Freelance Gradebook for Varndean College

IBRAHEEM HUSSAIN

Contents

[Analysis 6](#_Toc36921668)

[Described and justified the features that make the problem solvable by computational methods, explaining why it is amenable to a computational approach. 6](#_Toc36921669)

[Problem identification: 6](#_Toc36921670)

[Computational Methods 7](#_Toc36921671)

[Problem Solving: 7](#_Toc36921672)

[Security 7](#_Toc36921673)

[Data Mining: 7](#_Toc36921674)

[Visualisation: 7](#_Toc36921675)

[Concurrency: 7](#_Toc36921676)

[Thinking ahead: 7](#_Toc36921677)

[Abstraction: 7](#_Toc36921678)

[Questionnaire: 8](#_Toc36921679)

[Stakeholders: 9](#_Toc36921680)

[Identified suitable stakeholders for the project and described them explaining how they will make use of the proposed solution and why it is appropriate to their needs. 9](#_Toc36921681)

[Research on stakeholders 11](#_Toc36921682)

[Researched the problem in depth looking at existing solutions to similar problems, identifying and justifying suitable approaches based on this research. 15](#_Toc36921683)

[Existing solutions: 15](#_Toc36921684)

[Essential Features 19](#_Toc36921685)

[Identified the essential features of the proposed computational solution explaining these choices. 19](#_Toc36921686)

[Limitations 20](#_Toc36921687)

[Identified and explained with justification any limitations of the proposed solution. 20](#_Toc36921688)

[Hardware and software requirements 20](#_Toc36921689)

[Specified and justified the requirements for the solution including (as appropriate) any hardware and software requirements. 20](#_Toc36921690)

[Success criteria 21](#_Toc36921691)

[Identified and justified measurable success criteria for the proposed solution 21](#_Toc36921692)

[Design 24](#_Toc36921693)

[Breaking down the problem 24](#_Toc36921694)

[Broken the problem down systematically into a series of smaller problems suitable for computational solutions, explaining and justifying the process. 24](#_Toc36921695)

[Create user (register) form 25](#_Toc36921696)

[Login form 25](#_Toc36921697)

[Grade form 25](#_Toc36921698)

[Pseudocode 26](#_Toc36921699)

[Defined in detail the structure of the solution to be developed. 26](#_Toc36921700)

[Described the solution fully using appropriate and accurate algorithms justifying how these algorithms form a complete solution to the problem. 26](#_Toc36921701)

[Registering: 26](#_Toc36921702)

[Logging in: 27](#_Toc36921703)

[Grade form: 28](#_Toc36921704)

[Database Connection 30](#_Toc36921705)

[Usability features 32](#_Toc36921706)

[Described, justifying choices made, the usability features to be included in the solution. 32](#_Toc36921707)

[Login design: 32](#_Toc36921708)

[Register form design: 34](#_Toc36921709)

[Student gradebook form: 36](#_Toc36921710)

[Teacher gradebook form: 37](#_Toc36921711)

[Key variables 38](#_Toc36921712)

[Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) justifying and explaining any necessary validation. 38](#_Toc36921713)

[Validation: 40](#_Toc36921714)

[Test Data 42](#_Toc36921715)

[Identified and justified the test data to be used during the iterative development of the solution. 42](#_Toc36921716)

[Test data table 42](#_Toc36921717)

[Registering and logging in 42](#_Toc36921718)

[Gradebook 42](#_Toc36921719)

[Post development test data 43](#_Toc36921720)

[Identified and justified any further data to be used in the post development phase. 43](#_Toc36921721)

[Test plan 44](#_Toc36921722)

[Developing the coded solution 47](#_Toc36921723)

[Provided evidence of each stage of the iterative development process for a coded solution relating this to the break down of the problem from the analysis stage and explaining what they did and justifying why. – Explanation of below illustrations 47](#_Toc36921724)

[Provided evidence of prototype versions of their solution for each stage of the process. 47](#_Toc36921725)

[The solution will be well structured and modular in nature. 47](#_Toc36921726)

[Code will be annotated to aid future maintenance of the system. 47](#_Toc36921727)

[All variables and structures will be appropriately named. 47](#_Toc36921728)

[There will be evidence of validation for all key elements of the solution. 47](#_Toc36921729)

[The development will show review at all key stages in the process. 47](#_Toc36921730)

[Provided evidence of testing at each stage of the iterative development process. 47](#_Toc36921731)

[Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken. 47](#_Toc36921732)

[Designing the forms: 48](#_Toc36921733)

[Testing: 48](#_Toc36921734)

[Review at all key stages 51](#_Toc36921735)

[The development will show review at all key stages in the process. 51](#_Toc36921736)

[Database.cs code 52](#_Toc36921737)

[Database Review 52](#_Toc36921738)

[The development will show review at all key stages in the process. 52](#_Toc36921739)

[Login Form Code 52](#_Toc36921740)

[Login Review 53](#_Toc36921741)

[The development will show review at all key stages in the process. 53](#_Toc36921742)

[Register form code 55](#_Toc36921743)

[Register review 56](#_Toc36921744)

[The development will show review at all key stages in the process. 56](#_Toc36921745)

[Testing: 58](#_Toc36921746)

[Testing Review 58](#_Toc36921747)

[The development will show review at all key stages in the process. 58](#_Toc36921748)

[Main Form Design: 59](#_Toc36921749)

[SQL Server 2019 59](#_Toc36921750)

[Server Review 60](#_Toc36921751)

[The development will show review at all key stages in the process. 60](#_Toc36921752)

[Testing: 61](#_Toc36921753)

[Creating an account and logging in: 61](#_Toc36921754)

[Login testing review 62](#_Toc36921755)

[The development will show review at all key stages in the process. 62](#_Toc36921756)

[Developing the gradebook: 63](#_Toc36921757)

[Testing: 64](#_Toc36921758)

[Saving info to the database: 64](#_Toc36921759)

[Database test review 65](#_Toc36921760)

[The development will show review at all key stages in the process. 65](#_Toc36921761)

[Failed tests 65](#_Toc36921762)

[Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken. 65](#_Toc36921763)

[The solution will be well structured and modular in nature. 65](#_Toc36921764)

[Code annotation 66](#_Toc36921765)

[Code will be annotated to aid future maintenance of the system. 66](#_Toc36921766)

[Variable naming and structure 67](#_Toc36921767)

[All variables and structures will be appropriately named. 67](#_Toc36921768)

[Key validation 68](#_Toc36921769)

[There will be evidence of validation for all key elements of the solution. 68](#_Toc36921770)

[Annotated validation test: 69](#_Toc36921771)

[Failed test fix (for validation): 70](#_Toc36921772)

[Annotated validation test: 72](#_Toc36921773)

[Failed text fix (for validation): 73](#_Toc36921774)

[Annotated validation test: 75](#_Toc36921775)

[Failed text fix (for validation): 75](#_Toc36921776)

[Validation review 76](#_Toc36921777)

[The development will show review at all key stages in the process. 76](#_Toc36921778)

[Final Testing Review 77](#_Toc36921779)

[The development will show review at all key stages in the process. 77](#_Toc36921780)

[Additional Testing 78](#_Toc36921781)

[Provided evidence of testing at each stage of the iterative development process 78](#_Toc36921782)

[FAILED TESTS 83](#_Toc36921783)

[Provided evidence of any failed tests and the remedial actions taken with full justification taken. 83](#_Toc36921784)

[Evaluation 89](#_Toc36921785)

[Using test evidence to cross evidence with the success criteria to evaluate the solution. Explain how the evidence shows that the criteria have been fully, partially or not met in each case. 89](#_Toc36921786)

[Provided comments on how any partially or not met criteria could be addressed in further development. 89](#_Toc36921787)

[Success of the usability features 97](#_Toc36921788)

[Provided evidence of the usability features justifying their success, partial success or failure as effective usability features. 97](#_Toc36921789)

[How partially or unmet usability features could be addressed in further development 98](#_Toc36921790)

[Provided comments on how any issues with partially or unmet usability features could be addressed in further development. 98](#_Toc36921791)

[Maintenance 99](#_Toc36921792)

[Considered maintenance issues and limitations of the solution. 99](#_Toc36921793)

[Limitations and improvements 99](#_Toc36921794)

[Described how the program could be developed to deal with limitations and potential improvements / changes. 99](#_Toc36921795)

# Analysis

## Described and justified the features that make the problem solvable by computational methods, explaining why it is amenable to a computational approach.

## Problem identification:

Varndean College is a school located in Brighton and Hove which has many students. Currently, the grading is undertaken by a web based online system called “Moodle”. Since the introduction of Moodle, the grading process for teachers is difficult because Moodle’s architecture is hard to navigate. Students also experience issues in using Moodle. In order for a student to see their grades, they travel through many links in the website. Students find difficulty viewing the grades of different subjects as they must view each subject independently. Calculating average grades is very time consuming. Moodle also has a very clunky interface.  
On Moodle, teachers are required to set homework, however, due to the complicated interface, the teachers usually opt for alternative solutions, such as Google Classroom.  
The average grading calculation is complicated for teachers to use on Moodle, this is remedied by using spreadsheet software to calculate the grades for their classes. Using multiple systems is time consuming which could be *better used* to plan lessons enabling a more productive environment.

Google classroom is a web service, made by Google for schools, which aims to simplify making, giving out, and grading assignments without paper. The problem with Google Classroom is that students are confused when teachers assign points to students’ homework instead of actual lettered grades in Google Classroom.

The problem with spreadsheets is that they are 1 dimensional so they only have one purpose in a grading system which is to put the grades into an easy to read format, the teachers require multi-faceted system, which has been designed, developed and evaluated to meet the Varndean school teaching requirements.

Teachers at Varndean College have requested for me to create a new system which joins essential features such as storing grades by subject, academic year and names, this will allow more productive time for teachers as the use of multiple software will not be required and will not have to constantly switch between software. A grade book has also been requested. The grades must be able to be sorted by date and there must be an average grade for all of the dates and their respective tests.

The main problems with using Moodle, Google Classroom and spreadsheet software are as follows:

* It is time consuming because the solutions aren’t fully compatible with each other.
  + This will put a strain on planning time, making inefficient use of time resource which could be better used towards teaching content which could impact the students’ grades:
  + More time spent addressing student issue instead of planning will make improve the school’s average grades.
  + Headteachers may consider too much planning time a waste of resources and inefficient use of time Teachers, resulting in undue pressure for classroom teachers.
* Google classroom, Moodle and Spreadsheets (Office 365) require a reliable internet connection.
  + If teachers do not have an internet connection available, they will not be able to plan significantly impacting the quality of teaching.

## Computational Methods

### Problem Solving:

One problem is how I would store the grades. I will be able to store the grades on a database which I will create.

Another problem is how the user is supposed to enter their grades. This will be done through inputting the grades in the program then updating the grades in the database.

With Moodle, the pages for where the students’ grades are kept are hard to find. To solve this, I will put them all in one place.

### Security

I can store all the student’s data in an encrypted file, or in a compressed file, so that unauthorised users may not be able to access it. I can also create a login for the teacher for extra security.  
By using a sorting algorithm like quick sort, I can sort all of the raw data of the students’ scores in order (from lowest to highest), so that they can be outputted on a graph with ease.

### Data Mining:

An average score for students can be displayed for all the students for each topic on a line graph.  
This will help identify patterns. Specifically, where the average is low, to show which most of the topics the students are finding hard.

