic, and also change the cont boolean to true. After lots of dry running I managed to get the error messages opening at the right times and the database method to work at the right time.

\textbf{insert\\_data\\_first method (DatabaseClass 4.3.1)}

\begin{python}

def insert\_data\_first(self, task, correct\_count):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select TaskID from Student where TaskID = '{0}'".format(task))

info = cursor.fetchall()

if len(info) != 0:

sql = "UPDATE Student SET Qone = '{0}' WHERE TaskID = '{1}' AND Qone < '{2}'".format(correct\_count, task, correct\_count)

else:

sql = "insert into Student(TaskID, Qone, Qtwo, Qthree, Qfour) values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(task, correct\_count, str(0), str(0), str(0))

self.execute\_sql(sql)

\end{python}

I had a large issue with getting the insert\\_data\\_second method to be able to update and add new values in a record which already existed with no values in some of the entities. I decided to just add 0 values, as technically this would always be true for the user's score until they completed the second page anyway, and it would prove convenient when adding the update statements to the insert\\_data\\_second method, which would be able to update over 0 values or existing 1, 2, 3 etc. values. It did however take a long time to find this solution.

\textbf{selected\\_mark\\_2 method (HomeworkPage2ParentClass (4.3.19)}

\begin{python}

def selected\_mark\_2(self, attempts\_remaining\_a):

self.correct\_count\_2 = 0

if self.answer\_2.currentText() == "20":

self.correct\_count\_2 += 1

self.mark\_2.setText("Correct!")

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

else:

self.attempts\_remaining\_a -= 1

self.mark\_2.setText("Mark it|{0}".format(self.attempts\_remaining\_a))

if self.attempts\_remaining\_a == 0:

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

error\_message = ErrorMessage5()

error\_message.show()

error\_message.\_raise()

return self.attempts\_remaining\_a, self.correct\_count\_2

\end{python}

It was quite difficult to find the right indentation for the input widget disabling lines of code here. There were quite a few different things too juggle and maintain logic with this check algorithm - for example, counting the number of attempts remaining and only when that reaches 0 actually disabling the right widgets, and popping the error messages at the right time as well. It took a bit of playing around with the indentation but it eventually worked as intended.

\subsection{Self-created Algorithms}

\textbf{insert\\_data\\_first method (Database Class 4.3.1)}

\begin{python}

def insert\_data\_first(self, task, correct\_count):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select TaskID from Student where TaskID = '{0}'".format(task))

info = cursor.fetchall()

if len(info) != 0:

sql = "UPDATE Student SET Qone = '{0}' WHERE TaskID = '{1}' AND Qone < '{2}'".format(correct\_count, task, correct\_count)

else:

sql = "insert into Student(TaskID, Qone, Qtwo, Qthree, Qfour) values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(task, correct\_count, str(0), str(0), str(0))

self.execute\_sql(sql)

\end{python}

1 - The variables which represent the task name and the score the user achieved from a homework first page task are passed into the method.

2 - The database file intended for use, "student\\_database.db", is connected to by the controller.

3 - The cursor variable is the structure controller for the database.

4 - This SQL statement searches the databsae for a record with the same task name as the value being passed through as task.

5 - The data retrieved from the SQL statement is assigned to a list variable.

6 - If a record with the same task name is found, the algorithm will proceed to update it.

7 - This SQL statement updates the value of the Qone entity if the new value is greater than the old one. Otherwise it is just left.

8 - If a record is not found the algorithm will proceed to make a new one.

9 - This SQL statement inserts a new record into the Student table and applies a value to all entities so the next SQL statement can always be an update statement.

10 - The execute\\_sql method is run which contains the code which applies the SQL changes to the Student table in the database file.

\textbf{insert\\_data\\_second method (Database Class 4.3.1)}

\begin{python}

def insert\_data\_second(self, task, count\_2, count\_3, count\_4):

with sqlite3.connect(self.\_db\_name) as db:

sql = "UPDATE Student SET Qtwo = '{0}' WHERE TaskID = '{1}' AND Qtwo < '{2}'".format(count\_2, task, count\_2)

self.execute\_sql(sql)

sql\_2 = "UPDATE Student SET Qthree = '{0}' WHERE TaskID = '{1}' AND Qthree < '{2}'".format(count\_3, task, count\_3)

self.execute\_sql(sql\_2)

sql\_3 = "UPDATE Student SET Qfour = '{0}' WHERE TaskID = '{1}' AND Qfour < '{2}'".format(count\_4, task, count\_4)

self.execute\_sql(sql\_3)

\end{python}

1 - The variables representing the scores the user achieved from a homework second page task are passed into the method.

2 - The database file intended for use, "student\\_database.db", is connected to by the controller.

3 - This SQL statement takes the score from question 2 and updates either the clean 0 value or the previously obtained lower score of the Qtwo entity.

4 - The execute\\_sql method is run which contains the code which applies the SQL changes to the Student table in the database file.

5 - This SQL statement takes the score from question 3 and updates either the clean 0 value or the previously obtained lower score of the Qtwo entity.

6 - The execute\\_sql method is run which contains the code which applies the SQL changes to the Student table in the database file.

7 - This SQL statement takes the score from question 4 and updates either the clean 0 value or the previously obtained lower score of the Qtwo entity.

8 - The execute\\_sql method is run which contains the code which applies the SQL changes to the Student table in the database file.

\textbf{selected\\_submit method (Reportwidget 4.3.3)}

\begin{python}

def selected\_submit(self):

\_count = 0

data = self.task\_box.currentText()

score\_data = self.score\_box.currentText()

report = g\_database.get\_query(data, score\_data)

for count in range(24):

self.db.setItem(count, 0, QTableWidgetItem(None))

self.db.setItem(count, 1, QTableWidgetItem(None))

self.db.setItem(count, 2, QTableWidgetItem(None))

self.db.setItem(count, 3, QTableWidgetItem(None))

self.db.setItem(count, 4, QTableWidgetItem(None))

for record in report:

self.db.setItem(\_count, 0, QTableWidgetItem(record[0]))

self.db.setItem(\_count, 1, QTableWidgetItem(str(record[1])))

self.db.setItem(\_count, 2, QTableWidgetItem(str(record[2])))

self.db.setItem(\_count, 3, QTableWidgetItem(str(record[3])))

self.db.setItem(\_count, 4, QTableWidgetItem(str(record[4])))

\_count += 1

\end{python}

2 - A stepper variable is added to increment the row count for displaying the data in the QTableWidget.

3 - The contents of the task QComboBox is assigned to a variable which can be passed into the get\\_query method.

4 - The contents of the score QComboBox is assigned to a variable which can be passed into the get\\_query method.

5 - A list variable is assigned to the data which will be returned from the get\\_query method.

6 - A for loop is used to clear all values in the QTableWidget, which will only have twenty-four rows, one for each possible task that could be recorded.

7 - The value none is set for each space in the QTableWidget, removing all values from it.

12 - For every record fetched from the database, a for loop is used to add each record to the QTableWidget.

13 - The \\_count stepper variable is used to increment the row count so that each record is displayed on the next row rather than constantly over-writing the first row.

18 - The stepper variable increments.

\textbf{check\\_selected method (ParentLessonPage2 4.3.14)}

\begin{python}

def check\_selected(self):

if self.answer.text() == self.answer\_lesson:

self.answer.setText("{0} Correct".format(self.answer\_lesson))

else:

self.answer.setText("Incorrect")

self.answer.setReadOnly(True)

self.check.setEnabled(False)

\end{python}

2 - The algorithm checks to see if the contents of the line edit, i.e. the user's input, is the same as the hard-coded answer in the subclass lesson being used. If it is, it proceeds to tell the user they are correct, whilst leaving their input how it was (3).

4 - If they are not correct, it tells the user they are incorrect (5) and removes their wrong input.

6 - Disables the line edit so the user can no longer enter an input as they are already wrong and have been told the answer.

7 - Disables the check button so they cannot re-check the line edit which will now contain the text "Incorrect", which obviously won't change anyway.

\textbf{check\\_selected method (ParentHomeworkPage1Class (4.3.18)}

\begin{python}

def check\_selected(self):

self.allow\_cont = False

self.correct\_count = 0

if self.answer\_a.text() == self.answer\_1\_a:

self.answer\_a.setText("{0} Correct".format(self.answer\_a.text()))

self.correct\_count += 1

else:

self.answer\_a.setText("Incorrect")

if self.answer\_b.text() == self.answer\_1\_b:

self.answer\_b.setText("{0} Correct".format(self.answer\_b.text()))

self.correct\_count += 1

else:

self.answer\_b.setText("Incorrect")

if self.answer\_c.text() == self.answer\_1\_c:

self.answer\_c.setText("{0} Correct".format(self.answer\_c.text()))

self.correct\_count += 1

else:

self.answer\_c.setText("Incorrect")

if self.answer\_d.text() == self.answer\_1\_d:

self.answer\_d.setText("{0} Correct".format(self.answer\_d.text()))

self.correct\_count += 1

else:

self.answer\_d.setText("Incorrect")

if self.answer\_e.text() == self.answer\_1\_e:

self.answer\_e.setText("{0} Correct".format(self.answer\_e.text()))

self.correct\_count += 1

else:

self.answer\_e.setText("Incorrect")

if self.answer\_f.text() == self.answer\_1\_f:

self.answer\_f.setText("{0} Correct".format(self.answer\_f.text()))

self.correct\_count += 1

else:

self.answer\_f.setText("Incorrect")

for a in self.answers:

a.setReadOnly(True)

self.check.setEnabled(False)

self.reset.setEnabled(False)

self.allow\_cont = True

\end{python}

2 - Sets a boolean variable which determines whether or not the system will switch to the next screen based on whether or not the questions have all been answered.

3 - Sets a counter variable to keep track of how many of the line edit user inputs are correct.

4 - Checks to see of the contents of the first line edit, i.e. the user's input, is the same as the hard-coded answer in the homework subclass.

5 - If the input is correct, the user is told they are correct and a mark is awarded.

6 - The counter representing the number of marks is incremented.

7 - If the user input is wrong, the user is told they are wrong (8).

34 - Each line edit is in a list so the code used to change all of them the same way is only written once. A for loop is used for this.

35 - Each line edit is disabled so the user can no longer change their answer.

36 - The check button is disabled so they can't check either their right answer, which would then be interpreted as incorrect due to the additional text, or their wrong answer which would just keep printing "Incorrect" anyway.

37 - The reset button is disabled so they cannot reset all of the right answers or the incorrect messages as they have run out of attempts.

38 - Now that all of the answers have been checked, whether they were right or not, the allow\\_cont boolean is changed to true to allow the user to continue to the next screen.

\textbf{next\\_selected method (ParentHomeworkPage1Class 4.3.18)}

\begin{python}

def next\_selected(self):

cont = False

while not cont:

for a in self.answers:

if a.text() == "":

error\_message = ErrorMessage8()

error\_message.show()

error\_message.\_raise()

cont = False

cont = True

if self.allow\_cont:

g\_database.insert\_data\_first(self.task, self.correct\_count)

self.open\_page\_2()

self.hide()

else:

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

\end{python}

2 - A boolean variable is set which determines whether or not the system will switch screen based on whether or not the questions have all been answered.

3 - While the boolean is false, the algorithm will proceed to check if the line edit is blank, for each of the six line edits in the list (4, 5)

6 - If a line edit is blank, an error message will appear, and the algorithm will stop and allow the user to input an answer.

9 - The cont boolean remains false.

10 - If none of the line edits are blank, that means they have all been checked and the screen can switch.

11 - If the boolean variable from the check method is true, the insert\\_data\\_first method is run, saving the scores of the task as a record (12).

13 - The second screen in the stack is displayed.

15 - If the check method has not been run and the line edits have not been checked, it won't allow the user to proceed until they give an input.

16 - An error message is displayed asking the user to finish the questions.

\textbf{reset\\_selected method (ParentHomeworkPage1Class 4.3.18)}

\begin{python}

def reset\_selected(self):

self.answer\_a.setText(None)

self.answer\_b.setText(None)

self.answer\_c.setText(None)

self.answer\_d.setText(None)

self.answer\_e.setText(None)

self.answer\_f.setText(None)

\end{python}

2 - Simply sets the value of each line edit to none, effectively resetting the question so the user doesn't have to manually delete each input.

\textbf{check\\_button\\_1 method (HomeworkPage2ParentClass 4.3.19)}

\begin{python}

def check\_button\_1(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_1.text() == self.answer\_question\_4:

self.\_button\_1.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_1.setText("Incorrect")

self.\_button\_1.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

\end{python}

1 - The variable representing the number of attempts the user has left is passed through - This is so that the number doesn't reset if one of the other five methods for the other buttons is run (couldn't be subclassed) and it isn't in the method itself as that would effectively require it to be reset with every click.

2 - The variable representing the user's score is in the method because it needs to be reset everytime it is run otherwise it would just accumulate beyond a reasonable value.

3 - Checks to see if the text on the button is the same as the hard-coded answer in the homework subclass.

4 - If it is, it tells the user they are correct.

5 - All the buttons are disabled as they are no longer needed for input.

11 - The user is informed of their score.

12 - The user has a mark added.

13 - If the button text is not the same, the user is told they are incorrect (14).

15 - The incorrect button is disabled so they can't choose the same one again.

16 - The user loses an attempt.

17 - The button displaying the number of attempts the user has left is updated.

18 - If the user has no attempts left (attempts\\_remaining reaches 0), all of the multiple choice buttons are disabled (19).

25 - The button displaying the number of attempts the user has left is updated.

26 - The variable representing the number of attempts remaining is returned to be used in the check method for the other five buttons.

\textbf{selected\\_mark\\_2 method (HomeworkPage2ParentClass 4.3.19)}

\begin{python}

def selected\_mark\_2(self, attempts\_remaining\_a):

self.correct\_count\_2 = 0

if self.answer\_2.currentText() == "20":

self.correct\_count\_2 += 1

self.mark\_2.setText("Correct!")

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

else:

self.attempts\_remaining\_a -= 1

self.mark\_2.setText("Mark it|{0}".format(self.attempts\_remaining\_a))

if self.attempts\_remaining\_a == 0:

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

error\_message = ErrorMessage5()

error\_message.show()

error\_message.\_raise()

return self.attempts\_remaining\_a, self.correct\_count\_2

\end{python}

1 - The number representing the number of attempts the user has remaining is passed through so it isn't reset every time the method is run as it would if the variable was in the method.

2 - The correct\\_count variable counts the users score.

3 - Checks to see if the value of the text in the QComboBox is the same as the hard-coded answer in the homework subclass.

4 - If it is the same, the user recieves a mark.

5 - The user is told they are correct.

6 - The check button is disabled so they cannot exploit marks.

7 - The QComboBox is disabled so they cannot change their input.

8 - If the answer is not correct, the user loses an attempt (9).

10 - The text on the button which has the number or attempts remaining is updated.

11 - If the user has no attempts left, all of the inputs are disabled (12, 13).

14 - An error message tells the user they are wrong regardless of how many attempts they have left.

17 - The attempts\\_remaining is returned so it doesn't reset and is the same if the method is run again. The correct\\_count is returned to be passed into the insert\\_data\\_second method to be recorded in the database.

\textbf{selected\\_finish method (HomeworkPage2ParentClass 4.3.19)}

\begin{python}

def selected\_finish(self):

if self.attempts\_button.text() != "1 mark!" and self.attempts\_button.text() != "No more attempts":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_2.text() != "Correct!" and self.mark\_2.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_3.text() != "Correct!" and self.mark\_3.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

else:

g\_database.insert\_data\_second(self.task, self.correct\_count\_2, self.correct\_count\_3, self.correct\_count\_4)

self.parent.close()

\end{python}

2 - If the text of the attempts button displays neither of the texts it would display had the user used all of their attempts, it pops an error (3) as this means they have not answered the question or lost all of their attempts. The same goes for the other two questions.

14 - If all of the buttons have the text which signify that the questions have been properly answered, then the system proceeds to save the records to the database.

16 - The stack widget closes and the user is returned to the home screen.

\section{Settings}

There are no settings which the user or maintainer needs to be aware of; only Python 3 or later, and PyQt4 need to be installed on the computer for the entire program to run with no errors. Nothing will need to be changed or installed in order to develop additional content for the system in patches. Any changes that could possibly be made to this system would be possible using Python 3 and PyQt4. Anything using another software package would likely void some of the client's requirements.

\section{Acknowledgements}

\subsection{Pictures}

\begin{itemize}

\item \url{https://www.python.org/community/logos/} - Python Powered Logo - First Screen

\item \url{https://www.mathsisfun.com/pythagoras.html} - Pythagoras Picture - Home Screen

\item \url{http://mathinsight.org/vector\_introduction} - Vector Picture - Home Screen

\item \url{http://www.clipartpanda.com/categories/green-smiley-face-png} - Green Face - Home Screen

\item \url{https://commons.wikimedia.org/wiki/File:Triangle\_model\_of\_love.png} - 1st Triangle - Lesson Topic Menu

\item \url{https://en.wikipedia.org/wiki/Equilateral\_triangle} - 2nd Triangle - Lesson Topic Menu

\item \url{http://www.bbc.co.uk/bitesize/standard/maths\_i/measure/pythagoras/revision/1/} - 3rd Triangle - Lesson Topic Menu

\item \url{https://en.wikipedia.org/wiki/Triangle\_inequality} - 4th Triangle - Lesson Topic Menu

\item \url{https://en.wikipedia.org/wiki/Equilateral\_triangle} - 5th Triangle - Lesson Topic Menu

\item \url{http://passyworldofmathematics.com/trigonometric-ratios/} - 1st Picture - Trigonometry 1 Lesson Menu

\item \url{https://www.mathsisfun.com/algebra/sohcahtoa.html} - 2nd Picture - Trigonometry 1 Lesson Menu

\item \url{http://www.derivativesinvesting.net/article/262371065/trigonometry-finding-unknown-sides/} - SOHCAHTOA Lesson Page 2

\item \url{http://maths.nayland.school.nz/Year\_11/AS1.7\_Triangles/3\_trig\_angle.htm} - 1st Picture - Trigonometry 2 Lesson Menu

\item \url{http://revisionmaths.com/gcse-maths-revision/trigonometry/pythagorass-theorem} - 2nd Picture - Trigonometry 2 Lesson Menu

\item \url{http://www.pbs.org/wgbh/nova/proof/puzzle/theoremsans.html} - 1st Picture - Pythagoras Lesson Menu

\item \url{http://www.bbc.co.uk/schools/gcsebitesize/maths/geometry/pythagoras3drev1.shtml} - 2nd Picture - Pythagoras Lesson Menu

\item \url{http://www.kshitij-iitjee.com/Adding-Vectors-Subtracting-Vectors-equality-of-vectors} - 1st Picture - Vectors Lesson Menu

\item \url{http://www.maths.usyd.edu.au/u/MOW/vectors/vectors-3/v-3-3.html} - 2nd Picture - Vectors Lesson Menu

\item \url{https://en.wikipedia.org/wiki/Triangle} - 1st Picture - Summary Lesson Menu

\item \url{http://solvemymaths.com/2015/01/27/complements-7-the-pythagorean-theorem/} - 2nd Picture - Summary Lesson Menu

\item \url{http://stackoverflow.com/questions/18874339/deform-a-triangle-along-vector-to-get-a-specific-angle} - 3rd Picture - Summary Lesson Menu

\item \url{http://www.wikihow.com/Remember-the-Trigonometric-Table} - 1st Picture - Homework Topic Menu

\item \url{http://www.mathsaccelerator.com/measurement/trigonometry-cosine-rule} - 2nd Picture - Homework Topic Menu

\item \url{https://en.wikipedia.org/wiki/Pythagorean\_theorem} - 3rd Picture - Homework Topic Menu

\item \url{https://www.math.hmc.edu/calculus/tutorials/vectoranalysis/} - 4th Picture - Homework Topic Menu

\item \url{http://www.cafepress.co.uk/mf/30421745/i-love-trigonometry\_golf-shirt} - 5th Picture - Homework Topic Menu

\item \url{https://images.google.com/ - Unknown website} - 1st Picture - Trigonometry 1 Homework Menu

\item \url{https://images.google.com/ - Unknown website} - 2nd Picture - Trigonometry 1 Homework Menu

\item \url{https://images.google.com/ - Unknown website} - 3rd Picture - Trigonometry 1 Homework Menu

\item \url{http://passyworldofmathematics.com/lessons/page/5/} - 4th Picture - Trigonometry 1 Homework Menu

\item \url{http://passyworldofmathematics.com/lessons/page/5/} - 5th Picture - Trigonometry 1 Homework Menu

\item \url{http://passyworldofmathematics.com/lessons/page/5/} - 6th Picture - Trigonometry 1 Homework Menu

\item \url{https://images.google.com/ - Unknown website} - 1st Picture - Trigonometry 2 Homework Menu

\item \url{https://images.google.com/ - Unknown website} - 2nd Picture - Trigonometry 2 Homework Menu

\item \url{https://images.google.com/ - Unknown website} - 3rd Picture - Trigonometry 2 Homework Menu

\item \url{https://triglaws.wordpress.com/2011/03/08/problem-1-estimate-the-elevation/} - 4th Picture - Trigonometry 2 Homework Menu

\item \url{http://math.tutorcircle.com/geometry/right-triangle.html} - 5th Picture - Trigonometry 2 Homework Menu

\item \url{http://www.glogster.com/meredithgoh/emaths-project-trigonometry/g-6mshrggqu3136b3go4lu9a0} - 6th Picture - Trigonometry 2 Homework Menu

\item \url{http://www.bbc.co.uk/education/guides/zfbqtfr/revision/3} - 1st Picture - Pythagoras Homework Menu

\item \url{http://cribbd.com/question/how-do-i-use-trig-in-3d} - 2nd Picture - Pythagoras Homework Menu

\item \url{http://maths.nayland.school.nz/Year\_11/AS1.7\_Triangles/4\_trig\_apps.htm}

\item \url{http://phasesexperimental.weebly.com/trigonometry.html} - 1st Picture - Finding Angles Lesson Page 1

\item \url{http://www.bbc.co.uk/bitesize/standard/maths\_ii/trigonometry/equations/revision/1/} - 2nd Picture - Finding Angles Lesson Page 1

\item \url{http://www.onlinemathlearning.com/trig-graphs.html} - 1st Picture - Finding Angles Lesson Page 2

\end{itemize}

\subsection{Code Segments}

\textbf{The following code was taken from my current computing teacher's example on how to create a working database structure.}

\begin{python}

class Database:

def \_\_init\_\_(self, db\_name):

self.\_db\_name = db\_name

self.table\_name = "Student"

self.create\_table(self.table\_name)

def execute\_sql(self, sql):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute(sql)

def GetAllNames(self):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select \* from Student")

students = cursor.fetchall()

return students

count = 0

for student in students:

self.database.setItem(count, 0, QTableWidgetItem(student[0]))

self.database.setItem(count, 1, QTableWidgetItem(str(student[1])))

self.database.setItem(count, 2, QTableWidgetItem(str(student[2])))

self.database.setItem(count, 3, QTableWidgetItem(str(student[3])))

self.database.setItem(count, 4, QTableWidgetItem(str(student[4])))

count += 1

\end{python}

\textbf{The following code was taken from a video on this webpage: }

\url{http://www.pythonschool.net/databases/creating-the-data-model/}

\begin{python}

def create\_table(self, table\_name):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select name from sqlite\_master where name=?",(table\_name,))

result = cursor.fetchall()

keep\_table = True

if len(result) == 1:

response = input("The table {0} already exists, do you wish to recreate it (y/n): ".format(table\_name))

if response == "y":

keep\_table = False

print("The {0} table will be recreated - all existing data will be lost".format(table\_name))

cursor.execute("drop table if exists {0}".format(table\_name))

db.commit()

else:

print("The existing table was kept")

else:

keep\_table = False

if not keep\_table:

sql = """create table Student

(TaskID text,

Qone integer,

Qtwo integer,

Qthree integer,

Qfour integer,

primary key(TaskID))"""

cursor.execute(sql)

db.commit()

\end{python}

\textbf{The following code was taken and modified from this website: }

\url{http://www.qtforum.org/article/38014/qdockwidget-background.html}

\begin{python}

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

\end{python}

\textbf{The following code was taken from this website: }

\url{http://stackoverflow.com/questions/24659239/how-to-change-qpushbutton-text-and-background-color}

\begin{python}

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

\end{python}

\section{Code Listing}

\begin{landscape}

\subsection{MyWindow Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from student\_account\_home import \* #Contains the home screen class, needed to switch from the first screen to the home screen

from error\_messages import \* #Contains all the QErrorMessage classes.

from first\_screen\_widget import \* #This file has the class which creates the first screen

#in the initial stack

import sys

#This window is the main window of the system; it contains two widgets in a

#stack and accepts a pyqtsignal, then transfers to the next page.

class MyWindow(QMainWindow):

#The signal which activates the next window.

NameEntered = pyqtSignal()

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the first widget which opens when the program is run.

self.login\_widget = FirstScreen()

#This is the second widget in the stack.

self.student\_home = UserAccountWidget(self)

#Sets the layout to a stack.

self.stack = QStackedLayout()

#Adds the two widgets to a stack.

self.stack.addWidget(self.login\_widget)

self.stack.addWidget(self.student\_home)

self.widget = QWidget()

#This sets the stack layout as the program's layout.

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

#This is the connection for the continue button which uses the pyqtSignal().

#When pressed, it switches to the second window in the stack.

self.login\_widget.NameEntered.connect(self.enter\_program)

#The method which is run when the above connection is made.

def enter\_program(self):

#Sets the displayed window to self.student\_home.

self.stack.setCurrentIndex(1)

#Maximises the screen.

self.student\_home.showMaximized()

self.student\_home.raise\_()

#This makes the program start by default when run.

if \_\_name\_\_ == "\_\_main\_\_":

app = QApplication(sys.argv)

#Assigns the window to the MyWindow class made above.

window = MyWindow()

#Forces the window to be shown.

window.show()

window.raise\_()

#Maximises the window.

window.showMaximized()

app.exec\_()

\end{python}

\subsection{Database Class}

\begin{python}

import sqlite3 #This imports all of the built in sqlite3 code

#This is the class which generates the database

class Database:

#Constructor

def \_\_init\_\_(self, db\_name):

#Assigns the db\_name variable to whatever name is passed in when the class is called (in this case "student\_database.db")

self.\_db\_name = db\_name

#The name of the table - will be used to access this database using SQL queries

self.table\_name = "Student"

#Calls the method which creates the table called Student

self.create\_table(self.table\_name)

#This method is called everytime data is saved in the database so the code doesn't have to be written every time

def execute\_sql(self, sql):

#Opens a connection to student\_database.db

with sqlite3.connect(self.\_db\_name) as db:

#This line makes any addition to the database remain in the database, rather than just deleting it when the program stops running

cursor = db.cursor()

#This line runs the sql query which is passed in each time the method is called

cursor.execute(sql)

#This is the method called when the program is run - each time it can be kept or over-written

def create\_table(self, table\_name):

#connects to the student\_database.db

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

#Executes the SQL statement to check if the database already exists

cursor.execute("select name from sqlite\_master where name=?",(table\_name,))

#This fetches all of the results from the query (the database called Student) and assigns it to a variable so it can be used elsewhere

result = cursor.fetchall()

#Sets a boolean variable used to determine if the table will be deleted or not

keep\_table = True

#Checks if there is one table already called Student

if len(result) == 1:

#Gives the user the option to recreate the table

response = input("The table {0} already exists, do you wish to recreate it (y/n): ".format(table\_name))

#If they choose yes, the boolean changes and the table will be deleted

if response == "y":

keep\_table = False

print("The {0} table will be recreated - all existing data will be lost".format(table\_name))

#This SQL statement deletes the entire table

cursor.execute("drop table if exists {0}".format(table\_name))

#Commits the changes

db.commit()

else:

print("The existing table was kept")

else:

keep\_table = False

#If the user deleted the previous Student table another one will be created to replace it, exactly the same except for the data inside the table

if not keep\_table:

#This SQL statement creates the Student table with the following attributes

sql = """create table Student

(TaskID text,

Qone integer,

Qtwo integer,

Qthree integer,

Qfour integer,

primary key(TaskID))"""

#Executes the SQL statement by running the method at the top

cursor.execute(sql)

#Commits the changes (makes sure they stay)

db.commit()

#This method is called in the parent homework widget, and it saves the task name of the homework and the score from the first question,

#then fills the remaining columns with 0 so that the next method will work

def insert\_data\_first(self, task, correct\_count):

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

cursor.execute("select TaskID from Student where TaskID = '{0}'".format(task))

info = cursor.fetchall()

if len(info) != 0:

sql = "UPDATE Student SET Qone = '{0}' WHERE TaskID = '{1}' AND Qone < '{2}'".format(correct\_count, task, correct\_count)

else:

#Inserts the values

sql = "insert into Student(TaskID, Qone, Qtwo, Qthree, Qfour) values ('{0}', '{1}', '{2}', '{3}', '{4}')".format(task, correct\_count, str(0), str(0), str(0))

#Executes the SQL statement above by running the method at the top

self.execute\_sql(sql)

#This method is called in the parent homework page 2 widget, which saves the scores for the remaining questions

def insert\_data\_second(self, task, count\_2, count\_3, count\_4):

#Connects to the student\_database.db

with sqlite3.connect(self.\_db\_name) as db:

#These statements all update the value of the corresponding question score where the saved task is the same as the current task,

#so it will only overwrite the task being attempted

sql = "UPDATE Student SET Qtwo = '{0}' WHERE TaskID = '{1}' AND Qtwo < '{2}'".format(count\_2, task, count\_2)

self.execute\_sql(sql)

sql\_2 = "UPDATE Student SET Qthree = '{0}' WHERE TaskID = '{1}' AND Qthree < '{2}'".format(count\_3, task, count\_3)

self.execute\_sql(sql\_2)

sql\_3 = "UPDATE Student SET Qfour = '{0}' WHERE TaskID = '{1}' AND Qfour < '{2}'".format(count\_4, task, count\_4)

self.execute\_sql(sql\_3)

## sql\_4 = "UPDATE Student SET Total = '{0}%' WHERE TaskID = '{1}'".format(total, task)

## self.execute\_sql(sql\_4)

#This method fetches all of the relevant data from the database when the user makes a query i.e. searches for all the results for a task or score range

def get\_query(self, data, score\_data):

#Connects to the database

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

#Executes the SQL statement to search the database for anything with the same value as the data or score\_data variables

cursor.execute("select \* from Student WHERE TaskID = '{0}' or Qone = '{1}'".format(data, score\_data))

report = cursor.fetchall()

#Returns the results of the query so that they can be manipulated i.e. displayed in the secondary QTableWidget

return report

#This method is used in the database widget to display all data in the database as soon as it is opened

def GetAllNames(self):

#Connects to student\_database.db

with sqlite3.connect(self.\_db\_name) as db:

cursor = db.cursor()

#This SQL statement selects everything in the Student table

cursor.execute("select \* from Student")

#Everything in the cursor from the previous line is fetched and assigned a variable so that the data can be used

students = cursor.fetchall()

#Returns all data from the Student table

return students

#This variable is the database, constructed from the Database class, and is called whenever changes to the database are to be made

g\_database = Database("student\_database.db")

\end{python}

\subsection{DatabaseWidget class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from database\_class import \* #Needed to fetch all of the information from the database

from report\_widget import \* #Needed to connect to the report screen which opens when the report button is clicked

#This class is the template for the screen with the database view and the button for the report widget

class DatabaseWidget(QWidget):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#Maximises the screen

self.showMaximized()

#These four lines of code make the background of the widget white

pal = QPalette()

#Sets the chosen colour to white

pal.setColor(QPalette.Background, Qt.white)

#The screen will automatically be filled with the chosen colour

self.setAutoFillBackground(True)

self.setPalette(pal)

#This QLabel is the title on the screen

self.title = QLabel("Progress")

#This line sets the title font size and house style

self.title.setFont(QFont("Courier", 40))

#This QPushButton closes the screen

self.back = QPushButton("Return")

#Sets the minimum width of the QPushButton so that it will take up at least the required portion of the screen

self.back.setMinimumWidth(60)

#Sets the minimum height of the QPushButton so that it will take up at least the required portion of the screen

self.back.setMinimumHeight(100)

#Sets the font size and house style of the text in the button

self.back.setFont(QFont("Courier", 40))

#This overrides the background and text colour for the return button

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

#This button connects to the report window

self.report = QPushButton("Report")

self.report.setMinimumWidth(60)

self.report.setMinimumHeight(100)

self.report.setFont(QFont("Courier", 40))

#This QTableWidget is the table which the data from the database is presented in

self.database = QTableWidget()

#Sets the number of rows in the table - there are only 24 possible tasks, rach of which can only be recorded once and overwritten upon improvement

self.database.setRowCount(24)

#Sets the number of columns in the table - there are 5 headings, 5 attributes to record

self.database.setColumnCount(5)

#This is the header which appears in each column in the database

self.database\_header = ("Task Name", "Question 1", "Question 2", "Question 3", "Question 4")#"Total"

#This applies the header to the QTableWidget

self.database.setHorizontalHeaderLabels(self.database\_header)

#This sets the background colour and text colour for all of the QPushButtons on the page

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

#This sets the colour of the selected boxes in the QTableWidget

self.database.setStyleSheet("QTableView {selection-background-color: #A3C1DA;}")

#This overrides the background and text colour for the return button

#This calls a method in database\_class which fetches all of the data in the database

students = g\_database.GetAllNames()

#Every bit of data fetched from the database is allocated to the corresponding column in the QTableWidget

count = 0

for student in students:

#The TaskNames are put in the first column

self.database.setItem(count, 0, QTableWidgetItem(student[0]))

#The first question scores are put in the second column, etc.

self.database.setItem(count, 1, QTableWidgetItem(str(student[1])))

self.database.setItem(count, 2, QTableWidgetItem(str(student[2])))

self.database.setItem(count, 3, QTableWidgetItem(str(student[3])))

self.database.setItem(count, 4, QTableWidgetItem(str(student[4])))

count += 1

#Sets the layout of the page to a QGridLayout() so that every widget can be positioned where I want them easily

self.layout = QGridLayout()

#These four lines add the four widgets to the layout so they will appear when the screen is displayed.

self.layout.addWidget(self.title, 0, 0) #These numbers are for positioning the widgets e.g This one will be in the top left of the screen

self.layout.addWidget(self.database, 0, 1)

self.layout.addWidget(self.back, 4, 0)

self.layout.addWidget(self.report, 4, 1)

#Sets the layout as the layout to be displayed

self.setLayout(self.layout)

#When back is clicked the selected\_back method will be executed

self.back.clicked.connect(self.selected\_back)

#When report is clicked the selected\_report method will be executed

self.report.clicked.connect(self.selected\_report)

#Executed when back is clicked

def selected\_back(self):

#The entire progress screen disappears

self.close()

#Executed when report is clicked

def selected\_report(self):

#Assigns a variable to the ReportWidget() class

report\_widget = ReportWidget()

#The report widget will appear in front of the progress window

report\_widget.show()

report\_widget.\_raise()

\end{python}

\subsection{ReportWidget Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

from database\_widget import \* #This contains the data currently being displayed to the user

from database\_class import \* #This contains the methods which fetch the information from the database

#This is the template for the widget which is used to query the database

class ReportWidget(QWidget):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#Maximises the screen

self.showMaximized()

#This is just a label with the title of the window

self.header = QLabel("Report")

#Sets the font size and house style of the label

self.header.setFont(QFont("Courier", 30))

self.task\_box\_label = QLabel("Please select a task\nto query: ")

self.task\_box\_label.setFont(QFont("Courier", 25))

#This combo box contains all of the options which can be chosen from when

#querying the database for a task name

self.task\_box = QComboBox()

#Sets the size of the combo box

self.task\_box.setMinimumWidth(60)

self.task\_box.setMinimumHeight(100)

#Sets the font size and house style of the text in the combo box

self.task\_box.setFont(QFont("Courier", 30))

#Sets the background colour of the combo box

self.task\_box.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

#Adds the data which can be queried as options

self.task\_box.addItem("")

self.task\_box.addItem("Sides Easy")

self.task\_box.addItem("Sides Medium")

self.task\_box.addItem("Sides Hard")

self.task\_box.addItem("SOHCAHTOA Easy")

self.task\_box.addItem("SOHCAHTOA Medium")

self.task\_box.addItem("SOHCAHTOA Hard")

self.task\_box.addItem("Finding Angles Easy")

self.task\_box.addItem("Finding Angles Medium")

self.task\_box.addItem("Finding Angles Hard")

self.task\_box.addItem("3D Trigonometry Easy")

self.task\_box.addItem("3D Trigonometry Medium")

self.task\_box.addItem("3D Trigonometry Hard")

self.task\_box.addItem("Pythagoras' Theorem Easy")

self.task\_box.addItem("Pythagoras' Theorem Medium")

self.task\_box.addItem("Pythagoras' Theorem Hard")

self.task\_box.addItem("3D Pythagoras Easy")

self.task\_box.addItem("3D Pythagoras Medium")

self.task\_box.addItem("3D Pythagoras Hard")

self.task\_box.addItem("Vectors Easy")

self.task\_box.addItem("Vectors Medium")

self.task\_box.addItem("Vectors Hard")

self.task\_box.addItem("Easy Summary")

self.task\_box.addItem("Medium Summary")

self.task\_box.addItem("Hard Summary")

self.score\_box\_label = QLabel("Please input the maximum\nscore you would like\nto query: ")

self.score\_box\_label.setFont(QFont("Courier", 25))

#Essentially the same as the other combo box except with score ranges to select from

self.score\_box = QComboBox()

self.score\_box.setMinimumWidth(60)

self.score\_box.setMinimumHeight(100)

self.score\_box.setFont(QFont("Courier", 30))

self.score\_box.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.score\_box.addItem(None)

self.score\_box.addItem("6")

self.score\_box.addItem("5")

self.score\_box.addItem("4")

self.score\_box.addItem("3")

self.score\_box.addItem("2")

self.score\_box.addItem("1")

self.score\_box.addItem("0")

#This button closes the window

self.back = QPushButton("Return")

self.back.setMinimumWidth(60)

self.back.setMinimumHeight(100)

self.back.setFont(QFont("Courier", 30))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

#This button initiates the query method - fetches all the relevant data

self.submit = QPushButton("Query")

self.submit.setMinimumWidth(60)

self.submit.setMinimumHeight(100)

self.submit.setFont(QFont("Courier", 30))

self.submit.setStyleSheet("QPushButton {background-color: green; color: white;}")

#This is the table which displays the data which the user has queried when it is found

self.db = QTableWidget()

#Sets the number of rows in the table - only 24 possible tasks to find

self.db.setRowCount(24)

#Sets the number of columns in the table - there are 5 headers

self.db.setColumnCount(5)

#Sets the headers so that they match the database

self.db\_header = ("TaskName", "Question 1", "Question 2", "Question 3", "Question 4")

#Applies the header to the table

self.db.setHorizontalHeaderLabels(self.db\_header)

self.db.setStyleSheet("QTableWidget {selection-background-color: #A3C1DA;}")

#Sets the layout to a QGridLayout so that the widgets can be positioned easily

self.layout = QGridLayout()

##Sets layout as the layout to be used

self.setLayout(self.layout)

#Adds all of the widgets to the layout

self.layout.addWidget(self.db, 0, 0) #These numbers position the widget in the layout

self.layout.addWidget(self.task\_box\_label, 0, 1)

self.layout.addWidget(self.task\_box, 1, 1)

self.layout.addWidget(self.score\_box\_label, 2, 1)

self.layout.addWidget(self.score\_box, 3, 1)

self.layout.addWidget(self.back, 4, 0)

self.layout.addWidget(self.submit, 4, 1)

#The connections for returning to the previous screen or querying the database

self.back.clicked.connect(self.selected\_back)

self.submit.clicked.connect(self.selected\_submit)

#Closes the window

def selected\_back(self):

self.close()

#This method takes the input in the combo boxes are uses it to search the database

def selected\_submit(self):

\_count = 0

#data is the combo box selection and is passed into the database method

data = self.task\_box.currentText()

score\_data = self.score\_box.currentText()

#This method is in the database\_class

report = g\_database.get\_query(data, score\_data)

#This clears the contents of the table with each new query so it does'nt continue to

#display data that is no longer relevant

for count in range(24):

self.db.setItem(count, 0, QTableWidgetItem(None))

self.db.setItem(count, 1, QTableWidgetItem(None))

self.db.setItem(count, 2, QTableWidgetItem(None))

self.db.setItem(count, 3, QTableWidgetItem(None))

self.db.setItem(count, 4, QTableWidgetItem(None))

#The report variable represents all that was fetched from the database

for record in report:

#Each piece of information is displayed in the QTableWidget under the right headers

#so that it looks exactly the same as the actual database would

self.db.setItem(\_count, 0, QTableWidgetItem(record[0]))

self.db.setItem(\_count, 1, QTableWidgetItem(str(record[1])))

self.db.setItem(\_count, 2, QTableWidgetItem(str(record[2])))

self.db.setItem(\_count, 3, QTableWidgetItem(str(record[3])))

self.db.setItem(\_count, 4, QTableWidgetItem(str(record[4])))

\_count += 1

\end{python}

\subsection{FirstScreen Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

#This is the template for the very first screen which appears when the program is run

class FirstScreen(QWidget):

#This is the signal for the connection which switches to the second screen in the stack

#When the button is clicked, NameEntered becomes true and the connection is executed

NameEntered = pyqtSignal()

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#These four lines of code set the background of the screen to white

pal = QPalette()

#This line selects the colour for the background

pal.setColor(QPalette.Background, Qt.white)

#This line makes the background automatically change to the chosen colour whenever it is displayed

self.setAutoFillBackground(True)

#Sets the screen's palette to the one selected above

self.setPalette(pal)

self.message = QLabel("Welcome to the Triangle Geometry Education Program")

#This changes the QLabel's font size and house style

self.message.setFont(QFont("Courier", 40))

#This aligns the QLabel to the centre of the width of the screen

self.message.setAlignment(Qt.AlignCenter)

#This is the button which connects to the home screen, the second screen in the stack

self.cont = QPushButton("Continue")

#This sets the minimum height and width for the QPushButton so that different screen sizes can all work

self.cont.setMinimumHeight(110)

self.cont.setMinimumWidth(60)

#Changes the font size and house style of the text in the QPushButton

self.cont.setFont(QFont("Courier", 40))

#This is a picture

self.pic = QLabel()

#The picture is imported from the file below

self.pic.setPixmap(QPixmap("powered\_by\_python"))

#This centralises the picture in its designated area

self.pic.setAlignment(Qt.AlignCenter)

#Sets the layout to a QGridLayout so that all the widgets can be positioned easily

self.layout = QGridLayout()

#Sets layout as the layot used in the widget

self.setLayout(self.layout)

#Adds the widgets to the layout

self.layout.addWidget(self.pic, 0, 0) #The numbers here position the widgets

self.layout.addWidget(self.message, 1, 0)

self.layout.addWidget(self.cont, 2, 0)

#This sets the QPushButton's background colour and font colour

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

#This is the connection for the button, when clicked the enter method is executed

self.cont.clicked.connect(self.enter)

#This is run when the continue button is clicked

def enter(self):

#This is the pyqtSignal - when NameEntered is changed to true, the signal is emitted, and this switches the screento the home screen

self.NameEntered.emit()

\end{python}

\subsection{UserAccountWidget Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

import sys #This is needed for the exit button - to close the entire program

#The only built-in function that works exactly how it needs to here is from sys

from lesson\_menu\_widget import \* #Contains the lesson menu for when the lessons button is clicked

from homework\_menu\_widget import \* #Contains the homework menu for when the homework button is clicked

from database\_widget import \* #Contains the database widget for when the progress button is clicked

#This class is the template for the home screen; the second screen in the first stack

#when the program is first run

class UserAccountWidget(QWidget):

#Constructor

def \_\_init\_\_(self, parent):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent\_window = parent

#Sets the background colour of the window to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#This button connects to the lesson menu

self.lessons = QPushButton("Lessons")

#Sets the size of the buttons

self.lessons.setMinimumWidth(90)

self.lessons.setMinimumHeight(110)

#Sets the font size and house style of the text in the buttons

self.lessons.setFont(QFont("Courier", 40))

#This button connects to the homework menu

self.homework = QPushButton("Homework")

self.homework.setMinimumWidth(90)

self.homework.setMinimumHeight(110)

self.homework.setFont(QFont("Courier", 40))

#This button connects to the progress screen

self.progress = QPushButton("Progress")

self.progress.setMinimumWidth(90)

self.progress.setMinimumHeight(110)

self.progress.setFont(QFont("Courier", 40))

self.lessons\_label = QLabel("To view lessons\nand learn more,\nclick here! ")

#Sets the font size and house style in of the QLabel text

self.lessons\_label.setFont(QFont("Courier", 25))

self.homework\_label = QLabel("To access the\nhomework set for\nyou to complete,\nclick here! ")

self.homework\_label.setFont(QFont("Courier", 25))

self.database\_label = QLabel("To view your\nprogress so far,\nclick here! ")

self.database\_label.setFont(QFont("Courier", 25))

self.log\_out = QPushButton("Exit Program")

self.log\_out.setMinimumWidth(90)

self.log\_out.setMinimumHeight(110)

self.log\_out.setFont(QFont("Courier", 40))

#Overrides the style of the buttons in the window - exit button distinguished from

#other buttons

self.log\_out.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

#These are pictures - one for each button

self.picture = QLabel()

#This imports the picture from the below file

self.picture.setPixmap(QPixmap("student\_account\_home\_pic"))

#This aligns the picture to the centre of its designated area

self.picture.setAlignment(Qt.AlignCenter)

self.homework\_pic = QLabel()

self.homework\_pic.setPixmap(QPixmap("student\_home\_homework"))

self.homework\_pic.setAlignment(Qt.AlignCenter)

self.smiler = QLabel()

self.smiler.setPixmap(QPixmap("smile"))

self.smiler.setAlignment(Qt.AlignCenter)

#Sets the background colour and font colour of all of the buttons in the window

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

#Sets the layout to a QGridLayout so that the widgets are positioned easily

self.layout = QGridLayout()

#Adds the widgets to the layout

self.layout.addWidget(self.lessons, 0, 1)#These numbers position the widgets in the layout

self.layout.addWidget(self.picture, 0, 2)

self.layout.addWidget(self.homework, 1, 1)

self.layout.addWidget(self.progress, 2, 1)

self.layout.addWidget(self.lessons\_label, 0, 0)

self.layout.addWidget(self.homework\_label, 1, 0)

self.layout.addWidget(self.database\_label, 2, 0)

self.layout.addWidget(self.picture, 1, 3)

self.layout.addWidget(self.log\_out, 2, 3)

self.layout.addWidget(self.smiler, 2, 2)

self.layout.addWidget(self.homework\_pic, 1, 2)

#Sets layout as the layout to be used

self.setLayout(self.layout)

#These are the connections which connect to the next screens when the buttons are clicked

self.lessons.clicked.connect(self.selected\_lessons)

self.homework.clicked.connect(self.selected\_homework)

self.progress.clicked.connect(self.selected\_progress)

self.log\_out.clicked.connect(self.log\_out\_selected)

#This is where the 'import sys' is needed - it stops the entire program without

#displaying a message asking the user if they are sure they want to exit

def log\_out\_selected(self):

sys.exit()

#These are the methods from the connections

def selected\_lessons(self):

#Assigns a variable to the widget

lessonmenuwidget = LessonMenuWidget()

#This displays the window in front of the previous window

lessonmenuwidget.show()

lessonmenuwidget.\_raise()

lessonmenuwidget.showMaximized()

## self.parent\_window.close()

def selected\_homework(self):

homeworkmenuwidget = HomeworkMenuWidget()

homeworkmenuwidget.show()

homeworkmenuwidget.\_raise()

homeworkmenuwidget.showMaximized()

def selected\_progress(self):

databasewidget = DatabaseWidget()

databasewidget.show()

databasewidget.\_raise()

databasewidget.showMaximized()

\end{python}

\subsection{LessonMenuWidget Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

from derived\_lesson\_menus import \* #This contains all of the lesson menus which the buttons in the class connect to

#This is the template for the lesson topic menu (the one before the menus for each topic)

class LessonMenuWidget(QMainWindow):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#Maximises the screen when it is displayed

self.showMaximized()

#Sets the background colour of the window to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#These are the buttons which each connect to a different child menu

self.t1 = QPushButton("Trigonometry 1")

#Sets the size of hte buttons - having a minimum

# and maximum size helps prevent overlapping on different sizes of screen

self.t1.setMinimumWidth(90)

self.t1.setMinimumHeight(110)

#This sets the font size and house style of the text in the QPushButton

self.t1.setFont(QFont("Courier", 40))

self.t1\_pic = QLabel()

#Imports the image from the file below

self.t1\_pic.setPixmap(QPixmap("t1\_pic"))

#Aligns the image in the centre of its designated space

self.t1\_pic.setAlignment(Qt.AlignCenter)

self.t2 = QPushButton("Trigonometry 2")

self.t2.setMinimumWidth(90)

self.t2.setMinimumHeight(110)

self.t2.setFont(QFont("Courier", 40))

self.t2\_pic = QLabel()

self.t2\_pic.setPixmap(QPixmap("t2\_pic"))

self.t2\_pic.setAlignment(Qt.AlignCenter)

self.pyt = QPushButton("Pythagoras")

self.pyt.setMinimumWidth(90)

self.pyt.setMinimumHeight(110)

self.pyt.setFont(QFont("Courier", 40))

self.pyt\_pic = QLabel()

self.pyt\_pic.setPixmap(QPixmap("pyt\_pic"))

self.pyt\_pic.setAlignment(Qt.AlignCenter)

self.pytrig = QPushButton("Vectors")

self.pytrig.setMinimumWidth(90)

self.pytrig.setMinimumHeight(110)

self.pytrig.setFont(QFont("Courier", 40))

self.pytrig\_pic = QLabel()

self.pytrig\_pic.setPixmap(QPixmap("pytrig\_pic"))

self.pytrig\_pic.setAlignment(Qt.AlignCenter)

self.sum = QPushButton("Summary")

self.sum.setMinimumWidth(90)

self.sum.setMinimumHeight(110)

self.sum.setFont(QFont("Courier", 40))

self.sum\_pic = QLabel()

self.sum\_pic.setPixmap(QPixmap("sum\_pic"))

self.sum\_pic.setAlignment(Qt.AlignCenter)

#This button returns to the previous window, unlike the other buttons,

#so it is a different colour to make it clear that it serves a different

#purpose

self.back = QPushButton("Return")

self.back.setMinimumWidth(90)

self.back.setMinimumHeight(110)

self.back.setFont(QFont("Courier", 40))

#This overrides the style of the other buttons

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.lesson\_label = QLabel("Lessons")

self.lesson\_label.setFont(QFont("Courier", 40))

self.select = QLabel("Please select a topic: ")

self.select.setFont(QFont("Courier", 25))

self.title\_pic = QLabel()

self.title\_pic.setPixmap(QPixmap("title\_lessons"))

#This sets the background colour and font colour of all the QPushButtons

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

#Sets the layout as a QGridLayout so all the widgets can be positioned easily

self.layout = QGridLayout()

#Adds all of the widgets to the layout

self.layout.addWidget(self.title\_pic, 0, 0) #These numbers position the widgets

self.layout.addWidget(self.t1\_pic, 1, 0)

self.layout.addWidget(self.t1, 1, 1)

self.layout.addWidget(self.t2, 2, 0)

self.layout.addWidget(self.t2\_pic, 2, 1)

self.layout.addWidget(self.pyt\_pic, 3, 0)

self.layout.addWidget(self.pyt, 3, 1)

self.layout.addWidget(self.pytrig, 4, 0)

self.layout.addWidget(self.pytrig\_pic, 4, 1)

self.layout.addWidget(self.sum\_pic, 5, 0)

self.layout.addWidget(self.sum, 5, 1)

self.layout.addWidget(self.back, 6, 0)

#These 3 lines set \_centralwidget as the layout to be used

#It needs to be declared as a QWidget because the class it's in is a QMainWindow

self.\_centralwidget = QWidget()

self.\_centralwidget.setLayout(self.layout)

self.setCentralWidget(self.\_centralwidget)

#The connections for the buttons

self.t1.clicked.connect(self.selected\_t1) #These are the methods executed when the button is clicked

self.t2.clicked.connect(self.selected\_t2)

self.pyt.clicked.connect(self.selected\_pyt)

self.pytrig.clicked.connect(self.selected\_pytrig)

self.sum.clicked.connect(self.selected\_sum)

self.back.clicked.connect(self.selected\_back)

#These open the selected menus and display them

def selected\_t1(self):

trig\_1\_widget = Trigonometry1()

trig\_1\_widget.show()

trig\_1\_widget.\_raise()

trig\_1\_widget.showMaximized()

def selected\_t2(self):

trig\_2\_widget = Trigonometry2()

trig\_2\_widget.show()

trig\_2\_widget.\_raise()

trig\_2\_widget.showMaximized()

def selected\_pyt(self):

pythagoras\_widget = Pythagoras()

pythagoras\_widget.show()

pythagoras\_widget.\_raise()

pythagoras\_widget.showMaximized()

def selected\_pytrig(self):

pyth\_trig\_widget = PythagTrig()

pyth\_trig\_widget.show()

pyth\_trig\_widget.\_raise()

pyth\_trig\_widget.showMaximized()

def selected\_sum(self):

summary\_widget = Summary()

summary\_widget.show()

summary\_widget.\_raise()

summary\_widget.showMaximized()

#This closes the window and returns the user to the previous window

def selected\_back(self):

self.close()

\end{python}

\subsection{HomeworkMenuWidget Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from derived\_homework\_menus import \* #This contains the topic specific menu classes which open when the buttons are clicked

#This is the template for the homework topic menu (before the individual homework menus)

class HomeworkMenuWidget(QMainWindow):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#Maximises the screen

self.showMaximized()

#Changes the background colour of the window to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

self.title = QLabel()

self.title.setFont(QFont("Courier", 40))

#These buttons all connect to a child menu class, which inherit from the HomeworkMenuParentClass

self.ht1 = QPushButton("Trigonometry 1")

#Sets the minimum width and height of the button to reduce problems with different screen sizes

self.ht1.setMinimumWidth(90)

self.ht1.setMinimumHeight(110)

#Sets the font size and house style of the text in the QPushButton

self.ht1.setFont(QFont("Courier", 40))

self.ht2 = QPushButton("Trigonometry 2")

self.ht2.setMinimumWidth(90)

self.ht2.setMinimumHeight(110)

self.ht2.setFont(QFont("Courier", 40))

self.hpyt = QPushButton("Pythagoras")

self.hpyt.setMinimumWidth(90)

self.hpyt.setMinimumHeight(110)

self.hpyt.setFont(QFont("Courier", 40))

self.hpytrig = QPushButton("Vectors")

self.hpytrig.setMinimumWidth(90)

self.hpytrig.setMinimumHeight(110)

self.hpytrig.setFont(QFont("Courier", 40))

self.hsum = QPushButton("Summary")

self.hsum.setMinimumWidth(90)

self.hsum.setMinimumHeight(110)

self.hsum.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumWidth(90)

self.back.setMinimumHeight(110)

self.back.setFont(QFont("Courier", 40))

#Overrides the style of the other buttons so the user can distinguish it from the homework menu buttons easily

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

#Each button has a picture with it

self.ht1\_pic = QLabel()

#This imports the picture from the file name below

self.ht1\_pic.setPixmap(QPixmap("homework\_trig\_1\_pic"))

#This aligns the picture to the centre of its designated area

self.ht1\_pic.setAlignment(Qt.AlignCenter)

self.ht2\_pic = QLabel()

self.ht2\_pic.setPixmap(QPixmap("homework\_trig\_2\_pic"))

self.ht2\_pic.setAlignment(Qt.AlignCenter)

self.hpyt\_pic = QLabel()

self.hpyt\_pic.setPixmap(QPixmap("homework\_pythag\_pic"))

self.hpyt\_pic.setAlignment(Qt.AlignCenter)

self.hpytrig\_pic = QLabel()

self.hpytrig\_pic.setPixmap(QPixmap("homework\_vectors\_pic"))

self.hpytrig\_pic.setAlignment(Qt.AlignCenter)

self.hsum\_pic = QLabel()

self.hsum\_pic.setPixmap(QPixmap("homework\_summary\_pic"))

self.hsum\_pic.setAlignment(Qt.AlignCenter)

#This sets the bacground colour and font colour of all of the buttons in the window

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue; font-size: 20;}")

#This sets the layout as a QGridLayout so that each widget can be positioned easily

self.layout = QGridLayout()

#These add the widgets to the layout

self.layout.addWidget(self.title, 0, 0) #These numbers position the widget in the window

self.layout.addWidget(self.ht1\_pic, 2, 0)

self.layout.addWidget(self.ht1, 2, 1)

self.layout.addWidget(self.ht2, 3, 0)

self.layout.addWidget(self.ht2\_pic, 3, 1)

self.layout.addWidget(self.hpyt\_pic, 4, 0)

self.layout.addWidget(self.hpyt, 4, 1)

self.layout.addWidget(self.hpytrig, 5, 0)

self.layout.addWidget(self.hpytrig\_pic, 5, 1)

self.layout.addWidget(self.hsum\_pic, 6, 0)

self.layout.addWidget(self.hsum, 6, 1)

self.layout.addWidget(self.back, 7, 0)

#This chunk of code sets the centralwidget as the layout - it has to use a QWidget because this class is a QMainWindow

self.\_centralwidget = QWidget()

self.\_centralwidget.setLayout(self.layout)

self.setCentralWidget(self.\_centralwidget)

#These are the connections for the buttons - the methods are executed when they are clicked

self.ht1.clicked.connect(self.selected\_ht1)

self.ht2.clicked.connect(self.selected\_ht2)

self.hpyt.clicked.connect(self.selected\_hpyt)

self.hpytrig.clicked.connect(self.selected\_hpytrig)

self.hsum.clicked.connect(self.selected\_hsum)

self.back.clicked.connect(self.selected\_back)

#These are the methods executed when the buttons are clicked

def selected\_ht1(self):

#This assigns a variable to a widget then opens and displays the widget

trigonometry\_1\_homework = Trigonometry1HW()

trigonometry\_1\_homework.show()

trigonometry\_1\_homework.\_raise()

trigonometry\_1\_homework.showMaximized()

def selected\_ht2(self):

trigonometry\_2\_homework = Trigonometry2HW()

trigonometry\_2\_homework.show()

trigonometry\_2\_homework.\_raise()

trigonometry\_2\_homework.showMaximized()

def selected\_hpyt(self):

pythagoras\_homework = PythagorasHW()

pythagoras\_homework.show()

pythagoras\_homework.\_raise()

pythagoras\_homework.showMaximized()

def selected\_hpytrig(self):

pythag\_trig\_homework = PythagTrigonometryHW()

pythag\_trig\_homework.show()

pythag\_trig\_homework.\_raise()

pythag\_trig\_homework.showMaximized()

def selected\_hsum(self):

summary\_homework = SummaryHW()

summary\_homework.show()

summary\_homework.\_raise()

summary\_homework.showMaximized()

def selected\_back(self):

#This closes the entire window

self.close()

\end{python}

\subsection{ParentLessonMenu Class}

\begin{python}

from PyQt4.QtCore import \* #These two lines import all of the built in PyQt code

from PyQt4.QtGui import \*

#This is the parent class which provides the default attributes for the 5 child lesson menus

class ParentLessonMenu(QWidget):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This maximises the window of each child class

self.showMaximized()

#This sets the background colour of each window

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#The QLabel is declared here because all 5 child menus will have one

#The QPixmap is set in the child class

self.title = QLabel()

#All of the widgets have either 2 or 3 options to select, so 3 buttons are created

#here, then either 2 or 3 can be addded to the layout in the child class

#The text is set in the child class

self.button\_1 = QPushButton()

#Sets the size of the button

self.button\_1.setMinimumHeight(110)

self.button\_1.setMinimumWidth(60)

#Sets the font size and house style of the text in the button

self.button\_1.setFont(QFont("Courier", 40))

self.button\_2 = QPushButton()

self.button\_2.setMinimumHeight(110)

self.button\_2.setMinimumWidth(60)

self.button\_2.setFont(QFont("Courier", 40))

self.button\_3 = QPushButton()

self.button\_3.setMinimumHeight(110)

self.button\_3.setMinimumWidth(60)

self.button\_3.setFont(QFont("Courier", 40))

self.back = QPushButton("Return")

self.back.setMinimumHeight(100)

self.back.setMinimumWidth(60)

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

#Sets the background colour and font colour of all the buttons in all of the child classes

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue}")

#Sets the layout to a QGridLayout so the widgets can be positioned easily

self.layout = QGridLayout()

#Sets layout as the layout to be used

self.setLayout(self.layout)

#The back button is added here because it is in the same place in every child class

self.layout.addWidget(self.back, 3, 0) #These numbers position the widget at the bottm

#it doesn't matter whether there are two or three buttons above,

#as the program ignores gaps by default

#This connection is the same in each child class so it is written here, as is the method

self.back.clicked.connect(self.selected\_back)

#This closes the window and returns the user to the previous screen

def selected\_back(self):

self.close()

\end{python}

\subsection{ParentHomeworkMenuClass Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

#This is the template for the parent homework menu, which provides default attributes for 5 child homework menu classes

class ParentHomeworkMenuClass(QWidget):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This maximises the screen - each child class will be maximised when displayed

self.showMaximized()

#This sets the background colour to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#Each child class has a title in the form of a picture, so the QLabel is defined here and the QPixmap is defined in each separate child class

self.title = QLabel()

#Each of the child classes have either 3 or 6 buttons which connect to a homework

#The buttons are defined here, and all have the same characteristics, and the text is added in the child classes

#This way either 3 or 6 buttons can be added to the layout in each child class

self.button\_1 = QPushButton()

self.button\_1.setMinimumHeight(110)

self.button\_1.setMinimumWidth(60)

self.button\_1.setFont(QFont("Courier", 40))

self.button\_2 = QPushButton()

self.button\_2.setMinimumHeight(110)

self.button\_2.setMinimumWidth(60)

self.button\_2.setFont(QFont("Courier", 40))

self.button\_3 = QPushButton()

self.button\_3.setMinimumHeight(110)

self.button\_3.setMinimumWidth(60)

self.button\_3.setFont(QFont("Courier", 40))

self.button\_4 = QPushButton()

self.button\_4.setMinimumHeight(110)

self.button\_4.setMinimumWidth(60)

self.button\_4.setFont(QFont("Courier", 40))

self.button\_5 = QPushButton()

self.button\_5.setMinimumHeight(110)

self.button\_5.setMinimumWidth(60)

self.button\_5.setFont(QFont("Courier", 40))

self.button\_6 = QPushButton()

self.button\_6.setMinimumHeight(110)

self.button\_6.setMinimumWidth(60)

self.button\_6.setFont(QFont("Courier", 40))

#This button appears in every child class

self.back = QPushButton("Return")

self.back.setMinimumHeight(100)

self.back.setMinimumWidth(60)

self.back.setFont(QFont("Courier", 40))

#The colour of this one is different so it can be distinguished between the buttons that take you to a homework

self.back.setStyleSheet("QPushButton {background-color: red; color: white;}")

#Each child class has 1 picture for every homework button, so the QLabels are created here,

#and the desired amount of them can be added to the layout, depending on the number of buttons. The QPixmap is defined in the child class

self.pic\_1 = QLabel()

#This aligns the picture to the centre of its designated area, which is required for every picture in every child class

self.pic\_1.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setAlignment(Qt.AlignCenter)

self.pic\_3 = QLabel()

self.pic\_3.setAlignment(Qt.AlignCenter)

self.pic\_4 = QLabel()

self.pic\_4.setAlignment(Qt.AlignCenter)

self.pic\_5 = QLabel()

self.pic\_5.setAlignment(Qt.AlignCenter)

self.pic\_6 = QLabel()

self.pic\_6.setAlignment(Qt.AlignCenter)

#This sets the background colour and font colour for every button in each child class, except for the back button which is overridden

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue}")

#This sets the layout to a QGridLayout so that each widget can be positioned easily

self.layout = QGridLayout()

#This sets layout as the layout to be used

self.setLayout(self.layout)

#Only these two widgets are added here, because they are the same across each child class

self.layout.addWidget(self.title, 0, 0)

#The back button will be on the bottom one space beneath the next widget regardless of how many buttons are on the screen. No extra gaps are included

self.layout.addWidget(self.back, 10, 0)

#The buttons and pictures are added in the child class because 4, 5 and 6 of each won't always be used

#This is in the parent class because it is the same for each child class; the window just closes and the previous menu is displayed

self.back.clicked.connect(self.selected\_back)

#The other connections are in the child classes because they all have to connect to different homeworks stacks

#The method which is executed when back is selected in any of the 5 child classes

def selected\_back(self):

#This closes the window - the previous window will still be there

self.close()

\end{python}

\subsection{Derived Lesson Menus}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from parent\_lesson\_menu import \* #This imports the parent class from which the following 5 classes inherit all of their default attributes

from lesson\_stacks import \* #This imports the lesson stacks for the connections to open when the buttons are clicked

\end{python}

\subsubsection{Trigonometry1 Class}

\begin{python}

#This is the template for the trigonometry 1 lesson menu, most of which is defined in the parent class ParentLessonMenu

class Trigonometry1(ParentLessonMenu):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#The buttons are created in the parent class

#The text is set here so that they can be different in each of the 5 child classes

self.button\_1.setText("Sides")

self.button\_2.setText("SOHCAHTOA")

#The QLabels are defined here because pictures aren't necessarily included in every child class

self.pic = QLabel()

self.pic.setPixmap(QPixmap("trig\_1\_pic"))

#Aligns the picture to the center of the area to which it is positioned

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("trig\_1\_pic\_2"))

self.pic\_2.setAlignment(Qt.AlignCenter)

#The widgets are added to the layout here so that there are no overlaps from the parent class

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_2, 1, 1)

self.layout.addWidget(self.button\_1, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

#The connections are added in the child class because each button connects to different stack widgets depending on which menu it is

self.button\_1.clicked.connect(self.SidesAHO)

self.button\_2.clicked.connect(self.SOHCAHTOA)

#Each of these methods, called in the above connections, opens a different stack widget which is why they are declared in the child classes

def SidesAHO(self):

sides\_aho = Trig1StackSides()

sides\_aho.show()

sides\_aho.\_raise()

def SOHCAHTOA(self):

sohcahtoa = Trig1StackSOHCAHTOA()

sohcahtoa.show()

sohcahtoa.\_raise()

\end{python}

\subsubsection{Trigonometry 2 Class}

\begin{python}

#Essentially the same as the above class except with different images, button text and lesson stack connections

class Trigonometry2(ParentLessonMenu):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.button\_1.setText("Finding Angles")

self.button\_2.setText("3D Trigonometry")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("trig\_2\_pic\_1"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("trig\_2\_pic\_3"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_1, 1, 1)

self.layout.addWidget(self.button\_2, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.layout.addWidget(self.back, 3, 0)

self.button\_1.clicked.connect(self.FindingAngles)

self.button\_2.clicked.connect(self.ThreeDTrigonometry)

def FindingAngles(self):

finding\_angles = Trig2StackFA()

finding\_angles.show()

finding\_angles.\_raise()

def ThreeDTrigonometry(self):

three\_d\_trig = Trig2StackTDT()

three\_d\_trig.show()

three\_d\_trig.\_raise()

\end{python}

\subsubsection{Pythagoras Class}

\begin{python}

class Pythagoras(ParentLessonMenu):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.button\_1.setText("Pythagoras' Theorem")

self.button\_2.setText("3D Pythagoras")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("pythag\_pic\_1"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("pythag\_pic\_2"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_1, 1, 1)

self.layout.addWidget(self.button\_2, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.button\_1.clicked.connect(self.PythagTheorem)

self.button\_2.clicked.connect(self.ThreeDPythagoras)

def PythagTheorem(self):

pythag\_theorem = PythagStackTheorem()

pythag\_theorem.show()

pythag\_theorem.\_raise()

def ThreeDPythagoras(self):

three\_d\_pythag = PythagStackTDP()

three\_d\_pythag.show()

three\_d\_pythag.\_raise()

\end{python}

\subsubsection{PythagTrig Class}

\begin{python}

class PythagTrig(ParentLessonMenu):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.button\_1.setText("Vectors 1")

self.button\_2.setText("Vectors 2")

self.button\_3.setText("Vectors 3")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("vectors\_pic\_1"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("vectors\_pic\_2"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_1, 1, 1)

self.layout.addWidget(self.button\_2, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.layout.addWidget(self.button\_3, 3, 1)

self.layout.addWidget(self.back, 4, 0)

self.button\_1.clicked.connect(self.Easy)

self.button\_2.clicked.connect(self.Medium)

self.button\_3.clicked.connect(self.Hard)

def Easy(self):

easy = EasyStack()

easy.show()

easy.\_raise()

def Medium(self):

medium = MediumStack()

medium.show()

medium.\_raise()

def Hard(self):

hard = HardStack()

hard.show()

hard.\_raise()

\end{python}

\subsubsection{Summary Class}

\begin{python}

class Summary(ParentLessonMenu):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.button\_1.setText("Easy")

self.button\_2.setText("Medium")

self.button\_3.setText("Hard")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("summary\_pic\_1"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("summary\_pic\_2"))

self.pic\_2.setAlignment(Qt.AlignCenter)

self.pic\_3 = QLabel()

self.pic\_3.setPixmap(QPixmap("summary\_pic\_3"))

self.pic\_3.setAlignment(Qt.AlignCenter)

self.layout.addWidget(self.title, 0, 0)

self.layout.addWidget(self.pic, 1, 0)

self.layout.addWidget(self.button\_1, 1, 1)

self.layout.addWidget(self.button\_2, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

self.layout.addWidget(self.pic\_3, 3, 0)

self.layout.addWidget(self.button\_3, 3, 1)

self.layout.addWidget(self.back, 4, 0)

self.button\_1.clicked.connect(self.ReviseTrig1)

self.button\_2.clicked.connect(self.ReviseTrig2)

self.button\_3.clicked.connect(self.ReviseTrig3)

def ReviseTrig1(self):

revise\_trig\_1 = Revise1Stack()

revise\_trig\_1.show()

revise\_trig\_1.\_raise()

def ReviseTrig2(self):

revise\_trig\_2 = Revise2Stack()

revise\_trig\_2.show()

revise\_trig\_2.\_raise()

def ReviseTrig3(self):

revise\_trig\_3 = Revise3Stack()

revise\_trig\_3.show()

revise\_trig\_3.\_raise()

\end{python}

\subsection{Derived Homework Menus}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from homework\_menu\_parent\_class import \* #This contains the parent template for each of the 5 menus to take from

from homework\_stacks import \* #This has the templates for the stacks which are opened from each button which is clicked

\end{python}

\subsubsection{Trigonometry1HW}

\begin{python}

#This is the template for the trigonometry 1 menu which takes most of its attributes from the ParentHomeworkMenuClass parent class

class Trigonometry1HW(ParentHomeworkMenuClass):

##constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is an image - the words trigonometry 1 in large bold black text

self.title.setPixmap(QPixmap("trig\_1\_title"))

#These buttons have already been created in the parent class

#The text is set here so that it can be different across each of the 5 child menus

self.button\_1.setText("Sides Easy")

self.button\_2.setText("Sides Medium")

self.button\_3.setText("Sides Hard")

self.button\_4.setText("SOHCAHTOA Easy")

self.button\_5.setText("SOHCAHTOA Medium")

self.button\_6.setText("SOHCAHTOA Hard")

#The QLabels have already been created in the parent class

#The pixmaps are set here so that they can be different across each of the 5 child classes

self.pic\_1.setPixmap(QPixmap("trig\_1\_pic\_1\_h"))

self.pic\_2.setPixmap(QPixmap("trig\_1\_pic\_2\_h"))

self.pic\_3.setPixmap(QPixmap("trig\_1\_pic\_3\_h"))

self.pic\_4.setPixmap(QPixmap("trig\_1\_pic\_4\_h"))

self.pic\_5.setPixmap(QPixmap("trig\_1\_pic\_5\_h"))

self.pic\_6.setPixmap(QPixmap("trig\_1\_pic\_6\_h"))

#The layouts are set in the child classes so that nothing overlaps from the parent class

self.layout.addWidget(self.button\_1, 1, 0)

self.layout.addWidget(self.pic\_1, 1, 1)

self.layout.addWidget(self.pic\_2, 2, 0)

self.layout.addWidget(self.button\_2, 2, 1)

self.layout.addWidget(self.button\_3, 3, 0)

self.layout.addWidget(self.pic\_3, 3, 1)

self.layout.addWidget(self.pic\_4, 4, 0)

self.layout.addWidget(self.button\_4, 4, 1)

self.layout.addWidget(self.button\_5, 5, 0)

self.layout.addWidget(self.pic\_5, 5, 1)

self.layout.addWidget(self.pic\_6, 6, 0)

self.layout.addWidget(self.button\_6, 6, 1)

#The connections are here because the stack widgets that each button needs to connect to are different in each child class

self.button\_1.clicked.connect(self.sides\_aho\_easy)

self.button\_2.clicked.connect(self.sides\_aho\_medium)

self.button\_3.clicked.connect(self.sides\_aho\_hard)

self.button\_4.clicked.connect(self.sohcahtoa\_easy)

self.button\_5.clicked.connect(self.sohcahtoa\_medium)

self.button\_6.clicked.connect(self.sohcahtoa\_hard)

#These are the methods which open the homework stack widgets when the corresponding button is clicked

def sides\_aho\_easy(self):

#Opens and displays the Sides Easy homework stack

sides\_aho\_1 = Trig1StackSidesEasy()

sides\_aho\_1.show()

sides\_aho\_1.\_raise()

def sides\_aho\_medium(self):

sides\_aho\_2 = Trig1StackSidesMedium()

sides\_aho\_2.show()

sides\_aho\_2.\_raise()

def sides\_aho\_hard(self):

sides\_aho\_3 = Trig1StackSidesHard()

sides\_aho\_3.show()

sides\_aho\_3.\_raise()

def sohcahtoa\_easy(self):

sohcahtoa\_1 = Trig1StackSOHCAHTOAEasy()

sohcahtoa\_1.show()

sohcahtoa\_1.\_raise()

def sohcahtoa\_medium(self):

sohcahtoa\_2 = Trig1StackSOHCAHTOAMedium()

sohcahtoa\_2.show()

sohcahtoa\_2.\_raise()

def sohcahtoa\_hard(self):

sohcahtoa\_3 = Trig1StackSOHCAHTOAHard()

sohcahtoa\_3.show()

sohcahtoa\_3.\_raise()

\end{python}

\subsubsection{Trigonometry2HW}

\begin{python}

#Everything is essentially the same as the above class except with different text on the buttons, different images and connections to different stack widgets

class Trigonometry2HW(ParentHomeworkMenuClass):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title.setPixmap(QPixmap("trig\_2\_title"))

self.button\_1.setText("Finding Angles Easy")

self.button\_2.setText("Finding Angles Medium")

self.button\_3.setText("Finding Angles Hard")

self.button\_4.setText("3D Trigonometry Easy")

self.button\_5.setText("3D Trigonometry Medium")

self.button\_6.setText("3D Trigonometry Hard")

self.pic\_1.setPixmap(QPixmap("trig\_2\_pic\_1\_h"))

self.pic\_2.setPixmap(QPixmap("trig\_2\_pic\_2\_h"))

self.pic\_3.setPixmap(QPixmap("trig\_2\_pic\_3\_h"))

self.pic\_4.setPixmap(QPixmap("trig\_2\_pic\_7\_h"))

self.pic\_5.setPixmap(QPixmap("trig\_2\_pic\_8\_h"))

self.pic\_6.setPixmap(QPixmap("trig\_2\_pic\_9\_h"))

self.layout.addWidget(self.button\_1, 1, 0)

self.layout.addWidget(self.pic\_1, 1, 1)

self.layout.addWidget(self.pic\_2, 2, 0)

self.layout.addWidget(self.button\_2, 2, 1)

self.layout.addWidget(self.button\_3, 3, 0)

self.layout.addWidget(self.pic\_3, 3, 1)

self.layout.addWidget(self.pic\_4, 4, 0)

self.layout.addWidget(self.button\_4, 4, 1)

self.layout.addWidget(self.button\_5, 5, 0)

self.layout.addWidget(self.pic\_5, 5, 1)

self.layout.addWidget(self.pic\_6, 6, 0)

self.layout.addWidget(self.button\_6, 6, 1)

self.button\_1.clicked.connect(self.finding\_angles\_easy)

self.button\_2.clicked.connect(self.finding\_angles\_medium)

self.button\_3.clicked.connect(self.finding\_angles\_hard)

self.button\_4.clicked.connect(self.three\_d\_trig\_easy)

self.button\_5.clicked.connect(self.three\_d\_trig\_medium)

self.button\_6.clicked.connect(self.three\_d\_trig\_hard)

def finding\_angles\_easy(self):

finding\_angles\_1 = Trig2StackFindingAnglesEasy()

finding\_angles\_1.show()

finding\_angles\_1.\_raise()

def finding\_angles\_medium(self):

finding\_angles\_2 = Trig2StackFindingAnglesMedium()

finding\_angles\_2.show()

finding\_angles\_2.\_raise()

def finding\_angles\_hard(self):

finding\_angles\_3 = Trig2StackFindingAnglesHard()

finding\_angles\_3.show()

finding\_angles\_3.\_raise()

def three\_d\_trig\_easy(self):

three\_d\_trig\_1 = Trig2StackTDTEasy()

three\_d\_trig\_1.show()

three\_d\_trig\_1.\_raise()

def three\_d\_trig\_medium(self):

three\_d\_trig\_2 = Trig2StackTDTMedium()

three\_d\_trig\_2.show()

three\_d\_trig\_2.\_raise()

def three\_d\_trig\_hard(self):

three\_d\_trig\_3 = Trig2StackTDTHard()

three\_d\_trig\_3.show()

three\_d\_trig\_3.\_raise()

\end{python}

\subsubsection{PythagorasHW}

\begin{python}

class PythagorasHW(ParentHomeworkMenuClass):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title.setPixmap(QPixmap("pythag\_title"))

self.button\_1.setText("Pythagoras' Theorem Easy")

self.button\_2.setText("Pythagoras' Theorem Medium")

self.button\_3.setText("Pythagoras' Theorem Hard")

self.button\_4.setText("3D Pythagoras Easy")

self.button\_5.setText("3D Pythagoras Medium")

self.button\_6.setText("3D Pythagoras Hard")

self.pic\_1.setPixmap(QPixmap("pythag\_pic\_1\_h"))

self.pic\_2.setPixmap(QPixmap("pythag\_pic\_2\_h"))

self.pic\_3.setPixmap(QPixmap("pythag\_pic\_3\_h"))

self.layout.addWidget(self.button\_1, 1, 0)

self.layout.addWidget(self.pic\_1, 1, 1)

self.layout.addWidget(self.pic\_2, 2, 0)

self.layout.addWidget(self.button\_2, 2, 1)

self.layout.addWidget(self.button\_3, 3, 0)

self.layout.addWidget(self.pic\_3, 3, 1)

self.layout.addWidget(self.button\_4, 4, 1)

self.layout.addWidget(self.button\_5, 5, 0)

self.layout.addWidget(self.button\_6, 6, 1)

self.button\_1.clicked.connect(self.pythag\_theorem\_easy)

self.button\_2.clicked.connect(self.pythag\_theorem\_medium)

self.button\_3.clicked.connect(self.pythag\_theorem\_hard)

self.button\_4.clicked.connect(self.three\_d\_pythag\_easy)

self.button\_5.clicked.connect(self.three\_d\_pythag\_medium)

self.button\_6.clicked.connect(self.three\_d\_pythag\_hard)

def pythag\_theorem\_easy(self):

pythag\_theorem\_1 = PythagStackTheoremEasy()

pythag\_theorem\_1.show()

pythag\_theorem\_1.\_raise()

def pythag\_theorem\_medium(self):

pythag\_theorem\_2 = PythagStackTheoremMediun()

pythag\_theorem\_2.show()

pythag\_theorem\_2.\_raise()

def pythag\_theorem\_hard(self):

pythag\_theorem\_3 = PythagStackTheoremHard()

pythag\_theorem\_3.show()

pythag\_theorem\_3.\_raise()

def three\_d\_pythag\_easy(self):

three\_d\_pythag\_1 = PythagStackTDPEasy()

three\_d\_pythag\_1.show()

three\_d\_pythag\_1.\_raise()

def three\_d\_pythag\_medium(self):

three\_d\_pythag\_2 = PythagStackTDPMedium()

three\_d\_pythag\_2.show()

three\_d\_pythag\_2.\_raise()

def three\_d\_pythag\_hard(self):

three\_d\_pythag\_3 = PythagStackTDPHard()

three\_d\_pythag\_3.show()

three\_d\_pythag\_3.\_raise()

\end{python}

\subsubsection{PythagTrigonometryHW}

\begin{python}

class PythagTrigonometryHW(ParentHomeworkMenuClass):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title.setPixmap(QPixmap("vectors\_title"))

self.button\_1.setText("Vectors Easy")

self.button\_2.setText("Vectors Medium")

self.button\_3.setText("Vectors Hard")

self.layout.addWidget(self.button\_1, 0, 1)

self.layout.addWidget(self.button\_2, 1, 0)

self.layout.addWidget(self.button\_3, 2, 1)

self.button\_1.clicked.connect(self.pythag\_trig\_problems\_easy)

self.button\_2.clicked.connect(self.pythag\_trig\_problems\_medium)

self.button\_3.clicked.connect(self.pythag\_trig\_problems\_hard)

def pythag\_trig\_problems\_easy(self):

pythag\_trig\_problems\_1 = VectorsStackEasy()

pythag\_trig\_problems\_1.show()

pythag\_trig\_problems\_1.\_raise()

def pythag\_trig\_problems\_medium(self):

pythag\_trig\_problems\_2 = VectorsStackMedium()

pythag\_trig\_problems\_2.show()

pythag\_trig\_problems\_2.\_raise()

def pythag\_trig\_problems\_hard(self):

pythag\_trig\_problems\_3 = VectorsStackHard()

pythag\_trig\_problems\_3.show()

pythag\_trig\_problems\_3.\_raise()

\end{python}

\subsubsection{SummaryHW}

\begin{python}

class SummaryHW(ParentHomeworkMenuClass):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.title.setPixmap(QPixmap("summary\_title"))

self.button\_1.setText("Easy Summary")

self.button\_2.setText("Medium Summary")

self.button\_3.setText("Hard Summary")

self.layout.addWidget(self.button\_1, 0, 1)

self.layout.addWidget(self.button\_2, 1, 0)

self.layout.addWidget(self.button\_3, 2, 1)

self.button\_1.clicked.connect(self.easy\_summary)

self.button\_2.clicked.connect(self.medium\_summary)

self.button\_3.clicked.connect(self.hard\_summary)

def easy\_summary(self):

summary\_1 = ReviseTrigStackEasy()

summary\_1.show()

summary\_1.\_raise()

def medium\_summary(self):

summary\_2 = ReviseTrigStackMedium()

summary\_2.show()

summary\_2.\_raise()

def hard\_summary(self):

summary\_3 = ReviseTrigStackHard()

summary\_3.show()

summary\_3.\_raise()

\end{python}

\subsection{Lesson Stacks}

\begin{python}

from PyQt4.QtCore import \* #These two lines import all of the built in PyQt code

from PyQt4.QtGui import \*

from lesson\_widgets\_page\_1 import \* #This contains the first widget of each stack widget

from lesson\_page\_2 import \* #This contains the second widget of each stack widget

\end{python}

\subsubsection{Trig1StackSiddes}

\begin{python}

#These are the templates for the stack widgets which contain a first and a second page each for a lesson

class Trig1StackSides(QMainWindow):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This maximises the window

self.showMaximized()

#Assigns variables to the first and second widgets so that when ths specific stack is run

#the lesson pages will be relevant to each other

self.first\_widget = SidesAHOWidget(self)

self.second\_widget = SidesAHOWidgetPage2(self)

#Sets the stack a a QStackedLayout so that the two windows can be switched between

#in the same window

self.stack = QStackedLayout()

#Adds the widgets to the stack layout

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

#Creates a QWidget, as this class is a QMainWindow, and sets the stack as

#the layout to be used

self.widget = QWidget()

self.widget.setLayout(self.stack)

#Sets the QWidget as the central widget so it will actually appear

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSOHCAHTOA}

\begin{python}

class Trig1StackSOHCAHTOA(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SOHCAHTOAWidget(self)

self.second\_widget = SOHCAHTOAWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackFA}

\begin{python}

class Trig2StackFA(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = FindingAnglesWidget(self)

self.second\_widget = FindingAnglesWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackTDT}

\begin{python}

class Trig2StackTDT(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDTrigonometryWidget(self)

self.second\_widget = ThreeDTrigonometryWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTheorem}

\begin{python}

class PythagStackTheorem(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = PythagTheoremWidget(self)

self.second\_widget = PythagTheoremWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTDP}

\begin{python}

class PythagStackTDP(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDPythagorasWidget(self)

self.second\_widget = ThreeDPythagorasWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{EasyStack}

\begin{python}

class EasyStack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = EasyWidget(self)

self.second\_widget = EasyWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{MediumStack}

\begin{python}

class MediumStack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = MediumWidget(self)

self.second\_widget = MediumWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{HardStack}

\begin{python}

class HardStack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = HardWidget(self)

self.second\_widget = HardWidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Revise1Stack}

\begin{python}

class Revise1Stack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ReviseTrig1Widget(self)

self.second\_widget = ReviseTrig1WidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Revise2Stack}

\begin{python}

class Revise2Stack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ReviseTrig2Widget(self)

self.second\_widget = ReviseTrig2WidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Revise3Stack}

\begin{python}

class Revise3Stack(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ReviseTrig3Widget(self)

self.second\_widget = ReviseTrig3WidgetPage2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsection{Homework Stacks}

\begin{python}

from PyQt4.QtCore import \* #These two lines import the built in PyQt code

from PyQt4.QtGui import \*

from homework\_widgets import \* #Contains the first widgets to add to each stack

from homework\_widgets\_page\_2 import \* #Contains the second widgets to add to each stack

\end{python}

\subsubsection{Trig1StackSidesEasy}

\begin{python}

class Trig1StackSidesEasy(QMainWindow):

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#Maximises the stack widget for both windows contained in it

self.showMaximized()

#Assigns the first sides easy widget from the homework\_widgets file for the stack layout

self.first\_widget = SidesAHOEasyWidget(self)

#Assigns the second sides easy widget from the homework\_widgets\_page\_2 file for the stack layout

self.second\_widget = SidesAHOEasyWidget2(self)

#All of the first widgets and second widgets share files so that only 1 import is necessary for each, and they share a lot of code

#Furthermore the code is'nt too long

#Sets the layout to a stack layout to allow for the addition of a page 1 and 2 to a single window to switch between them

self.stack = QStackedLayout()

#Adds the two widgets to the stack

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

#Sets the layout to be used as a QWidget and adds the stack to it

self.widget = QWidget()

self.widget.setLayout(self.stack)

#Sets the layout to be used as the central widget so that it appears

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSidesMedium}

\begin{python}

class Trig1StackSidesMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SidesAHOMediumWidget(self)

self.second\_widget = SidesAHOMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSidesHard}

\begin{python}

class Trig1StackSidesHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SidesAHOHardWidget(self)

self.second\_widget = SidesAHOHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSOHCAHTOAEasy}

\begin{python}

class Trig1StackSOHCAHTOAEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SOHCAHTOAEasyWidget(self)

self.second\_widget = SOHCAHTOAEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSOHCAHTOAMedium}

\begin{python}

class Trig1StackSOHCAHTOAMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SOHCAHTOAMediumWidget(self)

self.second\_widget = SOHCAHTOAMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig1StackSOHCAHTOAHard}

\begin{python}

class Trig1StackSOHCAHTOAHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = SOHCAHTOAHardWidget(self)

self.second\_widget = SOHCAHTOAHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackFindingAnglesEasy}

\begin{python}

class Trig2StackFindingAnglesEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = FindingAnglesEasyWidget(self)

self.second\_widget = FindingAnglesEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackFindingAnglesMedium}

\begin{python}

class Trig2StackFindingAnglesMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = FindingAnglesMediumWidget(self)

self.second\_widget = FindingAnglesMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackFindingAnglesHard}

\begin{python}

class Trig2StackFindingAnglesHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = FindingAnglesHardWidget(self)

self.second\_widget = FindingAnglesHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackTDTEasy}

\begin{python}

class Trig2StackTDTEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDTrigEasyWidget(self)

self.second\_widget = ThreeDTrigEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackTDTMedium}

\begin{python}

class Trig2StackTDTMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDTrigMediumWidget(self)

self.second\_widget = ThreeDTrigMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{Trig2StackTDTHard}

\begin{python}

class Trig2StackTDTHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDTrigHardWidget(self)

self.second\_widget = ThreeDTrigHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTheoremEasy}

\begin{python}

class PythagStackTheoremEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = PythagTheoremEasyWidget(self)

self.second\_widget = PythagTheoremEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTheoremMediun}

\begin{python}

class PythagStackTheoremMediun(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = PythagTheoremMediumWidget(self)

self.second\_widget = PythagTheoremMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTheoremHard}

\begin{python}

class PythagStackTheoremHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = PythagTheoremHardWidget(self)

self.second\_widget = PythagTheoremHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTDPEasy}

\begin{python}

class PythagStackTDPEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDPythagorasEasyWidget(self)

self.second\_widget = ThreeDPythagorasEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTDPMedium}

\begin{python}

class PythagStackTDPMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDPythagorasMediumWidget(self)

self.second\_widget = ThreeDPythagorasMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{PythagStackTDPHard}

\begin{python}

class PythagStackTDPHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = ThreeDPythagorasHardWidget(self)

self.second\_widget = ThreeDPythagorasHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{VectorsStackEasy}

\begin{python}

class VectorsStackEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = VectorsEasyWidget(self)

self.second\_widget = VectorsEasyWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{VectorsStackMedium}

\begin{python}

class VectorsStackMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = VectorsMediumWidget(self)

self.second\_widget = VectorsMediumWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{VectorsStackHard}

\begin{python}

class VectorsStackHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = VectorsHardWidget(self)

self.second\_widget = VectorsHardWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{ReviseTrigStackEasy}

\begin{python}

class ReviseTrigStackEasy(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = EasySummaryWidget(self)

self.second\_widget = EasySummaryWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{ReviseTrigStackMedium}

\begin{python}

class ReviseTrigStackMedium(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = MediumSummaryWidget(self)

self.second\_widget = MediumSummaryWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsubsection{ReviseTrigStackHard}

\begin{python}

class ReviseTrigStackHard(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.showMaximized()

self.first\_widget = HardSummaryWidget(self)

self.second\_widget = HardSummaryWidget2(self)

self.stack = QStackedLayout()

self.stack.addWidget(self.first\_widget)

self.stack.addWidget(self.second\_widget)

self.widget = QWidget()

self.widget.setLayout(self.stack)

self.setCentralWidget(self.widget)

\end{python}

\subsection{ParentLessonLayout Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

#This is the parent class which provides the default attributes to all of the child lesson page 1 widgets

class ParentLessonLayout(QWidget):

#Constructor

def \_\_init\_\_(self, parent = None):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#This sets the title which is included in all child classes; the QPixmap is set in the child classes so they can all be different

self.title = QLabel()

#Sets the background colour of the window to white in all of the child classes

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#All of these widgets are used in each of the child classes; they are created and formatted here then the text is set in the child classes

self.back = QPushButton("Return")

#Sets the size of the button

self.back.setMinimumHeight(50)

self.back.setMinimumWidth(60)

#Changes the font size and house style of the QPushButton

self.back.setFont(QFont("Courier", 40))

self.back.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.next = QPushButton("Next")

self.next.setMinimumHeight(50)

self.next.setMinimumWidth(60)

self.next.setFont(QFont("Courier", 40))

self.lesson\_1 = QTextEdit()

self.lesson\_1.setMinimumHeight(400)

self.lesson\_1.setMinimumWidth(80)

self.lesson\_1.setFont(QFont("Courier", 20))

#Prevents the user from being able to change the text of the lessons

self.lesson\_1.setReadOnly(True)

self.lesson\_2 = QTextEdit()

self.lesson\_2.setMinimumHeight(400)

self.lesson\_2.setMinimumWidth(80)

self.lesson\_2.setFont(QFont("Courier", 20))

self.lesson\_2.setReadOnly(True)

#Sets the background colour and font colour of all of the QPushButtons in each child class

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

#Sets the layout to a QGridLayout so that each widget can be positioned easily

self.layout = QGridLayout()

#Adds all of the widgets to the layout

self.layout.addWidget(self.title, 0, 0) #These numbers position the widgets in the window

self.layout.addWidget(self.lesson\_1, 1, 0)

self.layout.addWidget(self.lesson\_2, 1, 1)

self.layout.addWidget(self.back, 3, 0)

self.layout.addWidget(self.next, 3, 1)

#Sets layout as the layout to be used

self.setLayout(self.layout)

#These are the connections which are the same in each child class;

#any that are different will be in the child classes, for example connecting to different windows

self.back.clicked.connect(self.selected\_back)

self.next.clicked.connect(self.selected\_next\_page)

#This method closes the entire stack window

def selected\_back(self):

self.parent.close()

\end{python}

\subsection{ParentLessonPage2 Class}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

#This is the parent class which provides the default attributes for all of these child classes

class ParentLessonPage2(QWidget):

#Constructor

def \_\_init\_\_(self, parent = None):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#Sets the background colour of the window to white for all of the child classes

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#All of the widgets are created here, as they are used in each child class

#The text unique attributes are set in the child classes

self.answer = QLineEdit()

#Sets the size of the widgets

self.answer.setMinimumWidth(80)

self.answer.setMinimumHeight(110)

#Sets the font size and house style of the text

self.answer.setFont(QFont("Courier", 40))

self.previous = QPushButton("Previous")

self.previous.setMinimumHeight(110)

self.previous.setMinimumWidth(60)

self.previous.setFont(QFont("Courier", 40))

self.previous.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.check = QPushButton("Check Answer")

self.check.setMinimumHeight(110)

self.check.setMinimumWidth(60)

self.check.setFont(QFont("Courier", 40))

self.check.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.finish = QPushButton("Finish")

self.finish.setMinimumHeight(110)

self.finish.setMinimumWidth(60)

self.finish.setFont(QFont("Courier", 40))

self.finish.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

self.text\_1 = QTextEdit()

self.text\_1.setMinimumWidth(80)

self.text\_1.setMinimumHeight(110)

self.text\_1.setFont(QFont("Courier", 20))

self.text\_1.setReadOnly(True)

self.text\_2 = QTextEdit()

self.text\_2.setMinimumWidth(80)

self.text\_2.setMinimumHeight(110)

self.text\_2.setFont(QFont("Courier", 20))

self.text\_2.setReadOnly(True)

#Sets the background colour and font colour of all the buttons in all of the child classes

#with just one line

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

#Sets the layout as a QGridLayout so all of the widgets can be positioned easily

self.layout = QGridLayout()

#Sets layout as the layout to be used

self.setLayout(self.layout)

#Adds the widgets to the layout

self.layout.addWidget(self.text\_1, 0, 0) #These numbers position the widgets in the layout

self.layout.addWidget(self.text\_2, 0, 1)

self.layout.addWidget(self.previous, 3, 0)

self.layout.addWidget(self.answer, 3, 1)

self.layout.addWidget(self.finish, 4, 0)

self.layout.addWidget(self.check, 4, 1)

#The connections for navigation and checking the practice answer

self.previous.clicked.connect(self.previous\_selected)

self.check.clicked.connect(self.check\_selected)

self.finish.clicked.connect(self.finish\_selected)

#Switches back to the previous screen in the stack window (page 1)

def previous\_selected(self):

self.parent.stack.setCurrentIndex(0)

#This checks to see if the practice answer is correct

def check\_selected(self):

#If it's correct it just tells them they are correct

#Nothing is saved from the lessons

if self.answer.text() == self.answer\_lesson:

self.answer.setText("{0} Correct".format(self.answer\_lesson))

else:

self.answer.setText("Incorrect")

#Disables the buttons so they can't edit their answer

self.answer.setReadOnly(True)

self.check.setEnabled(False)

#Closes the entire stack window

def finish\_selected(self):

self.parent.close()

\end{python}

\subsection{Lesson Page 1 Classes}

\begin{python}

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

from lesson\_layout\_parent\_class import \* #This is the parent class for all of the first lesson pages, it provides all of the default attributes

\end{python}

\subsubsection{SidesAHOWidget}

\begin{python}

#This is the child template for the first page of the Sides lesson

class SidesAHOWidget(ParentLessonLayout):

#Constructor

def \_\_init\_\_(self, parent):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This declares the parent variable so that it can be called to switch the screen in the stack

self.parent = parent

#The QLabel is declared in the parent class and the picture is set here so that it can be different across each child class

self.title.setPixmap(QPixmap("sides\_lesson\_title"))

#These TextEdits are declared in the parent class and the text is set here so that it can be different across each child class

self.lesson\_1.setText("Every triangle has 3 sides, and each side has a name.\nThe HYPOTENUSE is the longest side, and is always oppposite the right-angle of a triangle.\nThe length can be found using Pythagoras' Theorem of a\u00b2 + b\u00b2 = c\u00b2.\nSine function: sin(x) = Opposite {0} Hypotenuse".format(chr(247)))

self.lesson\_2.setText("The OPPOSITE is the side opposite the angle being used.\nCosine function: cos(x) = Adjacent {0} Hypotenuse".format(chr(247)))

#The connection is in the parent class, because each setCurrentIndex(1) (each second screen in the stack) is declared in the lesson stack template which is already in use

def selected\_next\_page(self):

#Sets the stack to the next screen

self.parent.stack.setCurrentIndex(1)

\end{python}

\subsubsection{SOHCAHTOAWidget}

\begin{python}

class SOHCAHTOAWidget(ParentLessonLayout):

def \_\_init\_\_(self, parent):

super().\_\_init\_\_()

self.parent = parent

self.title.setPixmap(QPixmap("sohcahtoa\_lesson\_title"))

self.lesson\_1.setText("SOHCAHTOA stands for: \nSine Opposite Hypotenuse\nCosine Adjacent Hypotenuse\nTan Opposite Adjacent.\nThis is the best way to remember the three different rules for working out trigonometry problems.They each turn into a FORMULA TRIANGLE, a triangle which shows the formula inside in a memorable format.")

self.lesson\_2.setText("Method\n1. Label the three sides O, A and H\n2. Write down from memory SOHCAHTOA\n3. Decide which two sides are involved: O,H, A,H or O,A\n4. Turn the one you choose into one of the formula triangles to the left of the screen\n5. Cover up the thing you want to find and write down whatever is left showing\n6. Translate into numbers and work it out\n7. Finally, check that your answer is sensible")

self.pic = QLabel()

self.pic.setPixmap(QPixmap("formula\_triangles.png"))

self.layout.addWidget(self.pic, 2, 0)

def selected\_next\_page(self):

self.parent.stack.setCurrentIndex(1)

\end{python}

\subsubsection{FindingAnglesWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigonometryWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagorasWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{EasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{MediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{HardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig1Widget}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig2Widget}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig3Widget}

\begin{python}

#Finish

\end{python}

\subsection{Lesson Page 2 Classes}

\begin{python}

from PyQt4.QtCore import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

from lesson\_page\_2\_parent\_class import \* #This contains the parent class which provides the default attributes

#for all of these child classes

\end{python}

\subsubsection{SidesAHOWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAWidgetPage2}

\begin{python}

#These are the child classes which inherit from lesson\_pge\_2\_parent\_class

class SOHCAHTOAWidgetPage2(ParentLessonPage2):

#Constructor

def \_\_init\_\_(self, parent):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

self.answer.setText("m")

#The widget variables are created in the parent class then the text is overridden in each child class

self.text\_1.setText("Example 1:\n1. Label O, A, H\n2. Write down SOHCAHTOA\n3. Two sides are involved: O,H\n4. So use O {0} S x H\n5. We want to find H so cover it up to leave H = (O {0} S(0))\n6. Translate: Press 15 {0} SIN(35) = 26.151702, so ans = 26.2m\n7. Check it's sensible: Yes, it's about twice as big as 15, as the diagram suggests.".format(chr(247)))

#Sets the size of the text box

self.text\_1.setMinimumHeight(380)

self.text\_2.setText("You have to figure out yourself which formula to use to find this answer.\nHere's a hint: cut the triangle down the middle and it becomes a right-angled triangle.\n \n \n \n \n \n \nPut your answer in the box below:")

self.text\_2.setMinimumHeight(380)

self.pic = QLabel()

self.pic.setPixmap(QPixmap("sohcahtoa\_lesson\_pic\_2.png"))

self.pic.setAlignment(Qt.AlignCenter)

self.pic\_2 = QLabel()

self.pic\_2.setPixmap(QPixmap("sohcahtoa\_lesson\_pic\_3.png"))

self.pic\_2.setAlignment(Qt.AlignCenter)

#Adds the widgets to the layout - the layout is assigned and set in the parent class

self.layout.addWidget(self.pic, 2, 0)

self.layout.addWidget(self.pic\_2, 2, 1)

#A hard-coded answer for the test question

self.answer\_lesson = "26.5m"

\end{python}

\subsubsection{FindingAnglesWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigonometryWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagorasWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{EasyWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{MediumWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{HardWidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig1WidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig2WidgetPage2}

\begin{python}

#Finish

\end{python}

\subsubsection{ReviseTrig3WidgetPage2}

\begin{python}

#Finish

\end{python}

\subsection{ParentHomeworkPage1Class Class}

\begin{python}

from PyQt4.QtCore import \* #These two lines import the built in PyQt code

from PyQt4.QtGui import \*

from database\_class import \* #Contains all of the methods required to manipulate the database

from error\_messages import \* #Contains all of the error message classes for when the user makes a mistake

#The parent template for the first page of all homeworks - all homework child classes inherit their default attributes from here

class ParentHomeworkPage1Class(QWidget):

#Constructor

def \_\_init\_\_(self, parent = None):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#This maximises the screen

self.showMaximized()

#This variable is overridden in each child class - it is the task name string value which is stored in the database

self.task = ""

#Decides whether or not the user has actually submitted answers to question 1

#and authorises access to the next page

self.allow\_cont = False

#Sets the background colour of each child window to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#All of these buttons appear in all of the child classes - they are created here then the text is set in the child classesso htey can all be different

self.next = QPushButton("Next")