### Visualisation:

Creating a bar chart which tracks the sum all the student’s test scores by date. The chart will display the dates on which day the tests were taken and what score they achieved. Under the chart will be a mean score and the range. The range will be used to compare their current/most recent score to their first score. This is a disadvantage, because there won’t be an option to compare 2 specific dates together, I will implement that later.

Behaviour can be assessed thought an emoticon, which, based on the student’s behaviour will change from a happy face to a sad face. The behaviour can be logged on a lesson to lesson basis through manual input from the teacher (the teacher will pick a face depending on the student’s behaviour depending on the day).

### Concurrency:

Reusing graph code to compare, and/or display the students’ scores.

### Thinking ahead:

I can reuse the code to create the graphs, but change the file input, so that the raw data can change depending on which file I am using for the graph.

### Abstraction:

I will not add useless features, like games into the program.

## Questionnaire:

Who would like to use my project/Who is your target audience?

Mainly teachers and tutors to compare each of their students’ scores, so they can spend a longer time teaching the harder topics and a shorter time teaching the easier topics.

Students may also use a limited number of features (compared to the teachers) of this project to view their own test results.

Will you be able to sort the scores?

Yes, you will be able to sort the scores by the date, or by scholarly term.

On what kind of scale is your project aimed at?

It is aimed for separate classes specifically, not whole schools. In the future, I will add that capability. For example, I will add a feature which will compare the different classes’ scores, or a feature which allows one to compare the scores from each different year. It will be displayed on a graph.  
(E.g. Avg. marks for 2019 is 60, 2018 is 98)

## Stakeholders:

### Identified suitable stakeholders for the project and described them explaining how they will make use of the proposed solution and why it is appropriate to their needs.

What is a stakeholder?

A stakeholder is an entity in a system who is interested in the success in the system. These are split into two main classifications, Internal (direct input or influence on the data output) and External (Do not have direct influence on the data output but uses the results to make an assessment or judgement.

* Teachers – Internal - Software users -
* Students – Internal - Software users -
* School / Head teacher – Internal – Objective focused Reviewers
* Exam Board – External - Oversee Reviewers – used for reviewing students grades whilst at college and comparing them to the external exams set by the exam board to determine students’ progress between college set exams and external exams
* Further education / University – External bodies – The output of grades from the programme can be used to be submitted to Universities as predicted grades used for their selection criteria.
* Local Authority – External - Grade data can be used to compare between different colleges to assist in determining the quality of teaching. This will be functional once the programme has been rolled out to other colleges.

This software is designed mainly for internal stakeholders, but external stakeholders are able to use this software as well. Teacher and students are both internal users who will use the software for what is proposed for being used for. The school or head teachers are both internal users. They differ to students and teachers by focussing on objectives such as improving the grades of the school.

External stakeholders will not use the software to add grades. They will use the software to view grades. Exam boards are external stakeholders and their purpose of using the software is to oversee the grades and to review them to give the students their accurate grades for their tests. This is important because students need to know the grades that they will get. Another external stakeholder is the further education bodes. They are supposed to check your grades in order to see if you are eligible to attend their university.

The local authority is another body who are stakeholders in my project. They can use the grades which the teachers input to compare the grades between each college to tell which college has the best quality of teaching. Once other colleges have access to this program, the local authority can compare the grades between colleges.

The teachers of Varndean College keep having to switch software to software to record grades. There have been complaints that it takes long to switch and login between these software and that they could use this time to plan to teach. They have told me that due to the repetitiveness of this, their teaching morale has been lowered. Avoidable mistakes are made using the software, the teachers have to use traditional methods for solutions when teaching, such as paper based teaching.

Some teachers have noted that they could pay for all-in-one software, however, since school budgeting is tight, and most all-in-one software is subscription based, they worry that the school will not be able to fund these kinds of software, which is one reason why the Varndean College have opted for free solutions such as Moodle and Google Classroom. Since I am a student at the school, I can develop a program to track grades for free by developing software iteratively to meet the needs of the school.

Whilst researching the teachers have requested to me that they need a gradebook which tracks grades with a functional login system which both students and teachers can use.

#### The School/The Headteacher

The school/the headmaster will also be stakeholders. With the teachers’ morale reducing, eventually, teachers may feel they do not have the software to make their job easier, so they may be more inclined to leave the school, with the school suffering a loss in quality teaching staff. With my software, there will be more time-efficiency than using less time-consuming software.

#### Teachers

The teachers will be stakeholders. The teachers will use the proposed software the most, by using it everyday to track the students’ progress and compare what they need to focus on when teaching. They will use it to store grades, Write notes as to how the students are performing in class and in tests. The program I am making should be easy one to use.

#### Students

The students will be stakeholders. Students will need to use the software to view their grades and report their grades back to their parents/carers. They will also be directly affected by the creation of the program because they will not be using Moodle and Google Classroom to see what grades the teacher has given them for homework or school exams.

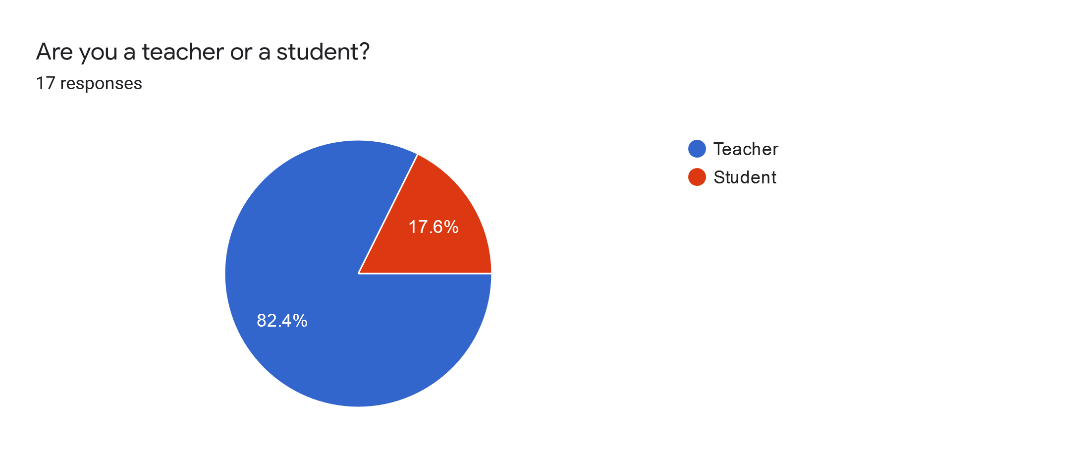
As the software is used the software can be further developed and advanced with future pool of school students. The benefit of this is that it will allow new student to develop their programming skills whilst developing the software and keep on top of new and changing curriculums. This is better than subscription based software, as it does not cost anything and new technologies can be adopted such as graph databases, machine learning and augmented Data Management.

The code has to be well scripted with comments easily read for future development.

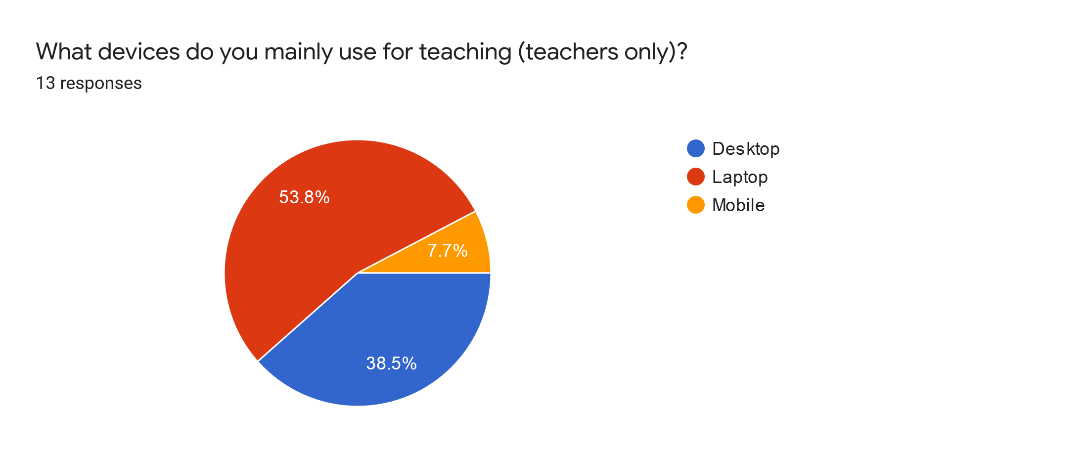
### Research on stakeholders

I have created a survey for stakeholders and have recorded and will annotate their responses. I have mainly sent this survey to teachers as they will be the main users of the software. Although this is a small sample of teachers, it is a good representation of a large samle of teachers so I can tell which features to add.

Here you can see that I have asked the respondents if they are a teacher or student. Most of them are teachers. This means that I will apply a design to the program which will be more suited for teachers (a professional design) as well as accommodating design choices which students’ prefer.



Next you can see that I have asked teacher respondents about which device they mainly use for teaching. The most picked option was a laptop. The second most picked option was a desktop. The least picked was a mobile phone. With these choices below, I have decided to make my software compatible for laptop and desktop only. I am planning on adding mobile support post development



The next question asked respondents what features they would like to be added in the program. I asked this so I could gauge their opinions to see which additional features they want. This question is optional because some students or teachers couldn’t think of any features which they needed other than recording grades. As you can see, the first response is “A gradebook”, which will be the main part of the program.

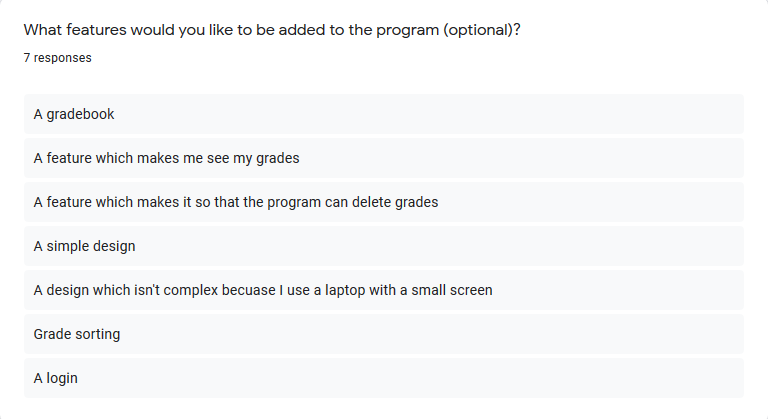
The next response is similar to the first one and is telling me that they need to be able to see their grades. In addition to just a gradebook, respondents have told me that they want a deleting feature. I will implement this only for teachers. To prevent students form deleting their percentage grades access will be read only for students.

A simple design will be implemented. This is so that users will not be confused when using the software. The next point asks for a simple design because of the device they use for teaching.

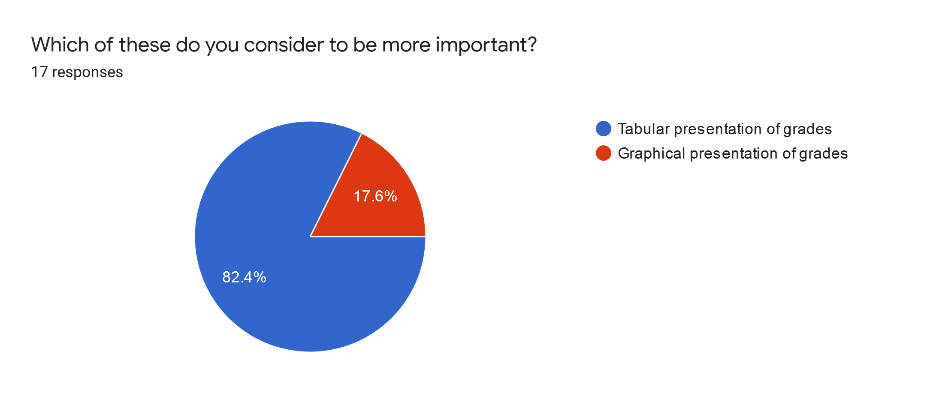
The last two points will also be implemented into the software which are storing and a login.

The sorting will be so that teachers will be able to see who the highest and lowest performing students.

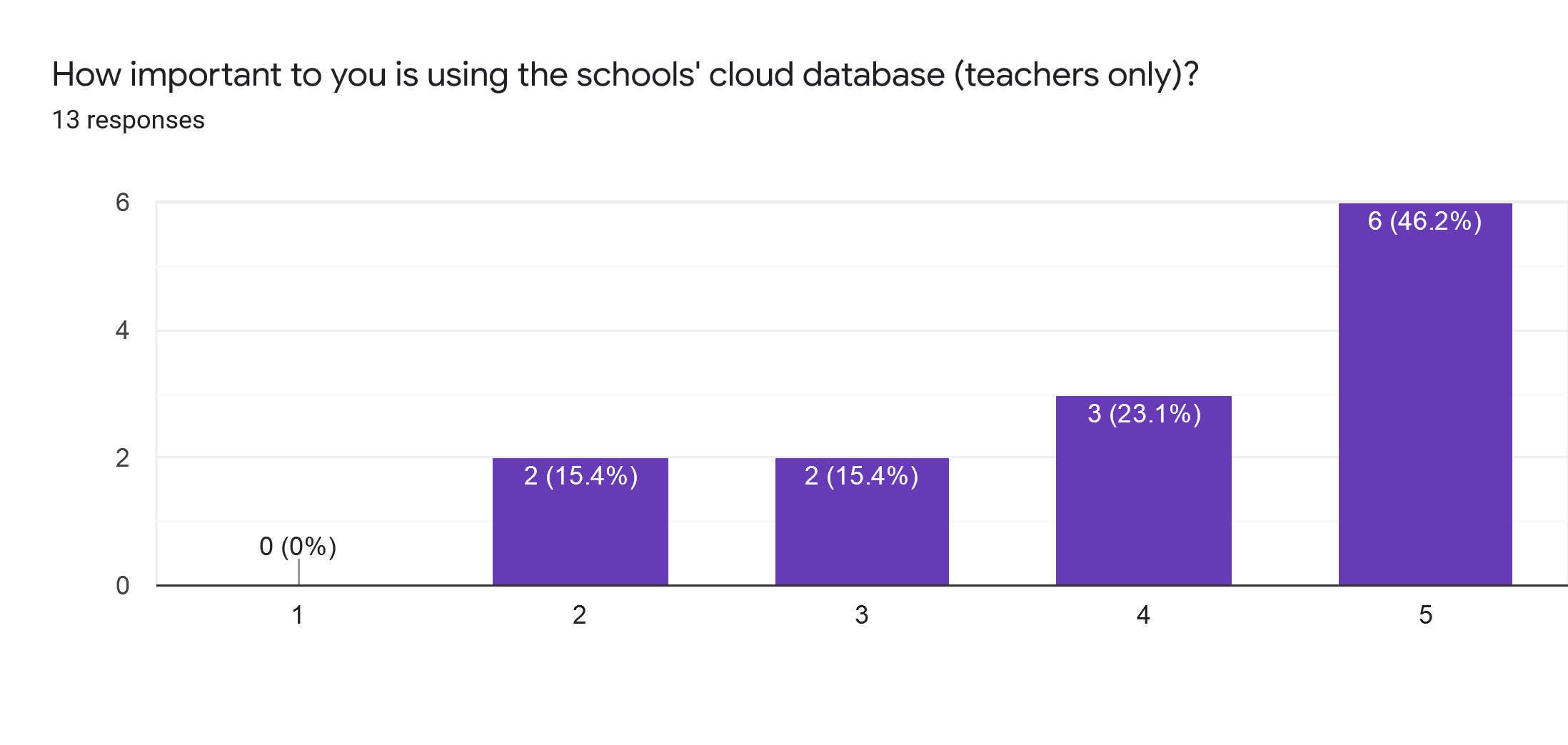
A login will be used to keep the records of the grades secure.



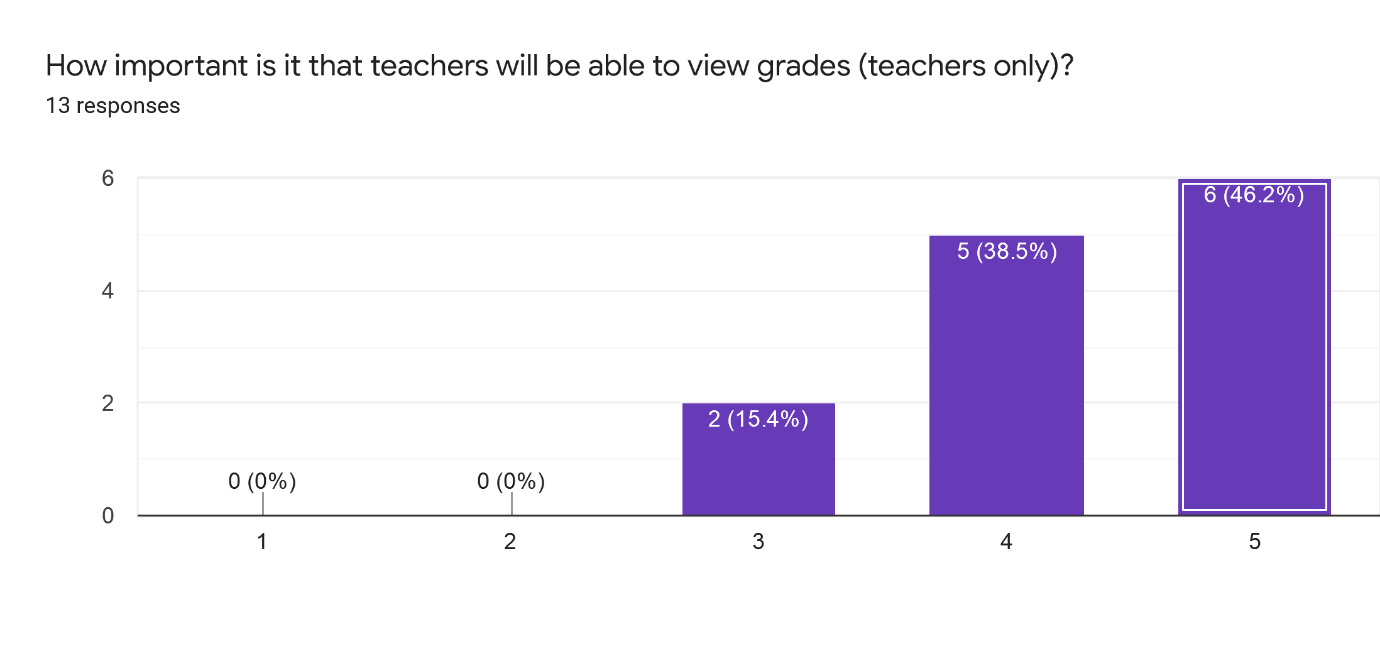
Next, I asked what kind of presentation of grades students and teachers wanted. 3 chose graphical presentation and 14 chose a tabular presentation. I have followed up and asked the students and teachers why they wanted a tabular presentation. The general consensus was that it is easier to sort and read the grades in a tabular presentation and that the graphical presentation will not contain enough information as graphs usually have 2 axis, while tables can have many rows and columns.



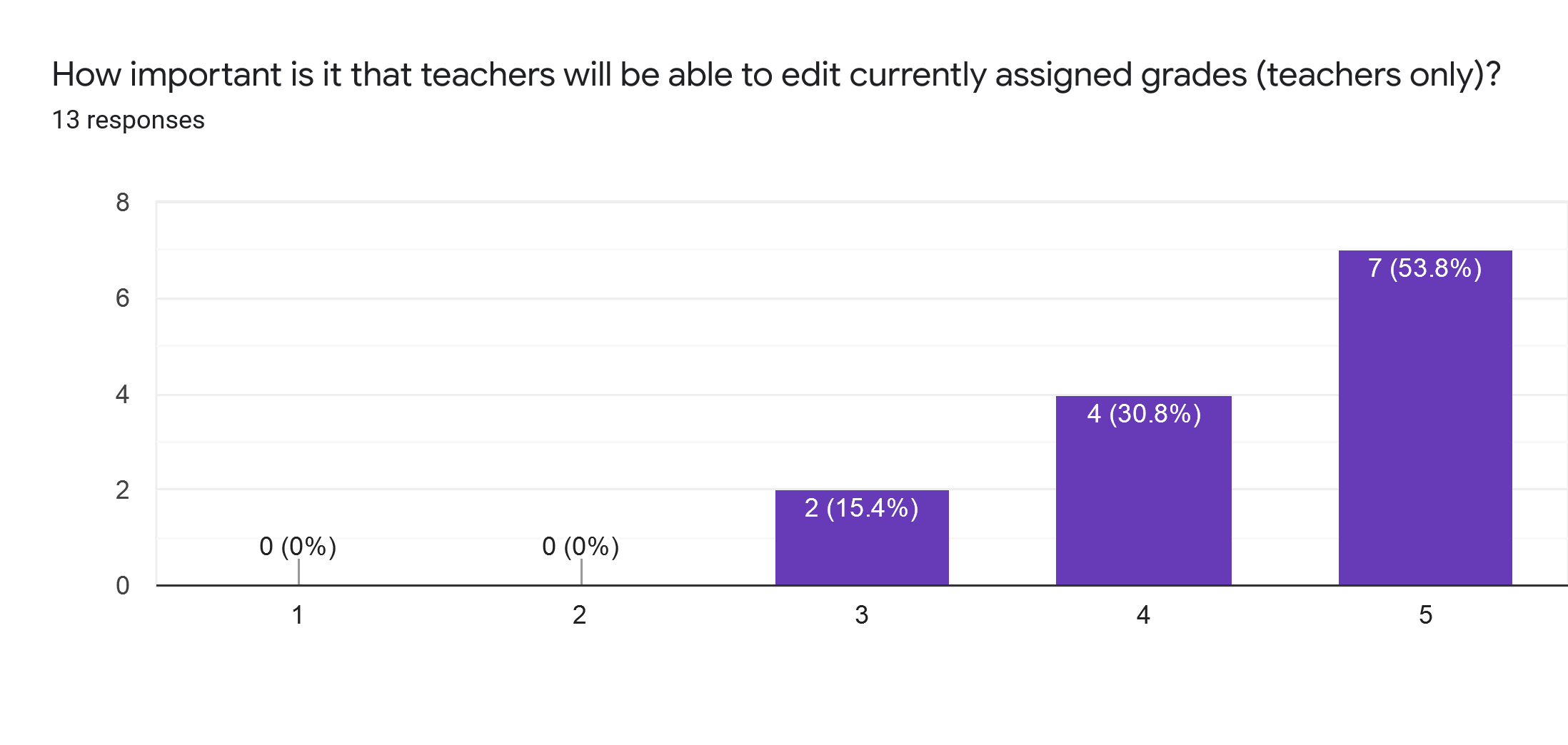
The next question asked teachers if they wanted access to the cloud database. I asked this so that I could tell if I should add a feature for the grades to be connected to the schools’ cloud database. I have only set this question for teachers because students don’t have access to the cloud database. Due to this restriction, The cloud database will only contain content for teachers only with “teacher only content” section. As you can see, just under half teachers selected it as most important (5), while none selected it as least impotant (1). Some teachers chose the middle options, but not enough to make a significant difference in the distribution of opinions. I will therefore include this.