#Sets the size of the button

self.next.setMinimumHeight(110)

self.next.setMinimumWidth(60)

#Sets the font size and house style of the text in the button

self.next.setFont(QFont("Courier", 40))

self.cancel = QPushButton("Cancel")

self.cancel.setMinimumHeight(110)

self.cancel.setMinimumWidth(60)

self.cancel.setFont(QFont("Courier", 40))

self.cancel.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

self.check = QPushButton("Check Answers")

self.check.setMinimumHeight(110)

self.check.setMinimumWidth(60)

self.check.setFont(QFont("Courier", 40))

self.check.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.reset = QPushButton("Reset Answers")

self.reset.setMinimumHeight(110)

self.reset.setMinimumWidth(60)

self.reset.setFont(QFont("Courier", 40))

self.reset.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.title = QLabel()

self.title.setFont(QFont("Courier", 40))

self.question\_1 = QLabel()

self.question\_1.setFont(QFont("Courier", 20))

self.question\_1\_shape = QLabel()

self.question\_1\_shape.setFont(QFont("Courier", 40))

self.answer\_a = QLineEdit()

self.answer\_a.setMinimumHeight(70)

self.answer\_a.setMinimumWidth(60)

self.answer\_a.setFont(QFont("Courier", 30))

self.answer\_b = QLineEdit()

self.answer\_b.setMinimumHeight(70)

self.answer\_b.setMinimumWidth(60)

self.answer\_b.setFont(QFont("Courier", 30))

self.answer\_c = QLineEdit()

self.answer\_c.setMinimumHeight(70)

self.answer\_c.setMinimumWidth(60)

self.answer\_c.setFont(QFont("Courier", 30))

self.answer\_d = QLineEdit()

self.answer\_d.setMinimumHeight(70)

self.answer\_d.setMinimumWidth(60)

self.answer\_d.setFont(QFont("Courier", 30))

self.answer\_e = QLineEdit()

self.answer\_e.setMinimumHeight(70)

self.answer\_e.setMinimumWidth(60)

self.answer\_e.setFont(QFont("Courier", 30))

self.answer\_f = QLineEdit()

self.answer\_f.setMinimumHeight(70)

self.answer\_f.setMinimumWidth(60)

self.answer\_f.setFont(QFont("Courier", 30))

self.q1a = QLabel("")

self.q1a.setFont(QFont("Courier", 20))

self.q1b = QLabel("")

self.q1b.setFont(QFont("Courier", 20))

self.q1c = QLabel("")

self.q1c.setFont(QFont("Courier", 20))

self.q1d = QLabel("")

self.q1d.setFont(QFont("Courier", 20))

self.q1e = QLabel("")

self.q1e.setFont(QFont("Courier", 20))

self.q1f = QLabel("")

self.q1f.setFont(QFont("Courier", 20))

self.score\_box = QLabel("Score: X/X")

self.score\_box.setFont(QFont("Courier", 30))

#Sets the background colour and font colour of all the buttons in each child window

self.setStyleSheet("QPushButton {background-color: #A3C1DA; color: blue;}")

#Sets the layout of the window to a QGridLayout so the widgets can all be positioned easily

self.layout = QGridLayout()

#These add all of the widgets to the layout

self.layout.addWidget(self.title, 0, 0) #These numbers position the widgets in the window

self.layout.addWidget(self.question\_1, 1, 0)

self.layout.addWidget(self.q1a, 1, 1)

self.layout.addWidget(self.reset, 1, 2)

self.layout.addWidget(self.question\_1\_shape, 2, 0)

self.layout.addWidget(self.answer\_a, 2, 1)

self.layout.addWidget(self.q1b, 3, 1)

self.layout.addWidget(self.answer\_b, 4, 1)

self.layout.addWidget(self.q1c, 5, 1)

self.layout.addWidget(self.answer\_c, 6, 1)

self.layout.addWidget(self.q1d, 7, 1)

self.layout.addWidget(self.answer\_d, 8, 1)

self.layout.addWidget(self.q1e, 9, 1)

self.layout.addWidget(self.answer\_e, 10, 1)

self.layout.addWidget(self.q1f, 11, 1)

self.layout.addWidget(self.answer\_f, 12, 1)

self.layout.addWidget(self.cancel, 13, 0)

self.layout.addWidget(self.check, 13, 1)

self.layout.addWidget(self.next, 13, 2)

#Sets layout as the layout to be used

self.setLayout(self.layout)

#The connections for the buttons which have the same purpose in each child class - some will be in the child classes if they have any unique purposes

self.check.clicked.connect(self.check\_selected)

self.reset.clicked.connect(self.reset\_selected)

self.cancel.clicked.connect(self.cancel\_selected)

self.next.clicked.connect(self.next\_selected)

#Adds each answer to a list so that whenever each of the 6 variables needs the same thing to happen to them only 1 line

#of code in a for loop is needed rather than 6 lines on their own

self.answers = []

self.answers.append(self.answer\_a)

self.answers.append(self.answer\_b)

self.answers.append(self.answer\_c)

self.answers.append(self.answer\_d)

self.answers.append(self.answer\_e)

self.answers.append(self.answer\_f)

#This method checks the contents of all 6 line edits to see if the user's answers are correct

def check\_selected(self):

#Sets the correct count to 0 so that it is reset if the user does more than 1 homework in one sitting

#The buttons are disabled so they cannot reset the score after completing the question

self.allow\_cont = False

self.correct\_count = 0

## raise Exception("Need to override check\_selected from ParentHomeworkPage1Class")

#For each line edit, if the user's answer is the same as the hard-coded correct answer then they get a mark

if self.answer\_a.text() == self.answer\_1\_a:

self.answer\_a.setText("{0} Correct".format(self.answer\_a.text()))

self.correct\_count += 1

#If they are wrong they are told they are wrong

else:

self.answer\_a.setText("Incorrect")

if self.answer\_b.text() == self.answer\_1\_b:

self.answer\_b.setText("{0} Correct".format(self.answer\_b.text()))

self.correct\_count += 1

else:

self.answer\_b.setText("Incorrect")

if self.answer\_c.text() == self.answer\_1\_c:

self.answer\_c.setText("{0} Correct".format(self.answer\_c.text()))

self.correct\_count += 1

else:

self.answer\_c.setText("Incorrect")

if self.answer\_d.text() == self.answer\_1\_d:

self.answer\_d.setText("{0} Correct".format(self.answer\_d.text()))

self.correct\_count += 1

else:

self.answer\_d.setText("Incorrect")

if self.answer\_e.text() == self.answer\_1\_e:

self.answer\_e.setText("{0} Correct".format(self.answer\_e.text()))

self.correct\_count += 1

else:

self.answer\_e.setText("Incorrect")

if self.answer\_f.text() == self.answer\_1\_f:

self.answer\_f.setText("{0} Correct".format(self.answer\_f.text()))

self.correct\_count += 1

else:

self.answer\_f.setText("Incorrect")

#When all answers have been checked all of the line edits, the check button and the reset button are disabled to they have to accept their score

for a in self.answers:

a.setReadOnly(True)

self.check.setEnabled(False)

self.reset.setEnabled(False)

#Allows the program to continue to page 2

self.allow\_cont = True

#This method switches to the second screen in the stack after checking that the question has been answered and saving the score to the database

def next\_selected(self):

#if cont is false then a question has been missed

cont = False

while not cont:

#Checks to see if any answer has been put in the line edit

for a in self.answers:

#If the text is blank then it has been missed

if a.text() == "":

#Displays an error message asking the user to enter a value

error\_message = ErrorMessage8()

error\_message.show()

error\_message.\_raise()

cont = False

#If all of the answers have been entered the correct count is saved to the database and the second page is set to the top of the stack

cont = True

if self.allow\_cont:

g\_database.insert\_data\_first(self.task, self.correct\_count)

self.open\_page\_2()

self.hide()

else:

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

#Resets all of the answers in the line edits immediately rather than the user deleting each wrong answer individually, before they check the answers

def reset\_selected(self):

self.answer\_a.setText(None)

self.answer\_b.setText(None)

self.answer\_c.setText(None)

self.answer\_d.setText(None)

self.answer\_e.setText(None)

self.answer\_f.setText(None)

#Closes the stack widget and returns the user to the homework menu

def cancel\_selected(self):

self.parent.close()

#Switches to the second window in the stack

def open\_page\_2(self):

self.parent.stack.setCurrentIndex(1)

\end{python}

\subsection{HomeworkPage2ParentClass Class}

\begin{python}

from PyQt4.QtCore import \*

from PyQt4.QtGui import \*

from database\_class import \* #This contains the methods used to add to the database

from error\_messages import \* #This contains the default error message classes used when the user makes an error

#This is the template for the parent class which all 24 child classes inherit their default attributes from

class HomeworkPage2ParentClass(QWidget):

#Constructor

def \_\_init\_\_(self, parent = None):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#This is overridden in each child class so the user will know which task goes with the score

self.task = ""

#This maximises the screen - each child class will be maximised when opened

self.showMaximized()

#This sets the background colour of each child class to white

pal = QPalette()

pal.setColor(QPalette.Background, Qt.white)

self.setAutoFillBackground(True)

self.setPalette(pal)

#These 6 buttons are used in every child class, so they are created here - they

#are always the same size and have the same style (overridden from the style of the other buttons)

#They are the multiple choice buttons for question 4

self.\_button\_1 = QPushButton()

#Sets the size of the buttons so they fit in amongst the many other widgets

self.\_button\_1.setMaximumWidth(200)

self.\_button\_1.setMinimumWidth(110)

self.\_button\_1.setMinimumHeight(110)

#Sets the background colour and font colour of the button - override from the style sheet of the whole widget

self.\_button\_1.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_2 = QPushButton()

self.\_button\_2.setMaximumWidth(200)

self.\_button\_2.setMinimumWidth(110)

self.\_button\_2.setMinimumHeight(110)

self.\_button\_2.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_3 = QPushButton()

self.\_button\_3.setMaximumWidth(200)

self.\_button\_3.setMinimumWidth(110)

self.\_button\_3.setMinimumHeight(110)

self.\_button\_3.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_4 = QPushButton()

self.\_button\_4.setMaximumWidth(200)

self.\_button\_4.setMinimumWidth(110)

self.\_button\_4.setMinimumHeight(110)

self.\_button\_4.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_5 = QPushButton()

self.\_button\_5.setMaximumWidth(200)

self.\_button\_5.setMinimumWidth(110)

self.\_button\_5.setMinimumHeight(110)

self.\_button\_5.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.\_button\_6 = QPushButton()

self.\_button\_6.setMaximumWidth(200)

self.\_button\_6.setMinimumWidth(110)

self.\_button\_6.setMinimumHeight(110)

self.\_button\_6.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.question\_2 = QLabel()

self.question\_2.setFont(QFont("Courier", 30))

self.shape\_2 = QLabel()

self.shape\_2.setFont(QFont("Courier", 30))

self.question\_3 = QLabel()

self.question\_3.setFont(QFont("Courier", 30))

self.shape\_3 = QLabel()

self.shape\_3.setFont(QFont("Courier", 30))

self.question\_4 = QLabel()

self.question\_4.setFont(QFont("Courier", 30))

#This is the combo box which provides the value to check to see if the input for the answer is right or not

self.answer\_2 = QComboBox()

self.answer\_2.setMinimumWidth(60)

self.answer\_2.setMinimumHeight(110)

self.answer\_2.setFont(QFont("Courier", 40))

self.answer\_2.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

#This button is connected to the method to check if the contents of the above combo box is right or not

self.mark\_2 = QPushButton("Mark it | 2")

self.mark\_2.setMinimumWidth(60)

self.mark\_2.setMinimumHeight(110)

self.mark\_2.setFont(QFont("Courier", 40))

self.mark\_2.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

self.answer\_3 = QComboBox()

self.answer\_3.setMinimumWidth(60)

self.answer\_3.setMinimumHeight(110)

self.answer\_3.setFont(QFont("Courier", 40))

self.answer\_3.setStyleSheet("QComboBox {background-color: lavender; color: purple;}")

self.mark\_3 = QPushButton("Mark it | 2")

self.mark\_3.setMinimumWidth(60)

self.mark\_3.setMinimumHeight(110)

self.mark\_3.setFont(QFont("Courier", 40))

self.mark\_3.setStyleSheet("QPushButton {background-color: yellow; color: black; font-size: 20;}")

#Switches to the first widget in the stack

self.previous = QPushButton("Previous")

self.previous.setMinimumWidth(60)

self.previous.setMinimumHeight(110)

self.previous.setFont(QFont("Courier", 40))

self.previous.setStyleSheet("QPushButton {background-color: red; color: white; font-size: 20;}")

#Close the stack

self.finish = QPushButton("Finish")

self.finish.setMinimumWidth(60)

self.finish.setMinimumHeight(110)

self.finish.setFont(QFont("Courier", 40))

self.finish.setStyleSheet("QPushButton {background-color: green; color: white; font-size: 20;}")

#Tells the user how many attempts they have left at question 4

self.attempts\_button = QPushButton("3 Attempts left")

self.attempts\_button.setMinimumHeight(60)

self.attempts\_button.setMinimumWidth(90)

self.attempts\_button.setMaximumWidth(200)

self.attempts\_button.setStyleSheet("QPushButton {background-color: white; font-color: black;}")

self.attempts\_button.setEnabled(False)

#Sets the layout to a QGridLayout so the widgets can be positioned easily

self.layout = QGridLayout()

#Adds all of the widgets to the layout

self.layout.addWidget(self.question\_2, 0, 0)

self.layout.addWidget(self.shape\_2, 0, 1)

self.layout.addWidget(self.answer\_2, 1, 0)

self.layout.addWidget(self.mark\_2, 1, 1)

self.layout.addWidget(self.\_button\_1, 1, 2)

self.layout.addWidget(self.\_button\_2, 1, 3)

self.layout.addWidget(self.\_button\_3, 2, 2)

self.layout.addWidget(self.question\_3, 2, 0)

self.layout.addWidget(self.\_button\_4, 2, 3)

self.layout.addWidget(self.\_button\_5, 3, 2)

self.layout.addWidget(self.\_button\_6, 3, 3)

self.layout.addWidget(self.shape\_3, 2, 1)

self.layout.addWidget(self.attempts\_button, 4, 2)

self.layout.addWidget(self.answer\_3, 3, 0)

self.layout.addWidget(self.mark\_3, 3, 1)

self.layout.addWidget(self.question\_4, 0, 2)

self.layout.addWidget(self.previous, 5, 0)

self.layout.addWidget(self.finish, 5, 3)

#Sets layout as the layout to be used in the window

self.setLayout(self.layout)

#All of the connections for checking the inputs and navigating the stack

self.mark\_2.clicked.connect(self.selected\_mark\_2)

self.mark\_3.clicked.connect(self.selected\_mark\_3)

self.previous.clicked.connect(self.selected\_previous)

self.finish.clicked.connect(self.selected\_finish)

self.\_button\_1.clicked.connect(self.check\_button\_1)

self.\_button\_2.clicked.connect(self.check\_button\_2)

self.\_button\_3.clicked.connect(self.check\_button\_3)

self.\_button\_4.clicked.connect(self.check\_button\_4)

self.\_button\_5.clicked.connect(self.check\_button\_5)

self.\_button\_6.clicked.connect(self.check\_button\_6)

#The variables used to measure how many attempts the user has for each question, so the sytem knows when to disable the appropriate buttons

self.attempts\_remaining\_a = 2

self.attempts\_remaining\_b = 2

self.attempts\_remaining\_c = 3

#These are the counts which represent the number of marks the user has, and these values are saved in the database

self.correct\_count\_2 = 0

self.correct\_count\_3 = 0

self.correct\_count\_4 = 0

#This is overridden in each child class - it is the value to check the multiple choice buttons answer against

self.answer\_question\_4 = None

#These are the methods to check if the selected button is the correct button for question 4

def check\_button\_1(self, attempts\_remaining\_c):

#Resets the correct count so that it doesn't increment across tasks

self.correct\_count\_4 = 0

#The value to check is taken from the text on the QPushButton

#If it is the same then the user is awarded a mark and all of the buttons are disabled so they can't keep changing the answer and clicking aimlessly,

#or risk crashing the program

if self.\_button\_1.text() == self.answer\_question\_4:

self.\_button\_1.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

#If they are wrong then an attempt is removed and the button they clicked is disabled so they can't waste an attempt when they know it's wrong

else:

self.\_button\_1.setText("Incorrect")

self.\_button\_1.setEnabled(False)

self.attempts\_remaining\_c -= 1

#Changes the text of the mark it button so the user knows how many attempts they have left

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

#If they have used all of their attempts all of the buttons are disabled and they cannot do anything more with this question

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

def check\_button\_2(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_2.text() == self.answer\_question\_4:

self.\_button\_2.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_2.setText("Incorrect")

self.\_button\_2.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

def check\_button\_3(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_3.text() == self.answer\_question\_4:

self.\_button\_3.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_3.setText("Incorrect")

self.\_button\_3.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

def check\_button\_4(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_4.text() == self.answer\_question\_4:

self.\_button\_4.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_4.setText("Incorrect")

self.\_button\_4.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

def check\_button\_5(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_5.text() == self.answer\_question\_4:

self.\_button\_5.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_5.setText("Incorrect")

self.\_button\_5.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

def check\_button\_6(self, attempts\_remaining\_c):

self.correct\_count\_4 = 0

if self.\_button\_6.text() == self.answer\_question\_4:

self.\_button\_6.setText("Correct")

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("1 mark!")

self.correct\_count\_4 += 1

else:

self.\_button\_6.setText("Incorrect")

self.\_button\_6.setEnabled(False)

self.attempts\_remaining\_c -= 1

self.attempts\_button.setText("{0} attempts remaining".format(self.attempts\_remaining\_c))

if self.attempts\_remaining\_c == 0:

self.\_button\_1.setEnabled(False)

self.\_button\_2.setEnabled(False)

self.\_button\_3.setEnabled(False)

self.\_button\_4.setEnabled(False)

self.\_button\_5.setEnabled(False)

self.\_button\_6.setEnabled(False)

self.attempts\_button.setText("No more attempts")

return self.attempts\_remaining\_c

#These check the input in the combo boxes for questions 2 and 3 to see if the user is correct

def selected\_mark\_2(self, attempts\_remaining\_a):

#Resets the correct count so that it doesn't increment across tasks

self.correct\_count\_2 = 0

#Takes the value to be checked from the content of the combo box and compares it to the hard-coded answer

if self.answer\_2.currentText() == "20":

#If they are right they are awarded marks and the buttons are disabled as they have no further need of them (they can't get the same marks twice)

self.correct\_count\_2 += 1

self.mark\_2.setText("Correct!")

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

else:

#If they are wrong they lose an attempt until they have none left

self.attempts\_remaining\_a -= 1

self.mark\_2.setText("Mark it|{0}".format(self.attempts\_remaining\_a))

if self.attempts\_remaining\_a == 0:

#When they run out of attempts everything is disabled so they cannot continue with the question

self.mark\_2.setEnabled(False)

self.answer\_2.setEnabled(False)

#This error message informs the user that they are wrong

error\_message = ErrorMessage5()

error\_message.show()

error\_message.\_raise()

#Returns the attempts remaining so that it isn't reset every time the method is run, and the correct\_count to be saved to the database

return self.attempts\_remaining\_a, self.correct\_count\_2

def selected\_mark\_3(self, attempts\_remaining\_b):

self.correct\_count\_3 = 0

if self.answer\_3.currentText() == "20":

self.correct\_count\_3 += 1

self.mark\_3.setText("Correct!")

self.mark\_3.setEnabled(False)

self.answer\_3.setEnabled(False)

else:

self.attempts\_remaining\_b -= 1

self.mark\_3.setText("Mark it|{0}".format(self.attempts\_remaining\_b))

if self.attempts\_remaining\_b == 0:

self.mark\_3.setEnabled(False)

self.answer\_3.setEnabled(False)

error\_message = ErrorMessage5()

error\_message.show()

error\_message.\_raise()

return self.attempts\_remaining\_b, self.correct\_count\_3

def selected\_previous(self):

#Switches to the first screen in the stack

self.parent.stack.setCurrentIndex(0)

def selected\_finish(self):

#This checks to see whether or not all of the answers have been given

if self.attempts\_button.text() != "1 mark!" and self.attempts\_button.text() != "No more attempts":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_2.text() != "Correct!" and self.mark\_2.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

elif self.mark\_3.text() != "Correct!" and self.mark\_3.text() != "Mark it|0":

error\_message\_2 = ErrorMessage8()

error\_message\_2.show()

error\_message\_2.\_raise()

else:

#This method is in the database class and it updates all of the pre-recorded 0 values in the record for this task with the actual scores

g\_database.insert\_data\_second(self.task, self.correct\_count\_2, self.correct\_count\_3, self.correct\_count\_4)

#Closes the stack widget

self.parent.close()

\end{python}

\subsection{Homework Page 1 Classes}

\begin{python}

from PyQt4.QtGui import \* #These two lines import all of the built in PyQt code

from PyQt4.QtCore import \*

from homework\_parent\_class import \* #This contains the parent class which proides the default attributes for all of these child classes

from database\_class import \* #This contains the method which saves the task name and the first question score to the database

from error\_messages import \* #This contains the error message classes displayed when the user makes a mistake

\end{python}

\subsubsection{SidesAHOEasyWidget}

\begin{python}

#These are the child classes which inherit from the homework\_parent\_class

class SidesAHOEasyWidget(ParentHomeworkPage1Class):

#Constructor

def \_\_init\_\_(self, parent):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#This overrides the task name in each child class and is saved to the database to identify the task that has been ocmpleted

self.task = "Sides Easy"

self.title.setText("Sides Easy")

#The widget variables are all created in the parent class as they are used in each child class, then the

#text is set in the child classes so that they can all be different

self.question\_1.setText("Question 1: Look at the diagram below\nand answer the following questions: ")

self.q1a.setText("Which side is oppopsite angle A? ")

self.q1b.setText("Which side is adjacent to angle Q? ")

self.q1c.setText("Which side is the hypotenuse? ")

self.q1d.setText("Which formula would you use to find AB? ")

self.question\_1\_shape.setPixmap(QPixmap("sides\_easy\_q1"))

#These are the hard-coded answers to the questions which are checked in the check methods

self.answer\_1\_a = "BC"

self.answer\_1\_b = "AC"

self.answer\_1\_c = "AB"

self.answer\_1\_d = "cosine"

self.answer\_1\_e = "5"

self.answer\_1\_f = "6"

\end{python}

\subsubsection{SidesAHOMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{SidesAHOHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsEasyWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsMediumWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsHardWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{EasySummaryWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{MediumSummaryWidget}

\begin{python}

#Finish

\end{python}

\subsubsection{HardSummaryWidget}

\begin{python}

#Finish

\end{python}

\subsection{Homework Page 2 Classes}

\begin{python}

from PyQt4.QtCore import \* #These two lines import all of the built in PyQt code

from PyQt4.QtGui import \*

from homework\_page\_2\_parent\_class import \* #This contains the parent class which provides all of the default attributes for these child classes

from database\_class import \* #This contains the method which saves the scores to the database

\end{python}

\subsubsection{SidesAHOEasyWidget2}

\begin{python}

#These are the child classes which inherit all of their default attributes from homework\_page\_2\_parent\_class

class SidesAHOEasyWidget2(HomeworkPage2ParentClass):

#Constructor

def \_\_init\_\_(self, parent):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the parent variable which allows the child classes to inherit the connections to the second widget in the stacks

self.parent = parent

#Overriding the taskname from the parent class so the user knows which task goes with the score

self.task = "Sides Easy"

#The widget variables are all created in the parent class because they are used in all of the child classes

#and the text is set here so that it can be different

self.question\_2.setText("Question 2\nWhat is the length\nof b?")

## self.shape\_2.setText("sides easy")

self.question\_3.setText("Question 3\nWhat is the length\nof c?")

## self.shape\_3.setText("Shape")

self.question\_4.setText("Question 4\nWhat is the\nlength of a?")

#Adds some wrong answers and the right answer to the combo box

self.answer\_2.addItem("10")

self.answer\_2.addItem("20")

self.answer\_2.addItem("30")

self.answer\_3.addItem("10")

self.answer\_3.addItem("20")

self.answer\_3.addItem("30")

#Sets the text of the multiple choice buttons which is used to check the answer in the check methods

self.\_button\_1.setText("60")

self.\_button\_2.setText("50")

self.\_button\_3.setText("40")

self.\_button\_4.setText("30")

self.\_button\_5.setText("20")

self.\_button\_6.setText("10")

#This is the hard-coded answer which is checked against with the text in the buttons above

self.answer\_question\_4 = "40"

\end{python}

\subsubsection{SidesAHOMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{SidesAHOHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{SOHCAHTOAHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{FindingAnglesHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDTrigHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{PythagTheoremHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagorasEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagorasMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{ThreeDPythagorasHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsEasyWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsMediumWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{VectorsHardWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{EasySummaryWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{MediumSummaryWidget2}

\begin{python}

#Finish

\end{python}

\subsubsection{HardSummaryWidget2}

\begin{python}

#Finish

\end{python}

\subsection{Error Message Classes}

\begin{python}

#FINISH THESE - MAKE THEM RIGHT

from PyQt4.QtGui import \* #These two lines import the built in PyQt code

from PyQt4.QtCore import \*

\end{python}

\subsubsection{ErrorMessage2}

\begin{python}

#These are child class from the built in QErrorMessage class and these are called when the user inputs wrong data types or no value at all on the homeworks

class ErrorMessage2(QErrorMessage):

#Constructor

def \_\_init\_\_(self):

#Return a proxy object that delegates method calls to a parent or sibling class of type.

super().\_\_init\_\_()

#This is the message that is passed into the call for the message to be shown

message = "Invalid data type - please make sure you are inputting a decimal value"

#This shows the error message on the screen

QErrorMessage.showMessage(self, message)

\end{python}

\subsubsection{ErrorMessage3}

\begin{python}

class ErrorMessage3(QErrorMessage):

def \_\_init\_\_(self):

super().\_\_init\_\_()

message = "Invalid data type - please make sure you are inputting an integer value"

QErrorMessage.showMessage(self, message)

\end{python}

\subsubsection{ErrorMessage4}

\begin{python}

class ErrorMessage4(QErrorMessage):

def \_\_init\_\_(self):

super().\_\_init\_\_()

message = "Invalid data type - please make sure you are inputting a string value"

QErrorMessage.showMessage(self, message)

\end{python}

\subsubsection{ErrorMessage5}

\begin{python}

class ErrorMessage5(QErrorMessage):

def \_\_init\_\_(self):

super().\_\_init\_\_()

message = "That is incorrect - please try again. You have X more attempts"

QErrorMessage.showMessage(self, message)

\end{python}

\subsubsection{ErrorMessage8}

\begin{python}

class ErrorMessage8(QErrorMessage):

def \_\_init\_\_(self):

super().\_\_init\_\_()

message = "Please input your answers"

QErrorMessage.showMessage(self, message)

\end{python}

\end{landscape}54

\chapter{Testing}

\section{Test Plan}

\begin{landscape}

\subsection{Original Outline Plan}

\begin{center}

\begin{longtable}{|p{2cm}|p{5cm}|p{5cm}|p{4cm}|}

\hline

\textbf{Test Series} & \textbf{Purpose of Test Series} & \textbf{Testing Strategy} & \textbf{Strategy Rationale}\\ \hline

1 & Test the connections between all of the buttons on the user interfaces & Top-down testing & Each button will be tested to make sure it connects to the right screen \\ \hline

2 & Test all of the input spaces & Bottom-up testing & Each input will be tested once it is developed \\ \hline

3 & Test all information is stored in the correct place in the database & Black box testing & The database will be viewed in a database viewer to ensure that SQL queries are working and data is being stored in the database once they have been developed \\ \hline

4 & Test all of the algorithms to make sure that the program gives the correct output and marks, both mathematical or other & White box testing & Each algorithm will be tested once it is developed \\ \hline

5 & Test that the system fulfils the clients request & Acceptance testing & The system will be developed once it is completed to a usable standard \\ \hline

\end{longtable}

\end{center}

\subsection{Changes to Outline Plan}

\begin{center}

\begin{longtable}{|p{2cm}|p{5cm}|p{5cm}|p{4cm}|}

\hline

\textbf{Test Series} & \textbf{Purpose of Test Series} & \textbf{Testing Strategy} & \textbf{Strategy Rationale}\\ \hline

1 & Test the connections between all of the buttons on the user interfaces & Top-down testing & Each button will be tested to make sure it connects to the right screen \\ \hline

2 & Test all of the input spaces & Bottom-up testing & Each input will be tested once it is developed \\ \hline

3 & Test all information is stored in the correct place in the database & Black box testing & The database will be viewed in a database viewer to ensure that SQL queries are working and data is being stored in the database once they have been developed \\ \hline

4 & Test all of the algorithms to make sure that the program gives the correct output and marks, both mathematical or other & White box testing & Each algorithm will be tested once it is developed \\ \hline

5 & Test that the system fulfils the clients request & Acceptance testing & The system will be developed once it is completed to a usable standard \\ \hline

\end{longtable}

\end{center}

\subsection{Original Detailed Plan}

\textbf{Test numbers with \* afterwards are tests which have been omitted from the updated testing plan.}

\begin{center}

\begin{longtable}{|p{1.5cm}|p{2.5cm}|p{2.5cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|}

\hline

\textbf{Test Series} & \textbf{Purpose of Test} & \textbf{Test Description} & \textbf{Test Data} & \textbf{Test Data Type (Normal/ Erroneous/ Boundary)} & \textbf{Expected Result} & \textbf{Actual Result} & \textbf{Evidence}\\ \hline

1.001 \* & To test the student log in button on the first menu functions as intended & This should link to the student menu screen & Click the log in button & Normal & The student account screen should be displayed & & \\ \hline

1.002 \* & To test the teacher log in button on the first menu functions as intended & This should link to the administrator menu screen & Click the log in button & Normal & The administrator account screen should be displayed & & \\ \hline

1.003 & To test the lessons button on the student account screen functions as intended & This should link to the lesson topic menu screen & Click the lessons button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.004 & To test the homework button on the student account screen functions as intended & This should link to the homework topic menu screen & Click the homework button & Normal & The homework topic menu should be displayed & & \\ \hline

1.005 & To test the progress button on the student account screen functions as intended & This should link to the student's personal database display screen & Click the progress button & Normal & The student's personal database screen should be displayed & & \\ \hline

1.006 & To test the log out button on the student account screen functions as intended & This should close down the entire program & Click the log out button & Normal & The program should stop & & \\ \hline

1.007 & To test the Trigonometry 1 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.008 & To test the Trigonometry 2 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.009 & To test the Pythagoras button on the lesson topic menu screen functions as intended & This should link to the pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.010 & To test the Pythagoras and Trigonometry Problems button on the lesson topic menu screen functions as intended & This should link to the trigonometry and pythagoras problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.011 & To test the Summary button on the lesson topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.012 & To test the return button on the lesson topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.013 & To test the Sides button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the sides lesson & Click the Sides button & Normal & The first sides lesson screen should be displayed & & \\ \hline

1.014 & To test the SOHCAHTOA button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the SOHCAHTOA lesson & Click the SOHCAHTOA button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.015 & To test the return button on the Trigonometry 1 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.016 & To test the Finding Angles button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the finding angles lesson & Click the Finding angles button & Normal & The first finding angles lesson screen should be displayed & & \\ \hline

1.017 \* & To test the Inverted Angles button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the inverted angles lesson & Click the Inverted angles button & Normal & The first inverted angles lesson screen should be displayed & & \\ \hline

1.018 & To test the 3D Trigonometry button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the 3D Trigonometry lesson & Click the 3D Trigonometry button & Normal & The first 3D trigonometry lesson screen should be displayed & & \\ \hline

1.019 & To test the return button on the Trigonometry 2 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.020 & To test the Pythagoras Theorem button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the pythagoras theorem lesson & Click the pythagoras theorem button & Normal & The first pythagoras theorem lesson screen should be displayed & & \\ \hline

1.021 & To test the 3D Pythagoras button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the 3D pythagoras lesson & Click the 3D pythagoras button & Normal & The first 3D pythagoras lesson screen should be displayed & & \\ \hline

1.022 & To test the return button on the Pythagoras menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.023 & To test the Easy Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the easy pythagoras and trigonometry problems lesson & Click the easy pythagoras and trigonometry problems button & Normal & The first easy pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.024 & To test the Medium Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the medium pythagoras and trigonometry problems lesson & Click the medium pythagoras and trigonometry problems button & Normal & The first medium pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.025 & To test the Hard Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the hard pythagoras and trigonometry problems lesson & Click the hard pythagoras and trigonometry problems button & Normal & The first hard pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.026 & To test the return button on the Pythagoras and Trigonometry Problems menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.027 & To test the Revise Trigonometry 1 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 1 lesson & Click the revise trigonometry 1 button & Normal & The first revise trigonometry 1 lesson screen should be displayed & & \\ \hline

1.028 & To test the Revise Trigonometry 2 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 2 lesson & Click the revise trigonometry 2 button & Normal & The first revise trigonometry 2 lesson screen should be displayed & & \\ \hline

1.029 & To test the Revise Trigonometry 3 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 3 lesson & Click the revise trigonometry 3 button & Normal & The first revise trigonometry 3 lesson screen should be displayed & & \\ \hline

1.030 & To test the return button on the Summary menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.031 & To test the return button on the first Sides lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the Sides return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.032 & To test the next button on the first Sides lesson screen functions as intended & This should link to the second Sides lesson screen & Click the Sides next button & Normal & The second Sides lesson screen should be displayed & & \\ \hline

1.033 & To test the previous button on the second Sides lesson screen functions as intended & This should link back to the first Sides lesson screen & Click the Sides previous button & Normal & The first Sides lesson screen should be displayed & & \\ \hline

1.034 & To test the Check Answer button on the second Sides lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Sides check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.035 & To test the Finish button on the second Sides lesson screen functions as intended & This close the lesson and return to the lesson topic menu screen & Click the Sides finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.036 & To test the return button on the first SOHCAHTOA lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the SOHCAHTOA return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.037 & To test the next button on the first SOHCAHTOA lesson screen functions as intended & This should link to the second SOHCAHTOA lesson screen & Click the SOHCAHTOA next button & Normal & The second SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.038 & To test the previous button on the second SOHCAHTOA lesson screen functions as intended & This should link back to the first SOHCAHTOA lesson screen & Click the SOHCAHTOA previous button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.039 & To test the Check Answer button on the second SOHCAHTOA lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the SOHCAHTOA check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.040 & To test the Finish button on the second SOHCAHTOA lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the SOHCAHTOA finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.041 & To test the return button on the first Finding Angles lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the Finding Angles return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.042 & To test the next button on the first Finding Angles lesson screen functions as intended & This should link to the second Finding Angles lesson screen & Click the Finding Angles next button & Normal & The second Finding Angles lesson screen should be displayed & & \\ \hline

1.043 & To test the previous button on the second Finding Angles lesson screen functions as intended & This should link back to the first Finding Angles lesson screen & Click the Finding Angles previous button & Normal & The first Finding Angles lesson screen should be displayed & & \\ \hline

1.044 & To test the Check Answer button on the second Finding Angles lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Finding Angles check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.045 & To test the Finish button on the second Finding Angles lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Finding Angles finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.046 \* & To test the return button on the first Inverted Angles lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the Inverted Angles return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.047 \* & To test the next button on the first Inverted Angles lesson screen functions as intended & This should link to the second Inverted Angles lesson screen & Click the Inverted Angles next button & Normal & The second Inverted Angles lesson screen should be displayed & & \\ \hline

1.048 \* & To test the previous button on the second Inverted Angles lesson screen functions as intended & This should link back to the first Inverted Angles lesson screen & Click the Inverted Angles previous button & Normal & The first Inverted Angles lesson screen should be displayed & & \\ \hline

1.049 \* & To test the Check Answer button on the second Inverted Angles lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Inverted Angles check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.050 \* & To test the Finish button on the second Inverted Angles lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Inverted Angles finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.051 & To test the return button on the first 3D Trigonometry lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the 3D Trigonometry return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.052 & To test the next button on the first 3D Trigonometry lesson screen functions as intended & This should link to the second 3D Trigonometry lesson screen & Click the 3D Trigonometry next button & Normal & The second 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.053 & To test the previous button on the second 3D Trigonometry lesson screen functions as intended & This should link back to the first 3D Trigonometry lesson screen & Click the 3D Trigonometry previous button & Normal & The first 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.054 & To test the Check Answer button on the second 3D Trigonometry lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Trigonometry check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.055 & To test the Finish button on the second 3D Trigonometry lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Trigonometry finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.056 & To test the return button on the first Pythagoras Theorem lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the Pythagoras Theorem return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.057 & To test the next button on the first Pythagoras Theorem lesson screen functions as intended & This should link to the second Pythagoras Theorem lesson screen & Click the Pythagoras Theorem next button & Normal & The second Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.058 & To test the previous button on the second Pythagoras Theorem lesson screen functions as intended & This should link back to the first Pythagoras Theorem lesson screen & Click the Pythagoras Theorem previous button & Normal & The first Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.059 & To test the Check Answer button on the second Pythagoras Theorem lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras Theorem check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.060 & To test the Finish button on the second Pythagoras Theorem lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras Theorem finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.061 & To test the return button on the first 3D Pythagoras lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the 3D Pythagoras return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.062 & To test the next button on the first 3D Pythagoras lesson screen functions as intended & This should link to the second 3D Pythagoras lesson screen & Click the 3D Pythagoras next button & Normal & The second 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.063 & To test the previous button on the second 3D Pythagoras lesson screen functions as intended & This should link back to the first 3D Pythagoras lesson screen & Click the 3D Pythagoras previous button & Normal & The first 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.064 & To test the Check Answer button on the second 3D Pythagoras lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Pythagoras check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.065 & To test the Finish button on the second 3D Pythagoras lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Pythagoras finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.066 & To test the return button on the first Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Easy return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.067 & To test the next button on the first Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Easy lesson screen & Click the Pythagoras and Trigonometry Problems Easy next button & Normal & The second Pythagoras and Trigonometry Problems Easy lesson screen should be displayed & & \\ \hline

1.068 & To test the previous button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Easy lesson screen & Click the Pythagoras and Trigonometry Problems Easy previous button & Normal & The first Pythagoras and Trigonometry Problems Easy lesson screen should be displayed & & \\ \hline

1.069 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Easy check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.070 & To test the Finish button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Easy finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.071 & To test the return button on the first Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Medium return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.072 & To test the next button on the first Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Medium lesson screen & Click the Pythagoras and Trigonometry Problems Medium next button & Normal & The second Pythagoras and Trigonometry Problems Medium lesson screen should be displayed & & \\ \hline

1.073 & To test the previous button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Medium lesson screen & Click the Pythagoras and Trigonometry Problems Medium previous button & Normal & The first Pythagoras and Trigonometry Problems Medium lesson screen should be displayed & & \\ \hline

1.074 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Medium check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.075 & To test the Finish button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Medium finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.076 & To test the return button on the first Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Hard return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.077 & To test the next button on the first Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Hard lesson screen & Click the Pythagoras and Trigonometry Problems Hard next button & Normal & The second Pythagoras and Trigonometry Problems Hard lesson screen should be displayed & & \\ \hline

1.078 & To test the previous button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Hard lesson screen & Click the Pythagoras and Trigonometry Problems Hard previous button & Normal & The first Pythagoras and Trigonometry Problems Hard lesson screen should be displayed & & \\ \hline

1.079 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Hard check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.080 & To test the Finish button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Hard finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.081 & To test the return button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 1 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.082 & To test the next button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link to the second Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 next button & Normal & The second Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.083 & To test the previous button on the second Revise Trigonometry 1 lesson screen functions as intended & This should link back to the first Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 previous button & Normal & The first Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.084 & To test the Check Answer button on the second Revise Trigonometry 1 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 1 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.085 & To test the Finish button on the second Revise Trigonometry 1 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 1 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.086 & To test the return button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 2 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.087 & To test the next button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link to the second Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 next button & Normal & The second Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.088 & To test the previous button on the second Revise Trigonometry 2 lesson screen functions as intended & This should link back to the first Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 previous button & Normal & The first Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.089 & To test the Check Answer button on the second Revise Trigonometry 2 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 2 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.090 & To test the Finish button on the second Revise Trigonometry 2 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 2 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.091 & To test the return button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 3 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.092 & To test the next button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link to the second Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 next button & Normal & The second Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.093 & To test the previous button on the second Revise Trigonometry 3 lesson screen functions as intended & This should link back to the first Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 previous button & Normal & The first Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.094 & To test the Check Answer button on the second Revise Trigonometry 3 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 3 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.095 & To test the Finish button on the second Revise Trigonometry 3 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 3 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.096 \* & To test each Set Homework button on the set homework list on the homework topic menu screen functions as intended & This should link to the corresponding homework that is named on the button - could be any homework in any order depending on what the teacher decides to set & Click the set homework button & Normal & The corresponding homework first screen should be displayed & & \\ \hline

1.097 & To test the Trigonometry 1 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.098 & To test the Trigonometry 2 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.099 & To test the Pythagoras button on the homework topic menu screen functions as intended & This should link to the Pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.100 & To test the Pythagoras and Trigonometry Problems button on the homework topic menu screen functions as intended & This should link to the Trigonometry and Pythagoras Problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.101 & To test the Summary button on the homework topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.102 & To test the return button on the homework topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.103 & To test the Sides Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Easy homework screen & click the Sides Easy button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.104 & To test the Sides Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Medium homework screen & click the Sides Medium button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.105 & To test the Sides Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Hard homework screen & click the Sides Hard button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.106 & To test the SOHCAHTOA Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Easy homework screen & click the SOHCAHTOA Easy button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.107 & To test the SOHCAHTOA Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Medium homework screen & click the SOHCAHTOA Medium button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.108 & To test the SOHCAHTOA Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Hard homework screen & click the SOHCAHTOA Hard button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.109 & To test the return button on the Trigonometry 1 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.110 & To test the Finding Angles Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Easy homework screen & click the Finding Angles Easy button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.111 & To test the Finding Angles Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Medium homework screen & click the Finding angles Medium button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.112 & To test the Finding Angles Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Hard homework screen & click the Finding angles Hard button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.113 \* & To test the Inverted Angles Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Easy homework screen & click the Inverted Angles Easy button & Normal & The first Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.114 \* & To test the Inverted Angles Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Medium homework screen & click the Inverted Angles Medium button & Normal & The first Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.115 \* & To test the Inverted Angles Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Hard homework screen & click the Inverted Angles Hard button & Normal & The first Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.116 & To test the 3D Trigonometry Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Easy homework screen & click the 3D Trigonometry Easy button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.117 & To test the 3D Trigonometry Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Medium homework screen & click the 3D Trigonometry Medium button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.118 & To test the 3D Trigonometry Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Hard homework screen & click the 3D Trigonometry Hard button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.119 & To test the return button on the Trigonometry 2 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.120 & To test the Pythagoras Theorem Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Easy homework screen & click the Pythagoras Theorem Easy button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.121 & To test the Pythagoras Theorem Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Medium homework screen & click the Pythagoras Theorem Medium button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.122 & To test the Pythagoras Theorem Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Hard homework screen & click the Pythagoras Theorem Hard button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.123 & To test the 3D Pythagoras Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Easy homework screen & click the 3D Pythagoras Easy button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.124 & To test the 3D Pythagoras Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Medium homework screen & click the 3D Pythagoras Medium button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.125 & To test the 3D Pythagoras Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Hard homework screen & click the 3D Pythagoras Hard button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.126 & To test the return button on the Pythagoras homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.127 & To test the Pythagoras and Trigonometry Problems Easy button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Easy homework screen & Click the Pythagoras and Trigonometry Problems Easy button & Normal & The first Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.128 & To test the Pythagoras and Trigonometry Problems Medium button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Medium homework screen & Click the Pythagoras and Trigonometry Problems Medium button & Normal & The first Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.129 & To test the Pythagoras and Trigonometry Problems Hard button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Hard homework screen & Click the Pythagoras and Trigonometry Problems Hard button & Normal & The first Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.130 & To test the return button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.131 & To test the Summary Easy button on the Summary homework menu screen functions as intended & This should link to the first Summary Easy homework screen & Click the Summary Easy button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.132 & To test the Summary Medium button on the Summary homework menu screen functions as intended & This should link to the first Summary Medium homework screen & Click the Summary Medium button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.133 & To test the Summary Hard button on the Summary homework menu screen functions as intended & This should link to the first Summary Hard homework screen & Click the Summary Hard button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.134 & To test the return button on the Summary homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.135 & To test the cancel button on the Sides Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.136 & To test the check answers button on the Sides Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.137 & To test the reset answers button on the Sides Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.138 & To test the next button on the Sides Easy first homework screen functions as intended & This should link to the second Sides Easy homework screen & Click the next button & Normal & The second Sides Easy homework screen should be displayed & & \\ \hline

1.139 & To test the first mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.140 & To test the second mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.141 & To test the third mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.142 & To test the previous button on the Sides Easy second homework screen functions as intended & This should link back to the first Sides Easy homework screen & Click the previous button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.143 & To test the finish button on the second Sides Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.144 & To test the cancel button on the Sides Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.145 & To test the check answers button on the Sides Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.146 & To test the reset answers button on the Sides Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.147 & To test the next button on the Sides Medium first homework screen functions as intended & This should link to the second Sides Medium homework screen & Click the next button & Normal & The second Sides Medium homework screen should be displayed & & \\ \hline

1.148 & To test the first mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.149 & To test the second mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.150 & To test the third mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.151 &To test the previous button on the Sides Medium second homework screen functions as intended & This should link back to the first Sides Medium homework screen & Click the previous button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.152 & To test the finish button on the second Sides Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.153 & To test the cancel button on the Sides Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.154 & To test the check answers button on the Sides Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.155 & To test the reset answers button on the Sides Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.156 & To test the next button on the Sides Hard first homework screen functions as intended & This should link to the second Sides Hard homework screen & Click the next button & Normal & The second Sides Hard homework screen should be displayed & & \\ \hline

1.157 & To test the first mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.158 & To test the second mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.159 & To test the third mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.160 & To test the previous button on the Sides Hard second homework screen functions as intended & This should link back to the first Sides Hard homework screen & Click the previous button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.161 & To test the finish button on the second Sides Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.162 & To test the cancel button on the SOHCAHTOA Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.163 & To test the check answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.164 & To test the reset answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.165 & To test the next button on the SOHCAHTOA Easy first homework screen functions as intended & This should link to the second SOHCAHTOA Easy homework screen & Click the next button & Normal & The second SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.166 & To test the first mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.167 & To test the second mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.168 & To test the third mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.169 & To test the previous button on the SOHCAHTOA Easy second homework screen functions as intended & This should link back to the first SOHCAHTOA Easy homework screen & Click the previous button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.170 & To test the finish button on the second SOHCAHTOA Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.171 & To test the cancel button on the SOHCAHTOA Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.172 & To test the check answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.173 & To test the reset answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.174 & To test the next button on the SOHCAHTOA Medium first homework screen functions as intended & This should link to the second SOHCAHTOA Medium homework screen & Click the next button & Normal & The second SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.175 & To test the first mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.176 & To test the second mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.177 & To test the third mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.178 & To test the previous button on the SOHCAHTOA Medium second homework screen functions as intended & This should link back to the first SOHCAHTOA Medium homework screen & Click the previous button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.179 & To test the finish button on the second SOHCAHTOA Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.180 & To test the cancel button on the SOHCAHTOA Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.181 & To test the check answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.182 & To test the reset answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.183 & To test the next button on the SOHCAHTOA Hard first homework screen functions as intended & This should link to the second SOHCAHTOA Hard homework screen & Click the next button & Normal & The second SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.184 & To test the first mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.185 & To test the second mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.186 & To test the third mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.187 & To test the previous button on the SOHCAHTOA Hard second homework screen functions as intended & This should link back to the first SOHCAHTOA Hard homework screen & Click the previous button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.188 & To test the finish button on the second SOHCAHTOA Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.189 & To test the cancel button on the Finding Angles Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.190 & To test the check answers button on the Finding Angles Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.191 & To test the reset answers button on the Finding Angles Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.192 & To test the next button on the Finding Angles Easy first homework screen functions as intended & This should link to the second Finding Angles Easy homework screen & Click the next button & Normal & The second Finding Angles Easy homework screen should be displayed & & \\ \hline

1.193 & To test the first mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.194 & To test the second mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.195 & To test the third mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.196 & To test the previous button on the Finding Angles Easy second homework screen functions as intended & This should link back to the first Finding Angles Easy homework screen & Click the previous button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.197 & To test the finish button on the second Finding Angles Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.198 & To test the cancel button on the Finding Angles Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.199 & To test the check answers button on the Finding Angles Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.200 & To test the reset answers button on the Finding Angles Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.201 & To test the next button on the Finding Angles Medium first homework screen functions as intended & This should link to the second Finding Angles Medium homework screen & Click the next button & Normal & The second Finding Angles Medium homework screen should be displayed & & \\ \hline

1.202 & To test the first mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.203 & To test the second mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.204 & To test the third mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.205 & To test the previous button on the Finding Angles Medium second homework screen functions as intended & This should link back to the first Finding Angles Medium homework screen & Click the previous button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.206 & To test the finish button on the second Finding Angles Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.207 & To test the cancel button on the Finding Angles Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.208 & To test the check answers button on the Finding Angles Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.209 & To test the reset answers button on the Finding Angles Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.210 & To test the next button on the Finding Angles Hard first homework screen functions as intended & This should link to the second Finding Angles Hard homework screen & Click the next button & Normal & The second Finding Angles Hard homework screen should be displayed & & \\ \hline

1.211 & To test the first mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.212 & To test the second mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.213 & To test the third mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.214 & To test the previous button on the Finding Angles Hard second homework screen functions as intended & This should link back to the first Finding Angles Hard homework screen & Click the previous button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.215 & To test the finish button on the second Finding Angles Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.216 \* & To test the cancel button on the Inverted Angles Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.217 \* & To test the check answers button on the Inverted Angles Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.218 \* & To test the reset answers button on the Inverted Angles Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.219 \* & To test the next button on the Inverted Angles Easy first homework screen functions as intended & This should link to the second Inverted Angles Easy homework screen & Click the next button & Normal & The second Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.220 \* & To test the first mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.221 \* & To test the second mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.222 \* & To test the third mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.223 \* & To test the previous button on the Inverted Angles Easy second homework screen functions as intended & This should link back to the first Inverted Angles Easy homework screen & Click the previous button & Normal & The first Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.224 \* & To test the finish button on the second Inverted Angles Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.225 \* & To test the cancel button on the Inverted Angles Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.226 \* & To test the check answers button on the Inverted Angles Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.227 \* & To test the reset answers button on the Inverted Angles Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.228 \* & To test the next button on the Inverted Angles Medium first homework screen functions as intended & This should link to the second Inverted Angles Medium homework screen & Click the next button & Normal & The second Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.229 \* & To test the first mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.230 \* & To test the second mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.231 \* & To test the third mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.232 \* & To test the previous button on the Inverted Angles Medium second homework screen functions as intended & This should link back to the first Inverted Angles Medium homework screen & Click the previous button & Normal & The first Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.233 \* & To test the finish button on the Inverted Angles Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.234 \* & To test the cancel button on the Inverted Angles Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.235 \* & To test the check answers button on the Inverted Angles Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.236 \* & To test the reset answers button on the Inverted Angles Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.237 \* & To test the next button on the Inverted Angles Hard first homework screen functions as intended & This should link to the second Inverted Angles Hard homework screen & Click the next button & Normal & The second Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.238 \* & To test the first mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.239 \* & To test the second mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.240 \* & To test the third mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.241 \* & To test the previous button on the Inverted Angles Hard second homework screen functions as intended & This should link back to the first Inverted Angles Hard homework screen & Click the previous button & Normal & The first Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.242 \* & To test the finish button on the second Inverted Angles Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.243 & To test the cancel button on the 3D Trigonometry Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.244 & To test the check answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.245 & To test the reset answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.246 & To test the next button on the 3D Trigonometry Easy first homework screen functions as intended & This should link to the second 3D Trigonometry Easy homework screen & Click the next button & Normal & The second 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.247 & To test the first mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.248 & To test the second mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.249 & To test the third mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.250 & To test the previous button on the 3D Trigonometry Easy second homework screen functions as intended & This should link back to the first 3D Trigonometry Easy homework screen & Click the previous button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.251 & To test the finish button on the second 3D Trigonometry Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.252 & To test the cancel button on the 3D Trigonometry Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.253 & To test the check answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.254 & To test the reset answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.255 & To test the next button on the 3D Trigonometry Medium first homework screen functions as intended & This should link to the second 3D Trigonometry Medium homework screen & Click the next button & Normal & The second 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.256 & To test the first mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.257 & To test the second mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.258 & To test the third mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.259 & To test the previous button on the 3D Trigonometry Medium second homework screen functions as intended & This should link back to the first 3D Trigonometry Medium homework screen & Click the previous button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.260 & To test the finish button on the second 3D Trigonometry Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.261 & To test the cancel button on the 3D Trigonometry Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.262 & To test the check answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.263 & To test the reset answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.264 & To test the next button on the 3D Trigonometry Hard first homework screen functions as intended & This should link to the second 3D Trigonometry Hard homework screen & Click the next button & Normal & The second 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.265 & To test the first mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.266 & To test the second mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.267 & To test the third mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.268 & To test the previous button on the 3D Trigonometry Hard second homework screen functions as intended & This should link back to the first 3D Trigonometry Hard homework screen & Click the previous button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.269 & To test the finish button on the second 3D Trigonometry Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.270 & To test the cancel button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.271 & To test the check answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.272 & To test the reset answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.273 & To test the next button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link to the second Pythagoras Theorem Easy homework screen & Click the next button & Normal & The second Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.274 & To test the first mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.275 & To test the second mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.276 & To test the third mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.277 & To test the previous button on the Pythagoras Theorem Easy second homework screen functions as intended & This should link back to the first Pythagoras Theorem Easy homework screen & Click the previous button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.278 & To test the finish button on the second Pythagoras Theorem Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.279 & To test the cancel button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.280 & To test the check answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.281 & To test the reset answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.282 & To test the next button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link to the second Pythagoras Theorem Medium homework screen & Click the next button & Normal & The second Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.283 & To test the first mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.284 & To test the second mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.285 & To test the third mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.286 & To test the previous button on the Pythagoras Theorem Medium second homework screen functions as intended & This should link back to the first Pythagoras Theorem Medium homework screen & Click the previous button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.287 & To test the finish button on the second Pythagoras Theorem Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.288 & To test the cancel button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.289 & To test the check answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.290 & To test the reset answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.291 & To test the next button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link to the second Pythagoras Theorem Hard homework screen & Click the next button & Normal & The second Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.292 & To test the first mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.293 & To test the second mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.294 & To test the third mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.295 & To test the previous button on the Pythagoras Theorem Hard second homework screen functions as intended & This should link back to the first Pythagoras Theorem Hard homework screen & Click the previous button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.296 & To test the finish button on the second Pythagoras Theorem Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.297 & To test the cancel button on the 3D Pythagoras Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.298 & To test the check answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.299 & To test the reset answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.300 & To test the next button on the 3D Pythagoras Easy first homework screen functions as intended & This should link to the second 3D Pythagoras Easy homework screen & Click the next button & Normal & The second 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.301 & To test the first mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.302 & To test the second mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.303 & To test the third mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.304 & To test the previous button on the 3D Pythagoras Easy second homework screen functions as intended & This should link back to the first 3D Pythagoras Easy homework screen & Click the previous button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.305 & To test the finish button on the second 3D Pythagoras Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.306 & To test the cancel button on the 3D Pythagoras Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.307 & To test the check answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.308 & To test the reset answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.309 & To test the next button on the 3D Pythagoras Medium first homework screen functions as intended & This should link to the second 3D Pythagoras Medium homework screen & Click the next button & Normal & The second 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.310 & To test the first mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.311 & To test the second mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.312 & To test the third mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.313 & To test the previous button on the 3D Pythagoras Medium second homework screen functions as intended & This should link back to the first 3D Pythagoras Medium homework screen & Click the previous button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.314 & To test the finish button on the second 3D Pythagoras Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.315 & To test the cancel button on the 3D Pythagoras Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.316 & To test the check answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.317 & To test the reset answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.318 & To test the next button on the 3D Pythagoras Hard first homework screen functions as intended & This should link to the second 3D Pythagoras Hard homework screen & Click the next button & Normal & The second 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.319 & To test the first mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.320 & To test the second mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.321 & To test the third mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.322 & To test the previous button on the 3D Pythagoras Hard second homework screen functions as intended & This should link back to the first 3D Pythagoras Hard homework screen & Click the previous button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.323 & To test the finish button on the second 3D Pythagoras Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.324 & To test the cancel button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.325 & To test the check answers button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.326 & To test the reset answers button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.327 & To test the next button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Easy homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.328 & To test the first mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.329 & To test the second mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.330 & To test the third mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.331 & To test the previous button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Easy homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.332 & To test the finish button on the second Pythagoras and Trigonometry Problems Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.333 & To test the cancel button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.334 & To test the check answers button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.335 & To test the reset answers button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.336 & To test the next button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Medium homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.337 & To test the first mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.338 & To test the second mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.339 & To test the third mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.340 & To test the previous button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Medium homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.341 & To test the finish button on the second Pythagoras and Trigonometry Problems Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.342 & To test the cancel button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.343 & To test the check answers button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.344 & To test the reset answers button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.345 & To test the next button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Hard homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.346 & To test the first mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.347 & To test the second mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.348 & To test the third mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.349 & To test the previous button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Hard homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.350 & To test the finish button on the second Pythagoras and Trigonometry Problems Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.351 & To test the cancel button on the Summary Easy first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.352 & To test the check answers button on the Summary Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.353 & To test the reset answers button on the Summary Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.354 & To test the next button on the Summary Easy first homework screen functions as intended & This should link to the second Summary Easy homework screen & Click the next button & Normal & The second Summary Easy homework screen should be displayed & & \\ \hline

1.355 & To test the first mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.356 & To test the second mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.357 & To test the third mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.358 & To test the previous button on the Summary Easy second homework screen functions as intended & This should link back to the first Summary Easy homework screen & Click the previous button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.359 & To test the finish button on the second Summary Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.360 & To test the cancel button on the Summary Medium first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.361 & To test the check answers button on the Summary Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.362 & To test the reset answers button on the Summary Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.363 & To test the next button on the Summary Medium first homework screen functions as intended & This should link to the second Summary Medium homework screen & Click the next button & Normal & The second Summary Medium homework screen should be displayed & & \\ \hline

1.364 & To test the first mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.365 & To test the second mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.366 & To test the third mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.367 & To test the previous button on the Summary Medium second homework screen functions as intended & This should link back to the first Summary Medium homework screen & Click the previous button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.368 & To test the finish button on the second Summary Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.369 & To test the cancel button on the Summary Hard first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.370 & To test the check answers button on the Summary Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.371 & To test the reset answers button on the Summary Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.372 & To test the next button on the Summary Hard first homework screen functions as intended & This should link to the second Summary Hard homework screen & Click the next button & Normal & The second Summary Hard homework screen should be displayed & & \\ \hline

1.373 & To test the first mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.374 & To test the second mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.375 & To test the third mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.376 & To test the previous button on the Summary Hard second homework screen functions as intended & This should link back to the first Summary Hard homework screen & Click the previous button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.377 & To test the finish button on the second Summary Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.378 \* & To test the Not Completed buttons on the progress screen function as intended & This should link to the first screen of the respective homework & Click the not completed button (Will have the name of the homework) & Normal & The first screen of the respective homework should be displayed & & \\ \hline

1.379 \* & To test the Not Enough Score buttons on the progress screen function as intended & This should link to the first screen of the respective homework & Click the not enough score button (Will have the name of the homework) & Normal & The first screen of the respective homework should be displayed & & \\ \hline

1.380 & To test the return button on the progress screen functions as intended & This should link back to the student account home screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.381 \* & To test the homework button on the admin account home screen functions as intended & This should link to the homework topic menu screen & Click the homework button & Normal & The homework topic menu should be displayed & & \\ \hline

1.382 \* & To test the results button on the admin account home screen functions as intended & This should link to the recent results menu & Click the results button & Normal & The recent results menu should be displayed & & \\ \hline

1.383 \* & To test the progress button on the admin account home screen functions as intended & This should link to the database viewer screen & Click the progress button & Normal & The database viewer screen should be displayed & & \\ \hline

1.384 \* & To test the log out button on the admin account home screen functions as intended & This should log the user off and close the application & Click the log out button & Normal & The program should stop running & & \\ \hline

1.385 \* & To test each Set Homework button on the admin set homework list on the homework topic menu screen functions as intended & This should link to the corresponding homework that is named on the button - could be any homework in any order depending on what the teacher decides to set & Click the set homework button & Normal & The corresponding homework results screen should be displayed & & \\ \hline

1.386 \* & To test the Trigonometry 1 button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.387 \* & To test the Trigonometry 2 button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.388 \* & To test the Pythagoras button on the admin homework topic menu screen functions as intended & This should link to the Pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.389 \* & To test the Pythagoras and Trigonometry Problems button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry and Pythagoras Problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.390 \* & To test the Summary button on the admin homework topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.391 \* & To test the return button on the homework topic menu functions as intended & This should link back to the admin account screen & Click the return button & Normal & The admin account screen should be displayed & & \\ \hline

1.392 \* & To test the Sides Easy button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.393 \* & To test the Sides Medium button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.394 \* & To test the Sides Hard button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.395 \* & To test the SOHCAHTOA Easy button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.396 \* & To test the SOHCAHTOA Medium button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.397 \* & To test the SOHCAHTOA Hard button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.398 \* & To test the return button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.399 \* & To test the Finding Angles Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the first homework set screen & click the Finding Angles Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.400 \* & To test the Finding Angles Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Finding angles Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.401 \* & To test the Finding Angles Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Finding angles Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.402 \* & To test the Inverted Angles Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.403 \* & To test the Inverted Angles Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.404 \* & To test the Inverted Angles Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.405 \* & To test the 3D Trigonometry Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.406 \* & To test the 3D Trigonometry Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.407 \* & To test the 3D Trigonometry Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.408 \* & To test the return button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.409 \* & To test the Pythagoras Theorem Easy button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.410 \* & To test the Pythagoras Theorem Medium button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.411 \* & To test the Pythagoras Theorem Hard button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.412 \* & To test the 3D Pythagoras Easy button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.413 \* & To test the 3D Pythagoras Medium button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.414 \* & To test the 3D Pythagoras Hard button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.415 \* & To test the return button on the Admin Pythagoras homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.416 \* & To test the Pythagoras and Trigonometry Problems Easy button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.417 \* & To test the Pythagoras and Trigonometry Problems Medium button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.418 \* & To test the Pythagoras and Trigonometry Problems Hard button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.419 \* & To test the return button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.420 \* & To test the Summary Easy button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.421 \* & To test the Summary Medium button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.422 \* & To test the Summary Hard button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.423 \* & To test the return button on the Admin Summary homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.424 \* & To test the return button on the homework set screen functions as intended & This should link back to the topic lesson menu which was selected before & Click the return button & Normal & The previous topic's homework menu should be displated & & \\ \hline

1.425 \* & To test the set button on the homework set screen functions as intended & This should add the selected homework to the list of each student in the selected class, and inform them when they next load the application, and finally register the deadline & Click the set button & Normal & The homework should be added to the list of each student in the class, and the deadline registered & & \\ \hline

1.426 \* & To test the Result buttons on the results menu screen functions as intended & This should link to the results of the corresponding homework in the database viewer for the entire class & Click the results button(s) & Normal & The results for the corresponding homework across the class should be displayed in the database viewer & & \\ \hline

1.427 \* & To test the return button on the results menu screen functions as intended & This should link back to the administrator account home screen & Click the return button & Normal & The administrator account home screen should be displayed & & \\ \hline

1.428 \* & To test the return button on the admin progress database viewer screen functions as intended & This should link back to the administrator account home screen & Click the return button & Normal & The administrator account home screen should be displayed & & \\ \hline

1.429 \* & To test the return button on the administrator report screen functions as intended & This should link back to the admin progress screen & Click the return button & Normal & The administrator progress screen should be displayed & & \\ \hline

1.430 \* & To test the query button on the administrator report screen functions as intended & This should link back to the admin progress screen and new queried information should be displayed & Click the query button & Normal & The admin progress screen should be displayed with updated data in the database widget & & \\ \hline

1.431 & To test the return button on the student report screen functions as intended & This should link back to the student progress screen & Click the return button & Normal & The student progress screen should be displayed & & \\ \hline

1.432 & To test the query button on the student report screen functions as intended & This should link back to the student progress screen and new queried information should be displayed & Click the query button & Normal & The student progress screen should be displayed with updated data in the database widget & & \\ \hline

1.433 \* & To test the add class button on the add names widget functions as intended & This should link to the add class screen & Click the add class button & Normal & The add class screen should be displayed & & \\ \hline

1.434 \* & To test the add name button on the add names widget functions as intended & This should add a name to the system and link back to the admin home screen & Click the add name button & Normal & The name should be stored and the admin home screen should be displayed & & \\ \hline

1.435 \* & To test the add another button on the add names widget functions as intended & This should add a name to the system and remain on the same page for the process to repeat & Click the add another button & Normal & The name should be added to the sytem and the input boxes wiped ready for another name & & \\ \hline

1.436 \* & To test the return button on the add names widget functions as intended & This should link back to the admin home screen & Click the return button & Normal & The admin home screen should be displayed & & \\ \hline

1.437 \* & To test the add stduents button on the add class widget functions as intended & This should link to the add names screen & Click the add students button & Normal & The add names screen should be displayed & & \\ \hline

1.438 \* & To test the add class button on the add class widget functions as intended & This should add a class to the system and then return to the admin home screen & Click the add class button & Normal & The class should be added to the system and the admin home screen should be displayed & & \\ \hline

1.439 \* & To test the add another button on the add class widget functions as intended & This should add a class to the system and remain on the same page for the process to repeat & Click the add another button & Normal & The class should be added to the sytem and the input boxes wiped ready for another class & & \\ \hline

1.440 \* & To test the return button on the add class widget functions as intended & This should link back to the admin home screen & Click the return button & Normal & The admin home screen should be displayed & & \\ \hline

2.001 \* & To test that the system can recognise a user's username when input correctly and allow access to their account & This should allow the user access to their account & Input User1 & Erroneous, Presence & The username should be accepted and the user allowed access (alongside a correct password) & & \\ \hline

2.002 \* & To test that the system can recognise a user's password when input correctly and allow access to their account & This should allow the user access to their account & Input d8g3h6g & Erroneous, Presence & The password should be accepted and the user allowed access to their account (alongside a correct username) & & \\ \hline

2.003 & To test that the input for the practice question on the Sides second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.004 & To test that the input for the practice question on the SOHCAHTOA second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.005 & To test that the input for the practice question on the Finding Angles second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.006 \* & To test that the input for the practice question on the Inverted Angles second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.007 & To test that the input for the practice question on the 3D Trigonometry second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.008 & To test that the input for the practice question on the Pythagoras' Theorem second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.009 & To test that the input for the practice question on the 3D Pythagoras second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.010 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Easy second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.011 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Medium second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.012 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Hard second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.013 & To test that the input for the practice question on the Revise Trigonometry 1 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.014 & To test that the input for the practice question on the Revise Trigonometry 2 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.015 & To test that the input for the practice question on the Revise Trigonometry 3 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.016 & To test that the text input for the generic question 1 part A is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.017 & To test that the text input for the generic question 1 part B is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.018 & To test that the text input for the generic question 1 part C is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon presisng the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.019 & To test that the text input for the generic question 1 part D is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.020 & To test that the text input for the generic question 1 part E is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.021 & To test that the text input for the generic question 1 part F is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.022 & To test that the combo box input for the generic question 2 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.023 & To test that the combo box input for the generic question 3 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.024 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.025 & To test that the text input for the generic question 1 part A is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.026 & To test that the text input for the generic question 1 part B is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.027 & To test that the text input for the generic question 1 part C is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.028 & To test that the text input for the generic question 1 part D is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.029 & To test that the text input for the generic question 1 part E is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.030 & To test that the text input for the generic question 1 part F is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.031 & To test that the combo box input for the generic question 2 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.032 & To test that the combo box input for the generic question 3 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.033 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.034 & To test that the text input for the generic question 1 part A is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.035 & To test that the text input for the generic question 1 part B is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.036 & To test that the text input for the generic question 1 part C is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.037 & To test that the text input for the generic question 1 part D is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.038 & To test that the text input for the generic question 1 part E is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.039 & To test that the text input for the generic question 1 part F is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.040 & To test that the combo box input for the generic question 2 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.041 & To test that the combo box input for the generic question 3 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.042 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.043 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.044 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.045 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.046 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.047 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.048 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.049 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.050 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.051 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.052 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.053 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.054 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.055 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.056 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.057 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.058 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.059 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.060 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.061 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.062 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.063 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.064 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.065 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.066 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.067 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.068 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.069 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.070 & To test that the text input for the generic question 1 part A is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.071 & To test that the text input for the generic question 1 part B is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.072 & To test that the text input for the generic question 1 part C is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.073 & To test that the text input for the generic question 1 part D is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.074 & To test that the text input for the generic question 1 part E is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.075 & To test that the text input for the generic question 1 part F is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.076 & To test that the combo box input for the generic question 2 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.077 & To test that the combo box input for the generic question 3 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.078 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.079 & To test that the text input for the generic question 1 part A is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.080 & To test that the text input for the generic question 1 part B is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.081 & To test that the text input for the generic question 1 part C is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.082 & To test that the text input for the generic question 1 part D is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.083 & To test that the text input for the generic question 1 part E is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.084 & To test that the text input for the generic question 1 part F is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.085 & To test that the combo box input for the generic question 2 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.086 & To test that the combo box input for the generic question 3 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.087 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.088 & To test that the text input for the generic question 1 part A is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.089 & To test that the text input for the generic question 1 part B is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.090 & To test that the text input for the generic question 1 part C is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.091 & To test that the text input for the generic question 1 part D is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.092 & To test that the text input for the generic question 1 part E is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.093 & To test that the text input for the generic question 1 part F is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.094 & To test that the combo box input for the generic question 2 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.095 & To test that the combo box input for the generic question 3 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.096 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.097 \* & To test that the text input for the generic question 1 part A is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.098 \* & To test that the text input for the generic question 1 part B is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.099 \* & To test that the text input for the generic question 1 part C is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.100 \* & To test that the text input for the generic question 1 part D is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.101 \* & To test that the text input for the generic question 1 part E is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.102 \* & To test that the text input for the generic question 1 part F is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.103 \* & To test that the combo box input for the generic question 2 is received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.104 \* & To test that the combo box input for the generic question 3 is received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.105 \* & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.106 \* & To test that the text input for the generic question 1 part A is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.107 \* & To test that the text input for the generic question 1 part B is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.108 \* & To test that the text input for the generic question 1 part C is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.109 \* & To test that the text input for the generic question 1 part D is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.110 \* & To test that the text input for the generic question 1 part E is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.111 \* & To test that the text input for the generic question 1 part F is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.112 \* & To test that the combo box input for the generic question 2 is received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.113 \* & To test that the combo box input for the generic question 3 is received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.114 \* & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.115 \* & To test that the text input for the generic question 1 part A is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.116 \* & To test that the text input for the generic question 1 part B is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.117 \* & To test that the text input for the generic question 1 part C is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.118 \* & To test that the text input for the generic question 1 part D is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.119 \* & To test that the text input for the generic question 1 part E is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.120 \* & To test that the text input for the generic question 1 part F is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.121 \* & To test that the combo box input for the generic question 2 is received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.122 \* & To test that the combo box input for the generic question 3 is received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.123 \* & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.124 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.125 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.126 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.127 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.128 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.129 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.130 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.131 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.132 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.133 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.134 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.135 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.136 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.137 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.138 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.139 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.140 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.141 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.142 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.143 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.144 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.145 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.146 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.147 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.148 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.149 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.150 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.151 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.152 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.153 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.154 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.155 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.156 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.157 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.158 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.159 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.160 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.161 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.162 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.163 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.164 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.165 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.166 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.167 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.168 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.169 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.170 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.171 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.172 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.173 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.174 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.175 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.176 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.177 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.178 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.179 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.180 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.181 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.182 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.183 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.184 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.185 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.186 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.187 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.188 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.189 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.190 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.191 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.192 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.193 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.194 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.195 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.196 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.197 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.198 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.199 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.200 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.201 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.202 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.203 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.204 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.205 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.206 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.207 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.208 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.209 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.210 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.211 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.212 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.213 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.214 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.215 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.216 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.217 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.218 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.219 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.220 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.221 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.222 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.223 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.224 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.225 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.226 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.227 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.228 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.229 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.230 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.231 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.232 & To test that the text input for the generic question 1 part A is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.233 & To test that the text input for the generic question 1 part B is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.234 & To test that the text input for the generic question 1 part C is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.235 & To test that the text input for the generic question 1 part D is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.236 & To test that the text input for the generic question 1 part E is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.237 & To test that the text input for the generic question 1 part F is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.238 & To test that the combo box input for the generic question 2 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.239 & To test that the combo box input for the generic question 3 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.240 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.241 & To test that the text input for the generic question 1 part A is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.242 & To test that the text input for the generic question 1 part B is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.243 & To test that the text input for the generic question 1 part C is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.244 & To test that the text input for the generic question 1 part D is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.245 & To test that the text input for the generic question 1 part E is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.246 & To test that the text input for the generic question 1 part F is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.247 & To test that the combo box input for the generic question 2 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.248 & To test that the combo box input for the generic question 3 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.249 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.250 & To test that the text input for the generic question 1 part A is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.251 & To test that the text input for the generic question 1 part B is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.252 & To test that the text input for the generic question 1 part C is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.253 & To test that the text input for the generic question 1 part D is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.254 & To test that the text input for the generic question 1 part E is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.255 & To test that the text input for the generic question 1 part F is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.256 & To test that the combo box input for the generic question 2 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.257 & To test that the combo box input for the generic question 3 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.258 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.259 \* & To test that the system can recognise an administrator's username when input correctly and allow access to their account & This should allow the administrator access to their account & Input Admin1 & Erroneous, Presence & The username should be accepted and the administrator allowed access (alongside a correct password) & & \\ \hline

2.260 \* & To test that the system can recognise an administrator's password when input correctly and allow access to their account & This should allow the administrator access to their account & Input h4j8d8s & Erroneous, Presence & The password should be accepted and the administrator allowed access to their account (alongside a correct username) & & \\ \hline

2.261 \* & To test that the system can register a selected class to be set homework from a combo box & This should set the homework for the students in the class and notify them on their accounts & 10A & Presence & The class should be registered as having to do that homework and each student should be notified & & \\ \hline

2.262 \* & To test that the system can register a chosen deadline for a class to complete a homework by from a combo box & This should set and register a date and notify the students of this date & 12/12/15 & Presence & The class should be notified of the deadline and it should be registered in the system & & \\ \hline

2.263 \* & To test that the system can register a score requirement for a selected homework from a combo box & This should register a minimum score for each student to achieve before they can properly submit the homework as complete & & \\ \hline

2.264 \* & To test that the feedback box for the Sides Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.265 \* & To test that the feedback box for the Sides Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.266 \* & To test that the feedback box for the Sides Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.267 \* & To test that the feedback box for the SOHCAHTOA Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.268 \* & To test that the feedback box for the SOHCHTOA Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.269 \* & To test that the feedback box for the SOHCAHTOA Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.270 \* & To test that the feedback box for the Finding Angles Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.271 \* & To test that the feedback box for the Finding Angles Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.272 \* & To test that the feedback box for the Finding Angles Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.273 \* & To test that the feedback box for the Inverted Angles Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.274 \* & To test that the feedback box for the Inverted Angles Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.275 \* & To test that the feedback box for the Inverted Angles Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.276 \* & To test that the feedback box for the 3D Trigonometry Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.277 \* & To test that the feedback box for the 3D Trigonometry Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.278 \* & To test that the feedback box for the 3D Trigonometry Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.279 \* & To test that the feedback box for the Pythagoras' Theorem Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.280 \* & To test that the feedback box for the Pythagoras' Theorem Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.281 \* & To test that the feedback box for the Pythagoras' Theorem Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.282 \* & To test that the feedback box for the 3D Pythagoras Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.283 \* & To test that the feedback box for the 3D Pythagoras Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.284 \* & To test that the feedback box for the 3D Pythagoras Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.285 \* & To test that the feedback box for the Pythagoras and Trigonometry Problems Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.286 \* & To test that the feedback box for the Pythagoras and Trigonometry Problems Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.287 \* & To test that the feedback box for the Pythagoras and Trigonometry Problems Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.288 \* & To test that the feedback box for the Summary Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.289 \* & To test that the feedback box for the Summary Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.290 \* & To test that the feedback box for the Summary Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.291 \* & To test the the class selection combo box on the admin report screen functions as intended & This should accept a class selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.292 \* & To test the the student selection combo box on the admin report screen functions as intended & This should accept a student selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.293 \* & To test the the task selection combo box on the admin report screen functions as intended & This should accept a task selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.294 \* & To test the the score selection combo box on the admin report screen functions as intended & This should accept a score selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.295 & To test the the task selection combo box on the student report screen functions as intended & This should accept a task selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.296 & To test the the score selection combo box on the student report screen functions as intended & This should accept a score selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.297 \* & To test the FirstName line edit on the add names widget functions as intended & This should accept the name given, validate it as a name and store it in the database & John & Erroneous, Presence & The name should be validated and added to the database & & \\ \hline

2.298 \* & To test the LastName line edit on the add names widget functions as intended & This should accept the name given, validate it as a name and store it in the database & Smith & Erroneous, Presence & The name should be validated and added to the database & & \\ \hline

2.299 \* & To test the ClassID line edit on the add class widget functions as intended & This should accept the class given, validate it as a ClassID and store it in the database & 10A & Erroneous, Presence & The class should be validated and added to the database & & \\ \hline

3.001 \* & To test that the ClassID is stored in the correct place in the database & This should be stored as a primary key and should not be displayed at any point in the system & 10A & Presence, Normal & The ClassID should be stored as a primary key in the Class database & & \\ \hline

3.002 \* & To test that the UserIDs are all stored in the correct place in the database & These should be stored as primary keys and should be displayed on account home screens & 0001 & Presence, Normal & The UserIDs should be stored as a primary key in the Class Database and in the User's database & & \\ \hline

3.003 \* & To test that the student's FirstNames are all stored in the correct place in the database & These should be stored and visible only by the user whose first name it is in their personal task database & John & Presence, Normal & The FirstNames should be stored in the User's database and visible together by the administrator or individually by the respective student & & \\ \hline

3.004 \* & To test that the student's LastNames are all stored in the correct place in the database & These should be stored and visible only by the user whose last name it is in their personal task database & Smith & Presence, Normal & The LastNames should be stored in the User's database and visible together by the administrator or individually by the respective student & & \\ \hline

3.005 \* & To test that the administrator's FirstName is stored in the correct place in the database & This should be stored and visible only by the administrator in the User's database & John & Presence, Normal & The FirstName should be stored in the User's database and visible only by the administrator & & \\ \hline

3.006 \* & To test that the administrator's LastName is stored in the correct place in the database & This should be stored and visible only by the administrator in the User's database & Smith & Presence, Normal & The LastName should be stored in the User's database and visible only by the administrator & & \\ \hline

3.007 \* & To test that all of the user's usernames are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user on their home screen or in their personal task database & User01 & Presence, Normal & These should be stored in the User's database and visible together by the adminsitrator or individually by the respective student & & \\ \hline

3.008 \* & To test that all of the user's passwords are stoed in the correct place in the database & These should be stored and only isible by the administrator & f8h4j7h & Presence, Normal & These should be stored in the User's database and visible only by the administrator behind their password & & \\ \hline

3.009 & To test that the TaskNames are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Sides Easy & Presence, Normal & These should be stored in the Tasks database and visible together by the administrator or individually by the respective student & & \\ \hline

3.010 \* & To test that the OverallPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 90\% & Presence, Normal & These should be stored in the Tasks database and the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.011 & To test that the IndividualPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 80\% & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.012 \* & To test that the Ratings are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Green & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.013 \* & To test that the Feedback is stored in the correct place in the database & This should be stored in and visible together by the administrator or individually by the receiving student & Very good & Presence, Normal & This should be stored in the Tasks database and visible in whole by the administrator or individually by the receiving student & & \\ \hline

4.001 \* & To test that the algorithm for initially inputting names and IDs into the system works as intended & The system should accept the administrators inputs and save them to the right place in the database, before generating an account space for each user along with a username and password & Input John Smith, 10A into the system & Presence, Erroneous & names should be accepted and stored, and usernames and passwords should be generated for each of them & & \\ \hline

4.002 \* & To test that the algorithm for generating usernames works as intended & For each user's name input into the system, a username consisting of part or all of the first name followed by a number should be generated and stored with the name in the database & Generate something like 'John1' & Erroneous & The username should be generated and stored with the names of each user, and should be recognised upon logging in & & \\ \hline

4.003 \* & To test that the algorithm for generating passwords works as intended & For each user's name input into the system, a password consisting of a letter, number, letter pattern of 7 characters should be generated and stored with the name in the database & Generate something like 'd7h3g6j' & Erroneous & The password should be generated and stored with the names of each user, and should be recognised upon logging in & & \\ \hline

4.004 \* & To test that the algorithm for validating a login works as intended & The system should recognise and accept a valid username and password, or reject an invalid username and password as stored in the database & enter John1, d7g4h5j into the login line edits & Erroneous & Correct login details will be accepted and incorrect login details will be rejected & & \\ \hline

4.005 \* & To test that the algorithm for calculating an answer to a pythagoras theorem related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.006 \* & To test that the alternative algorithm for calculating an answer to a pythagoras theorem related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.007 \* & To test that the algorithm for calculating an answer to a 3D pythagoras related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.008 \* & To test that the algorithm for calculating an answer to a sine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter sinA = 9 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.009 \* & To test that the algorithm for calculating an answer to a cosine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter cosineA = 7 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.010 \* & To test that the algorithm for calculating an answer to a tan rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter tanA = 8 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.011 \* & To test that the algorithm for calculating an answer to a finding angles using the cosine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter angle {$B = 45^o$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.012 \* & To test that the algorithm for calculating an answer to a finding angles using the tan rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter angle {$A = 30^0$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.013 \* & To test that the algorithm for calculating an answer to an inverted angles related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$invA = 20^0$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.014 \* & To test that the algorithm for calculating an answer to an area of triangle related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$area = 40cm^2$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

5.001 & To test that the entire system adequately meets the clients requests & The system should be as the client specified & Go through every aspect of the system with the client to ensure that they are satisfied & Normal & The client should be satisfied with the system and allow it to remain the same, or request some changes & & \\ \hline

5.002 \* & To test that all login details are accepted and recognised by the system, and that each log in has its own personal account to maintain progress and privacy & The system should accept the log in given (if it should be valid) and give the user access to their personal account & Input all login details one at a time & Erroneous & The system should allow access for each log in provided with each account being private & & \\ \hline

5.003 & To test that the right amount of information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right amount of information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right amount of information should be stored in the database & & \\ \hline

5.004 & To test that the right information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right information should be stored in the database & & \\ \hline

5.005 \* & To test that the local area network (LAN) connections across the system (for the database access) works as intended & Each account from any computer should be able to access all of the information which they have authorised access to from the server & Log in to multiple account on multiple computers in a LAN and attempt to store and read information & Normal & Each account and computer should be able to view their information from the server database and store new information & & \\ \hline

5.006 & To thoroughly ensure that the Data Protection Act 1998 is not breached at any point in the system, and that every measure is taken to prevent it from being breached in the future & The test will consist of checking every aspect of the database and algorithms involved in storing information & Check the algorithms work and store the right amount of the right information, and erase information after the legal minimum time frame & Normal & The system should comply with the Data Protection Act 1998 & & \\ \hline

\end{longtable}

\end{center}

\subsection{Changes to Detailed Plan}

\textbf{\*\* signifies tests which have been added since the test plan was updated.}

\begin{center}

\begin{longtable}{|p{1.5cm}|p{2.5cm}|p{2.5cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|}

\hline

1.003 & To test the lessons button on the student account screen functions as intended & This should link to the lesson topic menu screen & Click the lessons button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.004 & To test the homework button on the student account screen functions as intended & This should link to the homework topic menu screen & Click the homework button & Normal & The homework topic menu should be displayed & & \\ \hline

1.005 & To test the progress button on the student account screen functions as intended & This should link to the student's personal database display screen & Click the progress button & Normal & The student's personal database screen should be displayed & & \\ \hline

1.006 & To test the exit program button on the student account screen functions as intended & This should close down the entire program & Click the exit program button & Normal & The program should stop & & \\ \hline

1.007 & To test the Trigonometry 1 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.008 & To test the Trigonometry 2 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.009 & To test the Pythagoras button on the lesson topic menu screen functions as intended & This should link to the pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.010 & To test the Vectors button on the lesson topic menu screen functions as intended & This should link to the Vectors menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.011 & To test the Summary button on the lesson topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.012 & To test the return button on the lesson topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.013 & To test the Sides button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the sides lesson & Click the Sides button & Normal & The first sides lesson screen should be displayed & & \\ \hline

1.014 & To test the SOHCAHTOA button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the SOHCAHTOA lesson & Click the SOHCAHTOA button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.015 & To test the return button on the Trigonometry 1 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.016 & To test the Finding Angles button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the finding angles lesson & Click the Finding angles button & Normal & The first finding angles lesson screen should be displayed & & \\ \hline

1.018 & To test the 3D Trigonometry button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the 3D Trigonometry lesson & Click the 3D Trigonometry button & Normal & The first 3D trigonometry lesson screen should be displayed & & \\ \hline

1.019 & To test the return button on the Trigonometry 2 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.020 & To test the Pythagoras Theorem button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the pythagoras theorem lesson & Click the pythagoras theorem button & Normal & The first pythagoras theorem lesson screen should be displayed & & \\ \hline

1.021 & To test the 3D Pythagoras button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the 3D pythagoras lesson & Click the 3D pythagoras button & Normal & The first 3D pythagoras lesson screen should be displayed & & \\ \hline

1.022 & To test the return button on the Pythagoras menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.023 & To test the easy vectors button on the Vectors lesson menu functions as intended & This should link to the screen for the first page of the easy vectors lesson & Click the easy vectors button & Normal & The first easy vectors lesson screen should be displayed & & \\ \hline

1.024 & To test the Medium Vectors button on the Vectors lesson menu functions as intended & This should link to the screen for the first page of the medium Vectors lesson & Click the medium Vectors button & Normal & The first medium Vectors lesson screen should be displayed & & \\ \hline

1.025 & To test the Hard Vectors button on the Vectors lesson menu functions as intended & This should link to the screen for the first page of the hard Vectors lesson & Click the hard Vectors button & Normal & The first hard Vectors lesson screen should be displayed & & \\ \hline

1.026 & To test the return button on the Vectors menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.027 & To test the Revise Trigonometry 1 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 1 lesson & Click the revise trigonometry 1 button & Normal & The first revise trigonometry 1 lesson screen should be displayed & & \\ \hline

1.028 & To test the Revise Trigonometry 2 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 2 lesson & Click the revise trigonometry 2 button & Normal & The first revise trigonometry 2 lesson screen should be displayed & & \\ \hline

1.029 & To test the Revise Trigonometry 3 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 3 lesson & Click the revise trigonometry 3 button & Normal & The first revise trigonometry 3 lesson screen should be displayed & & \\ \hline

1.030 & To test the return button on the Summary menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.031 & To test the return button on the first Sides lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the Sides return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.032 & To test the next button on the first Sides lesson screen functions as intended & This should link to the second Sides lesson screen & Click the Sides next button & Normal & The second Sides lesson screen should be displayed & & \\ \hline

1.033 & To test the previous button on the second Sides lesson screen functions as intended & This should link back to the first Sides lesson screen & Click the Sides previous button & Normal & The first Sides lesson screen should be displayed & & \\ \hline

1.034 & To test the Check Answer button on the second Sides lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Sides check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.035 & To test the Finish button on the second Sides lesson screen functions as intended & This close the lesson and return to the lesson topic menu screen & Click the Sides finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.036 & To test the return button on the first SOHCAHTOA lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the SOHCAHTOA return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.037 & To test the next button on the first SOHCAHTOA lesson screen functions as intended & This should link to the second SOHCAHTOA lesson screen & Click the SOHCAHTOA next button & Normal & The second SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.038 & To test the previous button on the second SOHCAHTOA lesson screen functions as intended & This should link back to the first SOHCAHTOA lesson screen & Click the SOHCAHTOA previous button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.039 & To test the Check Answer button on the second SOHCAHTOA lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the SOHCAHTOA check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.040 & To test the Finish button on the second SOHCAHTOA lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the SOHCAHTOA finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.041 & To test the return button on the first Finding Angles lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the Finding Angles return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.042 & To test the next button on the first Finding Angles lesson screen functions as intended & This should link to the second Finding Angles lesson screen & Click the Finding Angles next button & Normal & The second Finding Angles lesson screen should be displayed & & \\ \hline

1.043 & To test the previous button on the second Finding Angles lesson screen functions as intended & This should link back to the first Finding Angles lesson screen & Click the Finding Angles previous button & Normal & The first Finding Angles lesson screen should be displayed & & \\ \hline

1.044 & To test the Check Answer button on the second Finding Angles lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Finding Angles check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.045 & To test the Finish button on the second Finding Angles lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Finding Angles finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.051 & To test the return button on the first 3D Trigonometry lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the 3D Trigonometry return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.052 & To test the next button on the first 3D Trigonometry lesson screen functions as intended & This should link to the second 3D Trigonometry lesson screen & Click the 3D Trigonometry next button & Normal & The second 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.053 & To test the previous button on the second 3D Trigonometry lesson screen functions as intended & This should link back to the first 3D Trigonometry lesson screen & Click the 3D Trigonometry previous button & Normal & The first 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.054 & To test the Check Answer button on the second 3D Trigonometry lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Trigonometry check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.055 & To test the Finish button on the second 3D Trigonometry lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Trigonometry finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.056 & To test the return button on the first Pythagoras Theorem lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the Pythagoras Theorem return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.057 & To test the next button on the first Pythagoras Theorem lesson screen functions as intended & This should link to the second Pythagoras Theorem lesson screen & Click the Pythagoras Theorem next button & Normal & The second Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.058 & To test the previous button on the second Pythagoras Theorem lesson screen functions as intended & This should link back to the first Pythagoras Theorem lesson screen & Click the Pythagoras Theorem previous button & Normal & The first Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.059 & To test the Check Answer button on the second Pythagoras Theorem lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras Theorem check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.060 & To test the Finish button on the second Pythagoras Theorem lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras Theorem finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.061 & To test the return button on the first 3D Pythagoras lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the 3D Pythagoras return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.062 & To test the next button on the first 3D Pythagoras lesson screen functions as intended & This should link to the second 3D Pythagoras lesson screen & Click the 3D Pythagoras next button & Normal & The second 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.063 & To test the previous button on the second 3D Pythagoras lesson screen functions as intended & This should link back to the first 3D Pythagoras lesson screen & Click the 3D Pythagoras previous button & Normal & The first 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.064 & To test the Check Answer button on the second 3D Pythagoras lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Pythagoras check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.065 & To test the Finish button on the second 3D Pythagoras lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Pythagoras finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.066 & To test the return button on the first Vectors Easy lesson screen functions as intended & This should link back to the Vectors menu screen & Click theVectors Easy return button & Normal & The Vectors menu screen should be displayed & & \\ \hline

1.067 & To test the next button on the first Vectors Easy lesson screen functions as intended & This should link to the second Vectors Easy lesson screen & Click the Vectors Easy next button & Normal & The second Vectors Easy lesson screen should be displayed & & \\ \hline

1.068 & To test the previous button on the second Vectors Easy lesson screen functions as intended & This should link back to the first Vectors Easy lesson screen & Click the Vectors Easy previous button & Normal & The first Vectors Easy lesson screen should be displayed & & \\ \hline

1.069 & To test the Check Answer button on the second Vectors Easy lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Vectors Easy check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.070 & To test the Finish button on the second Vectors Easy lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Vectors Easy finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.071 & To test the return button on the first Vectors Medium lesson screen functions as intended & This should link back to the Vectors menu screen & Click theVectors Medium return button & Normal & The Vectors menu screen should be displayed & & \\ \hline

1.072 & To test the next button on the first Vectors Medium lesson screen functions as intended & This should link to the second Vectors Medium lesson screen & Click the Vectors Medium next button & Normal & The second Vectors Medium lesson screen should be displayed & & \\ \hline

1.073 & To test the previous button on the second Vectors Medium lesson screen functions as intended & This should link back to the first Vectors Medium lesson screen & Click the Vectors Medium previous button & Normal & The first Vectors Medium lesson screen should be displayed & & \\ \hline

1.074 & To test the Check Answer button on the second Vectors Medium lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Vectors Medium check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.075 & To test the Finish button on the second Vectors Medium lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Vectors Medium finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.076 & To test the return button on the first Vectors Hard lesson screen functions as intended & This should link back to the Vectors menu screen & Click theVectors Hard return button & Normal & The Vectors menu screen should be displayed & & \\ \hline

1.077 & To test the next button on the first Vectors Hard lesson screen functions as intended & This should link to the second Vectors Hard lesson screen & Click the Vectors Hard next button & Normal & The second Vectors Hard lesson screen should be displayed & & \\ \hline

1.078 & To test the previous button on the second Vectors Hard lesson screen functions as intended & This should link back to the first Vectors Hard lesson screen & Click the Vectors Hard previous button & Normal & The first Vectors Hard lesson screen should be displayed & & \\ \hline

1.079 & To test the Check Answer button on the second Vectors Hard lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Vectors Hard check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.080 & To test the Finish button on the second Vectors Hard lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Vectors Hard finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.081 & To test the return button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 1 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.082 & To test the next button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link to the second Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 next button & Normal & The second Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.083 & To test the previous button on the second Revise Trigonometry 1 lesson screen functions as intended & This should link back to the first Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 previous button & Normal & The first Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.084 & To test the Check Answer button on the second Revise Trigonometry 1 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 1 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.085 & To test the Finish button on the second Revise Trigonometry 1 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 1 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.086 & To test the return button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 2 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.087 & To test the next button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link to the second Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 next button & Normal & The second Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.088 & To test the previous button on the second Revise Trigonometry 2 lesson screen functions as intended & This should link back to the first Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 previous button & Normal & The first Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.089 & To test the Check Answer button on the second Revise Trigonometry 2 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 2 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.090 & To test the Finish button on the second Revise Trigonometry 2 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 2 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.091 & To test the return button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 3 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.092 & To test the next button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link to the second Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 next button & Normal & The second Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.093 & To test the previous button on the second Revise Trigonometry 3 lesson screen functions as intended & This should link back to the first Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 previous button & Normal & The first Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.094 & To test the Check Answer button on the second Revise Trigonometry 3 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 3 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.095 & To test the Finish button on the second Revise Trigonometry 3 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 3 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.097 & To test the Trigonometry 1 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.098 & To test the Trigonometry 2 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.099 & To test the Pythagoras button on the homework topic menu screen functions as intended & This should link to the Pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.100 & To test the Vectors button on the homework topic menu screen functions as intended & This should link to the Vectors menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.101 & To test the Summary button on the homework topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.102 & To test the return button on the homework topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.103 & To test the Sides Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Easy homework screen & click the Sides Easy button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.104 & To test the Sides Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Medium homework screen & click the Sides Medium button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.105 & To test the Sides Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Hard homework screen & click the Sides Hard button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.106 & To test the SOHCAHTOA Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Easy homework screen & click the SOHCAHTOA Easy button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.107 & To test the SOHCAHTOA Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Medium homework screen & click the SOHCAHTOA Medium button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.108 & To test the SOHCAHTOA Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Hard homework screen & click the SOHCAHTOA Hard button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.109 & To test the return button on the Trigonometry 1 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.110 & To test the Finding Angles Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Easy homework screen & click the Finding Angles Easy button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.111 & To test the Finding Angles Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Medium homework screen & click the Finding angles Medium button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.112 & To test the Finding Angles Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Hard homework screen & click the Finding angles Hard button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.116 & To test the 3D Trigonometry Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Easy homework screen & click the 3D Trigonometry Easy button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.117 & To test the 3D Trigonometry Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Medium homework screen & click the 3D Trigonometry Medium button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.118 & To test the 3D Trigonometry Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Hard homework screen & click the 3D Trigonometry Hard button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.119 & To test the return button on the Trigonometry 2 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.120 & To test the Pythagoras Theorem Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Easy homework screen & click the Pythagoras Theorem Easy button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.121 & To test the Pythagoras Theorem Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Medium homework screen & click the Pythagoras Theorem Medium button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.122 & To test the Pythagoras Theorem Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Hard homework screen & click the Pythagoras Theorem Hard button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.123 & To test the 3D Pythagoras Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Easy homework screen & click the 3D Pythagoras Easy button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.124 & To test the 3D Pythagoras Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Medium homework screen & click the 3D Pythagoras Medium button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.125 & To test the 3D Pythagoras Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Hard homework screen & click the 3D Pythagoras Hard button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.126 & To test the return button on the Pythagoras homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.127 & To test the Vectors Easy button on the Vectors homework menu screen functions as intended & This should link to the first Vectors Easy homework screen & Click the Vectors Easy button & Normal & The first Vectors Easy homework screen should be displayed & & \\ \hline

1.128 & To test the Vectors Medium button on the Vectors homework menu screen functions as intended & This should link to the first Vectors Medium homework screen & Click the Vectors Medium button & Normal & The first Vectors Medium homework screen should be displayed & & \\ \hline

1.129 & To test the Vectors Hard button on the Vectors homework menu screen functions as intended & This should link to the first Vectors Hard homework screen & Click the Vectors Hard button & Normal & The first Vectors Hard homework screen should be displayed & & \\ \hline

1.130 & To test the return button on the Vectors homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.131 & To test the Summary Easy button on the Summary homework menu screen functions as intended & This should link to the first Summary Easy homework screen & Click the Summary Easy button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.132 & To test the Summary Medium button on the Summary homework menu screen functions as intended & This should link to the first Summary Medium homework screen & Click the Summary Medium button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.133 & To test the Summary Hard button on the Summary homework menu screen functions as intended & This should link to the first Summary Hard homework screen & Click the Summary Hard button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.134 & To test the return button on the Summary homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.135 & To test the cancel button on the Sides Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.136 & To test the check answers button on the Sides Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.137 & To test the reset answers button on the Sides Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.138 & To test the next button on the Sides Easy first homework screen functions as intended & This should link to the second Sides Easy homework screen & Click the next button & Normal & The second Sides Easy homework screen should be displayed & & \\ \hline

1.139 & To test the first mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.140 & To test the second mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.141 & To test the third mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.142 & To test the previous button on the Sides Easy second homework screen functions as intended & This should link back to the first Sides Easy homework screen & Click the previous button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.143 & To test the finish button on the second Sides Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.144 & To test the cancel button on the Sides Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.145 & To test the check answers button on the Sides Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.146 & To test the reset answers button on the Sides Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.147 & To test the next button on the Sides Medium first homework screen functions as intended & This should link to the second Sides Medium homework screen & Click the next button & Normal & The second Sides Medium homework screen should be displayed & & \\ \hline

1.148 & To test the first mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.149 & To test the second mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.150 & To test the third mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.151 & To test the previous button on the Sides Medium second homework screen functions as intended & This should link back to the first Sides Medium homework screen & Click the previous button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.152 & To test the finish button on the second Sides Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.153 & To test the cancel button on the Sides Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.154 & To test the check answers button on the Sides Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.155 & To test the reset answers button on the Sides Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.156 & To test the next button on the Sides Hard first homework screen functions as intended & This should link to the second Sides Hard homework screen & Click the next button & Normal & The second Sides Hard homework screen should be displayed & & \\ \hline

1.157 & To test the first mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.158 & To test the second mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.159 & To test the third mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.160 & To test the previous button on the Sides Hard second homework screen functions as intended & This should link back to the first Sides Hard homework screen & Click the previous button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.161 & To test the finish button on the second Sides Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.162 & To test the cancel button on the SOHCAHTOA Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.163 & To test the check answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.164 & To test the reset answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.165 & To test the next button on the SOHCAHTOA Easy first homework screen functions as intended & This should link to the second SOHCAHTOA Easy homework screen & Click the next button & Normal & The second SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.166 & To test the first mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.167 & To test the second mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.168 & To test the third mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.169 & To test the previous button on the SOHCAHTOA Easy second homework screen functions as intended & This should link back to the first SOHCAHTOA Easy homework screen & Click the previous button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.170 & To test the finish button on the second SOHCAHTOA Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.171 & To test the cancel button on the SOHCAHTOA Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.172 & To test the check answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.173 & To test the reset answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.174 & To test the next button on the SOHCAHTOA Medium first homework screen functions as intended & This should link to the second SOHCAHTOA Medium homework screen & Click the next button & Normal & The second SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.175 & To test the first mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.176 & To test the second mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.177 & To test the third mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.178 & To test the previous button on the SOHCAHTOA Medium second homework screen functions as intended & This should link back to the first SOHCAHTOA Medium homework screen & Click the previous button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.179 & To test the finish button on the second SOHCAHTOA Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.180 & To test the cancel button on the SOHCAHTOA Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.181 & To test the check answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.182 & To test the reset answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.183 & To test the next button on the SOHCAHTOA Hard first homework screen functions as intended & This should link to the second SOHCAHTOA Hard homework screen & Click the next button & Normal & The second SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.184 & To test the first mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.185 & To test the second mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.186 & To test the third mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.187 & To test the previous button on the SOHCAHTOA Hard second homework screen functions as intended & This should link back to the first SOHCAHTOA Hard homework screen & Click the previous button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.188 & To test the finish button on the second SOHCAHTOA Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.189 & To test the cancel button on the Finding Angles Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.190 & To test the check answers button on the Finding Angles Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.191 & To test the reset answers button on the Finding Angles Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.192 & To test the next button on the Finding Angles Easy first homework screen functions as intended & This should link to the second Finding Angles Easy homework screen & Click the next button & Normal & The second Finding Angles Easy homework screen should be displayed & & \\ \hline

1.193 & To test the first mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.194 & To test the second mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.195 & To test the third mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.196 & To test the previous button on the Finding Angles Easy second homework screen functions as intended & This should link back to the first Finding Angles Easy homework screen & Click the previous button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.197 & To test the finish button on the second Finding Angles Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.198 & To test the cancel button on the Finding Angles Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.199 & To test the check answers button on the Finding Angles Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.200 & To test the reset answers button on the Finding Angles Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.201 & To test the next button on the Finding Angles Medium first homework screen functions as intended & This should link to the second Finding Angles Medium homework screen & Click the next button & Normal & The second Finding Angles Medium homework screen should be displayed & & \\ \hline