The next question was also only for teachers and asked how important teachers would be able to view grades. As you can see, they deem it very important, as there are no responses which have chosen 1 or 2 (least important). Like the previous question, just under half teachers selected that it was the most important (5). This means that I will implement a feature which will allow users to view grades.



The next question asks teachers only how important it is to edit currently assigned grades. This is important because teachers may have to adjust a mark e.g where a student has retaken the test and attained a higher mark. The teachers think this is very important with more than half of responses being 5. I will definitely add this feature as teachers deem this important.



### Researched the problem in depth looking at existing solutions to similar problems, identifying and justifying suitable approaches based on this research.

## Existing solutions:

Physical copies - Written down in ink on paper

One existing solution is that you can do all the tasks of this program on paper.

The advantage is that this cannot be deleted but can be shredded or lost.

Other disadvantages are is it is inefficient, and you may have to buy storage (i.e. boxes, to store paper) if there are lots of documents, which is costly to store in a secure environment.

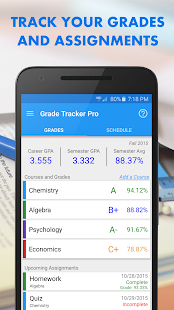
Time consuming when analysing grades for teachers as there will be lots of paper to analyse.

* You write down the grades on the paper.
* Advantages
* Paper can’t be deleted by accident. You can’t delete paper because paper is paper, and paper can’t be deleted
* Not much resource is needed. Only a pen and paper is needed.
* Disadvantages
* Paper can be lost because it is physical
* Paper will take up a lot of space because it is physical compared to a digital grade book
* Paper can be easily damaged because it is weak
* You can’t sort the grades which are in one piece of paper. You can sort grades digitally but not on paper.

Apps are computer programs or software applications designed to run on a mobile device

Available apps are:

Grade Tracker Pro (Free!):

Source:  
<https://play.google.com/store/apps/details?id=com.NsouthProductions.gradetrackerpro&hl=en>

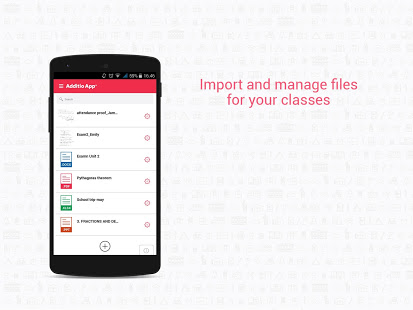
* This app tracks grades.
* You input your grades
* It can calculate your average grade.
* It can track your GPA
* It can predict your grades
* Advantages
* Grade Tracker Pro is an app so it can be on your phone. If it’s on your phone then the app is portable because your phone is portable
* It has lots of features. This is good because you do not have to switch apps a lot to get a feature which this app does not have
* Disadvantages
* This app is only suitable for students. This is because there is no feature to have multiple students or teachers.
* This app is only compatible with android. This is bad because people with other OSs for mobiles cannot use this.
* This is not available on computers. A small screen can make grades hard to read.
* This app only uses a Canadian and American grading system. This is bad because only people from 2 countries can use it. It is limited.
* Only college and university students can use this app because it is only designed for people who attend university and college. Only a limited amount of people can use this app.

This is a revision tracker app which can track your grades and gives you an average score.

One advantage is that it is an app and it is very portable so you can check your grades anytime.  
Most of the features are for university and college students in the US and Canada.  
It can also be used offline.  
It has a grade and GPA tracker. Other useful positive features include: Schedule management, custom grade scales, grade prediction and final exam prediction, data backup (manual), dropped grades (for dropped courses), weighted categories (give weight to different courses).

One key negative point is that, although a Canadian grading system and a US grading system is available on this App, there is not a grading system for the UK, so it is inappropriate for use at Varndean College.   
Other disadvantages are it is limited to older (university and college). The app doesn’t work on iOS either, so some students who own an iOS device will not be able to use it.

Teacher's Gradebook – Additio:

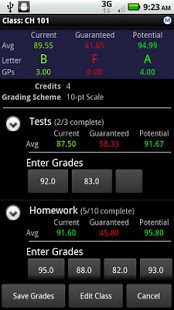
Source:  
https://www.additioapp.com/en/teachers/

This is an app which allows teachers to track their students’ progress.  
A positive is that you can record students’ attendance. It is a small file size (13MB) and only requires an old version of Android (4.0.3), the latest version is Android 9, it also has support for iOS. You can use mobile phones from December 16, 2011, which is when Android 4.0.3 was released. Both which are positives.

It has many features. Including: A digital notebook, lesson planner, reports, attendance records, timetable records, seating plan, calendar, importing data, link with Google Drive and Microsoft for Education and exporting data Microsoft Excel and to PDF.

A disadvantage is that there is only a web version for PC use. This may require an internet connection. Another disadvantage is that there doesn’t appear to be a version specifically for students, while my approach is designed to be used by both teachers and students. Another major disadvantage is that this version is subscription only. My solution is free of cost. Economically better for the Varndean college.

* This app is made for teachers
* It is made so that teachers can record students’ attendance and progress such as grades.
* It has a digital notebook, a lesson planner, a calendar
* Advantages
* It can be linked to Google Drive. This is good because it is compatible with a large cloud-based software.
* It is like an all in one teachers’ app. This is because it has lots of features suited to a teachers’ needs
* Disadvantages
* It is only made for teachers and there are no features for students. This restricts the amount of people who can use the application

GradeWizard:

Source:  
[**https://play.google.com/store/apps/details?id=com.jarcas.gradewizard.full**](https://play.google.com/store/apps/details?id=com.jarcas.gradewizard.full)

This is an app which can track your grades and give you an average score out of all your grades.  
Its features include: Predicted grades, your GPA, differentiate between letter grades and grade points, include weighted grades.  
A disadvantage is that it doesn’t contain as many features as other apps. Another disadvantage is that this program is ugly which will deter students from using it. It is only suitable for one student and not for many students, or teachers. There is no database where multiple users can store their grades.

* This app can track your grades
* It can predict grades and your GPA
* Advantages
* It has some good features
* Disadvantages
* It is ugly
* It is only for students. Teachers cannot use it
* There are not as many features as the other apps. Less people will use it

## Essential Features

### Identified the essential features of the proposed computational solution explaining these choices.

There will be 3 main essential forms:

* The login form
* The register form
* The gradebook form

First the program will have to open.

* This is so that the user who wants to use the program can use it.
  + To do this, I will create a login form which will be the first thing to open when the student registers.

There also needs to be a database to store the students’ information. The information stored will also be used to log into the software. I will store some information.

* The information stored will be:
  + Their first name
    - This is so the teacher can tell who has signed up
  + Last name
    - This is so the teacher can tell who has signed up
  + A username
    - This is so the user can use the login form
  + A password
    - This is so the user can use their password in the login form

The program will have to have a register system. The register will store the details which are written above in bullet points.

* This is so the students can login.
  + After the students login, they can use the main function of the program, which is the gradebook.

The proposed computational solution needs to be able so store grades into a database. To do this I will create another database to store the students’ grades. This is so that the teachers can record how well their teaching is by looking at the students’ progress. The database also needs to store the students’:

* First name
  + To tell who will get the grade
* Teachers’ name
  + To tell which teacher marked the test
* Grade percentage
  + To tell how well the student is performing
* Notes of the test which they took
  + To tell the student where to improve
* Academic year
  + To tell which year the student is in and how it reflects on their progress

## Limitations

### Identified and explained with justification any limitations of the proposed solution.

There are some limitations such as

* There isn’t any support for disabled people.
  + This will be added post development due to time constraints. This in particular will take a long time because there will be a lot of conditions to consider before making changes to the software.
    - The added accessibility features will be:
      * Text to speech. This will be used to read out information for blind people
      * Speech to text when inputting grades. This will be used by blind people to talk to the microphone to input information like grades.
* A timetable cannot be implemented.
  + This will be added post development due to time constraints. There will be 2 timetables. One for teachers and one for students.
    - This will tell what time the students will have their lessons and what time the teachers will teach.
* There is also no attendance tracker.
  + This will be added post development due to time constraints. The attendance tracker will tell the user how many classes the student has been attending (as a percentage) and how many they have missed.
    - This is good for teachers because they can see if and why a student is not attending.
* The design will be simple because that is what the stakeholders want.
  + A “less simple” design be added post development. It will still be simple but will have some more usability features to help the user use the software with ease.

## Hardware and software requirements

### Specified and justified the requirements for the solution including (as appropriate) any hardware and software requirements.

* The software requirements are that your computer must be on Windows 7, 8.1 or 10.
  + This is because the version of Visual Studio (Visual Studio 2019) which I am using only runs on these versions of Windows.
* The users must also have XAMPP installed.
  + This is so the users can save their information into the database.
* The teachers must install SQL Server Management Studio and they must install SQL Server 2019.
  + This is so they can manage the database with all of the grades in the database.
  + SQL Server will be used to install the server (if there isn’t a local server)
* The hardware requirements are the same as the one which is for Visual Studio 2019
  + Requires a 2.6 GHz or faster processor.
  + Quad core or better recommended.
  + Requires 4 GB of RAM or higher for load generation.
  + Requires 10 GB of hard disk space.

## Success criteria

### Identified and justified measurable success criteria for the proposed solution­­­­­­­­

Success criteria are the instructions I have to make and follow to ensure that everything about the program is fulfilled. They are broken down into smaller points and I have written down why I need them so that it will ensure a high-quality product.

1. The program opens.

a. The login form will be the first part to open in the program.

b. This is so the user can start to use the program.

1. You can enter text into the text boxes.

a. This is required so that the user will be able to enter their grades and their details and their login detail.

b. I will do this by creating textboxes to enter text.

1. You can click on the buttons.
   1. This is required for the user to add their details to the database in which is being held in.
   2. The button will have some code which (when clicked) will connect the program to the data and send the details.
   3. The buttons will be made in form design.
   4. They will be digital buttons.
2. There are buttons to move to and from the register for and the login forms.

a. If you don’t have an account, you may make one by going to the “Create an account” form using a button as a medium to get there.

1. A database must be created.
   1. The database is where the details of the users are stored.
   2. The database is where their grades of the users are stored.
   3. To do this, I will use PhpMyAdmin
   4. I will also use SQL Server 2019.
   5. SQL Server 2019 will be used to create a local server in which I can create the database.
   6. Along with SQL server 2019, I will use Microsoft SQL Server Management Studio (SSMS) to create a database with tables to store user details.
2. A connection to the database must be established
   1. This is so that the information entered may have a medium in which they can be stored.
   2. To do this, I will use the MySQL Connector package, which will allow me to connect to any database easily. I will also use XAMPP which will give me a port which I can use to connect to the database.
3. A table with columns corresponding to the fields in the “Create an account” form must be created.
   1. This is so the data can be sorted.
   2. The data needs to be sorted so the teachers and students can read it easily.
   3. This is a better solution than Moodle’s grading system as the interface there is clunky and hard to read (mentioned in the problem identification)
   4. I will do this through the PhpMyAdmin interface and name the columns with the same names as the ones in the “Create an account” form.
4. The text entered into the fields of the “Create an account” form must be put into a database as columns.
   1. The user must have somewhere where their login details can be stored so they are able to login.
   2. To do this, with the established connection mentioned in point 2, I will create tables in the database which correspond to the text box fields in the “create an account” form.
5. Text entered into separate fields must be put into their corresponding columns in the database columns which will have the same names as the names of the register form.
   1. This will be done in the register form
   2. This is to make distinctions between the user’s different details, like their name, or the grade which they attained.
   3. To do this, I will link the fields to the columns in the table with SQL.