1.202 & To test the first mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.203 & To test the second mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.204 & To test the third mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.205 & To test the previous button on the Finding Angles Medium second homework screen functions as intended & This should link back to the first Finding Angles Medium homework screen & Click the previous button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.206 & To test the finish button on the second Finding Angles Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.207 & To test the cancel button on the Finding Angles Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.208 & To test the check answers button on the Finding Angles Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.209 & To test the reset answers button on the Finding Angles Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.210 & To test the next button on the Finding Angles Hard first homework screen functions as intended & This should link to the second Finding Angles Hard homework screen & Click the next button & Normal & The second Finding Angles Hard homework screen should be displayed & & \\ \hline

1.211 & To test the first mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.212 & To test the second mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.213 & To test the third mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.214 & To test the previous button on the Finding Angles Hard second homework screen functions as intended & This should link back to the first Finding Angles Hard homework screen & Click the previous button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.215 & To test the finish button on the second Finding Angles Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.243 & To test the cancel button on the 3D Trigonometry Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.244 & To test the check answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.245 & To test the reset answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.246 & To test the next button on the 3D Trigonometry Easy first homework screen functions as intended & This should link to the second 3D Trigonometry Easy homework screen & Click the next button & Normal & The second 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.247 & To test the first mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.248 & To test the second mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.249 & To test the third mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.250 & To test the previous button on the 3D Trigonometry Easy second homework screen functions as intended & This should link back to the first 3D Trigonometry Easy homework screen & Click the previous button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.251 & To test the finish button on the second 3D Trigonometry Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.252 & To test the cancel button on the 3D Trigonometry Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.253 & To test the check answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.254 & To test the reset answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.255 & To test the next button on the 3D Trigonometry Medium first homework screen functions as intended & This should link to the second 3D Trigonometry Medium homework screen & Click the next button & Normal & The second 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.256 & To test the first mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.257 & To test the second mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.258 & To test the third mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.259 & To test the previous button on the 3D Trigonometry Medium second homework screen functions as intended & This should link back to the first 3D Trigonometry Medium homework screen & Click the previous button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.260 & To test the finish button on the second 3D Trigonometry Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.261 & To test the cancel button on the 3D Trigonometry Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.262 & To test the check answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.263 & To test the reset answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.264 & To test the next button on the 3D Trigonometry Hard first homework screen functions as intended & This should link to the second 3D Trigonometry Hard homework screen & Click the next button & Normal & The second 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.265 & To test the first mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.266 & To test the second mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.267 & To test the third mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.268 & To test the previous button on the 3D Trigonometry Hard second homework screen functions as intended & This should link back to the first 3D Trigonometry Hard homework screen & Click the previous button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.269 & To test the finish button on the second 3D Trigonometry Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.270 & To test the cancel button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.271 & To test the check answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.272 & To test the reset answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.273 & To test the next button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link to the second Pythagoras Theorem Easy homework screen & Click the next button & Normal & The second Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.274 & To test the first mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.275 & To test the second mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.276 & To test the third mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.277 & To test the previous button on the Pythagoras Theorem Easy second homework screen functions as intended & This should link back to the first Pythagoras Theorem Easy homework screen & Click the previous button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.278 & To test the finish button on the second Pythagoras Theorem Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.279 & To test the cancel button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.280 & To test the check answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.281 & To test the reset answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.282 & To test the next button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link to the second Pythagoras Theorem Medium homework screen & Click the next button & Normal & The second Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.283 & To test the first mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.284 & To test the second mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.285 & To test the third mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.286 & To test the previous button on the Pythagoras Theorem Medium second homework screen functions as intended & This should link back to the first Pythagoras Theorem Medium homework screen & Click the previous button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.287 & To test the finish button on the second Pythagoras Theorem Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.288 & To test the cancel button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.289 & To test the check answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.290 & To test the reset answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.291 & To test the next button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link to the second Pythagoras Theorem Hard homework screen & Click the next button & Normal & The second Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.292 & To test the first mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.293 & To test the second mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.294 & To test the third mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.295 & To test the previous button on the Pythagoras Theorem Hard second homework screen functions as intended & This should link back to the first Pythagoras Theorem Hard homework screen & Click the previous button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.296 & To test the finish button on the second Pythagoras Theorem Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.297 & To test the cancel button on the 3D Pythagoras Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.298 & To test the check answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.299 & To test the reset answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.300 & To test the next button on the 3D Pythagoras Easy first homework screen functions as intended & This should link to the second 3D Pythagoras Easy homework screen & Click the next button & Normal & The second 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.301 & To test the first mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.302 & To test the second mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.303 & To test the third mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.304 & To test the previous button on the 3D Pythagoras Easy second homework screen functions as intended & This should link back to the first 3D Pythagoras Easy homework screen & Click the previous button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.305 & To test the finish button on the second 3D Pythagoras Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.306 & To test the cancel button on the 3D Pythagoras Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.307 & To test the check answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.308 & To test the reset answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.309 & To test the next button on the 3D Pythagoras Medium first homework screen functions as intended & This should link to the second 3D Pythagoras Medium homework screen & Click the next button & Normal & The second 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.310 & To test the first mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.311 & To test the second mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.312 & To test the third mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.313 & To test the previous button on the 3D Pythagoras Medium second homework screen functions as intended & This should link back to the first 3D Pythagoras Medium homework screen & Click the previous button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.314 & To test the finish button on the second 3D Pythagoras Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.315 & To test the cancel button on the 3D Pythagoras Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.316 & To test the check answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.317 & To test the reset answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.318 & To test the next button on the 3D Pythagoras Hard first homework screen functions as intended & This should link to the second 3D Pythagoras Hard homework screen & Click the next button & Normal & The second 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.319 & To test the first mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.320 & To test the second mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.321 & To test the third mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.322 & To test the previous button on the 3D Pythagoras Hard second homework screen functions as intended & This should link back to the first 3D Pythagoras Hard homework screen & Click the previous button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.323 & To test the finish button on the second 3D Pythagoras Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.324 & To test the cancel button on the Vectors Easy first homework screen functions as intended & This should link back to the Vectors lesson menu screen & Click the cancel button & Normal & The Vectors homework menu should be displayed & & \\ \hline

1.325 & To test the check answers button on the Vectors Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.326 & To test the reset answers button on the Vectors Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.327 & To test the next button on the Vectors Easy first homework screen functions as intended & This should link to the second Vectors Easy homework screen & Click the next button & Normal & The second Vectors Easy homework screen should be displayed & & \\ \hline

1.328 & To test the first mark it button on the Vectors Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.329 & To test the second mark it button on the Vectors Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.330 & To test the third mark it button on the Vectors Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.331 & To test the previous button on the Vectors Easy second homework screen functions as intended & This should link back to the first Vectors Easy homework screen & Click the previous button & Normal & The first Vectors Easy homework screen should be displayed & & \\ \hline

1.332 & To test the finish button on the second Vectors Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.333 & To test the cancel button on the Vectors Medium first homework screen functions as intended & This should link back to the Vectors lesson menu screen & Click the cancel button & Normal & The Vectors homework menu should be displayed & & \\ \hline

1.334 & To test the check answers button on the Vectors Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.335 & To test the reset answers button on the Vectors Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.336 & To test the next button on the Vectors Medium first homework screen functions as intended & This should link to the second Vectors Medium homework screen & Click the next button & Normal & The second Vectors Medium homework screen should be displayed & & \\ \hline

1.337 & To test the first mark it button on the Vectors Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.338 & To test the second mark it button on the Vectors Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.339 & To test the third mark it button on the Vectors Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.340 & To test the previous button on the Vectors Medium second homework screen functions as intended & This should link back to the first Vectors Medium homework screen & Click the previous button & Normal & The first Vectors Medium homework screen should be displayed & & \\ \hline

1.341 & To test the finish button on the second Vectors Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.342 & To test the cancel button on the Vectors Hard first homework screen functions as intended & This should link back to the Vectors lesson menu screen & Click the cancel button & Normal & The Vectors homework menu should be displayed & & \\ \hline

1.343 & To test the check answers button on the Vectors Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.344 & To test the reset answers button on the Vectors Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.345 & To test the next button on the Vectors Hard first homework screen functions as intended & This should link to the second Vectors Hard homework screen & Click the next button & Normal & The second Vectors Hard homework screen should be displayed & & \\ \hline

1.346 & To test the first mark it button on the Vectors Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.347 & To test the second mark it button on the Vectors Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.348 & To test the third mark it button on the Vectors Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.349 & To test the previous button on the Vectors Hard second homework screen functions as intended & This should link back to the first Vectors Hard homework screen & Click the previous button & Normal & The first Vectors Hard homework screen should be displayed & & \\ \hline

1.350 & To test the finish button on the second Vectors Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.351 & To test the cancel button on the Summary Easy first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.352 & To test the check answers button on the Summary Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.353 & To test the reset answers button on the Summary Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.354 & To test the next button on the Summary Easy first homework screen functions as intended & This should link to the second Summary Easy homework screen & Click the next button & Normal & The second Summary Easy homework screen should be displayed & & \\ \hline

1.355 & To test the first mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.356 & To test the second mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.357 & To test the third mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.358 & To test the previous button on the Summary Easy second homework screen functions as intended & This should link back to the first Summary Easy homework screen & Click the previous button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.359 & To test the finish button on the second Summary Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.360 & To test the cancel button on the Summary Medium first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.361 & To test the check answers button on the Summary Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.362 & To test the reset answers button on the Summary Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.363 & To test the next button on the Summary Medium first homework screen functions as intended & This should link to the second Summary Medium homework screen & Click the next button & Normal & The second Summary Medium homework screen should be displayed & & \\ \hline

1.364 & To test the first mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.365 & To test the second mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.366 & To test the third mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.367 & To test the previous button on the Summary Medium second homework screen functions as intended & This should link back to the first Summary Medium homework screen & Click the previous button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.368 & To test the finish button on the second Summary Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.369 & To test the cancel button on the Summary Hard first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.370 & To test the check answers button on the Summary Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.371 & To test the reset answers button on the Summary Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.372 & To test the next button on the Summary Hard first homework screen functions as intended & This should link to the second Summary Hard homework screen & Click the next button & Normal & The second Summary Hard homework screen should be displayed & & \\ \hline

1.373 & To test the first mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.374 & To test the second mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.375 & To test the third mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.376 & To test the previous button on the Summary Hard second homework screen functions as intended & This should link back to the first Summary Hard homework screen & Click the previous button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.377 & To test the finish button on the second Summary Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.380 & To test the return button on the progress screen functions as intended & This should link back to the student account home screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.431 & To test the return button on the student report screen functions as intended & This should link back to the student progress screen & Click the return button & Normal & The student progress screen should be displayed & & \\ \hline

1.432 & To test the query button on the student report screen functions as intended & This should link back to the student progress screen and new queried information should be displayed & Click the query button & Normal & The student progress screen should be displayed with updated data in the database widget & & \\ \hline

1.441 \*\* & To test the continue button on the first screen functions as intended & This should link to the home screen & Click the continue button & Normal & The home screen should be displayed & & \\ \hline

1.442 \*\* & To test the report button on the first screen functions as intended & This should link to the report screen & Click the report button & Normal & The report screen should be displayed & & \\ \hline

2.003 & To test that the input for the practice question on the Sides second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.004 & To test that the input for the practice question on the SOHCAHTOA second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.005 & To test that the input for the practice question on the Finding Angles second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.007 & To test that the input for the practice question on the 3D Trigonometry second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.008 & To test that the input for the practice question on the Pythagoras' Theorem second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.009 & To test that the input for the practice question on the 3D Pythagoras second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.010 & To test that the input for the practice question on the Vectors Easy second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.011 & To test that the input for the practice question on the Vectors Medium second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.012 & To test that the input for the practice question on the Vectors Hard second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.013 & To test that the input for the practice question on the Revise Trigonometry 1 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.014 & To test that the input for the practice question on the Revise Trigonometry 2 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.015 & To test that the input for the practice question on the Revise Trigonometry 3 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.016 & To test that the text input for the generic question 1 part A is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.017 & To test that the text input for the generic question 1 part B is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.018 & To test that the text input for the generic question 1 part C is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon presisng the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.019 & To test that the text input for the generic question 1 part D is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.020 & To test that the text input for the generic question 1 part E is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.021 & To test that the text input for the generic question 1 part F is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.022 & To test that the combo box input for the generic question 2 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.023 & To test that the combo box input for the generic question 3 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.024 & To test that the button selection inputs for the generic question 4 are received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.025 & To test that the text input for the generic question 1 part A is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.026 & To test that the text input for the generic question 1 part B is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.027 & To test that the text input for the generic question 1 part C is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.028 & To test that the text input for the generic question 1 part D is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.029 & To test that the text input for the generic question 1 part E is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.030 & To test that the text input for the generic question 1 part F is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.031 & To test that the combo box input for the generic question 2 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.032 & To test that the combo box input for the generic question 3 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.033 & To test that the button selection inputs for the generic question 4 are received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.034 & To test that the text input for the generic question 1 part A is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.035 & To test that the text input for the generic question 1 part B is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.036 & To test that the text input for the generic question 1 part C is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.037 & To test that the text input for the generic question 1 part D is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.038 & To test that the text input for the generic question 1 part E is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.039 & To test that the text input for the generic question 1 part F is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.040 & To test that the combo box input for the generic question 2 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.041 & To test that the combo box input for the generic question 3 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.042 & To test that the button selection inputs for the generic question 4 are received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.043 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.044 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.045 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.046 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.047 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.048 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.049 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.050 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.051 & To test that the button selection inputs for the generic question 4 are received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.052 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.053 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.054 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.055 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.056 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.057 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.058 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.059 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.060 & To test that the button selection inputs for the generic question 4 are received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.061 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.062 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.063 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.064 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.065 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.066 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.067 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.068 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.069 & To test that the button selection inputs for the generic question 4 are received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.070 & To test that the text input for the generic question 1 part A is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.071 & To test that the text input for the generic question 1 part B is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.072 & To test that the text input for the generic question 1 part C is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.073 & To test that the text input for the generic question 1 part D is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.074 & To test that the text input for the generic question 1 part E is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.075 & To test that the text input for the generic question 1 part F is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.076 & To test that the combo box input for the generic question 2 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.077 & To test that the combo box input for the generic question 3 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.078 & To test that the button selection inputs for the generic question 4 are received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.079 & To test that the text input for the generic question 1 part A is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.080 & To test that the text input for the generic question 1 part B is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.081 & To test that the text input for the generic question 1 part C is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.082 & To test that the text input for the generic question 1 part D is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.083 & To test that the text input for the generic question 1 part E is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.084 & To test that the text input for the generic question 1 part F is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.085 & To test that the combo box input for the generic question 2 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.086 & To test that the combo box input for the generic question 3 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.087 & To test that the button selection inputs for the generic question 4 are received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.088 & To test that the text input for the generic question 1 part A is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.089 & To test that the text input for the generic question 1 part B is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.090 & To test that the text input for the generic question 1 part C is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.091 & To test that the text input for the generic question 1 part D is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.092 & To test that the text input for the generic question 1 part E is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.093 & To test that the text input for the generic question 1 part F is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.094 & To test that the combo box input for the generic question 2 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.095 & To test that the combo box input for the generic question 3 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.096 & To test that the button selection inputs for the generic question 4 are received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.124 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.125 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.126 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.127 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.128 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.129 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.130 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.131 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.132 & To test that the button selection inputs for the generic question 4 are received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.133 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.134 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.135 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.136 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.137 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.138 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.139 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.140 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.141 & To test that the button selection inputs for the generic question 4 are received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.142 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.143 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.144 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.145 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.146 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.147 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.148 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.149 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.150 & To test that the button selection inputs for the generic question 4 are received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.151 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.152 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.153 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.154 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.155 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.156 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.157 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.158 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.159 & To test that the button selection inputs for the generic question 4 are received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.160 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.161 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.162 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.163 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.164 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.165 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.166 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.167 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.168 & To test that the button selection inputs for the generic question 4 are received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.169 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.170 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.171 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.172 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.173 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.174 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.175 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.176 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.177 & To test that the button selection inputs for the generic question 4 are received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.178 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.179 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.180 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.181 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.182 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.183 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.184 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.185 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.186 & To test that the button selection inputs for the generic question 4 are received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.187 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.188 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.189 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.190 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.191 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.192 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.193 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.194 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.195 & To test that the button selection inputs for the generic question 4 are received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.196 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.197 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.198 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.199 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.200 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.201 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.202 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.203 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.204 & To test that the button selection inputs for the generic question 4 are received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.205 & To test that the text input for the generic question 1 part A is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.206 & To test that the text input for the generic question 1 part B is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.207 & To test that the text input for the generic question 1 part C is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.208 & To test that the text input for the generic question 1 part D is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.209 & To test that the text input for the generic question 1 part E is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.210 & To test that the text input for the generic question 1 part F is received on the Vectors Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.211 & To test that the combo box input for the generic question 2 is received on the Vectors Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.212 & To test that the combo box input for the generic question 3 is received on the Vectors Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.213 & To test that the button selection inputs for the generic question 4 are received on the Vectors Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.214 & To test that the text input for the generic question 1 part A is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.215 & To test that the text input for the generic question 1 part B is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.216 & To test that the text input for the generic question 1 part C is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.217 & To test that the text input for the generic question 1 part D is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.218 & To test that the text input for the generic question 1 part E is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.219 & To test that the text input for the generic question 1 part F is received on the Vectors Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.220 & To test that the combo box input for the generic question 2 is received on the Vectors Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.221 & To test that the combo box input for the generic question 3 is received on the Vectors Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.222 & To test that the button selection inputs for the generic question 4 are received on the Vectors Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.223 & To test that the text input for the generic question 1 part A is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.224 & To test that the text input for the generic question 1 part B is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.225 & To test that the text input for the generic question 1 part C is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.226 & To test that the text input for the generic question 1 part D is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.227 & To test that the text input for the generic question 1 part E is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.228 & To test that the text input for the generic question 1 part F is received on the Vectors Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.229 & To test that the combo box input for the generic question 2 is received on the Vectors Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.230 & To test that the combo box input for the generic question 3 is received on the Vectors Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.231 & To test that the button selection inputs for the generic question 4 are received on the Vectors Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.232 & To test that the text input for the generic question 1 part A is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.233 & To test that the text input for the generic question 1 part B is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.234 & To test that the text input for the generic question 1 part C is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.235 & To test that the text input for the generic question 1 part D is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.236 & To test that the text input for the generic question 1 part E is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.237 & To test that the text input for the generic question 1 part F is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.238 & To test that the combo box input for the generic question 2 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.239 & To test that the combo box input for the generic question 3 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.240 & To test that the button selection inputs for the generic question 4 are received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.241 & To test that the text input for the generic question 1 part A is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.242 & To test that the text input for the generic question 1 part B is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.243 & To test that the text input for the generic question 1 part C is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.244 & To test that the text input for the generic question 1 part D is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.245 & To test that the text input for the generic question 1 part E is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.246 & To test that the text input for the generic question 1 part F is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.247 & To test that the combo box input for the generic question 2 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.248 & To test that the combo box input for the generic question 3 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.249 & To test that the button selection inputs for the generic question 4 are received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.250 & To test that the text input for the generic question 1 part A is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.251 & To test that the text input for the generic question 1 part B is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.252 & To test that the text input for the generic question 1 part C is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.253 & To test that the text input for the generic question 1 part D is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.254 & To test that the text input for the generic question 1 part E is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.255 & To test that the text input for the generic question 1 part F is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.256 & To test that the combo box input for the generic question 2 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.257 & To test that the combo box input for the generic question 3 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.258 & To test that the button selection inputs for the generic question 4 are received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & Each button clicked & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.295 & To test the the task selection combo box on the student report screen functions as intended & This should accept a task selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.296 & To test the the score selection combo box on the student report screen functions as intended & This should accept a score selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

3.009 & To test that the TaskNames are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Sides Easy & Presence, Normal & These should be stored in the Tasks database and visible together by the administrator or individually by the respective student & & \\ \hline

3.010 & To test that the OverallPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 90\% & Presence, Normal & These should be stored in the Tasks database and the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.011 & To test that the IndividualPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 80\% & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.012 & To test that the Ratings are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Green & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

4.015 \*\* & To test that the parent algorithm for checking answers to the generic question 1 works as intended & This should check the answers given and inform the user whether or not they are correct & Click the check answers button & Normal, erroneous & The user should be given marks or corrected & & \\ \hline

4.016 \*\* & To test that the parent algorithm for checking answers to the combo box questions works as intended & This should check the answers selected in the combo boxes and inform the user whether or not they are correct & Click the mark it button & Normal, erroneous & The user should be given marks or corrected & & \\ \hline

4.017 \*\* & To test that the parent algorithm for checking the button selection questions works as intended & This should check the button selected is correct and inform the user if they are correct or not & Click the buttons & Normal & The user should be given marks or corrected & & \\ \hline

4.018 \*\* & To test that the parent algorithm for resetting the line edits works as intended & This should clear all of the text in the line edits & Click the reset button & Normal & The line edits should be empty & & \\ \hline

5.001 & To test that the entire system adequately meets the clients requests & The system should be as the client specified & Go through every aspect of the system with the client to ensure that they are satisfied & Normal & The client should be satisfied with the system and allow it to remain the same, or request some changes & & \\ \hline

5.003 & To test that the right amount of information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right amount of information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right amount of information should be stored in the database & & \\ \hline

5.004 & To test that the right information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right information should be stored in the database & & \\ \hline

5.006 & To thoroughly ensure that the Data Protection Act 1998 is not breached at any point in the system, and that every measure is taken to prevent it from being breached in the future & The test will consist of checking every aspect of the database and algorithms involved in storing information & Check the algorithms work and store the right amount of the right information, and erase information after the legal minimum time frame & Normal & The system should comply with the Data Protection Act 1998 & & \\ \hline

\end{longtable}

\end{center}

\section{Test Data}

\subsection{Original Test Data}

\textbf{Test numbers with \* afterwards are tests which have been omitted from the updated testing plan.}

\begin{center}

\begin{longtable}{|p{4cm}|p{4cm}|p{5cm}|} \hline

\textbf{Test Number} & \textbf{Test Data} & \textbf{Justification for choice of test data} \\ \hline

1.001 \* & john1 + d5g7f4s; sdgj + pa\$\$w0rd & Firstly to test that an actual login allows access, then to ensure that a false login does not allow access \\ \hline

1.002 \* & admin1 + g6h7j3d; ahsdg + p@ssw0rD & Firstly to test that an actual login allows access, then to ensure that a false login does not allow access \\ \hline

1.003 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.004 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.005 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.006 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.007 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.012 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.013 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.015 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.031 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.032 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.033 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.034 & 1 (or correct answer); a, 4, three, £; None & Firstly to test that the correct answer is recognised. Then make sure that the error exceptions work, and finally to test that the appropriate error message appears if nothing is entered before continuing \\ \hline

1.035 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.097 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.102 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.103 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.135 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.136 & 1, 2, 3, 4, 5, 6; a, b, 4, three, £, \$; None & Firstly to test that the correct answers are recognised. Then make sure that the error exceptions work, and finally to test that the appropriate error message appears if nothing is entered before continuing \\ \hline

1.137 & 12345; abcde; !"£\$\% & To make sure that the line edits are cleared regardless of what the input is \\ \hline

1.138 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.139 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.140 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.141 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.142 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.143 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.380 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.381 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.382 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.383 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.384 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.386 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.391 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.392 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.424 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.425 \* & Select a homework then click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.426 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.427 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.428 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.429 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.430 \* & Sides Easy, 70\% & It is coded using a connection; therefore the only possible input \\ \hline

1.431 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.432 & Sides Easy, 70\% & It is coded using a connection; therefore the only possible input \\ \hline

1.433 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.434 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.435 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.436 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.437 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.438 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.439 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.440 \* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

2.001 \* & john1, sdgj & Firstly to test that an actual username allows access, then to ensure that a false username does not allow access \\ \hline

2.002 \* & k8j6g8h, PASSW0rd & Firstly to test that an actual password allows access, then to ensure that a false password does not allow access \\ \hline

2.003 & 45; words & To test that the correct answer works and then that any different data types do not crash it \\ \hline

2.016 & 1; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.017 & 2; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.018 & 3; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.019 & 4; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.020 & 5; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.021 & 6; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.022 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

2.023 & 50 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

2.024 & Click each button in turn & To test that the intended correct button works and the others don't \\ \hline

2.259 \* & admin1; u53rn4m3 & To test that the actual username works and any other attempts or data types are rejected efficiently \\ \hline

2.260 \* & d7h4j9q; p@\$\$w0rd & To test that the actual password works and any other attempts or data types are rejected efficiently \\ \hline

2.261 \* & 10A & To test that the right class receives the homework \\ \hline

2.262 \* & 5th Feb 2016 3:30PM & To test that the deadline actually expires when it is supposed to and the correct class gets it \\ \hline

2.263 \* & 70\% & To make sure that the class actually does have to reach this score before their homework is classed as finished \\ \hline

2.264 \* & Very good work & To make sure that strings of any character within a 500 character length are accepted \\ \hline

2.291 \* & 10A & To test that the SQL finds the correct data - there will be no erroneous data due to limited options\\ \hline

2.292 \* & John Smith & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.293 \* & Sides Easy & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.294 \* & 70\% & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.295 & Sides Easy & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.296 & 70\% & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.297 \* & John; jo hn; 34538 & To test that strings with a capital first letter and no spaces are accepted and anything else is rejected \\ \hline

2.298 \* & Smith; sm ith; 28433 & To test that strings with a capital first letter and no spaces are accepted and anything else is rejected \\ \hline

2.299 \* & 10A; 10 a; dsjkgbsa & To test that two integers followed by an uppercase letter and no spaces are accepted and anything else is rejected \\ \hline

3.001 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.002 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.003 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.004 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.005 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.006 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.007 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.008 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.009 & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.010 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.011 & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.012 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.013 \* & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

4.001 \* & John Smith & To ensure that both names are validated as described above \\ \hline

4.002 \* & No input - expecting output & \\ \hline

4.003 \* & No input - expecting output & \\ \hline

4.004 \* & john1, d4g6h3d & To ensure that the generation worked as intended and that they are recognised in the system \\ \hline

4.005 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.006 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.007 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.008 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.009 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.010 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.011 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.012 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.013 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

4.014 \* & 45; dgf & Test that the correct answer is confirmed using the algorithm, and anything else is recognised as incorrect \\ \hline

5.001 & Load up the system, complete each lesson and homework, and close it, to make sure there are no bugs & Only real way to test the system as a whole \\ \hline

5.002 \* & all known logins; random words and character combiniations & To ensure that every login works and that the system cannot be accessed by any other means than with a login \\ \hline

5.003 & One run through a homework - make sure all data is included & To ensure that the right amount of memory space is taken up for each record \\ \hline

5.004 & One run through each homework & Simply check the database to make sure what I think should be under each header is in fact under each header \\ \hline

5.005 \* & Access the database from different computers at the same time & To make sure each computer has access and therefore that the server is working \\ \hline

5.006 & Check that no invalid logins allow acess to the system and that there are no other ways in & To ensure that no-one unauthorised can view the database \\ \hline

\end{longtable}

\end{center}

\subsection{Changes to Test Data}

\textbf{\*\* signifies tests which have been added since the test plan was updated.}

\begin{center}

\begin{longtable}{|p{4cm}|p{4cm}|p{5cm}|} \hline

\textbf{Test Number} & \textbf{Test Data} & \textbf{Justification for choice of test data} \\ \hline

1.003 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.004 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.005 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.006 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.007 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.012 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.013 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.015 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.031 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.032 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.033 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.034 & 1 (or correct answer); a, 4, three, £; None & Firstly to test that the correct answer is recognised. Then make sure that the error exceptions work, and finally to test that the appropriate error message appears if nothing is entered before continuing \\ \hline

1.035 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.097 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.102 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.103 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.135 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.136 & 1, 2, 3, 4, 5, 6; a, b, 4, three, £, \$; None & Firstly to test that the correct answers are recognised. Then make sure that the error exceptions work, and finally to test that the appropriate error message appears if nothing is entered before continuing \\ \hline

1.137 & 12345; abcde; !"£\$\% & To make sure that the line edits are cleared regardless of what the input is \\ \hline

1.138 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.139 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.140 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.141 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

1.142 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.143 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.380 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.431 & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.432 & Sides Easy, 70\% & It is coded using a connection; therefore the only possible input \\ \hline

1.441 \*\* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

1.442 \*\* & Click the button & It does not take a value, it is coded using a connection; therefore the only possible input \\ \hline

2.003 & 45; words & To test that the correct answer works and then that any different data types do not crash it \\ \hline

2.016 & 1; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.017 & 2; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.018 & 3; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.019 & 4; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.020 & 5; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.021 & 6; abc; None & To test that the correct answer is recognised and other data types do not crash it \\ \hline

2.022 & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

2.023 & 50 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

2.024 & Click each button in turn & To test that the intended correct button works and the others don't \\ \hline

2.295 & Sides Easy & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

2.296 & 70\% & To test that the SQL finds the correct data - there will be no erroneous data due to limited options \\ \hline

3.009 & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

3.011 & Anything & For this specific test it doesn't matter what is stored as long as it works and is in the correct place - validation is concerned in series 2 \\ \hline

4.015 \*\* & 1, 2, 3, 4, 5, 6; a, b, 4, three, £, \$; None & Firstly to test that the correct answers are recognised. Then make sure that the error exceptions work, and finally to test that the appropriate error message appears if nothing is entered before continuing \\ \hline

4.016 \*\* & 20 (or correct answer); 30 & To test that the correct option is recognised as the correct answer and the incorrect options are not \\ \hline

4.017 \*\* & Click each button in turn & To test that the intended correct button works and the others don't \\ \hline

4.018 \*\* & 12345; abcde; !"£\$\% & To make sure that the line edits are cleared regardless of what the input is \\ \hline

5.001 & Load up the system, complete each lesson and homework, and close it, to make sure there are no bugs & Only real way to test the system as a whole \\ \hline

5.003 & One run through a homework - make sure all data is included & To ensure that the right amount of memory space is taken up for each record \\ \hline

5.004 & One run through each homework & Simply check the database to make sure what I think should be under each header is in fact under each header \\ \hline

5.006 & Check that no invalid logins allow acess to the system and that there are no other ways in & To ensure that no-one unauthorised can view the database \\ \hline

\end{longtable}

\end{center}

\section{Annotated Samples}

\subsection{Actual Results}

\begin{center}

\begin{longtable}{|p{2.5cm}|p{4cm}|p{4cm}|p{4.5cm}|p{3cm}|} \hline

\textbf{Test Number} & \textbf{Expected Result} & \textbf{Test Data} & \textbf{Actual Result} & \textbf{Screenshot Numbers} \\ \hline

1.003 & The lessons menu should be displayed & Click the lessons button & The lessons menu was displayed, as expected & Figures 3.1, 3.2 \\ \hline

1.004 & The homework menu should be displayed & Click the homework button & The homework menu was displayed, as expected & Figures 3.3, 3.4 \\ \hline

1.005 & The progress window should be displayed & Click the progress button & The progress window was displayed, as expected & Figures 3.5, 3.6 \\ \hline

1.006 & The program should close entirely & Click the exit program button & The entire program closed, as expected & Figures 3.7, 3.8 \\ \hline

1.007 & The Trigonometry 1 lesson menu should be displayed & Click the Trigonometry 1 button & The Trigonometry 1 lesson menu was displayed, as expected & Figures 3.9, 3.10 \\ \hline

1.012 & The lesson menu should be closed and the home screen displayed & Click the return button & The lesson menu closed and the home screen was displayed, as expected & Figures 3.11, 3.12 \\ \hline

1.013 & The SOHCAHTOA first lesson screen should be displayed & Click the SOHCAHTOA button & The SOHCAHTOA first lesson screen was displayed, as expected & Figures 3.13, 3.14 \\ \hline

1.015 & The Trigonometry 1 lesson menu should be closed and the lesson menu displayed & Click the return button & The Trigonometry 1 lesson menu closed and the lesson menu was displayed, as expected & Figures 3.15, 3.16 \\ \hline

1.031 & The SOHCAHTOA first lesson screen should be closed and the Trigonometry 1 lesson menu should be displayed & Click the return button & The SOHCAHTOA first lesson screen was closed and the Trigonometry 1 lesson menu was displayed, as expected & Figures 3.17, 3.18 \\ \hline

1.032 & The SOHCAHTOA second lesson screen should replace the first SOHCAHTOA lesson screen in display & Click the next button & The second SOHCAHTOA lesson screen replaced the first SOHCAHTOA lesson screen, as expected & Figures 3.19, 3.20 \\ \hline

1.033 & The SOHCAHTOA first lesson screen should replace the second SOHCAHTOA lesson screen in display & Click the previous button & The first SOHCAHTOA lesson screen replaced the second SOHCAHTOA lesson screen, as expected & Figures 3.21, 3.22 \\ \hline

1.034 & The input typed in the line edit should be checked and the user told whether they were correct or not & Click the button & The correct answer was registered as correct and the wrong answer was registered as incorrect, as expected & Figures 3.23, 3.24 \\ \hline

1.035 & The stack window with the SOHCAHTOA lesson should close and the Trigonometry 1 lesson menu be displayed & Click the finish button & The SOHCAHTOA lesson stack was closed and the Trigonometry 1 lesson menu was displayed, as expected & Figures 3.25, 3.26 \\ \hline

1.097 & The Trigonometry 1 homework menu should be displayed & Click the trigonometry 1 button & The Trigonometry 1 homework menu was displayed, as expected & Figures 3.27, 3.28 \\ \hline

1.102 & The homework menu should close and the home screen should be displayed & Click the return button & The homework menu closed and the home screen was displayed, as expected & Figures 3.29, 3.30 \\ \hline

1.103 & The first sides easy homework screen should be displayed & Click the sides easy button & The first sides easy homework screen was displayed, as expected & Figures 3.31, 3.32 \\ \hline

1.135 & The sides easy homework stack should close and the trigonometry 1 homework menu displayed & Click the return button & The sides easy homework stack was closed and the trigonometry 1 homework menu was displayed, as expected & Figures 3.33, 3.34 \\ \hline

1.136 & The 6 line edits should be checked and the user told how many were right, and marks given & Click the check answers button & The line edits were checked, correct answers were recognised and incorrect answers were rejected, as expected & Figures 3.35, 3.36 \\ \hline

1.137 & The 6 line edits contents should all be reset to empty & Click the reset button & All 6 line edits were cleared, as expected & Figures 3.37, 3.38 \\ \hline

1.138 & The second sides easy homework screen should replace the first sides easy homework screen in the stack, and the score from the first question should be stored in the database & Click the next button & The second sides easy homework screen replaced the first sides easy homework screen in the stack, and the correct score count was stored in the database, as expected & Figures 3.39, 3.40 \\ \hline

1.139 & The input in the combo box should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click the mark it button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.41, 3.42 \\ \hline

1.140 & The input in the combo box should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click the mark it button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.43, 3.44 \\ \hline

1.141 & The correct button should be checked, the user informed if they are correct or not, and marks be added or attempts removed & Click each possible button & The correct answer was recognised and marks added, and the incorrect answer recognised and attempts removed afterwards, as expected & Figures 3.45, 3.46, 3.47 \\ \hline

1.142 & The first sides easy homework screen should replace the second sides easy homework screen in the stack & Click the previous button & The first sides easy homework screen replaced the second sides easy homework screen in the stack, as expected & Figures 3.48, 3.49 \\ \hline

1.143 & the sides easy homework stack should be closed and the home screen should be displayed; The scores from the questions should be stored in the database & Click the finish button & The sides easy homework stack was closed, the home screen was displayed, and the scores were saved to the database, as expected & Figures 3.50, 3.51 \\ \hline

1.380 & The progress screen should be closed and the home screen displayed & Click the return button & The progress screen was closed and the home screen was displayed, as expected & Figures 3.52, 3.53 \\ \hline

1.431 & The report screen should be closed and the progress screen displayed & Click the return button & The report screen was closed and the home screen was displayed, as expected & Figures 3.54, 3.55\\ \hline

1.432 & The relevant information should be fetched from the database and displayed in the same window & Click the query button & Relevant information was found and displayed in the database in hte same window, as expected & Figures 3.56, 3.57 \\ \hline

1.441 & The welcome screen should close and the home screen should be displayed & Click the continue button & The welcome screen was closed and the home screen was displayed, as expected & Figures 3.58, 3.59 \\ \hline

1.442 & The report screen should open and the progress screen hidden & Click the report button & The progress screen closed and the report screen was displayed, as expected & Figures 3.60, 3.61 \\ \hline

2.003 & If the input is correct, the word correct should be displayed, otherwise incorrect shoud be displayed & 5 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.62, 3.63 \\ \hline

2.016 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 1 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.017 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 2 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.018 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 3 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.019 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 4 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.020 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 5 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.021 & If the input is correct, the word correct should be displayed, otherwise incorrect should be displayed & 6 (or right answer); abc; None & Correct was displayed, incorrect was displayed and an error message for no answer was displayed respectively, as expected & Figures 3.64, 3.65 \\ \hline

2.022 & If the contents of the combo box is correct, correct should appear in the button next to it, otherwise an attempt will be removed & 20; 10 & For the correct answer, correct was printed, an incorrect for the incorrect answer, as expected & Figures 3.66, 3.67 \\ \hline

2.023 & If the contents of the combo box is correct, correct should appear in the button next to it, otherwise an attempt will be removed & 40; 15 & For the correct answer, correct was printed, an incorrect for the incorrect answer, as expected & Figures 3.68, 3.69 \\ \hline

2.024 & If the right button is clicked, display correct, otherwise display incorrect & Each button in order & When the right button was clicked, a mark was added, and when the wrong buttons were clicked attempts were removed, as expected & Figures 3.70, 3.71 \\ \hline

2.295 & The information relevant to the input should be fetched from the database and displayed & Sides Easy; Pythagoras Theorem Hard & The relevant task name was fetched and displayed, as expected & Figure 3.72 \\ \hline

2.296 & The information relevant to the input should be fetched from the database and displayed & 70\%; 80\% & The relevant scores were fetched and displayed, as expected & Figure 3.73 \\ \hline

3.009 & The task names should be stored under the 'Task Names' header in the database & Complete a task & The task name was stored under 'Task Names', as expected & Figure 3.74 \\ \hline

3.011 & The IndividualPercentScores should be stored under the 'QOne', 'QTwo', 'QThree', and 'QFour' headers respective to the question number in the database & Complete a task & The scores were under their respective headings, as expected & Figure 3.75 \\ \hline

4.015 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.76, 3.77, 3.78 \\ \hline

4.016 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.79, 3.80, 3.81 \\ \hline

4.017 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The correct outputs were given, as expected & Figures 3.82, 3.83, 3.84 \\ \hline

4.018 & The correct output should be displayed for a correct or incorrect or absent input & [Correct answer]; [Incorrect answer]; None & The appropriate outputs were given for every possile input tested & 3.85, 3.86 \\ \hline

5.001 & The client should be satisfied with the overall system & Show the client each aspect of the system & N/A & \\ \hline

5.003 & The task names and scores should be being saved to an sqlite database, 5 columns, as many rows as there are tasks & Complete a task & There are the right number of headings in the table all with the correct information being displayed in them, as expected & Figure 3.87 \\ \hline

5.004 & Only the task names and scores should be stored in the database & Complete a task & They are the only pieces of information being stored, as expected & Figure 3.88 \\ \hline

5.006 & No illegitimate information or personal information should be being stored & Complete a task (only source of information for the database) & No illegitimate information or personal information is being stored, only task names and scores, so the DPA cannot be breached anyway & Figure 3.89 \\ \hline

\end{longtable}

\end{center}

\end{landscape}

\subsection{Evidence}

\textbf{Test 1.003}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textit{Lessons button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_3}

\end{figure}

\textbf{Test 1.004}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textit{Homework button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_7}

\end{figure}

\textbf{Test 1.005}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textit{Progress button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_11}

\end{figure}

\textbf{Test 1.006}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textit{Exit program button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Program is closed}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_13}

\end{figure}

\textbf{Test 1.007}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_3}

\end{figure}

\textit{Trigonometry 1 button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

\textbf{Test 1.012}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_3}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textbf{Test 1.013}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

\textit{Sides button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_5}

\end{figure}

\textbf{Test 1.015}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_3}

\end{figure}

\textbf{Test 1.031}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_5}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

\textbf{Test 1.032}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_5}

\end{figure}

\textit{Next button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_6}

\end{figure}

\textbf{Test 1.033}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_6}

\end{figure}

\textit{Previous button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_5}

\end{figure}

\textbf{Test 1.034}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_14}

\end{figure}

\textit{Check answer button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_15}

\end{figure}

\textbf{Test 1.035}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_6}

\end{figure}

\textit{Finish button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_4}

\end{figure}

\textbf{Test 1.097}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_7}

\end{figure}

\textit{Trigonometry 1 button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_8}

\end{figure}

\textbf{Test 1.102}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_7}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textbf{Test 1.103}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_8}

\end{figure}

\textit{Sides easy button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

\textbf{Test 1.135}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_8}

\end{figure}

\textbf{Test 1.136}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_16}

\end{figure}

\textit{Check answers button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_17}

\end{figure}

\textbf{Test 1.137}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_16}

\end{figure}

\textit{Reset answers button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

\textbf{Test 1.138}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

\textit{Next button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_10}

\end{figure}

\textbf{Test 1.139}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_18}

\end{figure}

\textit{Mark it button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_19}

\end{figure}

\textbf{Test 1.140}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_20}

\end{figure}

\textit{Mark it button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_21}

\end{figure}

\textbf{Test 1.141}

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_22}

\end{figure}

\textit{Wrong button is clicked - incorrect message}

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_23}

\end{figure}

\textit{Right button is clicked - correct message}

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_24}

\end{figure}

\textbf{Test 1.142}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_10}

\end{figure}

\textit{Previous button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_9}

\end{figure}

\textbf{Test 1.143}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_10}

\end{figure}

\textit{Finish button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_8}

\end{figure}

\textbf{Test 1.380}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_11}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textbf{Test 1.431}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_12}

\end{figure}

\textit{Return button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_11}

\end{figure}

\textbf{Test 1.432}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_25}

\end{figure}

\textit{Query button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_26}

\end{figure}

\textbf{Test 1.441}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_1}

\end{figure}

\textit{Continue button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_2}

\end{figure}

\textbf{Test 1.442}

\begin{figure}[H]

\label{fig: First Screen}\caption{First screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_11}

\end{figure}

\textit{Report button is clicked: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_12}

\end{figure}

\textbf{Test 2.003}

\textit{Input is incorrect: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_27}

\end{figure}

\textit{Input is correct: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Second Correct input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_28}

\end{figure}

\textbf{Tests 2.016, 2.017, 2.018, 2.019, 2.020, 2.021}

\textit{Input is incorrect: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect inputs}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_29}

\end{figure}

\textit{Input is correct: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct inputs}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_30}

\end{figure}

\textbf{Test 2.022}

\textit{Input is incorrect: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_31}

\end{figure}

\textit{Input is correct: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_32}

\end{figure}

\textbf{Test 2.023}

\textit{Input is incorrect: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_33}

\end{figure}

\textit{Input is correct: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_34}

\end{figure}

\textbf{Test 2.024}

\textit{Input is incorrect: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_23}

\end{figure}

\textit{Input is correct: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct input}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_24}

\end{figure}

\textbf{Test 2.295}

\textit{Task is selected to query: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Task queried}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_35}

\end{figure}

\textbf{Test 2.296}

\textit{Score is selected to query: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Score queried}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_36}

\end{figure}

\textbf{Test 3.009}

\textit{Task name is stored under the task name header: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Task name stored}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_37}

\end{figure}

\textbf{Test 3.011}

\textit{The scores are stored under the score headers: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Scores stored}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_38}

\end{figure}

\textbf{Test 4.015}

\textit{Incorrect answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_39}

\end{figure}

\textit{Correct answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_40}

\end{figure}

\textit{No answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Absent answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_41}

\end{figure}

\textbf{Test 4.016}

\textit{Incorrect answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_42}

\end{figure}

\textit{Correct answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_43}

\end{figure}

\textit{No answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Absent answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_44}

\end{figure}

\textbf{Test 4.017}

\textit{Incorrect answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Incorrect answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_45}

\end{figure}

\textit{Correct answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Correct answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_46}

\end{figure}

\textit{No answer output: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Absent answer}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_44}

\end{figure}

\textbf{Test 4.018}

\textit{Before resetting: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Before reset}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_48}

\end{figure}

\textit{After resetting: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{After reset}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_49}

\end{figure}

\textbf{Test 5.001}

\textbf{Test 5.003}

\textit{The information is being stored in the right place here: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Database}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_50}

\end{figure}

\textbf{Test 5.004}

\textit{The right amount of information is being stored here: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Database}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_50}

\end{figure}

\textbf{Test 5.006}

\textit{No illegitimate information is stored here: }

\begin{figure}[H]

\label{fig: Second Screen}\caption{Database}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Testing/screen\_50}

\end{figure}

\section{Evaluation}

\subsection{Approach to Testing}

I decided to use a variation of testing methods in order to ensure that every aspect of this system works adequately and robustly, and meets as many client specifications as possible. For example, top-down testing was used to check that every button in the system works as intended, bottum up testing was used to test each input opportunity as they were created, black box testing was used to make sure all information was stored in the database correctly, white box testing was used to test each of the marking algorithm in the system worked efficiently, and finally acceptance testing was used to make sure that the client was satisfied with the system. For test series 1, not every button test was documented because there are too many of them and most of them share parent code, so testing at least one of each child button or unique button was enough to be sure that the same code was working everywhere.

\subsection{Problems Encountered}

\begin{itemize}

\item Initially, on the report widget, I tried to have a separate window pop up to output the results of the user's query, however I struggled to get some of the data variables to pass between files, so I had to have the QTableWidget on the same screen as the query menu. However, this turned out to be just as effective as the first attempted method.

\item For the generic question 4 on the homeworks, I wanted to put in drag and drop functionality to provide a variation of input types. However, there was a very unusual error where enabling a built in drag function would break the entire program, and it was beyond my abilities to either solve this problem, or find another way to implement the drag and drop functionality. So I replaced the drag and drop idea with multiple choice buttons, which admittedly is not as technically 'good', but there is still a range of input types, and this replacement works efficiently.

\item There was a significant problem with saving results to the database; I could not generate a total score variable because, again, I could not find how to pass the required variables across files, as the first and second screen's code are separate from each other. As a result of this, I could not store a Rating variable either, so I resorted to just including the task name along with each questions score.

\item The biggest problem was finding a way to create separate accounts, all accessible from different computers in a LAN, and having an administrator account capable of setting homeworks. It occured to me that this was far too much to code in the time I had, and that I had bitten off much more than I could chew, so I reformed the program down to a single user, self-assessing program, which still records data, but that data stays on one machine, used by one person. More than one person can use it; it can be installed on multiple offline machines, of course. The client will not be able to keep track of progress, they will have to rely on users to be honest or provide evidence.

\item The database kept having to be over-written, and would only hold one record, otherwise it would crash, if you tried to record results for the same task twice. I fixed this by implementing some SQL code which would fetch all information currently in the database and check to see if any records with the same TaskName value existed already. If so, another SQL statement would check the values of the current scores and the new scores and only overwrite them if the new ones were higher. Otherwise it would record the new task entirely. Now it works; there are no circumstances or results which would cause the program to crash upon attempting to record them. Either a new record is saved or an old one is overwritten.

\end{itemize}

\subsection{Strengths of Testing}

The main strength of my testing was testing the answer marking algorithms; these algorithms included awarding marks, removing remaining attempts and disabling buttons, and after testing every possible input, each of these algorithms works robustly and exactly how they were intended. Any wrong data type inputs are dealt with using error messages, and the buttons would be disabled after losing all attempts or getting the correct answer to prevent the user being able to attempt the same question more than once, overwrite the content of the buttons and crash the system upon trying to save information to the database. This also leads to the next strength, the bottum up input testing, as this also shows that no input can cause the program to crash at any point.

\subsection{Weaknesses of Testing}

The main weakness of my testing was simply the massive number of buttons which had to be tested. All of them were tested, and all of them work, even after up to three different methods of coding them, and most of them have a parent class, so the same code works in each place. This was a weakness because it took time, and not every test could be documented, as they were essentially the same. The other weakness of testing is the acceptance testing; after cutting out the entire administrator aspect of the program as well as LAN capabilites, it is unlikely that the client will be fully satisfied. With luck, they accept the system anyway, as it does meet some other main specifications.

\subsection{Reliability of Application}

My system is more or less reliable; all data stored is accurate and correct, and is stored in the right quantity. There are no validation errors when checking answers, as they are hard-coded, and testing has proven the algorithms reliable under every circumstance. The information in the lessons is correct, and all of the buttons work exactly as intended. Only one potentially system breaking issue was found, where the database could not save the same task twice, but that problem has been solved. Finally, the data in the database can only be manipulated by completing a task, there is no way to manually enter the database and change the data.

\subsection{Robustness of Application}

My program is very robust; there are error messages for every possible error, including wrong data types, or not being able to progress. The only error which pops up pretty much every time a new screen opens is the word \\_raise() not being recognised, however that makes absolutely no difference to anything and can be dealt with simply by putting in an error exception in every place where it could occur. There is no point in the program where an infinite loop can be entered.

\chapter{Design}

\section{Overall System Design}

\subsection{Short description of the main parts of the system}

\begin{itemize}

\item \textbf{GCSE Trigonometry and Pythagoras Education System:}

\begin{itemize}

\item General User Interface

\item Getting Username and Password

\item Entering Users Into the System

\item Navigating User's Personal Account User Interface

\item Setting Homework From Administrator

\item Running Individual Lessons

\item Running Individual Homeworks

\item Accepting User Inputs

\item Outputting Error Exception Messages

\item Recording Homework Results On Database

\item Viewing Results in Database

\item Querying the Database

\end{itemize}

\end{itemize}

\begin{itemize}

\item General User Interface

\begin{itemize}

\item This is the general layout of the interface across the system; the same layout pattern, same colour scheme, same positioning design.

\item This will provide the user with the ability to navigate the parts of the system that they have access to, and enable them to select tasks, submit homeworks, and access the database to view their personal and homework records.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Getting Username and Password

\begin{itemize}

\item This will come in the very first window every time the user or administrator wants to access the program; it will ask them to log in so that they can view their own personal account, with all of their homework progress and personal records.

\item It will mainly consist of two text boxes in the centre of the screen, one prompting a username and the other a password underneath.

\item If invalid information is input it will display an error message asking the user to retry. There will be no limit to the number of attempts a user can have; it won't lock them out of their computer if they keep getting it wrong.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Entering Users Into the System

\begin{itemize}

\item The administrator will be able to click a button from their home screen which will like them to an administration window from which they can add names and classes

\item They will input a name, then click submit to finish, or submit and add another to continue adding names, or select the add class button which will take them to a similar window to add a class

\end{itemize}

\end{itemize}

\begin{itemize}

\item Navigating User's Personal Account User Interface

\begin{itemize}

\item This is essentially the functionality between each window. The user will have a range of buttons to press which will then open the relevant window, and then will have the option to return, making each part of the system accessible.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Setting Homework From Administrator

\begin{itemize}

\item The function which makes the user aware of which homework has been set; the administrator will set it, the system will acknowledge it, and then notify the required users to do it.

\item The reverse will work similarly; the user will complete the homework and the administrator will be notified. They will also be notified if the user hasn't completed it before the due date.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Running Individual Lessons

\begin{itemize}

\item The module that runs for each lesson. Each lesson will have its own set of windows, containing the inputs, outputs and graphics to accurately represent what is being taught e.g Pythagoras Theorem.