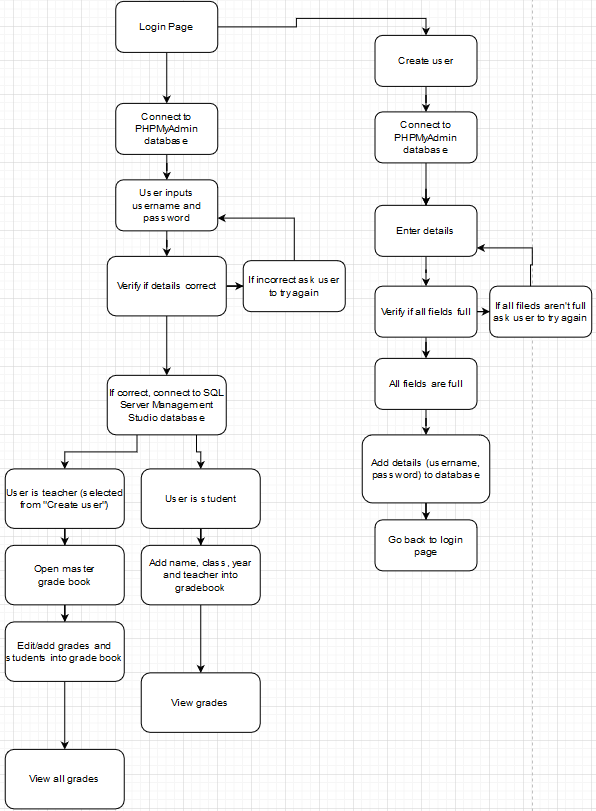
­­

1. The user must have an option to select if they are a teacher or a student in the register form
   1. This is so that they can have different user rights. The student will only be able to view their grades while the teacher will be able to add and edit their grades.
   2. I will do this by creating a comboBox which have a teacher selection and a student selection.
2. Pressing the “Create the account” button will put all field the information into the table.
   1. I will do this using the steps above.
3. All of the fields in the “Create an account” form must not be null.
   1. If students leave some fields null like their username, then they will not have a username.
   2. This is bad because they won’t be able to login.
4. After creating the account, the program will show the login page.
   1. This is so the user can login.
   2. To do this, I will make a function which will make the “Create an account” form close and the login form open.
5. The text boxes in the login form should accept input and the login button should be able to be clicked.
   1. This is so the user can login.
   2. I will do this by creating a button and a textbox for the user.
   3. The user must be able to select if they are a teacher or a student.
6. The username and password will be verified.
   1. This is so that only authorised people will be able to access the gradebook.
   2. I will do this by using SQL to verify if the inputs in the database and the login input boxes are the same.
7. A new form will be opened.
   1. This form will be where the student’s and teacher’s data will be stored.
   2. I will do this by creating a function which will link the login and the info form together.
8. The form will have multiple fields which you can type in.
   1. This is so that the user can type in their details such as their name and grade and which year they’re in.
   2. I will do this by making text boxes.
9. I need to create another database which uses MySQL as its language
   1. This is so that the information mentioned in 17a can be stored somewhere
   2. I will do this using SQL Server 2019
   3. Each user should have their own table for their own information.
   4. MySQL will be used to
10. The database needs to have database specific tables
    1. The table will be made in SSMS and will have rows and columns which will hold the grades and details of the student for the gradebook part of the project.
    2. These are required so that the information mentioned in 17a can be sorted appropriately
    3. I will do this in SQL server by creating a table in the database
11. The database must be connected to the application.
    1. I will use the SQLClient libraries in C# to do this.
    2. This is so the entered information will be able to go the database.
12. The information entered must be saved into the database
    1. This is so that the student can track their grades.
    2. I will do this by connecting the database to the application.
    3. When a “save” button is pressed, the database is updated with the relevant information.
13. There is a button which will allow the user to view their grades.
    1. This will allow the user to track their grades.
    2. I will do this by establishing a connection to the database.
    3. The information will be displayed in the application as a table.
14. If the user is a student, they can only view their own grades.
    1. This is for privacy reasons. Students may not like it if others can view their grade.
    2. I will do this by creating a separate table in SSMS for each user when they create an account.
    3. This will be done while the application and the database are linked.
15. Teachers will have more access and authority than students in viewing and editing the grades.
    1. This means they will be able to view edit grades of the students.
    2. I will do this by using SQL commands to edit the tables.
    3. Students will only be able to view their own grades but not edit them.

# Design

## Breaking down the problem

### Broken the problem down systematically into a series of smaller problems suitable for computational solutions, explaining and justifying the process.

  
In the picture above, you can see that I have designed the problem using a top-down design. It breaks down the problems and will tell me what to develop in what order.

There are three main sections in this program. The login, grade viewing/adding and the registering (create user) section. The login page will open at start up. If the user doesn’t have an account, then there will be a link between the login page and the registering page.

### Create user (register) form

First, the user will click on the register button to open the register form. Once they have done this, in the background, the program will connect to the phpMyAdmin database. The user will have multiple boxes to enter their details. They can also choose if they are a teacher or not. This is so they can have access to the master table where they can see and edit all the student’s grades. Next, when they click on the register button, the program will see if the user has filled in all of the fields or not. If they haven’t, then the program will ask them to try again. If they have, then the text in the textboxes will be saved to the database phpMyAdmin database. Then, the register form will close, and the login form will open so they can login.

### Login form

When the login form opens, the program will connect to the phpMyAdmin database. The user will input their username and password which they chose in the register form. The program will match the username and password to the ones in the database. If even one of them is incorrect, the program will come up with an error message. If not, then the program will move to the grade form.

### Grade form

When the user has successfully entered their login details, then the program will connect to the SQL Server Management Studio (SSMS) database. The program will now verify if the user is a teacher or a student (from the register form). If the user is a teacher, a master gradebook will open. The master gradebook is a gradebook with all users’ data inside of it. They will be able to view, add and remove data as they want. If the user is a student, they can add their own details (their name, grade, what year they’re in etc). They cannot remove anything or edit anything.

## Pseudocode

### Defined in detail the structure of the solution to be developed.

### Described the solution fully using appropriate and accurate algorithms justifying how these algorithms form a complete solution to the problem.

These pseudocodes are algorithms. The comments in the pseudocode describe how the algorithm is related to the solution. These pseudocode forms the basis to plan for the program code. The solution has been fully described below using pseudocode.

Registering:  
I have to make the program enter all of the details into the specific users table which I made for the database. I will do this be making subroutine which assign variables to the text box inputs. Then, I will connect the database to the program. While the connection is open, I will insert the values of the textboxes into the table using SQL. If some of the values are null, then it will ask you to try again. The SQL pseudocode is in quotation marks. It isn’t exact SQL, it is pseudocode SQL.

SUBROUTINE registerForm() { //Bracket 1 opening

//This is the register form which will add your information entered in the textboxes into the database table

WHILE registerButton().Click = TRUE { //Bracket 2 opening

STRING usernameInputBox = userInput

STRING passwordInputBox = passInput

STRING nameInputBox = nameInput

BOOLEAN comboBoxIsStudent = isStudent

CONNECTION con = NEW CONNECTION //Creates a connection with variable “con”

OPEN CONNECTION con “studentdatabase” WITH port 1234 {

//Bracket 3 opening. This will open the connection to the database.

IF usernInput OR passInput OR nameInput OR isStudent != NULL {

//Bracket 4 opening. This checks if all the values are not empty

SQL{ //Bracket 5 opening. This is the SQL command to open add the textbox values into the table. Code inside this bracket which isn’t in quotation marks isn’t SQL. It is normal pseudocode.

“INSERT INTO TABLE(loginInfo(username, password, name, studentYesNo) VALUES   
(userInput, passInput, nameInput, isStudent)))”

} //Bracket 5 closing. This is the closing bracket of the SQL statement.

} //Bracket 4 closing.

ELSE { //Bracket 6 opening.

OUTPUT “You must fill in all of the fields”

} //Bracket 6 closing. If all the boxes aren’t full then the program will tell you to fill in the  
 boxes  
} //Bracket 3 closing. This will close the connection to the database.  
} //Bracket 2 closing. The code within this bracket executes when the register button is clicked   
} //Bracket 1 closing. This ends the subroutine

### Logging in:

This pseudocode below is made to describe what happens when you open and click on the login buttons. To do this, I will assign variables to the text inside the textboxes. Then I will open the connection and use and SQL query to set the values of the table to a string. Next, the program will verify if the values are the same in the table and in the textboxes. If they are, then it should open the form which holds all of the grades. If not, then it should display a message telling them that the information which they typed was incorrect. The SQL pseudocode is in quotation marks. It isn’t actual SQL, it’s only pseudocode SQL.

SUBROUTINE loginForm() { //Bracket 1 opening

//This is the login form which will verify the contents of the database to the contents of the inputs of the text boxes

//The while statement occurs when the user clicks the login button.

WHILE loginButton().Click = TRUE{ //Bracket 2 opening

STRING loginUsernameTextboxInput = loginUsernameInput

STRING loginPasswordTextboxInput = loginPasswordInput

CONNECTION con = NEW CONNECTION

CONNECTION con OPEN “studentdatabase” WITH port 1234 {

//Bracket 3 opening. This will open the connection to the database

SQL{ //Bracket 4 opening. This SQL command will assign textbox values to variables. Code inside this bracket which isn’t in quotation marks isn’t SQL. It is normal pseudocode.

“SELECT \* FROM TABLE(loginInfo(username))” = NEW STRING loginUsernameVerify   
“SELECT \* FROM TABLE(loginInfo(password))” = NEW STRING loginPasswordVerify

} //Bracket 4 closing. This is the closing of the SQL statement.

IF loginUsernameInput = usernameVerify AND loginPasswordInput = loginPasswordVerify { //Bracket 5 opening

gradeForm().Open

}

//Bracket 3 closing. This selects the columns “username” and “password” in the database and assigns them to a variable. It will also close the connection to the database while keeping the values of the username and password. It is placed before loginForm().Close to close the connection before the form closes.

loginForm().Close

ELSE { //Bracket 6 opening.

OUTPUT “Incorrect username or password”

}

//Bracket 6 closing. If the values of the table and the input do not match, then a warning message appears

}   
//Bracket 5 closing. This is the end of the if statement. It checks if the username in the loginUsernameTextboxInput and the password in the loginPasswordTextboxInput are the same as the ones in the table in the database

} //Bracket 2 closing. This is the end of the loginButton().Click while loop

} //Bracket 1 closing. This is the end of the pseudocode.

### Grade form:

OK, to do this part of the program, I will open a connection to the SSMS database.

Then, I will make a function for what happens if you click on the view button. When you click on the view button, the code should execute some SQL code. The SQL will fetch all of the data from the table where all the grades and information is held. Then, it will store it in a variable. The variable will be assigned to a “dataGridView”, which is a box in the program which will be a medium for displaying the users’ grades and information. Then, the data will be put into the dataGridView, which will be displayed. The one below is made for the teacher, as they can select if the person they’re adding to the database is a teacher or a student.