\item These will only be opened, navigated and closed as nothing from these is recorded in the database; they are effectively read only.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Running Individual Homeworks

\begin{itemize}

\item The module that runs for each homework, saved so that each different homework task is always the same, in order to avoid confusion.

\item When the homework is run, inputs are accepted, outputs are given, and at the end the appropriate information is recorded in the database.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Accepting User Inputs

\begin{itemize}

\item Each input type in the system, be it the general user interface, a lesson or a homework, will be accepted.

\item Once accepted, it will be validated, and if invalid, the appropriate error message will be displayed.

\item Different types of inputs are:

\begin{itemize}

\item Text Boxes

\item Drag and Drop Images

\item Image Modification

\item Buttons

\end{itemize}

\item If valid, the system will either allow them access to their account, or add a mark to their homework task score.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Outputting Error Exception Messages

\begin{itemize}

\item If a user's input is invalid, the appropriate error message will be displayed.

\item Each error message will appear for 5 seconds and then disappear automatically.

\item These are the variations of error messages and their causes:

\begin{itemize}

\item Invalid Username

\item Invalid Password

\item Invalid Data Type

\item Wrong Answer 1{$^s$}{$^t$}

\item Wrong Answer 2{$^n$}{$^d$}

\item Unauthorised Access

\end{itemize}

\end{itemize}

\end{itemize}

\begin{itemize}

\item Recording Homework Results On Database

\begin{itemize}

\item Every time a user completes a homework, the results will be recorded in the database, including the \textbf{OverallPercentScore}, \textbf{IndividualPercentScore(s)}, and \textbf{Rating}.

\item The user will be able to view their results in their personal database section, and the administrator will be able to view these results in the entire class database for that task.

\end{itemize}

\end{itemize}

\begin{itemize}

\item Viewing Results in Database

\begin{itemize}

\item A student will be able to click a button on the home screen to take them to the database view, where they will clearly see all of their personal results

\item A teacher will be able to click a button on the home screen to take them to the database view, where they will clearly see all of the students results in any of their own classes, and they will be able to scroll through each student in alphabetical order on a scroll bar on the side to find individual's results

\item An administrator (teacher) can amend a database value if it is necessary, although it is unlikely that it will be

\item A return button will take them straight back to the home page to continue using the system

\end{itemize}

\end{itemize}

\begin{itemize}

\item Querying the Database

\begin{itemize}

\item An administrator or student eill be able to query the databases they can access by using the report window, where admins can select a class, student, task and/or score and be given all database entries with the relevant information. Students will only be able to query tasks and scores to comply with their access authorisation

\item The system will be queried using SQL statements provided by combining the user's selections from the report window

\end{itemize}

\end{itemize}

\subsection{System flowcharts showing an overview of the complete system}

\textbf{This flowchart represents the system accessible for a student/user: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/user\_flow\_1}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/user\_flow\_2}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/user\_flow\_3}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/user\_flow\_4}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/user\_flow\_5}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\textbf{This flowchart represents the system accessible for a teacher/administrator: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/admin\_flow\_1}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/admin\_flow\_2}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/admin\_flow\_3}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/admin\_flow\_4}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/admin\_flow\_5}

\label{fig:print\_function\_result}\caption{}

\end{figure}

\section{User Interface Designs}

%C:/Users/Jordan/git/COMP4Coursework2/Design

%C:/Users/Jordan/git/COMP4Coursework2/Design/

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Login Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_9}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage2}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_10}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Student Home Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_11}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Lesson Topic Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_12}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Lesson Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_13}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{First Lesson Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_14}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Second Lesson Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_15}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Topic Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_16}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_17}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{First Homework Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_18}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Second Homework Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_19}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Individual User Progress Database}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_20}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administrator Account Home}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_21}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Setting Topic Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_22}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Set Screen}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_23}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Results Menu}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_24}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Class Results Database}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_25}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage3}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_26}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_27}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_28}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_29}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administrator Query Window}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_030}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Student Query Window}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_031}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administration Names Window}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_032}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administration Class Window}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_033}

\end{figure}

\section{Hardware Specification}

This system will need to run on a standard desktop computer, as generally found in schools, using the Windows 7 operating system. It will need to have a screen resolution of 1920 x 1080, a 32 bit true colour scheme, a 16:9 aspect ratio, and an Intel HD Graphics 2500 card or better, running at 60p Hz. Windows 7 is required as it is still currently the operating system used by more or less every education facility, and the client will be working at such a facility. The 32 bit true colour scheme is usually the default set by the administrators, and they need a 16:9 aspect ratio, as I need to be able to make the windows fit properly on every machine at the facility, and the screens will not be resizable. The 1920 x 1080 resolution is desirable but not entirely necessary; it is achievable and will maintain the quality of the appearance of the program. The Intel HD Graphics 2500 is standard and should be able to run the system. A keyboard will be required for inputting text answers into boxes throughout the lessons, homework, and for logging in; any standard keyboard will be usable for this. Similarly, any standard mouse will be needed for navigating the GUI, clicking buttons, scrolling, using frop down boxes and dragging and dropping. A standard display will be used for the output of the system; just visual, no audio. The data for each user will be stored on a local server so that every user or administrator can view their own/all data from any of the machines at the facility. All of these requirements are able to be met by the client, who has it all available.

\section{Program Structure}

\subsection{Top-down design structure charts}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_30}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_31}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_32}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/figure\_2\_33}

\end{figure}

\subsection{Algorithms in pseudo-code for each data transformation process}

\textbf{Validates the username and password when the user tries to log in:}

\begin{algorithm}[H]

\caption{Takes the logins from the database and adds them to a list from which they can be validated.}

\begin{algorithmic}[1]

\SET{$with \ open("logins.txt", mode = "r") as \ logins$}

\For{$name$}{$logins$}

\SET{$list\\_.append(name)$}

\EndFor

\RECEIVE{"Please enter your username"}

\SET{$return \ username$}

\RECEIVE{"Please enter your password"}

\SET{$return \ password$}

\SET{$count$}{$0$}

\SET{$found$}{$False$}

\While{$found = False$}{$and \ count < len(list\\_)$}

\If{$list\\_[count] = str(username) \ and \ list\\_[count + 2] = str(password)$}

\SEND("Accepted")

\SET{$found$}{$True$}

\SET{$return \ found$}

\Else{}

\SEND{"Not accepted"}

\SET{$call \ Validation$}

\SET{$count \ += \ 1$}

\EndIf

\EndWhile

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Checks if the user's solution for the {$a^2 + b^2 = c^2$} is correct.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$side\\_a$}{$x$}

\SET{$side\\_b$}{$x$}

\SET{$side\\_c$}{$\sqrt{side\\_a^2 + side\\_b^2}$}

\SEND{$"Here \ is \ a \ right \ angled \ triangle. \ The \ length \ of \ side \ a \ is \ x$}

\SET{$centimetres, \ and \ the \ length \ of \ side \ b \ is \ x \ centimetres.$}

\SET{$Please \ calculate \ the \ length \ of \ side \ c"$}

\RECEIVE{$length$}{$"Please \ input \ the \ length \ of \ side \ c: "$}

\If {$length = side\\_c$}

\SEND{"+ 1 \ mark"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Alternative algorithm:}

\begin{algorithm}[H]

\caption{Same question and solution with a differently arranged formula.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$side\\_a$}{$x$}

\SET{$side\\_b$}{$x$}

\SET{$side\\_c^2$}{$side\\_a^2 + side\\_b^2$}

\SET{$side\\_c$}{$\sqrt{side\\_c^2}$}

\SEND{$"Here \ is \ a \ right \ angled \ triangle. \ The \ length \ of \ side \ a \ is \ x$}

\SET{$centimetres, \ and \ the \ length \ of \ side \ b \ is \ x \ centimetres.$}

\SET{$Please \ calculate \ the \ length \ of \ side \ c"$}

\RECEIVE{$length$}{$"Please \ input \ the \ length \ of \ side \ c: "$}

\If{$length = side\\_c$}

\SEND{"+ x \ marks"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{3D Pythagoras algorithm:}

\begin{algorithm}[H]

\caption{Similar algorithm, but continues to check the user's solution for a 3D pythagoras problem.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$left\\_side\\_a$}{$x$}

\SET{$middle\\_side\\_a$}{$x$}

\SET{$right\\_side\\_a$}{$x$}

\SET{$inside\\_side\\_a$}{$\sqrt{left\\_side\\_a^2 + middle\\_side\\_a^2}$}

\SET{$inside\\_side\\_b$}{$\sqrt{right\\_side\\_a^2 + inside\\_side\\_a^2}$}

\SEND{$"A \ magician \ stores \ his \ wand \ in \ a \ box.$}

\SET{$The \ box \ is \ xcm \ by \ xcm \ by \ xcm.$}

\SET{$The \ wand \ only \ just \ fits \ in \ wedged \ against \ opposite \ corners."$}

\RECEIVE{$length$}{$"How \ long \ is \ the \ wand?"$}

\If{$length = inside\\_side\\_b$}

\SEND{"+ \ x \ marks"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Trigonometry Algorithms:}

\begin{algorithm}[H]

\caption{Sine rule.}

\begin{algorithmic}[1]

\SET{$sinA$}{$\frac{opposite}{hypotenuse}$}{$\frac{a}{h}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Cosine rule.}

\begin{algorithmic}[1]

\SET{$cosA$}{$\frac{adjacent}{hypotenuse}$}{$\frac{b}{h}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Tan rule.}

\begin{algorithmic}[1]

\SET{$tanA$}{$\frac{opposite}{adjacent}$}{$\frac{a}{b}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The sine formula in use.}

\begin{algorithmic}[1]

\SET{$\frac{A}{sinA}$}{$\frac{B}{sinB}$}

\If{$\frac{A}{sinA} = \frac{B}{sinB}$}

\SEND{"Your solution is correct"}

\Else{}

\SEND{"Your solution is not correct"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The cosine formula in use.}

\begin{algorithmic}[1]

\SET{$a^2$}{$b^2 + c^2 \ - \ 2bc \ cosA$}

\RECEIVE{$side\\_a$}{$"Please \ input \ the \ length \ of \ side \ a: "$}

\If{$side\\_a = {b^2 + c^2 - 2bc \ cosA}$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The formula for finding angles in scalene triangles using the cosine rule.}

\begin{algorithmic}[1]

\SET{$cosA \ b^2 + c^2 - \frac{a^2}{2bc}$}

\SET{$C$}{$inv \ cos\frac{adjacent}{hypotenuse}$}

\RECEIVE{$angle\\_c$}{$"Please \ input \ the \ size \ of \ angle \ C:"$}

\If{$angle\\_c = {inv \ cos\frac{adjacent}{hypotenuse}}$}

\SEND{"Correct"}

\Else{}

\SEND{"Correct"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Formula for finding the area of a scalene triangle using the sine rule.}

\begin{algorithmic}[1]

\SET{$area$}{$\frac{1}{2} \ ab \ sinC$}

\RECEIVE{$area\\_1$}{"Please \ input \ the \ area \ of \ this \ scalene \ triangle:"}

\If{$area\\_1 = \frac{1}{2} \ ab \ sinC$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Formula for finding an angle using the tan rule.}

\begin{algorithmic}[1]

\SET{$tanA$}{$\frac{10}{15}$}

\SET{$\frac{10}{15}$}{$ \ $}{$0.67$}

\SET{$tan^-1(0.67)$}{$33.82^o$}

\SET{$tanA$}{$33.82^o$}

\RECEIVE{$tan\\_a$}{"Please \ input \ the \ size \ of \ the \ angle \ using \ the \ tan \ rule:"}

\If{$tan\\_a = {33.82^o}$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Algorithm for resetting the answers on the homework: }

\begin{algorithm}[H]

\caption{}

\begin{algorithmic}[1]

\For{$question$}{$screen$}

\SET{$question\\_text\\_box.value = 0$}

\EndFor

\end{algorithmic}

\end{algorithm}

\textbf{Algorithm for saving the results to the database: }

\begin{algorithm}[H]

\caption{}

\begin{algorithmic}[1]

\SET{$score\\_percentage\\_1$}{$75$}

\SET{$score\\_percentage\\_2$}{$50$}

\If{$button\\_pressed = finish\\_button$}

\For{$question$}{$screen\\_1$}

\SET{$database.topic.percentage\\_1.append$}{$score\\_percentage\\_1$}

\EndFor

\For{$question$}{$screen\\_2$}

\SET{$database.topic.percentage\\_2.append$}{$score\\_percentage\\_2$}

\EndFor

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Algorithm for setting the homeworks: }

\begin{algorithm}[H]

\caption{}

\begin{algorithmic}[1]

\If{$button\\_pressed = topic$}

\SET{$set\\_topic$}{$topic$}

\EndIf

\For{$student$}{$class$}

\SET{$topic.activate$}

\SEND{$You \ must \ now \ complete \ \{0\}.format\{topic\}$}

\EndFor

\end{algorithmic}

\end{algorithm}

\textbf{Administration algorithm:}

\begin{algorithm}[H]

\caption{}

\begin{algorithmic}[1]

\SET{$Boolean$} {$True$}

\While{$not$}{$False$}

\SEND{$"Please \ input \ the \ first \ name \ of \ the \ student:"$}

\RECEIVE{$FirstName$}

\SEND{$"Please \ input \ the \ last \ name \ of \ the \ student:"$}

\RECEIVE{$LastName$}

\SEND{$"Would \ you \ like \ to \ add \ another \ student?"$}

\RECEIVE{$Answer$}

\If{$Answer == Yes$}

\SET{$Boolean$}{$True$}

\Else

\SET{$Boolean$}{$False$}

\EndIf

\EndWhile

\end{algorithmic}

\end{algorithm}

\subsection{Object Diagrams}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/objectrelationships.png}

\caption{The relationships between each of the objects in the proposed system} \label{fig:print\_function\_result}

\end{figure}

\subsection{Class Definitions}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Course \\ \hline

Title \\

Subject \\ \hline

AddTitle \\

EditTitle \\

AddSubject \\

EditSubject \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Teacher \\ \hline

Surname \\

Title \\

Subject \\ \hline

AddSurname \\

EditSurname \\

AddTitle \\

EditTitle \\

AddSubject \\

EditSubject \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Student \\ \hline

FirstName \\

Surname \\

UserID \\

Password \\ \hline

AddFirstName \\

EditFirstName \\

AddSurname \\

EditSurname \\

AddUserID \\

EditUserID \\

AddPassword \\

EditPassword \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Section \\ \hline

TrigonometryLesson \\

TrigonometryHomework \\ \hline

PresentTrigonometryLesson \\

SetTrigonometryHomework \\

MarkTrigonometryHomework \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Section \\ \hline

PythagorasLesson \\

PythagorasHomework \\ \hline

PresentPythagorasLesson \\

SetPythagorasHomework \\

MarkPythagorasHomework \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Lesson \\ \hline

Examples \\

Questions \\ \hline

DisplayExamples \\

GiveExampleAnswers \\

GiveQuestionAnswers \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Homework \\ \hline

Questions \\

Answers \\ \hline

SetQuestions \\

CheckAnswers \\

DisplayCorrectAnswers \\

OutputCorrectMessage \\

OutputDataTypeErrorMessage \\

SubmitScore \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Lesson \\ \hline

Examples \\

Questions \\ \hline

DisplayExamples \\

GiveExampleAnswers \\

GiveQuestionAnswers \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Homework \\ \hline

Questions \\

Answers \\ \hline

SetQuestions \\

CheckAnswers \\

DisplayCorrectAnswers \\

OutputCorrectMessage \\

OutputDataTypeErrorMessage \\

SubmitScore \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

QuestionType \\ \hline

QuestionType \\

QuestionInput \\

QuestionOutput \\ \hline

InputSolution \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

TextQuestion \\ \hline

QuestionInputText \\

QuestionOutput \\ \hline

InputSolutionText \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

DragAndDropQuestion \\ \hline

QuestionDragImage \\

QuestionOutput \\ \hline

InputDragImage \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

ImageQuestion \\ \hline

QuestionSelectButton \\

QuestionOutput \\ \hline

InputSolutionButton \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\section{Prototyping}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Login Screen: The user inputs their username and password and selects log in, then either the student home scree, admin home screen or an error message will appear, depending on the input.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_1}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage2: If the user's input username or password is incorrect, this message will appear.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_2}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Student Home Screen: If the user's username is recognised as a student's name, the student version of the home screen will appear.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_3}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Lesson Topic MenC:/Users/Jordan Appears if the user selects the lessons button - displays the topics to choose from.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_4}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Lesson MenC:/Users/Jordan Will appear for any lesson topic selected, displaying the buttons for the topic specific lessons to choose from. Will look the same for each topic except for the button and window names.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_5}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{First Lesson Screen: Will appear with the lessons depending on which button was selected; all lessons will use a generic design plan.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_6}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Second Lesson Screen: Appears once next has been selected from the previous screen, continuing a generic design plan and displaying the next examples.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_7}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Topic MenC:/Users/Jordan Appears if the user selects the homework button - displays the topics to choose from.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_8}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework MenC:/Users/Jordan Will appear for any homework topic selected, displaying the buttons for the topic specific homeworks to choose from. Will look the same for each topic except for the button and window names.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_9}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{First Homework Screen: Will appear with the homework depending on which button was selected; all homeworks will use a generic design plan.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_10}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Second Homework Screen: Appears once next has been selected from the previous screen, continuing a generic design plan and displaying the next questions.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_11}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Individual User Progress Database: Appears if the user selects the progress button - displays their individual database in a table in the window.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_12}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administrator Account Home: If the user's username is recognised as an teacher's name, the administrator version of the home screen will appear.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_13}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Setting Topic MenC:/Users/Jordan Appears if the user selects the homework button - displays the topics to choose from to set as homework.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_14}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Homework Set Screen: Appears once a topic has been selected - once the set button is pressed this homework will be available on each student's homework to-do list.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_15}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Results MenC:/Users/Jordan Appears if the user selects the results button - displays a list of buttons leading to homework recently completed that needs to be checked.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_16}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Class Results Database: Once a homework is selected from the previous screen, the database will appear showing all the results for that specific homework.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_17}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage3: Appears if the user inputs a wrong data type into an answer box on a homework - should be a decimal.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_18}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage3: Appears if the user inputs a wrong data type into an answer box on a homework - should be an intege.r}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_19}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage3: Appears if the user inputs a wrong data type into an answer box on a homework - should be a string.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_20}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{ErrorMessage4: Appears if the user gets the wrong answer on an answer box on a homework.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_21}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administrator Report Widget: Appears if the user selects the 'report' button from the progress screen, and allows the user to quickly query the database by selecting criteria for an SQL search statement. Can also query an entire clsas or a student, unlike students whi have restricted use.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_22}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Student Report Widget: Appears if the user selects the 'report' button from the progress screen, and allows the user to quickly query the database by selecting criteria for an SQL search statement.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_23}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administration Names Widget: The administrator can use this window to add students into the system. Is accessible from the admin home screen.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_24}

\end{figure}

\begin{figure}[H]

\label{fig:print\_function\_result}\caption{Administration Class Widget: The administrator can use this window to add classes into the system. Is accessible from the admin home screen.}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/pyqt\_figure\_25}

\end{figure}

\section{Definition of Data Requirements}

\subsection{Identification of all data input items}

\begin{itemize}

\item Log In Inputs:

\begin{itemize}

\item Username

\item Password

\end{itemize}

\item Lesson Inputs:

\begin{itemize}

\item Lesson test answer

\item Check answers button

\end{itemize}

\item Homework Set Inputs:

\begin{itemize}

\item Class

\item Deadline

\item Score Requirements

\item Set homework button

\end{itemize}

\item Homework Inputs:

\begin{itemize}

\item Question 1a answer

\item Question 1b answer

\item Question 1c answer

\item Question 1d answer

\item Question 1e answer

\item Question 1f answer

\item Question 2 drop down selection

\item Question 3 drop down selection

\item Question 4 drag and drop inputs

\item Submit answers button

\item Homework Buttons:

\begin{itemize}

\item Reset answers button

\item Mark answers button

\end{itemize}

\end{itemize}

\item All Other Buttons:

\begin{itemize}

\item Log in button

\item Student Version:

\begin{itemize}

\item Lesson button

\item Homework button

\item Progress button

\item Lesson topic buttons

\item Lesson buttons

\item Homework topic buttons

\item Homework buttons

\item Set homework buttons

\end{itemize}

\item Teacher Version:

\begin{itemize}

\item Homework button

\item Results button

\item Progress button

\item Result selection buttons

\item Feedback

\end{itemize}

\item Report Screen:

\begin{itemize}

\item Class selection combo box

\item Student selection combo box

\item Task selection combo box

\item Score selection combo box

\end{itemize}

\item Administration:

\begin{itemize}

\item Student FirstName

\item Student LastName

\item ClassID

\end{itemize}

\item Previous button

\item Next button

\item Return button

\item Finish button

\item OK button

\item Log out button

\end{itemize}

\end{itemize}

\subsection{Identification of all data output items}

\begin{itemize}

\item Displays:

\begin{itemize}

\item All screens

\item Hard-coded lesson examples

\item Hard-coded homework questions

\item Correct answers

\end{itemize}

\item Database:

\begin{itemize}

\item FirstName

\item LastName

\item TaskName

\item OverallPercentageScore

\item IndividualPercentageScore (For each question in a homework)

\item Rating

\item Feedback (From teacher to student)

\end{itemize}

\item Error Messages:

\begin{itemize}

\item ErrorMessage2

\item ErrorMessage3 - Decimal

\item ErrorMessage3 - Integer

\item ErrorMessage3 - String

\item ErrorMessage4

\end{itemize}

\item Displayed On Home:

\begin{itemize}

\item Account name

\item Username

\item Average rating (Student)

\item Class (Teacher)

\end{itemize}

\end{itemize}

\subsection{Explanation of how data output items are generated}

\begin{center}

\begin{tabular}{|p{4cm}|p{6cm}|} \hline

\textbf{Output} & \textbf{How it's generated} \\ \hline

Lesson Examples & These are hard-coded and will be saved as overrides in each lesson's subclass \\ \hline

Homework Examples & These are hard-coded and will be saved as overrides in each homework's subclass \\ \hline

Correct Answers & These will be generated using algorithms which will solve the same problem the user is trying to solve, find the answer, and display it \\ \hline

FirstName & This will be saved to the database once the administrator inputs the class names \\ \hline

LastName & This will be saved to the database once the administrator inputs the class names \\ \hline

TaskName & This will be hard-coded as an attribute in each subclass \\ \hline

OverallPercentageScore & When a homework task is submitted, algorithms will check how many answers were correct, and obtain a percentage from them, then calculate the average from them \\ \hline

IndividualPercentageScore & When a homework task is submitted, algorithms will check how many answers were correct, and obtain a percentage from each question from the average of each question's parts \\ \hline

Rating & This will be decided using selection statements depending on what the OverallPercentageScore is \\ \hline

Feedback & The administrator will manually input this and the user will then be able to see it on their individual results page \\ \hline

ErrorMessage2 & This is a hard-coded QErrorMessage() which will appear if the user inputs an incorrect username or password \\ \hline

ErrorMessage3 & These are hard-coded QErrorMessage()s which will appear if the user inputs an incorrect data type as an answer, depending on what the data type is supposed to be \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{4cm}|p{6cm}|} \hline

\textbf{Output} & \textbf{How it's generated} \\ \hline

ErrorMessage4 & This is a hard-coded QErrorMessage() which will appear if the user inputs an incorrect answer to a homework question \\ \hline

Account Name & Will be a unique identifier saved by the system, only visible on the homescreens \\ \hline

Username & This is the same as the username used to log in, and will be obtained from the same place it's saved for validation, probably a notepad file \\ \hline

Average Rating & All of the user's past homework's ratings will be averaged and displayed on the homescreen \\ \hline

Class & This will be input by the administrator before inputting each students name \\ \hline

\end{tabular}

\end{center}

\subsection{Data Dictionary}

\begin{center}

\begin{tabular}{|p{3.4cm}|p{1.2cm}|p{2cm}|p{2cm}|p{2cm}|p{3.5cm}|}

\hline

\textbf{Name} & \textbf{Data Type} & \textbf{Length} & \textbf{Validation} & \textbf{Example Data} & \textbf{Comment} \\ \hline

UserID & Integer & 4 bits & 0001 to 9999 & 1546 & Unique to each user \\ \hline

Password & String and integers & 7 characters & letter followed by number followed by letter & f7h3j5f & The password generator uses mixed data types to avoid inappropriate passwords \\ \hline

FirstName & String & 15 characters & First letter upper case, rest lower case & John & Unique to each user, but could be shared by some \\ \hline

Surname & String & 25 characters & First letter upper case, rest lower case & Smith & Unique to each user, but could be shared by some \\ \hline

TaskName & String & 25 characters & & Trigonometry 2 & Hard-coded into the system \\ \hline

OverallPercentScore & Real & 3 characters & in range 0 - 100 & 76.5\% & The percentage of marks obtained in a test, decimal points allowed \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{3.4cm}|p{1.2cm}|p{2cm}|p{2cm}|p{2cm}|p{3.5cm}|}

\hline

\textbf{Name} & \textbf{Data Type} & \textbf{Length} & \textbf{Validation} & \textbf{Example Data} & \textbf{Comment} \\ \hline

IndividualPercentScore & Real & 3 characters & in range 0 - 100 & 45.5\% & The percentage of marks for an individual question, field will only appear in separate table for individual tasks \\ \hline

Rating & Blob & 64 kilobytes & \ & Green face graphic & Green, amber or red face graphic \\ \hline

Feedback & String & 500 characters & \ & Good work & This can consist of any characters as it is a personal message \\ \hline

AppointmentTime & Time & 5 characters & 24 hour format & 13:35 & Only relevant if the user has a true SeeAfterClass variable, set automatically bsed on the administrator's timetable but can be changed if necessary\\ \hline

SeeAfterClass & Boolean & 3 characters & Yes or No & Yes & If the user doesn't achieve a sufficient score, this variable will become true and alert the user \\ \hline

ErrorMessage1 & String & 50 characters & & Sorry, the name cannot have integers & An error message if the wrong data type is used to input a name \\ \hline

ErrorMessage2 & String & 50 characters & & Sorry, that is not a valid login & Tells the user if they have input the wrong username or password \\ \hline

ErrorMessage3 & String & 50 characters & & Please input a decimal, not an integer & Tells the user if their incorrect answer is the wrong data type \\ \hline

ErrorMessage4 & String & 50 characters & & That is incorrect, try one more time & Tells the user that their answer is incorrect and gives them one more attempt \\ \hline

CorrectAnswer & Integer, Real, String & 5 characters & Must be a decimal or whole number, or text & 25.5cm & Gives the user the correct answer if they get the question wrong too many times \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{3.4cm}|p{1.2cm}|p{2cm}|p{2cm}|p{2cm}|p{3.5cm}|}

\hline

\textbf{Name} & \textbf{Data Type} & \textbf{Length} & \textbf{Validation} & \textbf{Example Data} & \textbf{Comment} \\ \hline

Login data & String & 25 characters & & johnsmith1, f8j4h6k & The login information saved in the system to be loaded and checked with the user's inputs \\ \hline

Task data & String & 100 characters & & Trigonometry - sin rule, Level 7, Trigonometry & Contains all the information about the task, what difficulty it is, what type it is etc. \\ \hline

Set answers & Integer, Real, String & 5 characters & Must be a decimal or whole number, or text if in a text box & 45{$^o$} & Contains all the set answers for some of the tasks \\ \hline

Calculated answers & Integer, Real & 5 characters & Must be the same solution as the algorithm & 29.8 & Contains algorithms which find and validate the solution for randomly generated tasks \\ \hline

Account name & Integer & 4 characters & Must be a string of 4 integers, 0 - 9 & 1357 & This is the unique identifier for the database \\ \hline

Average rating & Blob & 1 blob & Will be a green, amber or red face & Green face blob & This is the rating from the average rating of each student's completed homework so far \\ \hline

Class & String & 3 - 6 characters & Must be the name of the class, same as is in the centre's system & 10A & Is input with the student names \\ \hline

\end{tabular}

\end{center}

\subsection{Identification of appropriate storage media}

My system files will need to be stored on the school's local server, as every part will need to be accessible from every computer, to save the teacher having to log into every computer one at a time to check work. this way, every computer can access every account, and if they have the permissions, every part of the database. If necessary, some files can be backed up onto an individual computer in the system, or onto a USB stick. As a result, all users and administrators will be able to share the files and use them at the same time.

\section{Database Design}

\subsection{Normalisation}

\subsubsection{ER Diagrams}

\begin{figure}[H]

\label{fig:print\_function\_result}

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Design/er\_diagram}

\end{figure}

\subsubsection{Entity Descriptions}

\begin{center}

\begin{tabular}{|p{3cm}|p{4cm}|p{4cm}|p{3cm}|} \hline

\textbf{Entity} & \textbf{Description} & \textbf{Attributes} & \textbf{Key} \\ \hline

Student & Each record shows how many results a student receives, and which class they are in & \textbf{Student}(\underline{StudentID}, \underline{ClassID}) & Primary Key(StudentID, ClassID) \\ \hline

Class & Shows which students are in the class and the teacher of that class, and how much homework the class has been set & \textbf{Class}(\underline{ClassID}, TeacherID) & Primary Key(ClassID) \\ \hline

Homework & Shows who each homework has been set for, and the results if the homework has been completed & \textbf{Homework}(\underline{HomeworkID}, HomeworkSet, HomeworkResults) & Primary Key(HomeworkID) \\ \hline

HomeworkSet & Shows who has been set the homework, when, and how many times/how long for & \textbf{HomeworkSet}(\underline{ClassID}, \underline{HomeworkID}, DueDate, SetDate) & Primary Key(ClassID, HomeworkID) \\ \hline

HomeworkResults & This record contains all of the results for a completed homework & \textbf{HomeworkResults} (\underline{HomeworkID}, \underline{StudentID}, \underline{QuestionID}, Rating, CompletedDate) & Primary Key(HomeworkID, StudentID, QuestionID) \\ \hline

Question & Shows how many questions were in a homework, and the percentage which are correct in each students results & \textbf{Question}(\underline{HomeworkID}, \underline{QuestionID}, QuestionText, Choice1, Choice2, ChoiceX, CorrectAnswer, TypeOfQuestion) & Primary Key(HomeworkID, QuestionID) \\ \hline

\end{tabular}

\end{center}

\subsubsection{UNF to 3NF}

\textbf{Unnormalised: }

\begin{center}

\begin{tabular}{|p{4cm}|} \hline

AccountID \\

UserID \\

ClassID

FirstName \\

LastName \\

TaskName \\

OverallPercentScore \\

IndividualPercentScore \\

Rating \\

Feedback \\ \hline

\end{tabular}

\end{center}

\textbf{1st Normalised Form: }

\begin{center}

\begin{tabular}{|p{4cm}|p{3cm}|} \hline

\textbf{Repeating Attributes} & \textbf{Non-repeating Attributes} \\ \hline

\textbf{\underline{UserID}} & \textbf{\underline{AccountID}} \\

\textbf{\underline{ClassID}} & \\

\textbf{\underline{TaskName}} & \\

FirstName & \\

LastName & \\

OverallPercentScore & \\

IndividualPercentScore & \\

Feedback & \\

Rating & \\ \hline

\end{tabular}

\end{center}

\textbf{2nd Normalised Form: }

\begin{center}

\begin{tabular}{|p{4cm}|p{3cm}|} \hline

\textbf{Repeating Attributes} & \textbf{Non-repeating Attributes} \\ \hline

\textbf{\underline{ClassID}} & \textbf{\underline{AccountID}} \\

\textbf{\underline{UserID}} & \\

FirstName & \\

LastName & \\

& \\

\textbf{\underline{TaskName}} & \\

OverallPercentScore & \\

IndividualPercentScore & \\

Feedback & \\

Rating & \\ \hline

\end{tabular}

\end{center}

\textbf{3rd Normalised Form: }

\begin{center}

\begin{tabular}{|p{4cm}|p{3cm}|} \hline

\textbf{Repeating Attributes} & \textbf{Non-repeating Attributes} \\ \hline

\textbf{\underline{ClassID}} & \textbf{\underline{AccountID}} \\

\textbf{\underline{UserID}} & \\

FirstName & \\

LastName & \\

& \\

\textbf{\underline{TaskName}} & \\

\underline{IndividualPercentScore} & \\

& \\

\textbf{\underline{IndividualPercentScore}} & \\

\underline{OverallPercentScore} & \\

& \\

\textbf{\underline{OverallPercentScore}} & \\

Rating & \\

Feedback & \\ \hline

\end{tabular}

\end{center}

\subsection{SQL Queries}

\begin{center}

\begin{tabular}{|p{8cm}|p{6cm}|} \hline

\textbf{SQL Query (Python Format)} & \textbf{Description} \\ \hline

sql = """create table Student & This is the SQL statement which \\

(UserID integer, & creates the initial table called \\

FirstName text, & Student, with all of the attributes \\

LastName text, & that need to be displayed to the users \\

TaskName text, & in the database \\

OverallPercentScore real, & \\

IndividualPercentScore real, &\\

Rating blob, & \\

Feedback text, & \\

primary key(UserID))""" & \\ \hline

"""insert into Student & This query adds values to the \\

(FirstName, LastName, UserID) values ('\{0\}', '\{1\}', '\{2\}') & FirstName, LastName and UserID \\

""".format(FirstName, LastName, UserID) & fields when the administrator adds names to the system \\ \hline

"""select \* from Student & This query is for student use and it \\

where TaskName = '\{0\}' and UserID = '\{1\}' & fetches all of the database information \\

""".format(TaskName, UserID) & for that user for a specific homework they have completed \\ \hline

"""select \* from Class & This query is for teacher use and it \\

where TaskName = '\{0\}' and ClassID = '\{1\}' & fetches all of the records saved for an\\

""".format(TaskName, ClassID) & entire class for a specific homework \\ \hline

"""select \* from Class & This query will fetch all of the records \\

where StudentID = '\{0\}'""".format(StudentID) & stored for a specific student, for teacher use \\ \hline

"""select \* from Class & This query can be used by the teacher \\

where TaskName = '\{0\}' and Rating = '\{1\}' & to search for all homework done in a \\

""".format(TaskName, Rating) & class which has not been done to a sufficient standard \\ \hline

"""select \* from Student & This query will select all of the \\

where StudentID = '\{0\}' and Rating = '\{1\}' & homework done by one student which \\

""".format(StudentID, Rating) & has not been done to a sufficient standard \\ \hline

"""update Student & This query will update the database \\

OverallPercentScore = '\{0\}' & with the results when a student \\

IndividualPercentScore = '\{1\}' & completes a homework \\

Rating = '\{2\}' & \\

where StudentID = '\{3\}' and TaskName = '\{4\}' & \\

""".format(OverallPercentScore, IndividualPercentScore, Rating, StudentID, TaskName) & \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{8cm}|p{6cm}|} \hline

\textbf{SQL Query (Python Format)} & \textbf{Description} \\ \hline

"""update Student & This query will update the records \\

Feedback = '\{0\}' & for a student when the \\

where StudentID = '\{1\}' and TaskName = '\{2\}' & teacher manually gives them \\

""".format(Feedback, StudentID, TaskName) & feedback \\ \hline

\end{tabular}

\end{center}

\section{Security and Integrity of the System and Data}

\subsection{Security and Integrity of Data}

As this system will store a small amount of personal data (FirstName, LastName, ClassID), the Data Protection Act of 1998 must be followed. Therefore the data must be kept up to date, and a way of editing the FirstName, LastName and ClassID records will be included to ensure that wrong data is not forced to be kept permanently, should it be input by mistake. To ensure that the data is not kept for longer than the DPA says it should be, any old data will be searched for and deleted whenever the system is started, should it be found. Furthermore, checks shall be implemented to ensure key data fields are not left empty, in order to maintain referential integrity. The only possible way to view the data stored by my system will be on the visual database in the graphical user interface, which will only be accessible following the input of a correct username and password. Finally, all of the data stored will be valid and feasible as drop down menus, and data type relevant error messages will be used to ensure the user is given a fair chance to input answers.

\subsection{System Security}

In order to prevent theft of information and tampering, all information in the system will only be accessible by providing a username and password, and each user will only be allowed to view data relevant to their own progression. Students will not be able to change any information in the database, but teachers will, if they accidentally include a typo when inputting names into the system. Furthermore, the number of people with access to the system will be limited, and those with key access trusted as professional teachers. The database will be encrypted to avoid people somehow accessing the database information by some other means, although only names will be stored as personal information. Lastly, all of the names input into the system will have a validation check to ensure they do not accidentally include numbers, extra capital letters, not enough capital letters or other characters not in the alphabet. Because the names stored will fall under the Data Protection Act, I will ensure that no data is sent to other countries, or even anywhere at all outside of the school, that the data is secure and only accessible by authorised people, that it is only stored by the school, that the data can be updated and destroyed to ensure protection, and that only necessary data will be collected in the first place.

\section{Validation}

\begin{center}

\begin{tabular}{|p{2cm}|p{3cm}|p{4cm}|p{4cm}|} \hline

\textbf{Item} & \textbf{Example} & \textbf{Validation applied} & \textbf{Comments} \\ \hline

ClassID & 10A & Presence check & The ClassID is the same as the classes actual class name, so no validation is required \\ \hline

FirstName & John & Data type check: First letter, capital, rest lower case, all string, no spaces & Checks that the name given is in fact a name \\ \hline

LastName & Smith & Data type check: First letter, capital, rest lower case, all string, no spaces & Checks that the name given is in fact a name \\ \hline

Username & User \ 1 & Format check: First 4 letters of LastName followed by a number. Validation check - Check it's in the system as a valid username & This ensures that the username is in the right format before checking if it's recognised as a valid username \\ \hline

Password & f5h3d7h & Format check: letter, number, letter, number, letter, number, letter, all lower case. Validation check - Check it's in the system and matches the username input beforehand & The password is randomly generated and attached to the username, in that format to avoid any inappropriate words being generated \\ \hline

Text answer input & 7 & ValueErrorCheck - integer & Checks that the value input is the correct data type \\ \hline

Text answer input & 7.5 & ValueErrorCheck - decimal & Checks that the value input is the correct data type \\ \hline

Text answer input & seven & ValueErrorCheck - string & Checks that the value input is the correct data type \\ \hline

ComboBox answer input & select 5 & Presence check & Checks that an option has actually been selected \\ \hline

Drag and drop answer input & all pictures dropped onto shape & Presence check & Checks to see if all of the drag icons have been dropped into the corresponding spaces \\ \hline

Correct & True & Presence check & This boolean corresponds to every answer and says whether or not it is correct \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{2cm}|p{3cm}|p{4cm}|p{4cm}|} \hline

\textbf{Item} & \textbf{Example} & \textbf{Validation applied} & \textbf{Comments} \\ \hline

Class selected & 10A and & Presence check & Makes sure a class has been selected from the combo box to set the homework for \\ \hline

Deadline & 14/12/15 & Presence check & Makes sure a date within a given set of boundaries has been selected from the combo box \\ \hline

Score Requirements & 75\% & Presence check & Makes sure a score percentage has been selected from the combo box \\ \hline

\end{tabular}

\end{center}

\section{Testing}

\begin{landscape}

\subsection{Outline Plan}

\begin{center}

\begin{tabular}{|p{2cm}|p{5cm}|p{5cm}|p{4cm}|}

\hline

\textbf{Test Series} & \textbf{Purpose of Test Series} & \textbf{Testing Strategy} & \textbf{Strategy Rationale}\\ \hline

1 & Test the connections between all of the buttons on the user interfaces & Top-down testing & Each button will be tested to make sure it connects to the right screen \\ \hline

2 & Test all of the input spaces & Bottom-up testing & Each input will be tested once it is developed \\ \hline

3 & Test all information is stored in the correct place in the database & Black box testing & The database will be viewed in a database viewer to ensure that SQL queries are working and data is being stored in the database once they have been developed \\ \hline

4 & Test all of the algorithms to make sure that the program gives the correct output and marks, both mathematical or other & White box testing & Each algorithm will be tested once it is developed \\ \hline

5 & Test that the system fulfils the clients request & Acceptance testing & The system will be developed once it is completed to a usable standard \\ \hline

\end{tabular}

\end{center}

\subsection{Detailed Plan}

\begin{center}

\begin{longtable}{|p{1.5cm}|p{2.5cm}|p{2.5cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|p{2cm}|}

\hline

\textbf{Test Series} & \textbf{Purpose of Test} & \textbf{Test Description} & \textbf{Test Data} & \textbf{Test Data Type (Normal/ Erroneous/ Boundary)} & \textbf{Expected Result} & \textbf{Actual Result} & \textbf{Evidence}\\ \hline

1.001 & To test the student log in button on the first menu functions as intended & This should link to the student menu screen & Click the log in button & Normal & The student account screen should be displayed & & \\ \hline

1.002 & To test the teacher log in button on the first menu functions as intended & This should link to the administrator menu screen & Click the log in button & Normal & The administrator account screen should be displayed & & \\ \hline

1.003 & To test the lessons button on the student account screen functions as intended & This should link to the lesson topic menu screen & Click the lessons button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.004 & To test the homework button on the student account screen functions as intended & This should link to the homework topic menu screen & Click the homework button & Normal & The homework topic menu should be displayed & & \\ \hline

1.005 & To test the progress button on the student account screen functions as intended & This should link to the student's personal database display screen & Click the progress button & Normal & The student's personal database screen should be displayed & & \\ \hline

1.006 & To test the log out button on the student account screen functions as intended & This should close down the entire program & Click the log out button & Normal & The program should stop & & \\ \hline

1.007 & To test the Trigonometry 1 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.008 & To test the Trigonometry 2 button on the lesson topic menu screen functions as intended & This should link to the trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.009 & To test the Pythagoras button on the lesson topic menu screen functions as intended & This should link to the pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.010 & To test the Pythagoras and Trigonometry Problems button on the lesson topic menu screen functions as intended & This should link to the trigonometry and pythagoras problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.011 & To test the Summary button on the lesson topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.012 & To test the return button on the lesson topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.013 & To test the Sides button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the sides lesson & Click the Sides button & Normal & The first sides lesson screen should be displayed & & \\ \hline

1.014 & To test the SOHCAHTOA button on the Trigonometry 1 lesson menu functions as intended & This should link to the screen for the first page of the SOHCAHTOA lesson & Click the SOHCAHTOA button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.015 & To test the return button on the Trigonometry 1 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.016 & To test the Finding Angles button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the finding angles lesson & Click the Finding angles button & Normal & The first finding angles lesson screen should be displayed & & \\ \hline

1.017 & To test the Inverted Angles button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the inverted angles lesson & Click the Inverted angles button & Normal & The first inverted angles lesson screen should be displayed & & \\ \hline

1.018 & To test the 3D Trigonometry button on the Trigonometry 2 lesson menu functions as intended & This should link to the screen for the first page of the 3D Trigonometry lesson & Click the 3D Trigonometry button & Normal & The first 3D trigonometry lesson screen should be displayed & & \\ \hline

1.019 & To test the return button on the Trigonometry 2 menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.020 & To test the Pythagoras Theorem button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the pythagoras theorem lesson & Click the pythagoras theorem button & Normal & The first pythagoras theorem lesson screen should be displayed & & \\ \hline

1.021 & To test the 3D Pythagoras button on the Pythagoras lesson menu functions as intended & This should link to the screen for the first page of the 3D pythagoras lesson & Click the 3D pythagoras button & Normal & The first 3D pythagoras lesson screen should be displayed & & \\ \hline

1.022 & To test the return button on the Pythagoras menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.023 & To test the Easy Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the easy pythagoras and trigonometry problems lesson & Click the easy pythagoras and trigonometry problems button & Normal & The first easy pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.024 & To test the Medium Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the medium pythagoras and trigonometry problems lesson & Click the medium pythagoras and trigonometry problems button & Normal & The first medium pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.025 & To test the Hard Pythagoras and Trigonometry Problems button on the Trigonometry and Pythagoras Problems lesson menu functions as intended & This should link to the screen for the first page of the hard pythagoras and trigonometry problems lesson & Click the hard pythagoras and trigonometry problems button & Normal & The first hard pythagoras and trigonometry problems lesson screen should be displayed & & \\ \hline

1.026 & To test the return button on the Pythagoras and Trigonometry Problems menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.027 & To test the Revise Trigonometry 1 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 1 lesson & Click the revise trigonometry 1 button & Normal & The first revise trigonometry 1 lesson screen should be displayed & & \\ \hline

1.028 & To test the Revise Trigonometry 2 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 2 lesson & Click the revise trigonometry 2 button & Normal & The first revise trigonometry 2 lesson screen should be displayed & & \\ \hline

1.029 & To test the Revise Trigonometry 3 button on the Summary lesson menu functions as intended & This should link to the screen for the first page of the revise trigonometry 3 lesson & Click the revise trigonometry 3 button & Normal & The first revise trigonometry 3 lesson screen should be displayed & & \\ \hline

1.030 & To test the return button on the Summary menu functions as intended & This should link back to the lesson topic menu screen & Click the return button & Normal & The lesson topic menu should be displayed & & \\ \hline

1.031 & To test the return button on the first Sides lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the Sides return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.032 & To test the next button on the first Sides lesson screen functions as intended & This should link to the second Sides lesson screen & Click the Sides next button & Normal & The second Sides lesson screen should be displayed & & \\ \hline

1.033 & To test the previous button on the second Sides lesson screen functions as intended & This should link back to the first Sides lesson screen & Click the Sides previous button & Normal & The first Sides lesson screen should be displayed & & \\ \hline

1.034 & To test the Check Answer button on the second Sides lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Sides check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.035 & To test the Finish button on the second Sides lesson screen functions as intended & This close the lesson and return to the lesson topic menu screen & Click the Sides finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.036 & To test the return button on the first SOHCAHTOA lesson screen functions as intended & This should link back to the Trigonometry 1 menu screen & Click the SOHCAHTOA return button & Normal & The Trigonometry 1 menu screen should be displayed & & \\ \hline

1.037 & To test the next button on the first SOHCAHTOA lesson screen functions as intended & This should link to the second SOHCAHTOA lesson screen & Click the SOHCAHTOA next button & Normal & The second SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.038 & To test the previous button on the second SOHCAHTOA lesson screen functions as intended & This should link back to the first SOHCAHTOA lesson screen & Click the SOHCAHTOA previous button & Normal & The first SOHCAHTOA lesson screen should be displayed & & \\ \hline

1.039 & To test the Check Answer button on the second SOHCAHTOA lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the SOHCAHTOA check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.040 & To test the Finish button on the second SOHCAHTOA lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the SOHCAHTOA finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.041 & To test the return button on the first Finding Angles lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the Finding Angles return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.042 & To test the next button on the first Finding Angles lesson screen functions as intended & This should link to the second Finding Angles lesson screen & Click the Finding Angles next button & Normal & The second Finding Angles lesson screen should be displayed & & \\ \hline

1.043 & To test the previous button on the second Finding Angles lesson screen functions as intended & This should link back to the first Finding Angles lesson screen & Click the Finding Angles previous button & Normal & The first Finding Angles lesson screen should be displayed & & \\ \hline

1.044 & To test the Check Answer button on the second Finding Angles lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Finding Angles check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.045 & To test the Finish button on the second Finding Angles lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Finding Angles finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.046 & To test the return button on the first Inverted Angles lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the Inverted Angles return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.047 & To test the next button on the first Inverted Angles lesson screen functions as intended & This should link to the second Inverted Angles lesson screen & Click the Inverted Angles next button & Normal & The second Inverted Angles lesson screen should be displayed & & \\ \hline

1.048 & To test the previous button on the second Inverted Angles lesson screen functions as intended & This should link back to the first Inverted Angles lesson screen & Click the Inverted Angles previous button & Normal & The first Inverted Angles lesson screen should be displayed & & \\ \hline

1.049 & To test the Check Answer button on the second Inverted Angles lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Inverted Angles check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.050 & To test the Finish button on the second Inverted Angles lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Inverted Angles finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.051 & To test the return button on the first 3D Trigonometry lesson screen functions as intended & This should link back to the Trigonometry 2 menu screen & Click the 3D Trigonometry return button & Normal & The Trigonometry 2 menu screen should be displayed & & \\ \hline

1.052 & To test the next button on the first 3D Trigonometry lesson screen functions as intended & This should link to the second 3D Trigonometry lesson screen & Click the 3D Trigonometry next button & Normal & The second 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.053 & To test the previous button on the second 3D Trigonometry lesson screen functions as intended & This should link back to the first 3D Trigonometry lesson screen & Click the 3D Trigonometry previous button & Normal & The first 3D Trigonometry lesson screen should be displayed & & \\ \hline

1.054 & To test the Check Answer button on the second 3D Trigonometry lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Trigonometry check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.055 & To test the Finish button on the second 3D Trigonometry lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Trigonometry finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.056 & To test the return button on the first Pythagoras Theorem lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the Pythagoras Theorem return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.057 & To test the next button on the first Pythagoras Theorem lesson screen functions as intended & This should link to the second Pythagoras Theorem lesson screen & Click the Pythagoras Theorem next button & Normal & The second Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.058 & To test the previous button on the second Pythagoras Theorem lesson screen functions as intended & This should link back to the first Pythagoras Theorem lesson screen & Click the Pythagoras Theorem previous button & Normal & The first Pythagoras Theorem lesson screen should be displayed & & \\ \hline

1.059 & To test the Check Answer button on the second Pythagoras Theorem lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras Theorem check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.060 & To test the Finish button on the second Pythagoras Theorem lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras Theorem finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.061 & To test the return button on the first 3D Pythagoras lesson screen functions as intended & This should link back to the Pythagoras menu screen & Click the 3D Pythagoras return button & Normal & The Pythagoras menu screen should be displayed & & \\ \hline

1.062 & To test the next button on the first 3D Pythagoras lesson screen functions as intended & This should link to the second 3D Pythagoras lesson screen & Click the 3D Pythagoras next button & Normal & The second 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.063 & To test the previous button on the second 3D Pythagoras lesson screen functions as intended & This should link back to the first 3D Pythagoras lesson screen & Click the 3D Pythagoras previous button & Normal & The first 3D Pythagoras lesson screen should be displayed & & \\ \hline

1.064 & To test the Check Answer button on the second 3D Pythagoras lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the 3D Pythagoras check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.065 & To test the Finish button on the second 3D Pythagoras lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the 3D Pythagoras finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.066 & To test the return button on the first Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Easy return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.067 & To test the next button on the first Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Easy lesson screen & Click the Pythagoras and Trigonometry Problems Easy next button & Normal & The second Pythagoras and Trigonometry Problems Easy lesson screen should be displayed & & \\ \hline

1.068 & To test the previous button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Easy lesson screen & Click the Pythagoras and Trigonometry Problems Easy previous button & Normal & The first Pythagoras and Trigonometry Problems Easy lesson screen should be displayed & & \\ \hline

1.069 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Easy check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.070 & To test the Finish button on the second Pythagoras and Trigonometry Problems Easy lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Easy finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.071 & To test the return button on the first Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Medium return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.072 & To test the next button on the first Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Medium lesson screen & Click the Pythagoras and Trigonometry Problems Medium next button & Normal & The second Pythagoras and Trigonometry Problems Medium lesson screen should be displayed & & \\ \hline

1.073 & To test the previous button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Medium lesson screen & Click the Pythagoras and Trigonometry Problems Medium previous button & Normal & The first Pythagoras and Trigonometry Problems Medium lesson screen should be displayed & & \\ \hline