To add data, I will make a button. Firstly, the values in the textboxes for where the data needs to be entered will be stored into variables with names corresponding to the text box names. Then, by using SQL, I will use INSERT to insert the data from the text boxes into the database columns using the variables which correspond to the text boxes names.

This subroutine is made to view and store the grades into the database. The SQL pseudocode is in quotation marks. It isn’t real SQL, it’s SQL pseudocode.

SUBROUTINE gradeForm() { //Bracket open 1.

CONNECTION con = NEW CONNECTION

CONNECTION con OPEN “studentdatabase” WITH port 1234

{ // Bracket open 2. Opens the connection to the database.

WHILE viewButton().Click = TRUE

{ //Bracket open 3. The code inside here is what happens when you click the view button.

SQL{ // Bracket open 4. This is the SQL which displays the table on the program. Code inside this bracket which isn’t in quotation marks isn’t SQL. It is normal pseudocode.

“SELECT \* FROM (gradedatabase(gradeTable))” = NEW gradeTable().dataGridView

OUTPUT gradeTable() INTO dataGridView

OUTPUT dataGridView

}

//Bracket close 4. This closes the SQL statement. It gives the table which holds the grades, a variable name in C#. It assigns the to a datagridview (which, in C#, can display a table from a database).

} //Bracket close 3. This closes the viewButton().Click while loop.

WHILE addButton().Click = TRUE

{ //Bracket open 5. This is the button you click when you want to add someone’s grades to the table

//The variable assignments below assign the text inside the input box, variable names

STRING nameInputBox = nameInput

STRING teacherInputBox = teacherInput

FLOAT gradeInputBox = gradeInput

STRING notesInputBox = notesInput

STRING testInputBox = testInput

BOOLEAN isStudentInput = isStudent

SQL

{ // Bracket open 6. Opening bracket for SQL. This is to add the data in the textboxes to the grade table using the insert function in SQL

“INSERT INTO TABLE(gradeTable(name, teacher, grade, notes) VALUES   
(nameInput, teacherInput, gradeInput, notesInput, testInput, isStudent)))”

} //Bracket close 6. Closes the SQL statement.  
} //Bracket close 5. This is the addButton().Click closing bracket  
} //Bracket close 2. Closes the connection.  
} //Bracket 1 closing. This is to end the subroutine

All three parts of the code above are commented for easy readability.

### Database Connection

The pseudocode below describes how the program will connect to the PhpMyAdmin database. The code above just opens the connection to the database, while the code below makes the connection. It also makes the opening to the connection above possible. This will run immediately after the program opens. The code below is where the PhpMyAdmin core connection to the database is established. It chooses to connect to PhpMyAdmin. Since lots of databases can be made in PhpMyAdmin, I have specified the name database which holds the register and login, so errors won’t appear and so it won’t connect to the first database in the list of databases inside of PhpMyAdmin. The database connection variable name will be called “connectToPhp”, as this is an appropriate name. I have set it so that while the program is running (while any of the forms are open), the connection to the database should be opened. Like before, the SQL code is pseudocode SQL.

Read the comments “//”to understand more.

CLASS phpconnect()

{ //Opening bracket 1. For the class.

IMPORT mysqlConnection PACKAGE //This allows the connector library to be used

SQL

{ //Opening bracket 2. Double apostrophes are where the SQL code is held.

//This code makes a variable names “connectToPhp” for the connection. The SQL is which database it should connect to inside of the PhpMyAdmin server.

SqlDatabaseConnection connectToPhp() = NEW SqlDatabaseConnection( “‘PhpMyAdmin’ DATABASE NAMED ‘Register\_login\_info’’)

} //Closing bracket 2. Closes SQL statements.

WHILE LoginForm().SHOW OR gradeForm().SHOW OR registerForm().SHOW = TRUE

{

//Opening bracket 3. The code in here will execute when the program is open (when any form is open)

OPEN CONNECTION connectToPhp()

} //Closing bracket 3. When the program is open, the connection to the PhpMyAdmin database will be opened

} // Closing bracket 1. For the class

The next pseudocode will be for connecting to the SSMS database.

The pseudocode below describes how the program will connect to the SSMS database. The code above just opens the connection to the SSMS database, while the code below makes the connection. It also makes the opening to the connection above possible. This will run immediately after the gradebook opens. The code below is where the core connection to the SSMS database is established. It chooses to connect to PhpMyAdmin. Since lots of databases can be made in SSMS, I have specified the name database which holds the register and login, so errors won’t appear and so it won’t connect to the first database in the list of databases inside of SSMS. Teachers will only have access to this database to edit code, which is why I have put a password in the database. The database connection variable name will be called “connectToSsms”, as this is an appropriate name. I have set it so that while the program is running (while only the grade form is opened), the connection to the database should be opened. Like before, the SQL code is pseudocode SQL.

Read the comments “//” to understand more.

CLASS ssmsConnect()   
{ //Opening bracket 1. For the class.

SQL

{ //Opening bracket 2. Double apostrophes are where the SQL code is held.

//This code makes a variable names “connectToSsms” for the connection. The SQL is which database it should connect to inside of the PhpMyAdmin server.

SqlDatabaseConnection connectToSsms() = NEW SqlDatabaseConnection( “‘SSMS’ DATABASE NAMED ‘gradeBook\_info’, PASSWORD=’password’”)

} //Closing bracket 2. Closes SQL statements.

WHILE gradeForm().SHOW = TRUE

{

//Opening bracket 3. The code in here will execute when the program is open (when any form is open)

OPEN CONNECTION connectToSsms()

} //Closing bracket 3. When the program is on the gradebook form, the connection to the SSMS database will be opened.

} // Closing bracket 1. For the class

The above algorithms are all the algorithms required for the program to function as it is made for.

## Usability features

### Described, justifying choices made, the usability features to be included in the solution.

### Login design:

Login form

Username:

Password:

Click here to create your account

LOGIN

The design above is what I plan for the login to look like and it is user friendly. This is what the user will see when they login. There are 3 labels.

* One to tell the user where the username text box is
* One to tell the user where the password text box is.
* The third one is the login form label which tell the user which form they’re in.

When the form is opened, in the background, the program will connect to the PHP database.

There is a label which tells the user that this is the login form. It is at the top because this program is in English and English is read from top to bottom; their eyes will lay on the top first.

There is a button which will redirect the user to the register form if they do not have an account.

I have placed this near the top. It will be the second thing the user will see. I have made it big so that the user cannot miss it. This means that they will know where to click when registering and they will not be lost in the login form.

Once the user has noticed they are in the login form, they can type in their username and password. I have put the username text box on top of the password text box because it is a logical flow. This gives the user a sense of ease when logging in.

Last is the login button.

* It is placed last because the user must type in their username and password first before clicking in the login button.
* I have also made it big so that the user will can tell where they need to click instead of searching for the login button.
* It is also placed halfway and centred in the form.

After this form has loaded, the program will connect to the PHP database where the program can verify the login details and its validity to the database.

When clicked, the buttons will run the code which is meant to run when they are clicked. When the program is opened, only the database is connected to the program.

The username textbox and password textbox will start off as being empty then they will be filled in by the user. The password textbox will be filled with asterisks to hide the users’ details.

The “Create an account” button will open to the “Register form” when pressed. It will then close the login form.

If the username and password is incorrect, a textbox will appear warning the user they have incorrect details and to try again by typing their correct username and password again.

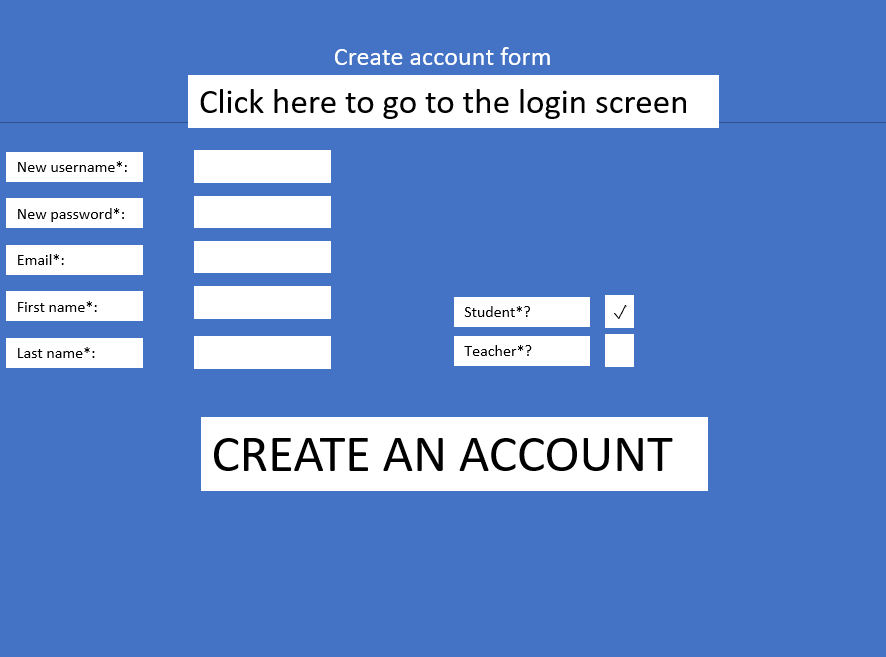
INCORRECT DETAILS. TRY AGAIN.

This will be a big box to warn the user. If it is small, the user may miss it and they will be confused as to why they couldn’t log in.

The tab button on the keyboard, will switch between each textbox in order.

All points above are usability features which will make the program more user friendly.

### Register form design:



The design above is what I plan for the register form to look like. This is what the user will see when they register. There are 8 labels.

There are labels to what form this is, what information they need to type in and labels asking if they are a student or a teacher.

* There is a label which tells the user that this is the create account form. It is at the top because this program is in English and English is read from top to bottom; their eyes will lay on the top first.
* The next labels explain the information which they need to type in
* There is a username label
* A password label
* An email label
* A “First name” label
* A “Last name” email
* A label which asks you to select if you want to tell the program that you’re a student
* A label which asks you to select if you want to tell the program that you’re a teacher

There is a button which would redirect the user to the login form if they accidentally clicked on the register form button, in the register form screen.

I have placed this near the top. It will be the second thing the user will see. I have made it big so that the user cannot miss it. This means that they will know where to click when going to log in and they will not be lost in the register form.

Once the user has noticed they are in the create an account form, they can type in their new username and password. I have put the username text box on top of the password text box as this is how username and password fields are conventionally laid out. This gives the user a sense of familiarity.

The next things the user should type in are their email, their first name and their last name. These aren’t in any particular order apart from the users’ names. The first name textbox and label are placed on top the last name textbox and label. This is because register forms are conventionally laid out this way and it will create a sense of familiarity for the user. One usually tells others their forename then their last name when greeting someone. I have made this so it seems like the user is talking to a real person which is a usability feature because if I put the last name first, they would get confused as to which textbox to use.

The last labels are the student and teacher labels. I have placed the student label first because there are more students than teachers. This is a usability feature because users will not have to read more than necessary to find the student label. The teacher label is placed under the student label.

I have made the create an account button as the button with the largest font so the user can tell easily where to click when logging in.

Initially, the checkboxes next to the student and teacher labels will be empty. I have illustrated a tick inside one of them to illustrate that they are checkboxes.

Furthermore, all textboxes will be empty on opening the “Create an account form”. If they had text inside of them, it would confuse the user.

The “create an account” button is placed at the bottom because the user is supposed to fill in all of the details before clicking on the button.

I will also place asterisks in my program to show that all of the fields are compulsory.

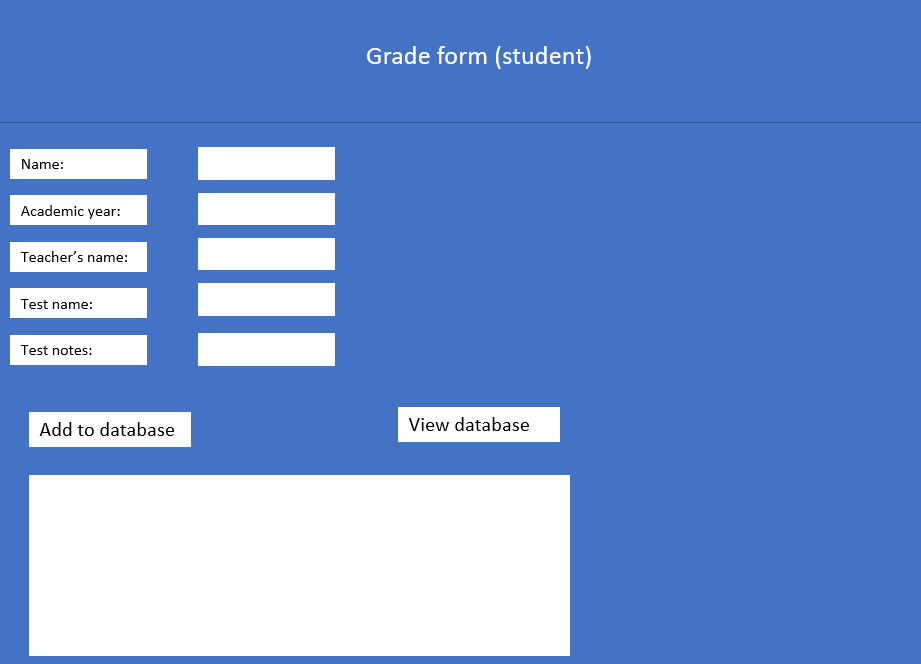
When the user presses tab, blinking cursor should switch between each textbox in order.

Unfilled fields will yield an error to prevent confusion so that the user knows that they have missed some fields.

FILL IN ALL OF THE FIELDS AND TRY AGAIN

All points above are usability features which will make the program more user friendly.

### Student gradebook form:

  
This is the gradebook for the student.

The design above is what I plan for the student gradebook form to look like. This is what the user will see when they register. There are 6 labels. There are labels to what form this is and what information they need to type in.

There is a label which tells the user that this is the create account form. It is at the top because this program is in English and English is read from top to bottom; their eyes will lay on the top first.

This will open after the user logs in successfully. Before this form opens, the program will connect to the SSMS database. When this form opens, all the fields should all be empty.

The user will enter their information into the fields and then click on “Add to database” to add that information to the database. Fields will be able to be left null.

There are “Add to database” button is placed under the text boxes. It is placed under the text boxes because the user is supposed to enter their details in first, then add their information.

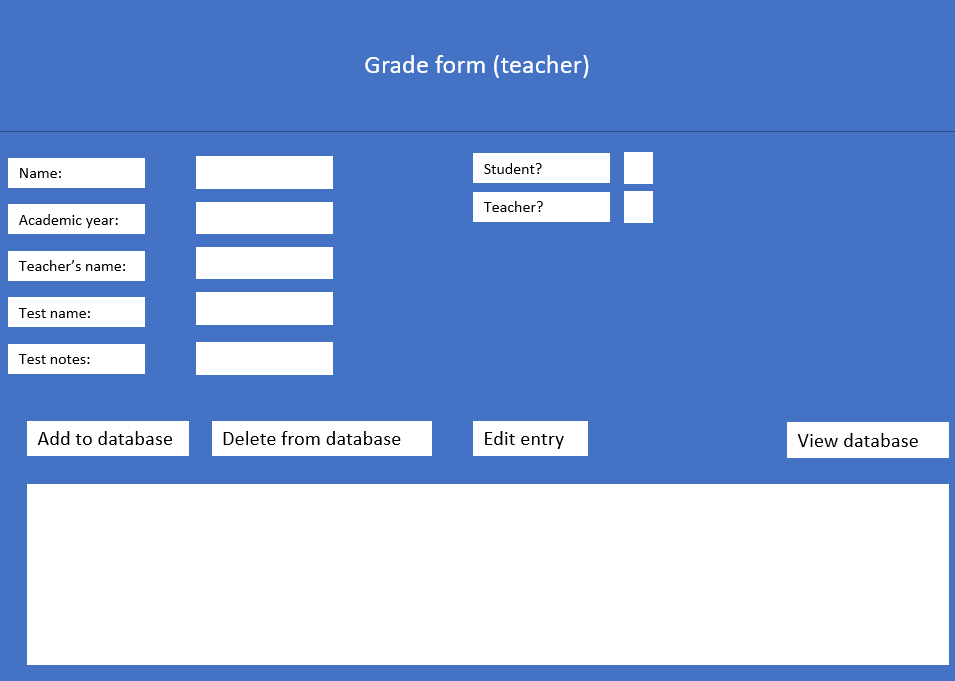
The “View database” button is placed next to the add to database button because after the student has added their details, they need to view them.

The blank data box at the bottom is where the table holding the users’ information will be shown after the user clicks on the view button. It is placed at the bottom because the user needs to enter all of their details and click the add and view buttons (which are all on top of the data box) before looking at the data box.

When the user presses tab, blinking cursor should switch between each textbox in order.

All points above are usability features which will make the program more user friendly.

### Teacher gradebook form:



This is the same as the user gradebook form and has the same usability features apart from that the teacher can delete and edit the database entries. The label telling which gradebook this is (student or teacher) has changed to teacher. There are also 2 checkboxes. Only 1 can be selected at a time. This is so the teacher can add either a student or a teacher into the database.

The student and teacher checkboxes are placed on top of each other, it wouldn’t make sense to move them apart because the teacher can only choose 1 option.

Apart from that, the usability features are the same as the students’ gradebook form.

## Key variables

### Identified and justified the key variables / data structures / classes (as appropriate to the proposed solution) justifying and explaining any necessary validation.

Most of the key variables have been described and justified why they are used in the pseudocode. Here is a table of key variables showing the Name of the variable, The Type of variable (Data structures) and the purpose and justification of the variable.

|  |  |  |
| --- | --- | --- |
| Name | Type - Data Structures | Purpose and Justification |
| **Register Form** | Form | To hold the code for the user to register |
| userInput | String | For the user to choose their own username in the register form. It will be saved to the database |
| passInput | String | For the user to choose what password they want for their account |
| nameInput | String | For the user to type their first name |
| isStudent | Boolean | It is for choosing if the user is a student or a teacher |
| registerButton | Button | This is a button which will save all of the values of the variables to the PhpMyAdmin database |
|  |  |  |
| **Login Form** | Form | To hold the code for the user to login to go to the form where they can add their grades |
| loginUsernameInput | String | Where the value of the username for the login form is held |
| loginPasswordInput | String | Where the value of the password for the login form is held |
| loginUsernameVerify | String | This holds all of the usernames in the PhpMyAdmin database |
| loginPasswordVerify | String | This holds all of the passwords in the PhpMyAdmin database |
|  |  |  |
| **gradeForm** | Form | This form is where the program will display the grades from the SQL Server Management Studio (SSMS) |
| nameInput | String | This is the name of the student. It will be saved into the SSMS database |
| teacherInput | String | This is the name of the teacher or the student’s teacher. It will be saved into the SSMS database |
| gradeInput | Float | This is the percentage of the grade which the student got in their test. The student will input this grade into the database with their other details. It will be saved into the SSMS database |
| isStudent | Boolean | The user will use this to choose if they are a student or not. If they choose true, then the user has chosen to be a student. If they choose false, then they have chosen to be a teacher |

### Validation:

Validation is when you check if the value of a variable contains a valid value before using it. For example, maybe it can't be zero length "" such as a username/ password or maybe it is a number that must be within a certain range 0 to 100 for example.

|  |  |  |
| --- | --- | --- |
| Variable name | Verification | Purpose and justification |
| **Register Form** | All of the fields must be full | So the teacher can tell if they have signed up and so that the users can choose a username and password. |
| userInput | It mustn’t be blank. It has to accept string values. | Users need to enter their username in the register form. It can’t be blank otherwise they will have no username. To do this I will set the column holding this information into one that only accepts strings. |
| passInput | It mustn’t be blank and it must accept string values. | Users need to enter their password in the register form. It can’t be blank otherwise they will have no username. |
| nameInput | Can only input string values. It mustn’t be blank. | Users need to enter their name for teachers to see. If it is blank, the teachers can’t see which student has signed up into the software. |
| isStudent | Must only accept 2 values (student or teacher). | The user needs to choose if they’re a student or a teacher. If there are more than 2 values, then the teacher will not know if it is a teachers’ account or a students’ account |
|  |  |  |
| **Login Form** | All values must be full | This is because the user needs to enter all of their information to login |
| loginUsernameInput | Same as userInput above | The values entered in userInput will be the same as loginUsernameInput |
| loginPasswordInput | Same as passInput above | The values entered in passInput will be the same as loginPasswordInput |
| loginUsernameVerify | Does not need verification | Same reason as loginUsernameInput because this will hold the same value |
|  |  |  |
|  |  |  |
| loginPasswordVerify | Does not need verification | Same reason as loginPasswordInput because this will hold the same value |
|  |  |  |
| **gradeForm** |  |  |
| nameInput | String values. Can be empty. | This is the sutdents’ name, so only string inputs (and no input) will be accepted |
| teacherInput | String values. Can be empty. | This is the teachers’ name, so only string inputs (and no input) will be accepted |
| gradeInput | Float values up to 100 | Percentage of grades cannot go over 100 |
| isStudent | Must only accept 2 values (student or teacher). | The user needs to choose if they’re a student or a teacher. If there are more than 2 values, then the teacher will not know if it is a teachers’ account or a students’ account |

## Test Data

### Identified and justified the test data to be used during the iterative development of the solution.

Test data is data which is used to test the program.

The test data will be plain text as the purpose of my program is to track grades. I will enter the text into the text boxes to test the functionality of the program during the iterative development section of the project. I will test if all of the text is updated into the database tables.

For the post development phase, the test data will be the same. It will be used to test if the databases are updated or not.

I have undertaken cloud database testing. The cloud environment used is my WD Mydrive device linked to my internal LAN home network. This has been used on test data. When the program has been rolled out to Varndean College to capture grades, a cloud server will be required for this part of the program to operate.

## Test data table

### Registering and logging in

This table shows which data is going to be used for the iterative development of the solution. It justifies which data is going to be used, the data type and why justification as to why the data is going to be used.

|  |  |  |
| --- | --- | --- |
| Test data | Purpose/Justification | Data type |
|  |  |  |
| TestUser | This will be username which I will use to register and login | String |
| Test first name | This will be the first name which I will use to register | String |
| test@test.com | This will be the email which I will use to register | String |
| Test last name | This will be the last name I will use to register | String |
| Test | This will be used as a password while registering an logging in | String |

For registering, I expect the test data to be added to the PHP database.

### Gradebook

|  |  |  |
| --- | --- | --- |
| Test data | Purpose | Data type |
|  |  |  |
| Name | This will be the name for the student | String |
| 7 | This is to enter in the “academic year” part of the form | Integer |
| Good test | I will type this in the field where the test notes are | String |
| Maths test | I will type this in the field where the input for the name of the test is | String |

For the gradebook, I expect the test data to be added to the SSMS database.