1.074 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Medium check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.075 & To test the Finish button on the second Pythagoras and Trigonometry Problems Medium lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Medium finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.076 & To test the return button on the first Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems menu screen & Click the Pythagoras and Trigonometry Hard return button & Normal & The Pythagoras and Trigonometry Problems menu screen should be displayed & & \\ \hline

1.077 & To test the next button on the first Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Hard lesson screen & Click the Pythagoras and Trigonometry Problems Hard next button & Normal & The second Pythagoras and Trigonometry Problems Hard lesson screen should be displayed & & \\ \hline

1.078 & To test the previous button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Hard lesson screen & Click the Pythagoras and Trigonometry Problems Hard previous button & Normal & The first Pythagoras and Trigonometry Problems Hard lesson screen should be displayed & & \\ \hline

1.079 & To test the Check Answer button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Pythagoras and Trigonometry Problems Hard check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.080 & To test the Finish button on the second Pythagoras and Trigonometry Problems Hard lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Pythagoras and Trigonometry Problems Hard finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.081 & To test the return button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 1 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.082 & To test the next button on the first Revise Trigonometry 1 lesson screen functions as intended & This should link to the second Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 next button & Normal & The second Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.083 & To test the previous button on the second Revise Trigonometry 1 lesson screen functions as intended & This should link back to the first Revise Trigonometry 1 lesson screen & Click the Revise Trigonometry 1 previous button & Normal & The first Revise Trigonometry 1 lesson screen should be displayed & & \\ \hline

1.084 & To test the Check Answer button on the second Revise Trigonometry 1 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 1 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.085 & To test the Finish button on the second Revise Trigonometry 1 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 1 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.086 & To test the return button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 2 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.087 & To test the next button on the first Revise Trigonometry 2 lesson screen functions as intended & This should link to the second Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 next button & Normal & The second Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.088 & To test the previous button on the second Revise Trigonometry 2 lesson screen functions as intended & This should link back to the first Revise Trigonometry 2 lesson screen & Click the Revise Trigonometry 2 previous button & Normal & The first Revise Trigonometry 2 lesson screen should be displayed & & \\ \hline

1.089 & To test the Check Answer button on the second Revise Trigonometry 2 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 2 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.090 & To test the Finish button on the second Revise Trigonometry 2 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 2 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.091 & To test the return button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link back to the Summary menu screen & Click the Revise Trigonometry 3 return button & Normal & The Summary menu screen should be displayed & & \\ \hline

1.092 & To test the next button on the first Revise Trigonometry 3 lesson screen functions as intended & This should link to the second Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 next button & Normal & The second Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.093 & To test the previous button on the second Revise Trigonometry 3 lesson screen functions as intended & This should link back to the first Revise Trigonometry 3 lesson screen & Click the Revise Trigonometry 3 previous button & Normal & The first Revise Trigonometry 3 lesson screen should be displayed & & \\ \hline

1.094 & To test the Check Answer button on the second Revise Trigonometry 3 lesson screen functions as intended & This should run an algorithm to check if the user's input is correct or not, and tell them & Click the Revise Trigonometry 3 check answers button & Normal & A truthful 'Correct' or 'Incorrect' message should be displayed & & \\ \hline

1.095 & To test the Finish button on the second Revise Trigonometry 3 lesson screen functions as intended & This should close the lesson and return to the lesson topic menu screen & Click the Revise Trigonometry 3 finish button & Normal & The Lesson Topic Menu screen should be displayed & & \\ \hline

1.096 & To test each Set Homework button on the set homework list on the homework topic menu screen functions as intended & This should link to the corresponding homework that is named on the button - could be any homework in any order depending on what the teacher decides to set & Click the set homework button & Normal & The corresponding homework first screen should be displayed & & \\ \hline

1.097 & To test the Trigonometry 1 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.098 & To test the Trigonometry 2 button on the homework topic menu screen functions as intended & This should link to the Trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.099 & To test the Pythagoras button on the homework topic menu screen functions as intended & This should link to the Pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.100 & To test the Pythagoras and Trigonometry Problems button on the homework topic menu screen functions as intended & This should link to the Trigonometry and Pythagoras Problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.101 & To test the Summary button on the homework topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.102 & To test the return button on the homework topic menu functions as intended & This should link back to the student account screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.103 & To test the Sides Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Easy homework screen & click the Sides Easy button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.104 & To test the Sides Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Medium homework screen & click the Sides Medium button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.105 & To test the Sides Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first Sides Hard homework screen & click the Sides Hard button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.106 & To test the SOHCAHTOA Easy button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Easy homework screen & click the SOHCAHTOA Easy button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.107 & To test the SOHCAHTOA Medium button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Medium homework screen & click the SOHCAHTOA Medium button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.108 & To test the SOHCAHTOA Hard button on the Trigonometry 1 homework menu screen functions as intended & This should link to the first SOHCAHTOA Hard homework screen & click the SOHCAHTOA Hard button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.109 & To test the return button on the Trigonometry 1 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.110 & To test the Finding Angles Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Easy homework screen & click the Finding Angles Easy button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.111 & To test the Finding Angles Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Medium homework screen & click the Finding angles Medium button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.112 & To test the Finding Angles Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Finding Angles Hard homework screen & click the Finding angles Hard button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.113 & To test the Inverted Angles Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Easy homework screen & click the Inverted Angles Easy button & Normal & The first Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.114 & To test the Inverted Angles Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Medium homework screen & click the Inverted Angles Medium button & Normal & The first Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.115 & To test the Inverted Angles Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first Inverted Angles Hard homework screen & click the Inverted Angles Hard button & Normal & The first Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.116 & To test the 3D Trigonometry Easy button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Easy homework screen & click the 3D Trigonometry Easy button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.117 & To test the 3D Trigonometry Medium button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Medium homework screen & click the 3D Trigonometry Medium button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.118 & To test the 3D Trigonometry Hard button on the Trigonometry 2 homework menu screen functions as intended & This should link to the first 3D Trigonometry Hard homework screen & click the 3D Trigonometry Hard button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.119 & To test the return button on the Trigonometry 2 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.120 & To test the Pythagoras Theorem Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Easy homework screen & click the Pythagoras Theorem Easy button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.121 & To test the Pythagoras Theorem Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Medium homework screen & click the Pythagoras Theorem Medium button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.122 & To test the Pythagoras Theorem Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first Pythagoras Theorem Hard homework screen & click the Pythagoras Theorem Hard button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.123 & To test the 3D Pythagoras Easy button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Easy homework screen & click the 3D Pythagoras Easy button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.124 & To test the 3D Pythagoras Medium button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Medium homework screen & click the 3D Pythagoras Medium button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.125 & To test the 3D Pythagoras Hard button on the Pythagoras homework menu screen functions as intended & This should link to the first 3D Pythagoras Hard homework screen & click the 3D Pythagoras Hard button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.126 & To test the return button on the Pythagoras homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.127 & To test the Pythagoras and Trigonometry Problems Easy button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Easy homework screen & Click the Pythagoras and Trigonometry Problems Easy button & Normal & The first Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.128 & To test the Pythagoras and Trigonometry Problems Medium button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Medium homework screen & Click the Pythagoras and Trigonometry Problems Medium button & Normal & The first Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.129 & To test the Pythagoras and Trigonometry Problems Hard button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the first Pythagoras and Trigonometry Problems Hard homework screen & Click the Pythagoras and Trigonometry Problems Hard button & Normal & The first Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.130 & To test the return button on the Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.131 & To test the Summary Easy button on the Summary homework menu screen functions as intended & This should link to the first Summary Easy homework screen & Click the Summary Easy button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.132 & To test the Summary Medium button on the Summary homework menu screen functions as intended & This should link to the first Summary Medium homework screen & Click the Summary Medium button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.133 & To test the Summary Hard button on the Summary homework menu screen functions as intended & This should link to the first Summary Hard homework screen & Click the Summary Hard button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.134 & To test the return button on the Summary homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.135 & To test the cancel button on the Sides Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.136 & To test the check answers button on the Sides Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.137 & To test the reset answers button on the Sides Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.138 & To test the next button on the Sides Easy first homework screen functions as intended & This should link to the second Sides Easy homework screen & Click the next button & Normal & The second Sides Easy homework screen should be displayed & & \\ \hline

1.139 & To test the first mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.140 & To test the second mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.141 & To test the third mark it button on the Sides Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.142 & To test the previous button on the Sides Easy second homework screen functions as intended & This should link back to the first Sides Easy homework screen & Click the previous button & Normal & The first Sides Easy homework screen should be displayed & & \\ \hline

1.143 & To test the finish button on the second Sides Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.144 & To test the cancel button on the Sides Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.145 & To test the check answers button on the Sides Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.146 & To test the reset answers button on the Sides Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.147 & To test the next button on the Sides Medium first homework screen functions as intended & This should link to the second Sides Medium homework screen & Click the next button & Normal & The second Sides Medium homework screen should be displayed & & \\ \hline

1.148 & To test the first mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.149 & To test the second mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.150 & To test the third mark it button on the Sides Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.151 &To test the previous button on the Sides Medium second homework screen functions as intended & This should link back to the first Sides Medium homework screen & Click the previous button & Normal & The first Sides Medium homework screen should be displayed & & \\ \hline

1.152 & To test the finish button on the second Sides Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.153 & To test the cancel button on the Sides Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.154 & To test the check answers button on the Sides Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.155 & To test the reset answers button on the Sides Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.156 & To test the next button on the Sides Hard first homework screen functions as intended & This should link to the second Sides Hard homework screen & Click the next button & Normal & The second Sides Hard homework screen should be displayed & & \\ \hline

1.157 & To test the first mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.158 & To test the second mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.159 & To test the third mark it button on the Sides Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.160 & To test the previous button on the Sides Hard second homework screen functions as intended & This should link back to the first Sides Hard homework screen & Click the previous button & Normal & The first Sides Hard homework screen should be displayed & & \\ \hline

1.161 & To test the finish button on the second Sides Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.162 & To test the cancel button on the SOHCAHTOA Easy first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.163 & To test the check answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.164 & To test the reset answers button on the SOHCAHTOA Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.165 & To test the next button on the SOHCAHTOA Easy first homework screen functions as intended & This should link to the second SOHCAHTOA Easy homework screen & Click the next button & Normal & The second SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.166 & To test the first mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.167 & To test the second mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.168 & To test the third mark it button on the SOHCAHTOA Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.169 & To test the previous button on the SOHCAHTOA Easy second homework screen functions as intended & This should link back to the first SOHCAHTOA Easy homework screen & Click the previous button & Normal & The first SOHCAHTOA Easy homework screen should be displayed & & \\ \hline

1.170 & To test the finish button on the second SOHCAHTOA Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.171 & To test the cancel button on the SOHCAHTOA Medium first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.172 & To test the check answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.173 & To test the reset answers button on the SOHCAHTOA Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.174 & To test the next button on the SOHCAHTOA Medium first homework screen functions as intended & This should link to the second SOHCAHTOA Medium homework screen & Click the next button & Normal & The second SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.175 & To test the first mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.176 & To test the second mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.177 & To test the third mark it button on the SOHCAHTOA Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.178 & To test the previous button on the SOHCAHTOA Medium second homework screen functions as intended & This should link back to the first SOHCAHTOA Medium homework screen & Click the previous button & Normal & The first SOHCAHTOA Medium homework screen should be displayed & & \\ \hline

1.179 & To test the finish button on the second SOHCAHTOA Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.180 & To test the cancel button on the SOHCAHTOA Hard first homework screen functions as intended & This should link back to the Trigonometry 1 lesson menu screen & Click the cancel button & Normal & The Trigonometry 1 homework menu should be displayed & & \\ \hline

1.181 & To test the check answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.182 & To test the reset answers button on the SOHCAHTOA Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.183 & To test the next button on the SOHCAHTOA Hard first homework screen functions as intended & This should link to the second SOHCAHTOA Hard homework screen & Click the next button & Normal & The second SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.184 & To test the first mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.185 & To test the second mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.186 & To test the third mark it button on the SOHCAHTOA Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.187 & To test the previous button on the SOHCAHTOA Hard second homework screen functions as intended & This should link back to the first SOHCAHTOA Hard homework screen & Click the previous button & Normal & The first SOHCAHTOA Hard homework screen should be displayed & & \\ \hline

1.188 & To test the finish button on the second SOHCAHTOA Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.189 & To test the cancel button on the Finding Angles Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.190 & To test the check answers button on the Finding Angles Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.191 & To test the reset answers button on the Finding Angles Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.192 & To test the next button on the Finding Angles Easy first homework screen functions as intended & This should link to the second Finding Angles Easy homework screen & Click the next button & Normal & The second Finding Angles Easy homework screen should be displayed & & \\ \hline

1.193 & To test the first mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.194 & To test the second mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.195 & To test the third mark it button on the Finding Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.196 & To test the previous button on the Finding Angles Easy second homework screen functions as intended & This should link back to the first Finding Angles Easy homework screen & Click the previous button & Normal & The first Finding Angles Easy homework screen should be displayed & & \\ \hline

1.197 & To test the finish button on the second Finding Angles Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.198 & To test the cancel button on the Finding Angles Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.199 & To test the check answers button on the Finding Angles Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.200 & To test the reset answers button on the Finding Angles Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.201 & To test the next button on the Finding Angles Medium first homework screen functions as intended & This should link to the second Finding Angles Medium homework screen & Click the next button & Normal & The second Finding Angles Medium homework screen should be displayed & & \\ \hline

1.202 & To test the first mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.203 & To test the second mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.204 & To test the third mark it button on the Finding Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.205 & To test the previous button on the Finding Angles Medium second homework screen functions as intended & This should link back to the first Finding Angles Medium homework screen & Click the previous button & Normal & The first Finding Angles Medium homework screen should be displayed & & \\ \hline

1.206 & To test the finish button on the second Finding Angles Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.207 & To test the cancel button on the Finding Angles Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.208 & To test the check answers button on the Finding Angles Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.209 & To test the reset answers button on the Finding Angles Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.210 & To test the next button on the Finding Angles Hard first homework screen functions as intended & This should link to the second Finding Angles Hard homework screen & Click the next button & Normal & The second Finding Angles Hard homework screen should be displayed & & \\ \hline

1.211 & To test the first mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.212 & To test the second mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.213 & To test the third mark it button on the Finding Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.214 & To test the previous button on the Finding Angles Hard second homework screen functions as intended & This should link back to the first Finding Angles Hard homework screen & Click the previous button & Normal & The first Finding Angles Hard homework screen should be displayed & & \\ \hline

1.215 & To test the finish button on the second Finding Angles Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.216 & To test the cancel button on the Inverted Angles Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.217 & To test the check answers button on the Inverted Angles Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.218 & To test the reset answers button on the Inverted Angles Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.219 & To test the next button on the Inverted Angles Easy first homework screen functions as intended & This should link to the second Inverted Angles Easy homework screen & Click the next button & Normal & The second Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.220 & To test the first mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.221 & To test the second mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.222 & To test the third mark it button on the Inverted Angles Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.223 & To test the previous button on the Inverted Angles Easy second homework screen functions as intended & This should link back to the first Inverted Angles Easy homework screen & Click the previous button & Normal & The first Inverted Angles Easy homework screen should be displayed & & \\ \hline

1.224 & To test the finish button on the second Inverted Angles Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.225 & To test the cancel button on the Inverted Angles Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.226 & To test the check answers button on the Inverted Angles Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.227 & To test the reset answers button on the Inverted Angles Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.228 & To test the next button on the Inverted Angles Medium first homework screen functions as intended & This should link to the second Inverted Angles Medium homework screen & Click the next button & Normal & The second Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.229 & To test the first mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.230 & To test the second mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.231 & To test the third mark it button on the Inverted Angles Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.232 & To test the previous button on the Inverted Angles Medium second homework screen functions as intended & This should link back to the first Inverted Angles Medium homework screen & Click the previous button & Normal & The first Inverted Angles Medium homework screen should be displayed & & \\ \hline

1.233 & To test the finish button on the Inverted Angles Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.234 & To test the cancel button on the Inverted Angles Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.235 & To test the check answers button on the Inverted Angles Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.236 & To test the reset answers button on the Inverted Angles Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.237 & To test the next button on the Inverted Angles Hard first homework screen functions as intended & This should link to the second Inverted Angles Hard homework screen & Click the next button & Normal & The second Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.238 & To test the first mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.239 & To test the second mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.240 & To test the third mark it button on the Inverted Angles Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.241 & To test the previous button on the Inverted Angles Hard second homework screen functions as intended & This should link back to the first Inverted Angles Hard homework screen & Click the previous button & Normal & The first Inverted Angles Hard homework screen should be displayed & & \\ \hline

1.242 & To test the finish button on the second Inverted Angles Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.243 & To test the cancel button on the 3D Trigonometry Easy first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.244 & To test the check answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.245 & To test the reset answers button on the 3D Trigonometry Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.246 & To test the next button on the 3D Trigonometry Easy first homework screen functions as intended & This should link to the second 3D Trigonometry Easy homework screen & Click the next button & Normal & The second 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.247 & To test the first mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.248 & To test the second mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.249 & To test the third mark it button on the 3D Trigonometry Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.250 & To test the previous button on the 3D Trigonometry Easy second homework screen functions as intended & This should link back to the first 3D Trigonometry Easy homework screen & Click the previous button & Normal & The first 3D Trigonometry Easy homework screen should be displayed & & \\ \hline

1.251 & To test the finish button on the second 3D Trigonometry Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.252 & To test the cancel button on the 3D Trigonometry Medium first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.253 & To test the check answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.254 & To test the reset answers button on the 3D Trigonometry Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.255 & To test the next button on the 3D Trigonometry Medium first homework screen functions as intended & This should link to the second 3D Trigonometry Medium homework screen & Click the next button & Normal & The second 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.256 & To test the first mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.257 & To test the second mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.258 & To test the third mark it button on the 3D Trigonometry Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.259 & To test the previous button on the 3D Trigonometry Medium second homework screen functions as intended & This should link back to the first 3D Trigonometry Medium homework screen & Click the previous button & Normal & The first 3D Trigonometry Medium homework screen should be displayed & & \\ \hline

1.260 & To test the finish button on the second 3D Trigonometry Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.261 & To test the cancel button on the 3D Trigonometry Hard first homework screen functions as intended & This should link back to the Trigonometry 2 lesson menu screen & Click the cancel button & Normal & The Trigonometry 2 homework menu should be displayed & & \\ \hline

1.262 & To test the check answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.263 & To test the reset answers button on the 3D Trigonometry Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.264 & To test the next button on the 3D Trigonometry Hard first homework screen functions as intended & This should link to the second 3D Trigonometry Hard homework screen & Click the next button & Normal & The second 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.265 & To test the first mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.266 & To test the second mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.267 & To test the third mark it button on the 3D Trigonometry Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.268 & To test the previous button on the 3D Trigonometry Hard second homework screen functions as intended & This should link back to the first 3D Trigonometry Hard homework screen & Click the previous button & Normal & The first 3D Trigonometry Hard homework screen should be displayed & & \\ \hline

1.269 & To test the finish button on the second 3D Trigonometry Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.270 & To test the cancel button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.271 & To test the check answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.272 & To test the reset answers button on the Pythagoras Theorem Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.273 & To test the next button on the Pythagoras Theorem Easy first homework screen functions as intended & This should link to the second Pythagoras Theorem Easy homework screen & Click the next button & Normal & The second Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.274 & To test the first mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.275 & To test the second mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.276 & To test the third mark it button on the Pythagoras Theorem Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.277 & To test the previous button on the Pythagoras Theorem Easy second homework screen functions as intended & This should link back to the first Pythagoras Theorem Easy homework screen & Click the previous button & Normal & The first Pythagoras Theorem Easy homework screen should be displayed & & \\ \hline

1.278 & To test the finish button on the second Pythagoras Theorem Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.279 & To test the cancel button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.280 & To test the check answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.281 & To test the reset answers button on the Pythagoras Theorem Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.282 & To test the next button on the Pythagoras Theorem Medium first homework screen functions as intended & This should link to the second Pythagoras Theorem Medium homework screen & Click the next button & Normal & The second Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.283 & To test the first mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.284 & To test the second mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.285 & To test the third mark it button on the Pythagoras Theorem Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.286 & To test the previous button on the Pythagoras Theorem Medium second homework screen functions as intended & This should link back to the first Pythagoras Theorem Medium homework screen & Click the previous button & Normal & The first Pythagoras Theorem Medium homework screen should be displayed & & \\ \hline

1.287 & To test the finish button on the second Pythagoras Theorem Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.288 & To test the cancel button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.289 & To test the check answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.290 & To test the reset answers button on the Pythagoras Theorem Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.291 & To test the next button on the Pythagoras Theorem Hard first homework screen functions as intended & This should link to the second Pythagoras Theorem Hard homework screen & Click the next button & Normal & The second Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.292 & To test the first mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.293 & To test the second mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.294 & To test the third mark it button on the Pythagoras Theorem Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.295 & To test the previous button on the Pythagoras Theorem Hard second homework screen functions as intended & This should link back to the first Pythagoras Theorem Hard homework screen & Click the previous button & Normal & The first Pythagoras Theorem Hard homework screen should be displayed & & \\ \hline

1.296 & To test the finish button on the second Pythagoras Theorem Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.297 & To test the cancel button on the 3D Pythagoras Easy first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.298 & To test the check answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.299 & To test the reset answers button on the 3D Pythagoras Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.300 & To test the next button on the 3D Pythagoras Easy first homework screen functions as intended & This should link to the second 3D Pythagoras Easy homework screen & Click the next button & Normal & The second 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.301 & To test the first mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.302 & To test the second mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.303 & To test the third mark it button on the 3D Pythagoras Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.304 & To test the previous button on the 3D Pythagoras Easy second homework screen functions as intended & This should link back to the first 3D Pythagoras Easy homework screen & Click the previous button & Normal & The first 3D Pythagoras Easy homework screen should be displayed & & \\ \hline

1.305 & To test the finish button on the second 3D Pythagoras Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.306 & To test the cancel button on the 3D Pythagoras Medium first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.307 & To test the check answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.308 & To test the reset answers button on the 3D Pythagoras Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.309 & To test the next button on the 3D Pythagoras Medium first homework screen functions as intended & This should link to the second 3D Pythagoras Medium homework screen & Click the next button & Normal & The second 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.310 & To test the first mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.311 & To test the second mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.312 & To test the third mark it button on the 3D Pythagoras Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.313 & To test the previous button on the 3D Pythagoras Medium second homework screen functions as intended & This should link back to the first 3D Pythagoras Medium homework screen & Click the previous button & Normal & The first 3D Pythagoras Medium homework screen should be displayed & & \\ \hline

1.314 & To test the finish button on the second 3D Pythagoras Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.315 & To test the cancel button on the 3D Pythagoras Hard first homework screen functions as intended & This should link back to the Pythagoras lesson menu screen & Click the cancel button & Normal & The Pythagoras homework menu should be displayed & & \\ \hline

1.316 & To test the check answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.317 & To test the reset answers button on the 3D Pythagoras Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.318 & To test the next button on the 3D Pythagoras Hard first homework screen functions as intended & This should link to the second 3D Pythagoras Hard homework screen & Click the next button & Normal & The second 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.319 & To test the first mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.320 & To test the second mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.321 & To test the third mark it button on the 3D Pythagoras Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.322 & To test the previous button on the 3D Pythagoras Hard second homework screen functions as intended & This should link back to the first 3D Pythagoras Hard homework screen & Click the previous button & Normal & The first 3D Pythagoras Hard homework screen should be displayed & & \\ \hline

1.323 & To test the finish button on the second 3D Pythagoras Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.324 & To test the cancel button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.325 & To test the check answers button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.326 & To test the reset answers button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.327 & To test the next button on the Pythagoras and Trigonometry Problems Easy first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Easy homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.328 & To test the first mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.329 & To test the second mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.330 & To test the third mark it button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.331 & To test the previous button on the Pythagoras and Trigonometry Problems Easy second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Easy homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Easy homework screen should be displayed & & \\ \hline

1.332 & To test the finish button on the second Pythagoras and Trigonometry Problems Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.333 & To test the cancel button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.334 & To test the check answers button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.335 & To test the reset answers button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.336 & To test the next button on the Pythagoras and Trigonometry Problems Medium first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Medium homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.337 & To test the first mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.338 & To test the second mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.339 & To test the third mark it button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.340 & To test the previous button on the Pythagoras and Trigonometry Problems Medium second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Medium homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Medium homework screen should be displayed & & \\ \hline

1.341 & To test the finish button on the second Pythagoras and Trigonometry Problems Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.342 & To test the cancel button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should link back to the Pythagoras and Trigonometry Problems lesson menu screen & Click the cancel button & Normal & The Pythagoras and Trigonometry Problems homework menu should be displayed & & \\ \hline

1.343 & To test the check answers button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.344 & To test the reset answers button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.345 & To test the next button on the Pythagoras and Trigonometry Problems Hard first homework screen functions as intended & This should link to the second Pythagoras and Trigonometry Problems Hard homework screen & Click the next button & Normal & The second Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.346 & To test the first mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.347 & To test the second mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.348 & To test the third mark it button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.349 & To test the previous button on the Pythagoras and Trigonometry Problems Hard second homework screen functions as intended & This should link back to the first Pythagoras and Trigonometry Problems Hard homework screen & Click the previous button & Normal & The first Pythagoras and Trigonometry Problems Hard homework screen should be displayed & & \\ \hline

1.350 & To test the finish button on the second Pythagoras and Trigonometry Problems Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.351 & To test the cancel button on the Summary Easy first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.352 & To test the check answers button on the Summary Easy first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.353 & To test the reset answers button on the Summary Easy first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.354 & To test the next button on the Summary Easy first homework screen functions as intended & This should link to the second Summary Easy homework screen & Click the next button & Normal & The second Summary Easy homework screen should be displayed & & \\ \hline

1.355 & To test the first mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.356 & To test the second mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.357 & To test the third mark it button on the Summary Easy second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.358 & To test the previous button on the Summary Easy second homework screen functions as intended & This should link back to the first Summary Easy homework screen & Click the previous button & Normal & The first Summary Easy homework screen should be displayed & & \\ \hline

1.359 & To test the finish button on the second Summary Easy homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.360 & To test the cancel button on the Summary Medium first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.361 & To test the check answers button on the Summary Medium first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.362 & To test the reset answers button on the Summary Medium first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.363 & To test the next button on the Summary Medium first homework screen functions as intended & This should link to the second Summary Medium homework screen & Click the next button & Normal & The second Summary Medium homework screen should be displayed & & \\ \hline

1.364 & To test the first mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.365 & To test the second mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.366 & To test the third mark it button on the Summary Medium second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.367 & To test the previous button on the Summary Medium second homework screen functions as intended & This should link back to the first Summary Medium homework screen & Click the previous button & Normal & The first Summary Medium homework screen should be displayed & & \\ \hline

1.368 & To test the finish button on the second Summary Medium homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.369 & To test the cancel button on the Summary Hard first homework screen functions as intended & This should link back to the Trigonometry Summary lesson menu screen & Click the cancel button & Normal & The Summary homework menu should be displayed & & \\ \hline

1.370 & To test the check answers button on the Summary Hard first homework screen functions as intended & This should use algorithms to check if the user's answers are correct or not, and tell them & Click the check answers button & Normal & It should inform the user whether or not their answers are correct & & \\ \hline

1.371 & To test the reset answers button on the Summary Hard first homework screen functions as intended & This should turn all of the current values in the input spaces to none & Click the reset answers button & Normal & It should remove any answers in the boxes or move drag icons back to their original position & & \\ \hline

1.372 & To test the next button on the Summary Hard first homework screen functions as intended & This should link to the second Summary Hard homework screen & Click the next button & Normal & The second Summary Hard homework screen should be displayed & & \\ \hline

1.373 & To test the first mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 2 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the first mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.374 & To test the second mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 3 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the second mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.375 & To test the third mark it button on the Summary Hard second homework screen functions as intended & This should run algorithms to check if the user's answers to question 4 are correct or not, and tell the user, and then save these as final submissions ready to be stored in the database & Click the third mark it button & Normal & The user should be told whether or not their answers are correct & & \\ \hline

1.376 & To test the previous button on the Summary Hard second homework screen functions as intended & This should link back to the first Summary Hard homework screen & Click the previous button & Normal & The first Summary Hard homework screen should be displayed & & \\ \hline

1.377 & To test the finish button on the second Summary Hard homework screen functions as intended & This should take the user's results, save them to the database, and return the user to the homework topic menu screen & Click the finish button & Normal & The results should be saved to the database and the homework topic menu screen should be displayed & & \\ \hline

1.378 & To test the Not Completed buttons on the progress screen function as intended & This should link to the first screen of the respective homework & Click the not completed button (Will have the name of the homework) & Normal & The first screen of the respective homework should be displayed & & \\ \hline

1.379 & To test the Not Enough Score buttons on the progress screen function as intended & This should link to the first screen of the respective homework & Click the not enough score button (Will have the name of the homework) & Normal & The first screen of the respective homework should be displayed & & \\ \hline

1.380 & To test the return button on the progress screen functions as intended & This should link back to the student account home screen & Click the return button & Normal & The student account screen should be displayed & & \\ \hline

1.381 & To test the homework button on the admin account home screen functions as intended & This should link to the homework topic menu screen & Click the homework button & Normal & The homework topic menu should be displayed & & \\ \hline

1.382 & To test the results button on the admin account home screen functions as intended & This should link to the recent results menu & Click the results button & Normal & The recent results menu should be displayed & & \\ \hline

1.383 & To test the progress button on the admin account home screen functions as intended & This should link to the database viewer screen & Click the progress button & Normal & The database viewer screen should be displayed & & \\ \hline

1.384 & To test the log out button on the admin account home screen functions as intended & This should log the user off and close the application & Click the log out button & Normal & The program should stop running & & \\ \hline

1.385 & To test each Set Homework button on the admin set homework list on the homework topic menu screen functions as intended & This should link to the corresponding homework that is named on the button - could be any homework in any order depending on what the teacher decides to set & Click the set homework button & Normal & The corresponding homework results screen should be displayed & & \\ \hline

1.386 & To test the Trigonometry 1 button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry 1 menu screen & Click the Trigonometry 1 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.387 & To test the Trigonometry 2 button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry 2 menu screen & Click the Trigonometry 2 button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.388 & To test the Pythagoras button on the admin homework topic menu screen functions as intended & This should link to the Pythagoras menu screen & Click the Pythagoras button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.389 & To test the Pythagoras and Trigonometry Problems button on the admin homework topic menu screen functions as intended & This should link to the Trigonometry and Pythagoras Problems menu screen & Click the Pythaogras and Trigonometry Problems button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.390 & To test the Summary button on the admin homework topic menu screen functions as intended & This should link to the summary menu screen & Click the Summary button & Normal & The topic's menu screen should be displayed & & \\ \hline

1.391 & To test the return button on the homework topic menu functions as intended & This should link back to the admin account screen & Click the return button & Normal & The admin account screen should be displayed & & \\ \hline

1.392 & To test the Sides Easy button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.393 & To test the Sides Medium button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.394 & To test the Sides Hard button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the Sides Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.395 & To test the SOHCAHTOA Easy button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.396 & To test the SOHCAHTOA Medium button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.397 & To test the SOHCAHTOA Hard button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link to the homework set screen & click the SOHCAHTOA Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.398 & To test the return button on the Admin Trigonometry 1 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.399 & To test the Finding Angles Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the first homework set screen & click the Finding Angles Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.400 & To test the Finding Angles Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Finding angles Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.401 & To test the Finding Angles Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Finding angles Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.402 & To test the Inverted Angles Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.403 & To test the Inverted Angles Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.404 & To test the Inverted Angles Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the Inverted Angles Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.405 & To test the 3D Trigonometry Easy button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.406 & To test the 3D Trigonometry Medium button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.407 & To test the 3D Trigonometry Hard button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link to the homework set screen & click the 3D Trigonometry Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.408 & To test the return button on the Admin Trigonometry 2 homework menu screen functions as intended & This should link back to the homework topic menu screen & click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.409 & To test the Pythagoras Theorem Easy button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.410 & To test the Pythagoras Theorem Medium button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.411 & To test the Pythagoras Theorem Hard button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the Pythagoras Theorem Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.412 & To test the 3D Pythagoras Easy button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.413 & To test the 3D Pythagoras Medium button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.414 & To test the 3D Pythagoras Hard button on the Admin Pythagoras homework menu screen functions as intended & This should link to the homework set screen & click the 3D Pythagoras Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.415 & To test the return button on the Admin Pythagoras homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.416 & To test the Pythagoras and Trigonometry Problems Easy button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.417 & To test the Pythagoras and Trigonometry Problems Medium button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.418 & To test the Pythagoras and Trigonometry Problems Hard button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link to the homework set screen & Click the Pythagoras and Trigonometry Problems Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.419 & To test the return button on the Admin Pythagoras and Trigonometry Problems homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.420 & To test the Summary Easy button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Easy button & Normal & The homework set screen should be displayed & & \\ \hline

1.421 & To test the Summary Medium button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Medium button & Normal & The homework set screen should be displayed & & \\ \hline

1.422 & To test the Summary Hard button on the Admin Summary homework menu screen functions as intended & This should link to the homework set screen & Click the Summary Hard button & Normal & The homework set screen should be displayed & & \\ \hline

1.423 & To test the return button on the Admin Summary homework menu screen functions as intended & This should link back to the homework topic menu screen & Click the return button & Normal & The homework topic menu screen should be displayed & & \\ \hline

1.424 & To test the return button on the homework set screen functions as intended & This should link back to the topic lesson menu which was selected before & Click the return button & Normal & The previous topic's homework menu should be displated & & \\ \hline

1.425 & To test the set button on the homework set screen functions as intended & This should add the selected homework to the list of each student in the selected class, and inform them when they next load the application, and finally register the deadline & Click the set button & Normal & The homework should be added to the list of each student in the class, and the deadline registered & & \\ \hline

1.426 & To test the Result buttons on the results menu screen functions as intended & This should link to the results of the corresponding homework in the database viewer for the entire class & Click the results button(s) & Normal & The results for the corresponding homework across the class should be displayed in the database viewer & & \\ \hline

1.427 & To test the return button on the results menu screen functions as intended & This should link back to the administrator account home screen & Click the return button & Normal & The administrator account home screen should be displayed & & \\ \hline

1.428 & To test the return button on the admin progress database viewer screen functions as intended & This should link back to the administrator account home screen & Click the return button & Normal & The administrator account home screen should be displayed & & \\ \hline

1.429 & To test the return button on the administrator report screen functions as intended & This should link back to the admin progress screen & Click the return button & Normal & The administrator progress screen should be displayed & & \\ \hline

1.430 & To test the query button on the administrator report screen functions as intended & This should link back to the admin progress screen and new queried information should be displayed & Click the query button & Normal & The admin progress screen should be displayed with updated data in the database widget & & \\ \hline

1.431 & To test the return button on the student report screen functions as intended & This should link back to the student progress screen & Click the return button & Normal & The student progress screen should be displayed & & \\ \hline

1.432 & To test the query button on the student report screen functions as intended & This should link back to the student progress screen and new queried information should be displayed & Click the query button & Normal & The student progress screen should be displayed with updated data in the database widget & & \\ \hline

1.433 & To test the add class button on the add names widget functions as intended & This should link to the add class screen & Click the add class button & Normal & The add class screen should be displayed & & \\ \hline

1.434 & To test the add name button on the add names widget functions as intended & This should add a name to the system and link back to the admin home screen & Click the add name button & Normal & The name should be stored and the admin home screen should be displayed & & \\ \hline

1.435 & To test the add another button on the add names widget functions as intended & This should add a name to the system and remain on the same page for the process to repeat & Click the add another button & Normal & The name should be added to the sytem and the input boxes wiped ready for another name & & \\ \hline

1.436 & To test the return button on the add names widget functions as intended & This should link back to the admin home screen & Click the return button & Normal & The admin home screen should be displayed & & \\ \hline

1.437 & To test the add stduents button on the add class widget functions as intended & This should link to the add names screen & Click the add students button & Normal & The add names screen should be displayed & & \\ \hline

1.438 & To test the add class button on the add class widget functions as intended & This should add a class to the system and then return to the admin home screen & Click the add class button & Normal & The class should be added to the system and the admin home screen should be displayed & & \\ \hline

1.439 & To test the add another button on the add class widget functions as intended & This should add a class to the system and remain on the same page for the process to repeat & Click the add another button & Normal & The class should be added to the sytem and the input boxes wiped ready for another class & & \\ \hline

1.440 & To test the return button on the add class widget functions as intended & This should link back to the admin home screen & Click the return button & Normal & The admin home screen should be displayed & & \\ \hline

2.001 & To test that the system can recognise a user's username when input correctly and allow access to their account & This should allow the user access to their account & Input User1 & Erroneous, Presence & The username should be accepted and the user allowed access (alongside a correct password) & & \\ \hline

2.002 & To test that the system can recognise a user's password when input correctly and allow access to their account & This should allow the user access to their account & Input d8g3h6g & Erroneous, Presence & The password should be accepted and the user allowed access to their account (alongside a correct username) & & \\ \hline

2.003 & To test that the input for the practice question on the Sides second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.004 & To test that the input for the practice question on the SOHCAHTOA second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.005 & To test that the input for the practice question on the Finding Angles second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.006 & To test that the input for the practice question on the Inverted Angles second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.007 & To test that the input for the practice question on the 3D Trigonometry second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.008 & To test that the input for the practice question on the Pythagoras' Theorem second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.009 & To test that the input for the practice question on the 3D Pythagoras second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.010 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Easy second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.011 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Medium second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneou, Presences & The user should be notified whether or not they were right & & \\ \hline

2.012 & To test that the input for the practice question on the Pythagoras and Trigonometry Problems Hard second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.013 & To test that the input for the practice question on the Revise Trigonometry 1 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.014 & To test that the input for the practice question on the Revise Trigonometry 2 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.015 & To test that the input for the practice question on the Revise Trigonometry 3 second lesson screen is properly checked as correct or incorrect & This should truthfully notify the user whether they were right or wrong & 45 degrees & Boundary, Erroneous, Presence & The user should be notified whether or not they were right & & \\ \hline

2.016 & To test that the text input for the generic question 1 part A is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.017 & To test that the text input for the generic question 1 part B is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.018 & To test that the text input for the generic question 1 part C is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon presisng the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.019 & To test that the text input for the generic question 1 part D is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.020 & To test that the text input for the generic question 1 part E is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.021 & To test that the text input for the generic question 1 part F is received on the Sides Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.022 & To test that the combo box input for the generic question 2 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.023 & To test that the combo box input for the generic question 3 is received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.024 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.025 & To test that the text input for the generic question 1 part A is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.026 & To test that the text input for the generic question 1 part B is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.027 & To test that the text input for the generic question 1 part C is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.028 & To test that the text input for the generic question 1 part D is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.029 & To test that the text input for the generic question 1 part E is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.030 & To test that the text input for the generic question 1 part F is received on the Sides Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.031 & To test that the combo box input for the generic question 2 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.032 & To test that the combo box input for the generic question 3 is received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.033 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.034 & To test that the text input for the generic question 1 part A is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.035 & To test that the text input for the generic question 1 part B is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.036 & To test that the text input for the generic question 1 part C is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.037 & To test that the text input for the generic question 1 part D is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.038 & To test that the text input for the generic question 1 part E is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.039 & To test that the text input for the generic question 1 part F is received on the Sides Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.040 & To test that the combo box input for the generic question 2 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.041 & To test that the combo box input for the generic question 3 is received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.042 & To test that the drag and drop inputs for the generic question 4 are received on the Sides Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.043 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.044 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.045 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.046 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.047 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.048 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.049 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.050 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.051 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.052 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.053 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.054 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.055 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.056 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.057 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.058 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.059 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.060 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.061 & To test that the text input for the generic question 1 part A is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.062 & To test that the text input for the generic question 1 part B is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.063 & To test that the text input for the generic question 1 part C is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.064 & To test that the text input for the generic question 1 part D is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.065 & To test that the text input for the generic question 1 part E is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.066 & To test that the text input for the generic question 1 part F is received on the SOHCAHTOA Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.067 & To test that the combo box input for the generic question 2 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.068 & To test that the combo box input for the generic question 3 is received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.069 & To test that the drag and drop inputs for the generic question 4 are received on the SOHCAHTOA Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.070 & To test that the text input for the generic question 1 part A is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.071 & To test that the text input for the generic question 1 part B is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.072 & To test that the text input for the generic question 1 part C is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.073 & To test that the text input for the generic question 1 part D is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.074 & To test that the text input for the generic question 1 part E is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.075 & To test that the text input for the generic question 1 part F is received on the Finding Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.076 & To test that the combo box input for the generic question 2 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.077 & To test that the combo box input for the generic question 3 is received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.078 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.079 & To test that the text input for the generic question 1 part A is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.080 & To test that the text input for the generic question 1 part B is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.081 & To test that the text input for the generic question 1 part C is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.082 & To test that the text input for the generic question 1 part D is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.083 & To test that the text input for the generic question 1 part E is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.084 & To test that the text input for the generic question 1 part F is received on the Finding Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.085 & To test that the combo box input for the generic question 2 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.086 & To test that the combo box input for the generic question 3 is received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.087 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.088 & To test that the text input for the generic question 1 part A is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.089 & To test that the text input for the generic question 1 part B is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.090 & To test that the text input for the generic question 1 part C is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.091 & To test that the text input for the generic question 1 part D is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.092 & To test that the text input for the generic question 1 part E is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.093 & To test that the text input for the generic question 1 part F is received on the Finding Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.094 & To test that the combo box input for the generic question 2 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.095 & To test that the combo box input for the generic question 3 is received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.096 & To test that the drag and drop inputs for the generic question 4 are received on the Finding Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.097 & To test that the text input for the generic question 1 part A is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.098 & To test that the text input for the generic question 1 part B is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.099 & To test that the text input for the generic question 1 part C is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.100 & To test that the text input for the generic question 1 part D is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.101 & To test that the text input for the generic question 1 part E is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.102 & To test that the text input for the generic question 1 part F is received on the Inverted Angles Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.103 & To test that the combo box input for the generic question 2 is received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.104 & To test that the combo box input for the generic question 3 is received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.105 & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.106 & To test that the text input for the generic question 1 part A is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.107 & To test that the text input for the generic question 1 part B is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.108 & To test that the text input for the generic question 1 part C is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.109 & To test that the text input for the generic question 1 part D is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.110 & To test that the text input for the generic question 1 part E is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.111 & To test that the text input for the generic question 1 part F is received on the Inverted Angles Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.112 & To test that the combo box input for the generic question 2 is received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.113 & To test that the combo box input for the generic question 3 is received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.114 & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.115 & To test that the text input for the generic question 1 part A is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.116 & To test that the text input for the generic question 1 part B is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.117 & To test that the text input for the generic question 1 part C is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.118 & To test that the text input for the generic question 1 part D is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.119 & To test that the text input for the generic question 1 part E is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.120 & To test that the text input for the generic question 1 part F is received on the Inverted Angles Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.121 & To test that the combo box input for the generic question 2 is received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.122 & To test that the combo box input for the generic question 3 is received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.123 & To test that the drag and drop inputs for the generic question 4 are received on the Inverted Angles Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.124 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.125 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.126 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.127 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.128 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.129 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.130 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.131 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.132 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.133 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.134 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.135 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.136 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.137 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.138 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.139 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.140 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.141 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.142 & To test that the text input for the generic question 1 part A is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.143 & To test that the text input for the generic question 1 part B is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.144 & To test that the text input for the generic question 1 part C is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.145 & To test that the text input for the generic question 1 part D is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.146 & To test that the text input for the generic question 1 part E is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.147 & To test that the text input for the generic question 1 part F is received on the 3D Trigonometry Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.148 & To test that the combo box input for the generic question 2 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.149 & To test that the combo box input for the generic question 3 is received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.150 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Trigonometry Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.151 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.152 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.153 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.154 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.155 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.156 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.157 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.158 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.159 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.160 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.161 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.162 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.163 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.164 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.165 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.166 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.167 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.168 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.169 & To test that the text input for the generic question 1 part A is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.170 & To test that the text input for the generic question 1 part B is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.171 & To test that the text input for the generic question 1 part C is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.172 & To test that the text input for the generic question 1 part D is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.173 & To test that the text input for the generic question 1 part E is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.174 & To test that the text input for the generic question 1 part F is received on the Pythagoras' Theorem Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.175 & To test that the combo box input for the generic question 2 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.176 & To test that the combo box input for the generic question 3 is received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.177 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras' Theorem Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.178 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.179 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.180 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.181 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.182 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.183 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.184 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.185 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.186 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.187 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.188 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.189 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.190 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.191 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.192 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.193 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.194 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.195 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.196 & To test that the text input for the generic question 1 part A is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.197 & To test that the text input for the generic question 1 part B is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.198 & To test that the text input for the generic question 1 part C is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.199 & To test that the text input for the generic question 1 part D is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.200 & To test that the text input for the generic question 1 part E is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.201 & To test that the text input for the generic question 1 part F is received on the 3D Pythagoras Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.202 & To test that the combo box input for the generic question 2 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.203 & To test that the combo box input for the generic question 3 is received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.204 & To test that the drag and drop inputs for the generic question 4 are received on the 3D Pythagoras Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.205 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.206 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.207 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.208 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.209 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.210 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.211 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.212 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.213 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.214 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.215 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.216 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.217 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.218 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.219 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.220 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.221 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.222 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.223 & To test that the text input for the generic question 1 part A is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.224 & To test that the text input for the generic question 1 part B is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.225 & To test that the text input for the generic question 1 part C is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.226 & To test that the text input for the generic question 1 part D is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.227 & To test that the text input for the generic question 1 part E is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.228 & To test that the text input for the generic question 1 part F is received on the Pythagoras and Trigonometry Problems Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.229 & To test that the combo box input for the generic question 2 is received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.230 & To test that the combo box input for the generic question 3 is received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.231 & To test that the drag and drop inputs for the generic question 4 are received on the Pythagoras and Trigonometry Problems Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.232 & To test that the text input for the generic question 1 part A is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.233 & To test that the text input for the generic question 1 part B is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.234 & To test that the text input for the generic question 1 part C is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.235 & To test that the text input for the generic question 1 part D is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.236 & To test that the text input for the generic question 1 part E is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.237 & To test that the text input for the generic question 1 part F is received on the Summary Easy first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.238 & To test that the combo box input for the generic question 2 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.239 & To test that the combo box input for the generic question 3 is received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.240 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Easy second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.241 & To test that the text input for the generic question 1 part A is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.242 & To test that the text input for the generic question 1 part B is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.243 & To test that the text input for the generic question 1 part C is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.244 & To test that the text input for the generic question 1 part D is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.245 & To test that the text input for the generic question 1 part E is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.246 & To test that the text input for the generic question 1 part F is received on the Summary Medium first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.247 & To test that the combo box input for the generic question 2 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.248 & To test that the combo box input for the generic question 3 is received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.249 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Medium second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.250 & To test that the text input for the generic question 1 part A is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.251 & To test that the text input for the generic question 1 part B is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.252 & To test that the text input for the generic question 1 part C is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.253 & To test that the text input for the generic question 1 part D is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.254 & To test that the text input for the generic question 1 part E is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.255 & To test that the text input for the generic question 1 part F is received on the Summary Hard first homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 45 degrees & Boundary, Erroneous, Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.256 & To test that the combo box input for the generic question 2 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.257 & To test that the combo box input for the generic question 3 is received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & 30 degrees & Presence & The user should either receive a mark or be told they were wrong & & \\ \hline

2.258 & To test that the drag and drop inputs for the generic question 4 are received on the Summary Hard second homework screen, checked and output is given as intended & The user should be truthfully informed whether or not they were right upon pressing the mark it button & First icon dragged to third space on diagram & Presence, Erroneous & The user should either receive marks or be told they were wrong & & \\ \hline

2.259 & To test that the system can recognise an administrator's username when input correctly and allow access to their account & This should allow the administrator access to their account & Input Admin1 & Erroneous, Presence & The username should be accepted and the administrator allowed access (alongside a correct password) & & \\ \hline

2.260 & To test that the system can recognise an administrator's password when input correctly and allow access to their account & This should allow the administrator access to their account & Input h4j8d8s & Erroneous, Presence & The password should be accepted and the administrator allowed access to their account (alongside a correct username) & & \\ \hline

2.261 & To test that the system can register a selected class to be set homework from a combo box & This should set the homework for the students in the class and notify them on their accounts & 10A & Presence & The class should be registered as having to do that homework and each student should be notified & & \\ \hline

2.262 & To test that the system can register a chosen deadline for a class to complete a homework by from a combo box & This should set and register a date and notify the students of this date & 12/12/15 & Presence & The class should be notified of the deadline and it should be registered in the system & & \\ \hline

2.263 & To test that the system can register a score requirement for a selected homework from a combo box & This should register a minimum score for each student to achieve before they can properly submit the homework as complete & & \\ \hline

2.264 & To test that the feedback box for the Sides Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.265 & To test that the feedback box for the Sides Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.266 & To test that the feedback box for the Sides Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.267 & To test that the feedback box for the SOHCAHTOA Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.268 & To test that the feedback box for the SOHCHTOA Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.269 & To test that the feedback box for the SOHCAHTOA Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.270 & To test that the feedback box for the Finding Angles Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.271 & To test that the feedback box for the Finding Angles Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.272 & To test that the feedback box for the Finding Angles Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.273 & To test that the feedback box for the Inverted Angles Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.274 & To test that the feedback box for the Inverted Angles Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.275 & To test that the feedback box for the Inverted Angles Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.276 & To test that the feedback box for the 3D Trigonometry Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.277 & To test that the feedback box for the 3D Trigonometry Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.278 & To test that the feedback box for the 3D Trigonometry Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.279 & To test that the feedback box for the Pythagoras' Theorem Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.280 & To test that the feedback box for the Pythagoras' Theorem Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.281 & To test that the feedback box for the Pythagoras' Theorem Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.282 & To test that the feedback box for the 3D Pythagoras Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.283 & To test that the feedback box for the 3D Pythagoras Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.284 & To test that the feedback box for the 3D Pythagoras Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.285 & To test that the feedback box for the Pythagoras and Trigonometry Problems Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.286 & To test that the feedback box for the Pythagoras and Trigonometry Problems Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.287 & To test that the feedback box for the Pythagoras and Trigonometry Problems Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.288 & To test that the feedback box for the Summary Easy homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.289 & To test that the feedback box for the Summary Medium homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.290 & To test that the feedback box for the Summary Hard homework functions as intended & This should accept a string of text from an administrator and make it available for the corresponding student to view & Very good work & Normal, Presence & The corresponding student should then be able to view this feedback & & \\ \hline

2.291 & To test the the class selection combo box on the admin report screen functions as intended & This should accept a class selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.292 & To test the the student selection combo box on the admin report screen functions as intended & This should accept a student selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.293 & To test the the task selection combo box on the admin report screen functions as intended & This should accept a task selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.294 & To test the the score selection combo box on the admin report screen functions as intended & This should accept a score selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.295 & To test the the task selection combo box on the student report screen functions as intended & This should accept a task selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.296 & To test the the score selection combo box on the student report screen functions as intended & This should accept a score selected from the combo box and use it in the SQL statement to query the database & 10A & Normal, Presence & The input should be used in the SQL statement to give relevant information & & \\ \hline

2.297 & To test the FirstName line edit on the add names widget functions as intended & This should accept the name given, validate it as a name and store it in the database & John & Erroneous, Presence & The name should be validated and added to the database & & \\ \hline

2.298 & To test the LastName line edit on the add names widget functions as intended & This should accept the name given, validate it as a name and store it in the database & Smith & Erroneous, Presence & The name should be validated and added to the database & & \\ \hline

2.299 & To test the ClassID line edit on the add class widget functions as intended & This should accept the class given, validate it as a ClassID and store it in the database & 10A & Erroneous, Presence & The class should be validated and added to the database & & \\ \hline

3.001 & To test that the ClassID is stored in the correct place in the database & This should be stored as a primary key and should not be displayed at any point in the system & 10A & Presence, Normal & The ClassID should be stored as a primary key in the Class database & & \\ \hline

3.002 & To test that the UserIDs are all stored in the correct place in the database & These should be stored as primary keys and should be displayed on account home screens & 0001 & Presence, Normal & The UserIDs should be stored as a primary key in the Class Database and in the User's database & & \\ \hline

3.003 & To test that the student's FirstNames are all stored in the correct place in the database & These should be stored and visible only by the user whose first name it is in their personal task database & John & Presence, Normal & The FirstNames should be stored in the User's database and visible together by the administrator or individually by the respective student & & \\ \hline

3.004 & To test that the student's LastNames are all stored in the correct place in the database & These should be stored and visible only by the user whose last name it is in their personal task database & Smith & Presence, Normal & The LastNames should be stored in the User's database and visible together by the administrator or individually by the respective student & & \\ \hline

3.005 & To test that the administrator's FirstName is stored in the correct place in the database & This should be stored and visible only by the administrator in the User's database & John & Presence, Normal & The FirstName should be stored in the User's database and visible only by the administrator & & \\ \hline

3.006 & To test that the administrator's LastName is stored in the correct place in the database & This should be stored and visible only by the administrator in the User's database & Smith & Presence, Normal & The LastName should be stored in the User's database and visible only by the administrator & & \\ \hline

3.007 & To test that all of the user's usernames are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user on their home screen or in their personal task database & User01 & Presence, Normal & These should be stored in the User's database and visible together by the adminsitrator or individually by the respective student & & \\ \hline

3.008 & To test that all of the user's passwords are stoed in the correct place in the database & These should be stored and only isible by the administrator & f8h4j7h & Presence, Normal & These should be stored in the User's database and visible only by the administrator behind their password & & \\ \hline

3.009 & To test that the TaskNames are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Sides Easy & Presence, Normal & These should be stored in the Tasks database and visible together by the administrator or individually by the respective student & & \\ \hline

3.010 & To test that the OverallPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 90\% & Presence, Normal & These should be stored in the Tasks database and the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.011 & To test that the IndividualPercentScores are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & 80\% & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.012 & To test that the Ratings are stored in the correct place in the database & These should be stored and visible together by the administrator or individually by the user who completed the task & Green & Presence, Normal & These should be stored in the Scores database, visible together by the administrator or individually by the respective student & & \\ \hline

3.013 & To test that the Feedback is stored in the correct place in the database & This should be stored in and visible together by the administrator or individually by the receiving student & Very good & Presence, Normal & This should be stored in the Tasks database and visible in whole by the administrator or individually by the receiving student & & \\ \hline

4.001 & To test that the algorithm for initially inputting names and IDs into the system works as intended & The system should accept the administrators inputs and save them to the right place in the database, before generating an account space for each user along with a username and password & Input John Smith, 10A into the system & Presence, Erroneous & names should be accepted and stored, and usernames and passwords should be generated for each of them & & \\ \hline

4.002 & To test that the algorithm for generating usernames works as intended & For each user's name input into the system, a username consisting of part or all of the first name followed by a number should be generated and stored with the name in the database & Generate something like 'John1' & Erroneous & The username should be generated and stored with the names of each user, and should be recognised upon logging in & & \\ \hline

4.003 & To test that the algorithm for generating passwords works as intended & For each user's name input into the system, a password consisting of a letter, number, letter pattern of 7 characters should be generated and stored with the name in the database & Generate something like 'd7h3g6j' & Erroneous & The password should be generated and stored with the names of each user, and should be recognised upon logging in & & \\ \hline

4.004 & To test that the algorithm for validating a login works as intended & The system should recognise and accept a valid username and password, or reject an invalid username and password as stored in the database & enter John1, d7g4h5j into the login line edits & Erroneous & Correct login details will be accepted and incorrect login details will be rejected & & \\ \hline

4.005 & To test that the algorithm for calculating an answer to a pythagoras theorem related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.006 & To test that the alternative algorithm for calculating an answer to a pythagoras theorem related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.007 & To test that the algorithm for calculating an answer to a 3D pythagoras related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$c^2$} = 5 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.008 & To test that the algorithm for calculating an answer to a sine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter sinA = 9 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.009 & To test that the algorithm for calculating an answer to a cosine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter cosineA = 7 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.010 & To test that the algorithm for calculating an answer to a tan rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter tanA = 8 into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.011 & To test that the algorithm for calculating an answer to a finding angles using the cosine rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter angle {$B = 45^o$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.012 & To test that the algorithm for calculating an answer to a finding angles using the tan rule related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter angle {$A = 30^0$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.013 & To test that the algorithm for calculating an answer to an inverted angles related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$invA = 20^0$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

4.014 & To test that the algorithm for calculating an answer to an area of triangle related question works as intended & The system should calculate an answer to the question given, and then compare it to the answer input by the user, finally outputting to the user whether or not they were correct & Enter {$area = 40cm^2$} into an input space & Erroneous, Boundary & The system should truthfully tell the user whether or not their answer was correct & & \\ \hline

5.001 & To test that the entire system adequately meets the clients requests & The system should be as the client specified & Go through every aspect of the system with the client to ensure that they are satisfied & Normal & The client should be satisfied with the system and allow it to remain the same, or request some changes & & \\ \hline

5.002 & To test that all login details are accepted and recognised by the system, and that each log in has its own personal account to maintain progress and privacy & The system should accept the log in given (if it should be valid) and give the user access to their personal account & Input all login details one at a time & Erroneous & The system should allow access for each log in provided with each account being private & & \\ \hline

5.003 & To test that the right amount of information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right amount of information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right amount of information should be stored in the database & & \\ \hline

5.004 & To test that the right information is saved to the database in every occasion where information is stored, in order to comply with the Data Protection Act 1998 & The right information should be stored when inputting names and a class into the system, and when submitting task results or feedback & Input test names into the system and complete all homework tasks, then give feedback from a temporary administrator account & Normal & The right information should be stored in the database & & \\ \hline

5.005 & To test that the local area network (LAN) connections across the system (for the database access) works as intended & Each account from any computer should be able to access all of the information which they have authorised access to from the server & Log in to multiple account on multiple computers in a LAN and attempt to store and read information & Normal & Each account and computer should be able to view their information from the server database and store new information & & \\ \hline

5.006 & To thoroughly ensure that the Data Protection Act 1998 is not breached at any point in the system, and that every measure is taken to prevent it from being breached in the future & The test will consist of checking every aspect of the database and algorithms involved in storing information & Check the algorithms work and store the right amount of the right information, and erase information after the legal minimum time frame & Normal & The system should comply with the Data Protection Act 1998 & & \\ \hline

\end{longtable}

\end{center}

\end{landscape}

\chapter{Analysis}

\section{Introduction}

\subsection{Client Identification}

The client for this project is Trish Marshall, who is a maths teacher at a secondary school who teaches all variations of maths in the current curriculum to GCSE students. She finds that the trigonometry resources she currently has are not as effective for teaching as she would like. The curriculum is changing from the 2016-17 academic year onwards, so the current resources are not up to date. She would like this program to enable her to use more up-to-date methods of teaching trigonometry to support the new curriculum.

\subsection{Define the current system}

The current system is a website called MyMaths which is used to teach all areas of maths from SAT level to A-Level. It is accessible online by anyone who signs up through a centre of education, for example a maths class. It consists of lessons which are interactive and give problems with solutions, followed by homework tasks which can be set by the teacher. The status of these tests and the results are recorded for each student in an online database accessible by the teacher. Homework can be set online and progress is recorded so the teacher can view the submissions from the students and take appropriate action following the deadline. The client also uses a smart board to demonstrate methods, and gives out textbooks to be read.

\subsection{Describe the problems}

The main problem with the current system is that it is not designed in a way that sufficiently challenges the student, for example, the button for the answer to a question will have the answer to the next question behind it, which presents a risk of the student accidentally double clicking and getting a lucky mark, or in some cases, they are just lazy and assume it's right. Furthermore, the sine, cosine and tan buttons are always put in that order, which minimises the amount if thinking a student will have to do to figure out which one is right for the problem they are solving, otherwise they get used to it already being in the same place for them.

In some of the examples used on the website MyMaths, the working out shows unnecessary stages which could sometimes put students off getting the right answer, such as when it gives an inaccurately rounded decimal number to represent a fraction when only the fraction is needed to find the solution.

MyMaths does not include a section with problems for any of the three rules, which limits the students ability to work out which rule will be needed, if all of the problems in one test use the same rule. Alongside this problem, MyMaths does not have a means of teaching the student how to know whether a problem will use the sine rule or the cosine rule. It also doesn't teach the sine, cosine and tan rules all together.

The current website is not designed to completely support the new GCSE maths curriculum which will be implemented starting from the 2016/2017 academic year, so will only be up to date for one more year, unless they change it.

The lessons on MyMaths teach the students how to calculate angles before how to calculate sides, which is a problem because in most cases you have to be able to use the inverse function of the rule to find the missing angle, for which you need to know which side is missing or which sides are in use, and what their values are.

The only way of inputting answers is by typing into boxes, which can become repetitive and boring.

The feedback system is not visual enough or convenient enough to use effectively, for example, there is no quick way of instructing a student to a detention or a meeting with their teacher.

\subsection{Section appendix}

The client interview was conducted by meeting in person and asking the questions directly, with access to the current system to show in detail precisely what the problems were, and what she needed improving upon. An up to date textbook was also present for referencing. The questionnaire was as follows:

\textbf{1. What do you require the proposed system to do?}

The system needs to interactively teach the updated GCSE trigonometry syllabus in a way that reduces the number of flaws compared to the current system. It needs to have a range of difficulty for the questions to teach a range of abilities, and it needs to cover every part of GCSE trigonometry in the most effective order. It must also provide an adequate amount of homework tasks to properly test their knowledge following the lessons, and these tasks should be submitted upon completion to be viewed by myself. It must also be quick and easy for me to view the student's progress with trigonometry, mostly by assessing their homework, with feedback and warnings being easy to give out when required. Finally, it must save a record of every task set for every student, and its completion progress and results, and these records need to be easily accessible. Every student and teacher needs their own individual login to avoid shared or inaccurate progression.

\textbf{2. What are the problems with the current way of doing things?}

Some of the biggest problems are with the way in which things are taught on the current system; in some cases, the method demonstrations are inaccurate and can mislead students away from the correct solution. Furthermore, the button for two consecutive answers are often in the same place, which can allow a student to get the right answer by assuming the answer is in that place after this problem occurring enough times prior to that task, or even by just double-clicking by accident. Answer buttons should be jumbled up, as this problem can be prevented, and can also solve the problem of the students not properly distinguishing the differences between the sine, cosine and tan rules, as they always appear in the same order and are taken for granted. The order in which lessons are taught is also problematic as sides should be taught before angles to allow for the student to learn how to use the inverse function of the rule which is often a key part of finding a missing angle, whereas MyMaths teaches angles before sides. Alongside this problem is that the sine, cosine and tan rules are all taught individually. If they were taught together it would give the student more experience in deciding which rule they actually have to use, rather than just being told. MyMaths also lacks a lesson on how to tell the whether the problem will use the sine or cosine rule. Lastly, the current system does not cover the new GCSE curriculum which will be implemented next year (2016-17), so more focus on the problem-solving aspect is required.

\textbf{To summarise: }

\begin{itemize}

\item The demonstrations used to show the methods used to find a solution are oftne inaccurate and can mislead students

\item These methods are also taught in an undesirable order which could potentially confuse students or cause them to miss out on important information

\item The positioning of some multiple choice buttons causes students to be able to memorise or guess the correct answer, or double click by accident and get two correct answers without any thought behind them. They need to be randomised so they move around

\item Another problem cuased by the button positioning is that they make it harder to distinguish the differences between the sin, cosine and tan rules as they do not provide a challenge

\item The order in which lessons are taught is not effective; students end up attempting problems which require knowledge that they have not yet acquired e.g doing angles before knowing the inverse function rules

\item The current system does not cover the new 2016-17 GCSE Maths curriculum

\item \textbf{See section \textit{1.2.1 - Input Forms, Output Forms, Report Formats} for visual details}

\end{itemize}

\textbf{3. What data or information is recorded in the current system?}

The currently recorded fields in the homework set table are: Name (Text), Task or activity (Text), Type (Text), Created (Date), Completed (Text), Start (Date, Foreign Key), Due (Date), Feedback (Text, Foreign Key).

The currently recorded fields in the results table are: Level (Integer), Topic (Text), Task name (Text), Number of tries (Integer), Start (Date, Foreign Key), Last tried (Date), Rating (Integer), Percentage (Real), Question number percentage (Real), Feedback (Text, Foreign Key).

The currently recorded field in the administration table is: Students belong to these classes (Text).

\textbf{4. Is the any data or information you require to be recorded in the proposed system? If so, how much?}

I require the following data to be recorded in the proposed system: User ID (Integer, Primary Key), First name (Text), Surname (Text), Overall percentage score (Real), Red/Amber/Green face (Graphic, Text, Foreign Key), Feedback (Text).

Also, in a separate table I need a field for each question to view the individual percentage so I know exactly what the student needs to work on, along with the feedback, red/amber/green face and whether or not the student needs to see me after class.

Example: Question 1 percentage (Real), Question 2 percentage (Real), Question 3 - x (Real), Red/Amber/Green face (Graphic, Text, Foreign Key), Appointment with teacher? (Text, Y/N), Time (Time, only if Appointment with teacher is Yes)

\textbf{5. If there is data, how frequently will it need to be updated?}

The data will need to be updated every time a student submits a finished or amended homework assignment.

\textbf{6. Will new records need to be added and old ones deleted?}

The old records will not need to be deleted, and new ones will be added as described before.

\textbf{7. How important is the data or information that is recorded?}

It is very important as it will allow both myself and the students to track progress and know what needs improving upon, and allows the me to record evidence of the amount of work and revision each student has done. However progress will not be actively monitored, only checked following deadlines.

\textbf{8. What processes or functions are performed by the current system?}

The current system saves data to the database which can be viewed by both the students and myself. It uses an interactive GUI so that the user can navigate between many pages and areas of the website, and practice tasks and homework tasks use text boxes for the user to fill in to submit an answer.

\textbf{9. What processes and functions are to be performed by the new system?}

The system must also save data to a database, but it must be much easier to access, navigate and use generally. The program will be navigable using a GUI, and lessons, homework tasks and the database will be accessible. The lessons and homework will be interactive and use text boxes for answers, as well as drag and drop graphics and boxes for showing working out.

\textbf{10. What special algorithms do these processes use?}

To save data to the database, the current system uses a read/write algorithm, writing to store, and reading to view in the database. The test questions themselves will use selection statements to determine if a submitted answer is the same as the expected correct answer, stored as a fixed variable. To log on to the website, the program uses a validation algorithm to check the user name and password is correct, and uses error exception algorithms to ask the user for the correct input if necessary. All the teaching tasks use mathematical formula algorithms.

\textbf{11. Which processes should be executed manually?}

Clicking the buttons, submitting answers, logging on, navigating the GUI.

\textbf{12. What are the inputs to the current system?}

User name and password, cursor clicks for navigation and selecting text boxes, keyboards for typing answers, cursor clicks for submitting answers.

\textbf{13. What inputs will be required for the proposed system?}

Cursor clicks for navigation, selecting text boxes, submitting answers and dragging graphics, and keyboard for typing answers, user name and password.

\textbf{14. What are the outputs from the current system?}

It outputs an error handling exception if the user name or password is incorrect, a message appears asking you to re-enter the correct inputs. It outputs a tick or a cross beside submitted answers, and graphics to represent how well you did on the database, such as a red or green face. It outputs your score percentage, your rating, and your feedback from the teacher.

\textbf{15. What outputs will be required from the proposed system?}

I want the system to display methods of working out next to an incorrect answer submission, rather than just a big cross. If a student inputs the wrong data type, it will produce an error exception hint saying "Please input an integer" or "Please input a decimal" or "Please input an angle". It will give a hint in the same way if a student gets the wrong answer using the correct data type, then when they get it wrong the second time it will display the answer as well as the method of working out, and give them marks for their working out accordingly rather than the big cross. The data stored in the database will need to be easily readable and the graphics should be appropriate.

\textbf{16. What computing resources does the client possess?}

The centre has many computer rooms, a smart board in the classroom, and some students can gain permission to bring in their own devices.

\textbf{17. Are there any security issues?}

I may have to gain permission from IT if you need to install your program on the school computers for testing on a class, but this shouldn't be an issue if you are trustworthy and your program can be checked.

\textbf{18. Should there be restricted access to particular areas?}

Students should not be allowed to view the progress of the whole class, only teachers. Teachers should also have exclusive access to manually setting homework and sending personal messages or warnings to the students.

\textbf{19. how are exceptions and errors handled in the current system?}

When logging in, if the user name or password is invalid it asks you to input the correct values. If an answer is wrong, or uses the wrong data type, it just takes off the marks and moves on.

\textbf{20. What errors and exceptions should be reported in the proposed system?}

If a user name or password is incorrect, a small box should pop up asking for the correct value. If a wrong data type is used to answer a question, a small box should pop up asking for the correct data type (e.g. "Please input an integer, decimal or angle") and will not dock marks or remaining attempts. If a student gets it wrong on their first attempt, a small box should pop up giving a hint, and dock one of the two attempts. If they get it wrong again, the correct answer along with the full method for working out will appear next to the box.

\textbf{21. How should they be reported?}

Small windows temporarily popping up on screen with messages, until the OK button or the cross in the top right is clicked.

\textbf{22. Are there any constraints on hardware, software, data, methods of working, cost, time, etc?}

We have access to plenty of computers, and our system is powerful enough to handle easily enough data. We don't have a deadline for you.

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/signature}

\end{figure}

\section{Investigation}

\subsection{The current system}

\subsubsection{Data sources and destinations}

\textit{User:Student, Administrator:Teacher}

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|}

\hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

User & Username & The user manually inputs their username using a keyboard & john\\_smith\\_b, string & Certification Path Verification Algorithm \\ \hline

User & Password & The user manually inputs their password using a keyboard & SFC, string & Certification Path Verification Algorithm \\ \hline

Administrator & First name & Is added manually when a class is being set up & John, string & Database \\ \hline

Administrator & Surname & Is added manually when a class is being set up & Smith, string & Database \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|}

\hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

User, Administrator & Task or Activity & Each task name is hard-coded & Trigonometry Test 1, string & Administrator, Database, User \\ \hline

System & Type & Type of task is hard-coded & Angles, string & Administrator, Database, User \\ \hline

Administrator & Created & The date the administrator sets the task is automatically saved when set & 14/10/15, date & Database \\ \hline

User & Completed & The date when the student completes the task is saved automatically & Yes, boolean & Administrator, Database \\ \hline

User & Start Date & The date when the student first attempts the task is saved automatically & 16/10/15, date & Administrator, Database \\ \hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

Administrator & Due & The date when the task is due is set manually by the administrator & 21/10/15, date & Database, User \\ \hline

Administrator & Feedback & Text typed by the administrator and saved manually & Good work, string & Database, User \\ \hline

System & Level & Level is hard-coded & 8, integer & Database, User \\ \hline

System & Topic & Topic is hard-coded & Tan Rule, string & Database, User \\ \hline

System & Task Name & Task name is hard-coded & Tan Rule Practice, string & Database, User \\ \hline

System & Number of Tries & Number of tries is automatically recorded, incrementing the total each time the task is opened and marked & 3, integer & Administrator, Database, User \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|}

\hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

User & Last Tried & The date of the most recent attempt is recorded and the last attempt is overwritten & 17/10/15, date & Administrator, Database, User \\ \hline

System & Rating & Depending on the student's score, a coloured face (red, amber or green) is automatically saved when a task is submitted & Amber face graphic, blob & Administrator, Database, User \\ \hline

User & Percentage & calculated from the amount of marks a student gets for each task & 85\%, real & Administrator, Database, User \\ \hline

User & Question Percentage & Percentage calculated from the amount of marks awarded to the student for each individual question & 90\%, real & Administrator, Database, User \\ \hline

Administrator & Students Belong To These Classes & All students added when setting up a class are stored in a separate table in the database & John Smith, string & Database \\ \hline

\end{tabular}

\end{center}

\subsubsection{Algorithms}

\textbf{Manual User names input program:}

\begin{algorithm}[H]

\caption{Inputting the names of the class into the list.}

\begin{algorithmic}[1]

%The administrator (teacher) enters the names of all the users (students).

\RECEIVE{$"Please\ enter\ the \ name \ of \ the \ user \ you \ would \ like \ to \ add"$}

%Each name is appended to the list

\SET{$class.append(name)$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Checking whether to add another user or save current list of users.}

\begin{algorithmic}[1]

%This is run after every name is input to check with the admin if they want to add another name or stop.

\RECEIVE{$"Would \ you \ like \ to \ add \ another \ user? \ (Y/N) "$}

\If{$exit.upper() = Y$}

%If they do, this runs the get\_name function again.

\SET{$get\\_ name$}

\ElsIf{$exit.upper() = N$}

%If they don't, it saves all the names currently in the list to the database.

\SET{$save\\_names()$}

\Else{}

%Makes sure they choose yes or no.

\SEND{"Please \ enter \ a \ valid \ value"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Saving the name list to the database.}

\begin{algorithmic}[1]

\SET{$with \ open("username.txt", mode = "r") as \ file$}

%This saves every name in the list to the file

\For{$name$}{$class\\_list$}

\SET{$file.write(name)$}

\EndFor

\end{algorithmic}

\end{algorithm}

\textbf{Username and password generator:}

\begin{algorithm}[H]

\caption{Generating the username from the list of names.}

\begin{algorithmic}[1]

%This for loop sets every users username as their name

\For{$name$}{$usernames$}

\SET{$username$}{$name$}

\SET{$usernames.append(username)$}

\EndFor

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Generating the password for each username.}

\begin{algorithmic}[1]

%This loop generates 7 different characters and combines them to form a password.

\For{$username$}{$usernames$}

\For{$count$}{$7$}

\SET{$password[count]$}{$random$}

\SET{$count$}{$0$}

\EndFor

\For{$count$}{$7$}

\SET{$password$}{$password + password[count]$}

\SET{$count += 1$}

\EndFor

\SET{$passwords.append(password)$}

\EndFor

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Combines the username and password and saves them together as a complete login.}

\begin{algorithmic}[1]

%This loop appends each username followed by a password, so the password follows the username in the list which is useful for validation.

\SET{$count$}{$0$}

\For{$username$}{$usernames$}

\SET{$login\\_user$}{$usernames[count]$}

\SET{$login\\_pass$}{$passwords[count]$}

\SET{$logins.append(login\\_user)$}

\SET{$logins.append(login\\_pass)$}

` \SET{$count += 1$}

\EndFor

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Saves the username and password as combined logins to the database.}

\begin{algorithmic}[1]

\SET{$with \ open("logins.txt", mode = "w") as \ logins\\_$}

\For{$login$}{$logins$}

\SET{$logins\\_.write(login)$}

\SET{$logins\\_.write("\backslash \ n")$}

\EndFor

\end{algorithmic}

\end{algorithm}

\textbf{Validates the username and password when the user tries to log in:}

\begin{algorithm}[H]

\caption{Takes the logins from the database and adds them to a list from which they can be validated.}

\begin{algorithmic}[1]

\SET{$with \ open("logins.txt", mode = "r") as \ logins$}

\For{$name$}{$logins$}

\SET{$list\\_.append(name)$}

\EndFor

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Asks the user for their username.}

\begin{algorithmic}

\RECEIVE{"Please enter your username"}

\SET{$return \ username$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Asks the user for their password.}

\begin{algorithmic}[1]

\RECEIVE{"Please enter your password"}

\SET{$return \ password$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Validates the given username and password.}

\begin{algorithmic}[1]

\SET{$count$}{$0$}

\SET{$found$}{$False$}

\While{$found = False$}{$and \ count < len(list\\_)$}

\If{$list\\_[count] = str(username) \ and \ list\\_[count + 2] = str(password)$}

\SEND("Accepted")

\SET{$found$}{$True$}

\SET{$return \ found$}

\Else{}

\SEND{"Not accepted"}

\SET{$call \ Validation$}

\SET{$count \ += \ 1$}

\EndIf

\EndWhile

\end{algorithmic}

\end{algorithm}

\textbf{Pythagoras algorithms:}

\begin{algorithm}[H]

\caption{Checks if the user's solution for the {$a^2 + b^2 = c^2$} is correct.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$side\\_a$}{$x$}

\SET{$side\\_b$}{$x$}

\SET{$side\\_c$}{$\sqrt{side\\_a^2 + side\\_b^2}$}

\SEND{$"Here \ is \ a \ right \ angled \ triangle. \ The \ length \ of \ side \ a \ is \ x$}

\SET{$centimetres, \ and \ the \ length \ of \ side \ b \ is \ x \ centimetres.$}

\SET{$Please \ calculate \ the \ length \ of \ side \ c"$}

\RECEIVE{$length$}{$"Please \ input \ the \ length \ of \ side \ c: "$}

\If {$length = side\\_c$}

\SEND{"+ 1 \ mark"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Alternative algorithm:}

\begin{algorithm}[H]

\caption{Same question and solution with a differently arranged formula.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$side\\_a$}{$x$}

\SET{$side\\_b$}{$x$}

\SET{$side\\_c^2$}{$side\\_a^2 + side\\_b^2$}

\SET{$side\\_c$}{$\sqrt{side\\_c^2}$}

\SEND{$"Here \ is \ a \ right \ angled \ triangle. \ The \ length \ of \ side \ a \ is \ x$}

\SET{$centimetres, \ and \ the \ length \ of \ side \ b \ is \ x \ centimetres.$}

\SET{$Please \ calculate \ the \ length \ of \ side \ c"$}

\RECEIVE{$length$}{$"Please \ input \ the \ length \ of \ side \ c: "$}

\If{$length = side\\_c$}

\SEND{"+ x \ marks"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{3D Pythagoras algorithm:}

\begin{algorithm}[H]

\caption{Similar algorithm, but continues to check the user's solution for a 3D pythagoras problem.}

\begin{algorithmic}[1]

\SET{$total\\_marks$}{$0$}

\SET{$left\\_side\\_a$}{$x$}

\SET{$middle\\_side\\_a$}{$x$}

\SET{$right\\_side\\_a$}{$x$}

\SET{$inside\\_side\\_a$}{$\sqrt{left\\_side\\_a^2 + middle\\_side\\_a^2}$}

\SET{$inside\\_side\\_b$}{$\sqrt{right\\_side\\_a^2 + inside\\_side\\_a^2}$}

\SEND{$"A \ magician \ stores \ his \ wand \ in \ a \ box.$}

\SET{$The \ box \ is \ xcm \ by \ xcm \ by \ xcm.$}

\SET{$The \ wand \ only \ just \ fits \ in \ wedged \ against \ opposite \ corners."$}

\RECEIVE{$length$}{$"How \ long \ is \ the \ wand?"$}

\If{$length = inside\\_side\\_b$}

\SEND{"+ \ x \ marks"}

\SET{$total\\_marks \ += \ x$}

\Else{}

\SEND{"The \ answer \ is \ x"}

\EndIf

\end{algorithmic}

\end{algorithm}

\textbf{Trigonometry Algorithms:}

\begin{algorithm}[H]

\caption{Sine rule.}

\begin{algorithmic}[1]

\SET{$sinA$}{$\frac{opposite}{hypotenuse}$}{$\frac{a}{h}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Cosine rule.}

\begin{algorithmic}[1]

\SET{$cosA$}{$\frac{adjacent}{hypotenuse}$}{$\frac{b}{h}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Tan rule.}

\begin{algorithmic}[1]

\SET{$tanA$}{$\frac{opposite}{adjacent}$}{$\frac{a}{b}$}

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The sine formula in use.}

\begin{algorithmic}[1]

\SET{$\frac{A}{sinA}$}{$\frac{B}{sinB}$}

\If{$\frac{A}{sinA} = \frac{B}{sinB}$}

\SEND{"Your solution is correct"}

\Else{}

\SEND{"Your solution is not correct"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The cosine formula in use.}

\begin{algorithmic}[1]

\SET{$a^2$}{$b^2 + c^2 \ - \ 2bc \ cosA$}

\RECEIVE{$side\\_a$}{$"Please \ input \ the \ length \ of \ side \ a: "$}

\If{$side\\_a = {b^2 + c^2 - 2bc \ cosA}$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{The formula for finding angles in scalene triangles using the cosine rule.}

\begin{algorithmic}[1]

\SET{$cosA \ b^2 + c^2 - \frac{a^2}{2bc}$}

\SET{$C$}{$inv \ cos\frac{adjacent}{hypotenuse}$}

\RECEIVE{$angle\\_c$}{$"Please \ input \ the \ size \ of \ angle \ C:"$}

\If{$angle\\_c = {inv \ cos\frac{adjacent}{hypotenuse}}$}

\SEND{"Correct"}

\Else{}

\SEND{"Correct"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Formula for finding the area of a scalene triangle using the sine rule.}

\begin{algorithmic}[1]

\SET{$area$}{$\frac{1}{2} \ ab \ sinC$}

\RECEIVE{$area\\_1$}{"Please \ input \ the \ area \ of \ this \ scalene \ triangle:"}

\If{$area\\_1 = \frac{1}{2} \ ab \ sinC$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\begin{algorithm}[H]

\caption{Formula for finding an angle using the tan rule.}

\begin{algorithmic}[1]

\SET{$tanA$}{$\frac{10}{15}$}

\SET{$\frac{10}{15}$}{$ \ $}{$0.67$}

\SET{$tan^-1(0.67)$}{$33.82^o$}

\SET{$tanA$}{$33.82^o$}

\RECEIVE{$tan\\_a$}{"Please \ input \ the \ size \ of \ the \ angle \ using \ the \ tan \ rule:"}

\If{$tan\\_a = {33.82^o}$}

\SEND{"Correct"}

\Else{}

\SEND{"Incorrect"}

\EndIf

\end{algorithmic}

\end{algorithm}

\subsubsection{Data flow diagram}

\textbf{Key: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowkey.png}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram1.png}

\caption{The process of how the administrator puts their classes names into the system} \label{fig:print\_function\_result}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram2.png}

\caption{The system validating the users login using the stored login information} \label{fig:print\_function\_result}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram3.png}

\caption{The entire process of a user using the system and the results of a homework being recorded, including the teacher setting the task} \label{fig:print\_function\_result}

\end{figure}

\subsubsection{Input Forms, Output Forms, Report Formats}

There are no physical forms that are used for input or output; the class names are added into the system manually by the teacher, copying from a register, and output forms always come in the form of an e-mail, or a meeting with parents. Specifically, if a student does not complete homework to a sufficient standard, the teacher manually e-mails the parents personally or calls them in for a meeting. No action is taken by the system beyond informing the teacher of the student's progress.

\textbf{An example register:}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/register.png}

\caption{A screenshot of what the register used to input the names correctly might look like} \label{fig:print\_function\_result}

\end{figure}

\textbf{An example e-mail to a parent:}

Dear Sir/Madam

I am writing to inform you that your son/daughter has unfortunately not been completing their homework to the standard that is expected of them.

They have many outstanding tasks to complete, and if they do not complete them very soon I am afraid I will have to call you in for a meeting.

Yours faithfully, \\

Ms Teacher

As for computerised input forms, these include the login screen, the lesson/task select window, and the generic text boxes for inputting answers. There are currently no error messages, so the only other output form is displaying the results following a submission, which are then saved to the database.

\textbf{Login screen:}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/login\_screen.png}

\end{figure}

\textbf{Task select:}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/task\_select.png}

\end{figure}

\textbf{Generic user inputs:}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/user\_input.png}

\end{figure}

\textbf{Results viewing:}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/results.png}

\end{figure}

\subsection{The proposed system}

\subsubsection{Data sources and destinations}

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|}

\hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

Administrator & User ID & The user ID is automatically generated when a class is added to the system & 0001, integer & Database, User \\ \hline

Administrator & Password & The user's password is automatically generated when a UserID is generated & d4g5s1g & Database, User \\ \hline

Administrator & First name & The first name of each user is manually input by the administrator & John, string & Database, User \\ \hline

Administrator & Surname & The surname of each user is manually input by the administrator & Smith, string & Database, User \\ \hline

User & Overall percentage score & The percentage of the questions in a task which the user got correct is recorded automatically upon submission by the system & 70\%, real & Administrator, Database, User \\ \hline

User & Individual question percentage score & The percentage score for each individual question is recorded automatically upon task submission & 50\%, real & Administrator, Database, User \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|}

\hline

\textbf{Source} & \textbf{Data} & \textbf{Process} & \textbf{Example Data} & \textbf{Destination} \\ \hline

System & Rating & A red, amber or green face is recorded for each submitted assignment to represent whether or not the user did well enough & Green face graphic, blob & Administrator, Database, User \\ \hline

Administrator & Feedback & The teacher gives feedback to the user manually & Well done, string & Database, User \\ \hline

Administrator & See after class & The administrator can manually set parameters so that the system knows to inform a student if they need to see the teacher after their next lesson e.g Below 50\% & Yes, boolean & User \\ \hline

Administrator & Time of appointment & The time for a meeting with the user if they need to see the teacher, input manually be the teacher & 11:00AM, time & Administrator, User \\ \hline

\end{tabular}

\end{center}

\subsubsection{Data flow diagram}

\textbf{Key: }

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowkey.png}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram1.png}

\caption{The process of how the administrator puts their classes names into the system} \label{fig:print\_function\_result}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram2.png}

\caption{The system validating the users login using the stored login information} \label{fig:print\_function\_result}

\end{figure}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/dataflowdiagram4.png}

\caption{The teacher sets the task, then the student proceeds to complete it, and their progress is recorded} \label{fig:print\_function\_result}

\end{figure}

\subsubsection{Data dictionary}

\begin{center}

\begin{tabular}{|p{3.4cm}|p{1.2cm}|p{2cm}|p{2cm}|p{2cm}|p{3.5cm}|}

\hline

\textbf{Name} & \textbf{Data Type} & \textbf{Length} & \textbf{Validation} & \textbf{Example Data} & \textbf{Comment} \\ \hline

UserID & Integer & 4 bits & 0001 to 9999 & 1546 & Unique to each user \\ \hline

Password & String and integers & 7 characters & letter followed by number followed by letter & f7h3j5f & The password generator uses mixed data types to avoid inappropriate passwords \\ \hline

FirstName & String & 15 characters & First letter upper case, rest lower case & John & Unique to each user, but could be shared by some \\ \hline

Surname & String & 25 characters & First letter upper case, rest lower case & Smith & Unique to each user, but could be shared by some \\ \hline

TaskName & String & 25 characters & & Trigonometry 2 & Hard-coded into the system \\ \hline

OverallPercentScore & Real & 3 characters & in range 0 - 100 & 76.5\% & The percentage of marks obtained in a test, decimal points allowed \\ \hline

IndividualPercentScore & Real & 3 characters & in range 0 - 100 & 45.5\% & The percentage of marks for an individual question, field will only appear in separate table for individual tasks \\ \hline

Rating & Blob & 64 kilobytes & \ & Green face graphic & Green, amber or red face graphic \\ \hline

Feedback & String & 500 characters & \ & Good work & This can consist of any characters as it is a personal message \\ \hline

AppointmentTime & Time & 5 characters & 24 hour format & 13:35 & Only relevant if the user has a true SeeAfterClass variable, set automatically bsed on the administrator's timetable but can be changed if necessary\\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{3.4cm}|p{1.2cm}|p{2cm}|p{2cm}|p{2cm}|p{3.5cm}|}

\hline

\textbf{Name} & \textbf{Data Type} & \textbf{Length} & \textbf{Validation} & \textbf{Example Data} & \textbf{Comment} \\ \hline

SeeAfterClass & Boolean & 3 characters & Yes or No & Yes & If the user doesn't achieve a sufficient score, this variable will become true and alert the user \\ \hline

ErrorMessage1 & String & 50 characters & & Sorry, the name cannot have integers & An error message if the wrong data type is used to input a name \\ \hline

ErrorMessage2 & String & 50 characters & & Sorry, that is not a valid login & Tells the user if they have input the wrong username or password \\ \hline

ErrorMessage3 & String & 50 characters & & Please input a decimal, not an integer & Tells the user if their incorrect answer is the wrong data type \\ \hline

ErrorMessage4 & String & 50 characters & & That is incorrect, try one more time & Tells the user that their answer is incorrect and gives them one more attempt \\ \hline

CorrectAnswer & Integer, Real, String & 5 characters & Must be a decimal or whole number, or text & 25.5cm & Gives the user the correct answer if they get the question wrong too many times \\ \hline

Login data & String & 25 characters & & johnsmith1, f8j4h6k & The login information saved in the system to be loaded and checked with the user's inputs \\ \hline

Task data & String & 100 characters & & Trigonometry - sin rule, Level 7, Trigonometry & Contains all the information about the task, what difficulty it is, what type it is etc. \\ \hline

Set answers & Integer, Real, String & 5 characters & Must be a decimal or whole number, or text if in a text box & 45{$^o$} & Contains all the set answers for some of the tasks \\ \hline

Calculated answers & Integer, Real & 5 characters & Must be the same solution as the algorithm & 29.8 & Contains algorithms which find and validate the solution for randomly generated tasks \\ \hline

\end{tabular}

\end{center}

\subsubsection{Volumetrics}

There are between 25 and 30 students in each class in years 10 and 11, and there are about 10 classes between the two years at the client's school. This equates to 300 students, so 300 records will be needed across the first year. These will not be needed once a year group moves on, so 150 will be overwritten by 150 more, meaning no expansion will be necessary. Obviously there won't be exactly 150 in each year, so there will be a boundary of 30 more or less per year. It is highly unlikely that more would be needed. There will be a total of 360 record spaces each year, half of which will be either overwritten or just deleted following a decline in numbers. Each of these 360 students will have a record with personal details, and then will also have space for 30 more records, easily enough for the amount of tasks they could complete. 360 \* 20 is 7600 task records, added to the personal records, making the total number of records in the master database 7960.

Each user's personal record will look like this:

\begin{center}

\begin{tabular}{|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|p{2.5cm}|} \hline

\textbf{First Name} & \textbf{Surname} & \textbf{UserID} & \text{Password} \\ \hline

John & Smith & john\\_smith\\_1 & s3g6h3f \\ \hline

\end{tabular}

\end{center}

Each user's task record will look like this:

\begin{center}

\begin{tabular}{|p{4.2cm}|p{2.5cm}|p{3.7cm}|p{4.1cm}|} \hline

\textbf{UserID} & \textbf{TaskName} & \textbf{OverallPercentScore} & \textbf{IndividualPercentScore1} \\ \hline

john\\_smith\\_1 & Trigonometry Higher & 75\% & 100\% \\ \hline

\textbf{IndividualPercentScore2} & \textbf{Rating} & \textbf{Feedback} \\ \hline

50\% & [Green face] & Very good \\ \hline

\end{tabular}

\end{center}

SeeAfterClass and TimeOfAppointment will not need to be recorded, only the user and the teacher temporarily need to know these.

If FirstName, Surname and UserID allow for up to 15 characters, and the Password is 7 characters, then 52 bytes will be needed for each user's personal record. 52 x 300 = 15600 bytes, or roughly 15.6 kilobytes for all of the personal records.

\textbf{UserID} + \textbf{Password} will be 30 bytes, and the \textbf{Feedback} box will have space for 500 characters, or 500 bytes. 500 + 30 = 530 bytes, or roughly 0.53 kilobytes. Three faces will be stored by the system, and the database will store an index which will find the face and display it when the display is loaded. 1 character each, for up to 7600 records, totalling 7.6 kilobytes. 0.53 kilobytes x 7600 is 4028 kilobytes, or 4.02 megabytes roughly. The values for \textbf{OverallPercentScore} and \textbf{IndividualPercentScore} will be calculated, so won't take up permanent memory space.

4.028 megabytes + 15.6 kilobytes = approximately 4.043 megabytes, which is the estimated size of the proposed database.

Therefore it will fit on a standard USB stick, and it should be possible to send by e-mail.

\section{Objectives}

\subsection{General Objectives}

\begin{itemize}

\item To have an interactive and easily navigable graphical user interface, applying a suitable colour scheme and layout

\item To make the database concise and adjustable

\item To create various lessons, with a wide range of challenges, which effectively teach students how to do trigonometry and Pythagoras

\item To create tasks which are relevant to the lessons to be completed by the user in order to test their progress

\item To allow this progress to be recorded in an easily accessible and readable database

\item To incorporate algorithms which find and/or check the solution given by the user accurately and give clear and easy to read outputs to correspond with said inputs

\item To have some access restrictions to certain levels of user

\item To make the program accessible only from various computers with permissions

\end{itemize}

\subsection{Specific Objectives}

\begin{itemize}

\item To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\item To include the following topics: Trigonometry, Pythagoras, 3D Trigonometry, 3D Pythagoras

\item To include a range of difficulty levels, which can challenge every user's level of ability

\item Use drag and drop, text boxes and drop down menus for inputs

\item To include interactive 2D graphics which give a clearer idea of the method being shown to the user

\item To have a database which can be accessed by different computers online

\item Use a specific, continuous and attractive colour scheme in every window

\item To have medium sized, highly visible icons

\item To have all input buttons randomised to avoid double clicking and guessing from memory

\item To have small error message windows which pop up and disappear on a timer

\item To include images and shapes which contrast the colour scheme so they are visible and readable

\end{itemize}

\subsection{Core Objectives}

\begin{itemize}

\item To create a teaching program that uses the new GCSE Maths curriculum, as lots of resources will soon be out of date

\item To make the database easy to access and easy to read

\item To include primarily trigonometry based topics, such as how to use the sine, cosine and tan rules

\item To include an initial, moderate difficulty in order to cater for a majority of students

\item To make the database functional and able to store the requested details

\end{itemize}

\subsection{Other Objectives}

\begin{itemize}

\item To position buttons, text boxes and drag and drop boxes in within the layout of the graphical user interface in such a way that cheating and lucky guessing can be minimised

\item To make the database adjustable if necessary

\item Use a more interesting range of input types like drawing boxes rather than just clicking and typing

\item To include a wider range of difficulties to challenge every student on the right level for them

\item To include a wider range of topics such as pythagoras, then 3D trigonometry and 3D pythagoras

\end{itemize}

\section{ER Diagrams and Descriptions}

\subsection{ER Diagram}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/er\_diagram.png}

\label{fig:print\_function\_result}

\end{figure}

\subsection{Entity Descriptions}

\begin{center}

\begin{tabular}{|p{3cm}|p{4cm}|p{4cm}|p{3cm}|} \hline

\textbf{Entity} & \textbf{Description} & \textbf{Attributes} & \textbf{Key} \\ \hline

Student & Each record shows how many results a student receives, and which class they are in & \textbf{Student}(\underline{StudentID}, \underline{ClassID}) & Primary Key(StudentID, ClassID) \\ \hline

Class & Shows which students are in the class and the teacher of that class, and how much homework the class has been set & \textbf{Class}(\underline{ClassID}, TeacherID) & Primary Key(ClassID) \\ \hline

Homework & Shows who each homework has been set for, and the results if the homework has been completed & \textbf{Homework}(\underline{HomeworkID}, HomeworkSet, HomeworkResults) & Primary Key(HomeworkID) \\ \hline

HomeworkSet & Shows who has been set the homework, when, and how many times/how long for & \textbf{HomeworkSet}(\underline{ClassID}, \underline{HomeworkID}, DueDate, SetDate) & Primary Key(ClassID, HomeworkID) \\ \hline

HomeworkResults & This record contains all of the results for a completed homework & \textbf{HomeworkResults} (\underline{HomeworkID}, \underline{StudentID}, \underline{QuestionID}, Rating, CompletedDate) & Primary Key(HomeworkID, StudentID, QuestionID) \\ \hline

Question & Shows how many questions were in a homework, and the percentage which are correct in each students results & \textbf{Question}(\underline{HomeworkID}, \underline{QuestionID}, QuestionText, Choice1, Choice2, ChoiceX, CorrectAnswer, TypeOfQuestion) & Primary Key(HomeworkID, QuestionID) \\ \hline

\end{tabular}

\end{center}

\section{Object Analysis}

\subsection{Object Listing}

\begin{itemize}

\item Course

\item Teacher

\item Student

\item Trigonometry Section

\item Pythagoras Section

\item Trigonometry Lesson

\item Trigonometry Homework

\item Pythagoras Lesson

\item Pythagoras Homework

\item QuestionType

\item TextQuestion

\item DragAndDropQuestion

\item ImageQuestion

\end{itemize}

\subsection{Relationship diagrams}

\begin{figure}[H]

\includegraphics[width=\textwidth]{C:/Users/Jordan/git/COMP4Coursework2/Analysis/objectrelationships.png}

\caption{The relationships between each of the objects in the proposed system} \label{fig:print\_function\_result}

\end{figure}

\subsection{Class definitions}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Course \\ \hline

Title \\

Subject \\ \hline

AddTitle \\

EditTitle \\

AddSubject \\

EditSubject \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Teacher \\ \hline

Surname \\

Title \\

Subject \\ \hline

AddSurname \\

EditSurname \\

AddTitle \\

EditTitle \\

AddSubject \\

EditSubject \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Student \\ \hline

FirstName \\

Surname \\

UserID \\

Password \\ \hline

AddFirstName \\

EditFirstName \\

AddSurname \\

EditSurname \\

AddUserID \\

EditUserID \\

AddPassword \\

EditPassword \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Section \\ \hline

TrigonometryLesson \\

TrigonometryHomework \\ \hline

PresentTrigonometryLesson \\

SetTrigonometryHomework \\

MarkTrigonometryHomework \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Section \\ \hline

PythagorasLesson \\

PythagorasHomework \\ \hline

PresentPythagorasLesson \\

SetPythagorasHomework \\

MarkPythagorasHomework \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Lesson \\ \hline

Examples \\

Questions \\ \hline

DisplayExamples \\

GiveExampleAnswers \\

GiveQuestionAnswers \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Trigonometry Homework \\ \hline

Questions \\

Answers \\ \hline

SetQuestions \\

CheckAnswers \\

DisplayCorrectAnswers \\

OutputCorrectMessage \\

OutputDataTypeErrorMessage \\

SubmitScore \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Lesson \\ \hline

Examples \\

Questions \\ \hline

DisplayExamples \\

GiveExampleAnswers \\

GiveQuestionAnswers \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

Pythagoras Homework \\ \hline

Questions \\

Answers \\ \hline

SetQuestions \\

CheckAnswers \\

DisplayCorrectAnswers \\

OutputCorrectMessage \\

OutputDataTypeErrorMessage \\

SubmitScore \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

QuestionType \\ \hline

QuestionType \\

QuestionInput \\

QuestionOutput \\ \hline

InputSolution \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

TextQuestion \\ \hline

QuestionInputText \\

QuestionOutput \\ \hline

InputSolutionText \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

DragAndDropQuestion \\ \hline

QuestionDragImage \\

QuestionOutput \\ \hline

InputDragImage \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\begin{center}

\begin{tabular}{|p{5cm}|} \hline

ImageQuestion \\ \hline

QuestionSelectButton \\

QuestionOutput \\ \hline

InputSolutionButton \\

CheckSolution \\

OutputError \\ \hline

\end{tabular}

\end{center}

\section{Constraints}

\subsection{Hardware}

The centre which my client works for have multiple classrooms with standard computers in, which are powerful enough to run a program of the proposed programs size. All of the hardware components are compatible with Python, and any other software required to use a Python-made program. There are enough computers at the facility for everyone to use them together. There are essentially no hardware constraints, and no additional hardware will be required.

\subsection{Software}

All of the computers at the center have the appropriate software installed to be able to view graphics and access the internet, so they can be used to record the user's progression in an online database. They do not have Python, but that can easily be installed if necessary. They will be able to access the download and/or installation for my program easily too. Otherwise there are no constraints.

\subsection{Time}

The deadline for this project officially is set by the teacher and then the exam board; the client does not mind when it is done, within reason, but would prefer if it was ready for the updated curriculum, which is after the teacher deadline anyway, so won't be a problem.

\subsection{User Knowledge}

The users have had sufficient experience using computers and navigating other websites and programs, so will have no trouble navigating the proposed system. There will be a guide if they do have any problems. The administrator will have had experience using a similar program in terms of setting homework and deadlines, and accessing the database will be easy. Otherwise, no more knowledge is necessary. There will be a clear and comprehensive user manual which will be available digitally for each user to read should they have any problems.

\subsection{Access restrictions}

Users will be allowed to access all lessons and homework, and their personal database records and task records for their own profile. They will not be able to see other user's database records. Only the administrator will be able to set homework and deadlines, and view every user's database records. Every user will be able to access a computer with the program installed on it during the lesson time or extra time in which they can access the computer rooms. They might not be able to access it from a home device if they choose not to install it. There may be restrictions in installing software on a school's computers, in which case fewer people will be able to test the program.

\section{Limitations}

\subsection{Areas which will not be included in computerisation}

The system to be developed is focused on the trigonometry and Pythagoras areas of GCSE Mathematics exclusively, so will not include all of the other areas of GCSE Maths, such as algebra, vectors, fractions, probability, etc. It also will only cover GCSE level, so no SAT or A-Level type problems will be included.

\subsection{Areas considered for future computerisation}

It is possible that some questions at the easier or moderate end of AS Level trigonometry could be included for those who really want to challenge themselves. Lessons and homework on the area of 2D ad 3D triangles might be included, expanding the range of triangle based subjects within GCSE Maths.

\section{Solutions}

\subsection{Alternative solutions}

\begin{itemize}

\item The database will be accessible online via the click of a button from the program home page, and will be easy to read, as space conserving as possible (on screen), and will be adjustable exclusively by the administrator if necessary. The database will be initially generated by a Python program which incorporates the SQL package, and then python-SQL commands will be used to make adjustments. The GUI of the database could be programmed using OpenGL, if the Python generated database itself cannot be made to look like the client's request.

\item The 2D graphics used in some of the lessons and homework could use OpenGL or JQuery because OpenGL is platform independent, a requirement for the client. It supports 2D transformations, translations and rotations, solid filling, and hidden line removal, all of the things which I will include in the lessons and homework. JQuery is able to support online graphics, on a web browser if need be.

\item PHP could be used because it has good support for working with SQL.

\item JQuery could be implemented somewhere because it supports browsers, which might be the basis of my database.

\item Mathematical formulas from the numpy package could be implemented into algorithms which will help check whether or not the user's solution is correct for a question. These will include Pythagoras' Theorem and SOHCAHTOA. They will also be used to generate random questions with solutions in certain homework or optional challenges.

\end{itemize}

\subsection{Justification of chosen solution}

\begin{itemize}

\item OpenGL is supported by Python very well, and is capable of producing high quality graphics. Python also supports SQL, so both of these languages can be incorporated into my mainly Python program effectively.

\item Python has a numpy package which supports mathematical algorithms, including Pythagoras and trigonometry algorithms, so will be useful for implementing them alongside the OpenGL graphics code for 2D operations and solution checks.

\item 2D graphics will be used in the first place because they make the program more interactive by including animations triggered by the user, and interactivity is a priority across all aspects of the program, including navigating the GUI, learning from the lessons, completing the homework and reading and adjusting the database.

\end{itemize}