For the register form, there will be a Boolean value which the user can click if they’re a teacher or a student. The test data will be a click of the mouse to select if the user is a teacher or a student.

For the teachers’ grade form, there will be a Boolean value which the user can click if want to add a teacher or a student. The test data will be a click. The test data will be a click of the mouse to select if the added entry is a teacher or a student.

## Post development test data

### Identified and justified any further data to be used in the post development phase.

The data used in further development will be the same as the data used during the iterative development. This is because there will not be any change in the inputs. All of the inputs will be text based (entering grades and names) into the system.

## Test plan

I created a test plan in order to document the development of the program.

For the next section I will get the success criteria from the analysis section and paste it into a spreadsheet. Then I will use the success criteria to test the program to see how much of this has been fulfilled. This is for the testing during iterative development, the validation and the review of key points.

|  |  |  |
| --- | --- | --- |
| **Key:** | **Works** | **Does not work** |
|  |  |  |
| **TESTING** |  | **Does it work?** |
| 1. The program opens. | |  |
| a. The login form will be the first part to open in the program. | |  |
| b. This is so the user can start to use the program. | |  |
| 2. You can enter text into the text boxes. | |  |
| a. This is required so that the user will be able to enter their details. | |  |
| b. I will do this by creating textboxes to enter text. | |  |
| 3. You can click on the buttons. | |  |
| a. This is required for the user to add their details to the database in which is being held in. | |  |
| 4. There are buttons to move to and from the register for and the login forms. | |  |
| a. If you don’t have an account, you may make one by going to the “Create an account” form using a button as a medium to get there. | |  |
| 5. A database must be created. | |  |
| a. The database is where the details of the users are stored. | |  |
| b. The database is where their grades of the users are stored. | |  |
| c. To do this, I will use PHPMyAdmin | |  |
| d. I will also use SQL Server 2019. | |  |
| e. SQL Server 2019 will be used to create a local server in which I can create the database. | |  |
| f. Along with SQL server 2019, I will use Microsoft SQL Server Management Studio (SSMS) to create a database with tables to store user details. | |  |
| 6. A connection to the database must be established | |  |
| a. This is so that the information entered may have a medium in which they can be stored. | |  |
| b. To do this, I will use the MySQL Connector package, which will allow me to connect to any database easily. I will also use XAMPP which will give me a port which I can use to connect to the database. | |  |
| 7. A table with columns corresponding to the fields in the “Create an account” form must be created. | |  |
| a. This is so the data can be sorted easily. | |  |
| b. I will do this through the PHPMyAdmin interface and name the columns with the same names as the ones in the “Create an account” form. | |  |
| 8. The text entered into the fields of the “Create an account” form must be put into a database as columns. | |  |
| a. The user must have somewhere where their login details can be stored so they are able to login. | |  |
| b. To do this, with the established connection mentioned in point 2, I will create tables in the database which correspond to the text box fields in the “create an account” form. | |  |
| 9. Text entered into separate fields must be put into their corresponding columns. | |  |
| a. This is to make distinctions between the user’s different details, like their name, or the grade which they attained. | |  |
| b. To do this, I will link the fields to the columns in the table with SQL. | |  |
| 10. The user must have a chance to select if they are a teacher or a student. | |  |
| a. This is so that they can have different priorities. The student will only be able to view their grades while the teacher will be able to add grades and edit their grades. | |  |
| b. I will do this by creating a comboBox which have a teacher selection and a student selection. | |  |
| 11. Pressing the “Create the account” button will put all of the information into the table. | |  |
| a. I will do this using the steps above. | |  |
| 12. All of the fields in the “Create an account” form must not be null. | |  |
| 13. After creating the account, the program will show the login page. | |  |
| a. This is so the user can login. | |  |
| b. To do this, I will make a function which will make the “Create an account” form close and the login form open. | |  |
| 14. The text boxes should accept input and the login button should be able to be clicked. | |  |
| a. This is so the user can login. | |  |
| b. I will do this by creating a button and textboxes for the user to login | |  |
| c. The user must be able to select if they are a teacher or a student. | |  |
| 15. The username and password will be verified. | |  |
| a. This is so that only authorised people will be able to access the gradebook | |  |
| b. I will do this by using SQL to verify if the inputs in the database and the login input boxes in the program are the same. | |  |
| 16. A new form will be opened. | |  |
| a. This form will be where the student’s and teacher’s data will be stored. | |  |
| b. I will do this by creating a function which will link the login and the info form together. | |  |
| 17. The form will have multiple fields which you can type in. | |  |
| a. This is so that the user can type in their details such as their name and grade and which year they’re in. | |  |
| b. I will do this by making text boxes. | |  |
| 18. I need to create another database | |  |
| a. This is so that the information mentioned in 17a can be stored somewhere | |  |
| b. I will do this using SQL Server 2019 | |  |
| c. Each user should have their own table for their own information. | |  |
| 19. The database needs to have a table | |  |
| a. These are required so that the information mentioned in 17a can be sorted appropriately | |  |
| b. I will do this in SQL server by creating a table in the database | |  |
| 20. The database must be connected to the application. | |  |
| a. I will use the SQLClient libraries in C# to do this. | |  |
| b. This is so the entered information will be able to go the database. | |  |
| 21. The information entered must be saved into the database | |  |
| a. This is so that the student can track their grades. | |  |
| b. I will do this by connecting the database to the application. | |  |
| c. When a “save” button is pressed, the database is updated with the relevant information. | |  |
| 22. There is a button which will allow the user to view their grades. | |  |
| a. This will allow the user to track their grades. | |  |
| b. I will do this by establishing a connection to the database. | |  |
| c. The information will be displayed in the application as a table. | |  |
| 23. If the user is a student, they can only view their own grades. | |  |
| a. This is for privacy reasons. | |  |
| b. I will do this by creating a separate table in SSMS for each user when they create an account. | |  |
| c. This will be done while the application and the database are linked. | |  |
| 24. Teachers will have more access than students. | |  |
| a. This means they will be able to view edit grades of the students. | |  |
| b. I will do this by using SQL commands to edit the tables. | |  |
| c. Students will only be able to view their own grades but not edit them. | |  |

# Developing the coded solution

### Provided evidence of each stage of the iterative development process for a coded solution relating this to the break down of the problem from the analysis stage and explaining what they did and justifying why. – Explanation of below illustrations

Provided evidence of prototype versions of their solution for each stage of the process. – Below illustrations

The solution will be well structured and modular in nature. – C# code files submitted separately.

Code will be annotated to aid future maintenance of the system. – In the screenshots of the code there are comments to explain what the code does. I have also explained what the code does under the screenshots

All variables and structures will be appropriately named. - C# code files submitted separately.

There will be evidence of validation for all key elements of the solution. – There is validation of the code working compared to the test plan

The development will show review at all key stages in the process. -I have reviewed what each section (key stages) of the program does after the points above.

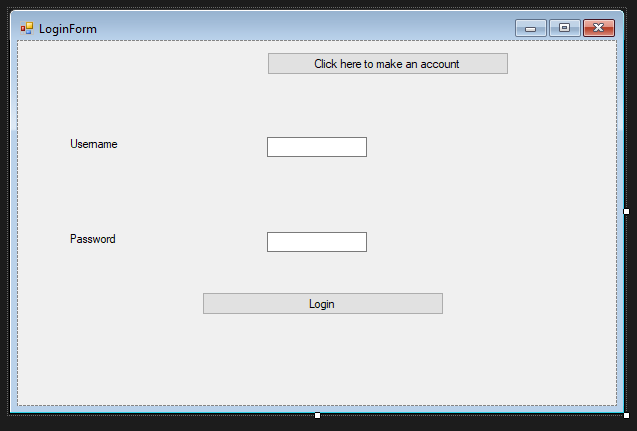
Provided evidence of testing at each stage of the iterative development process. – There is a heading which shows where the testing is

Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken. – Below I have shown the failed tests with explanations as to what I will do to fix it.

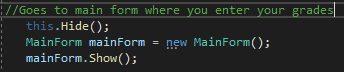
### Designing the forms:

As the program is being developed, I will record the code in each stage of the iterative development process.

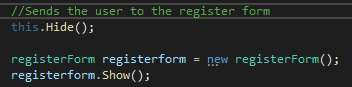
The login form design has been modelled off the login form design in the design section.



There is a box to enter your username and a box to enter your password. If you click on the login button, it will lead you to the main form where the user adds their grades. The code for this is illustrated below.



The “Click here to make an account” button will lead you the register form. The code for this is illustrated below.



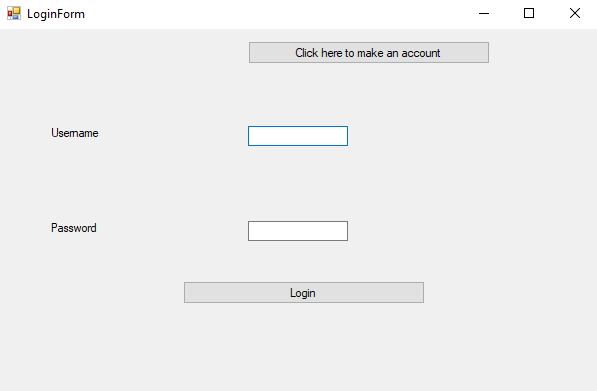
Typing in the password field should make asterisk appear instead of the letters so the others cannot see the password. I did this in the properties section of the textbox.



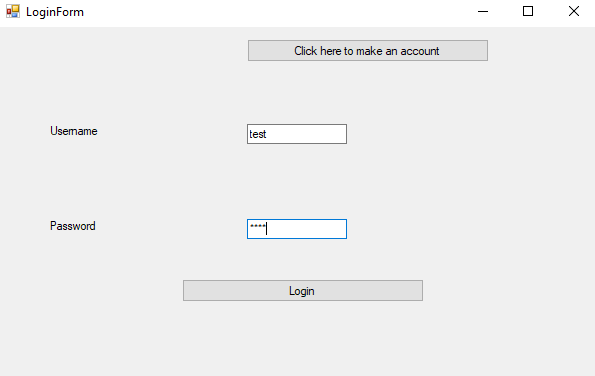
The picture above illustrates that anything which will be inputted into the textbox will be shown to anyone looking at the screen as a “\*”.

### Testing:

When I start the program, the login form opens.

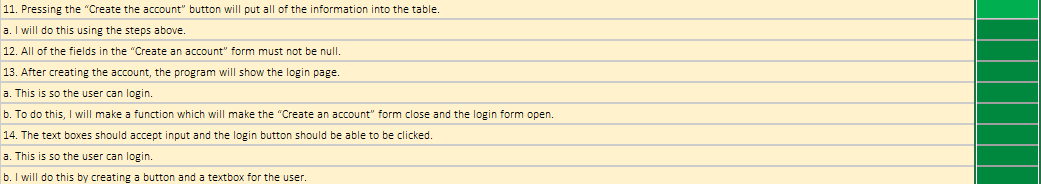


You can also enter text into the text boxes. The password is hidden.



The “testing” picture below is a picture of the success criteria from the analysis section in the form of a graph. There are 2 tabs, 1 with the success criteria and 1 which has the purpose of showing if the success criteria requirements have succeeded. If the program fulfils the success criteria requirements, then I will colour the tab green. The testing to inform development will be done using the green check boxes.

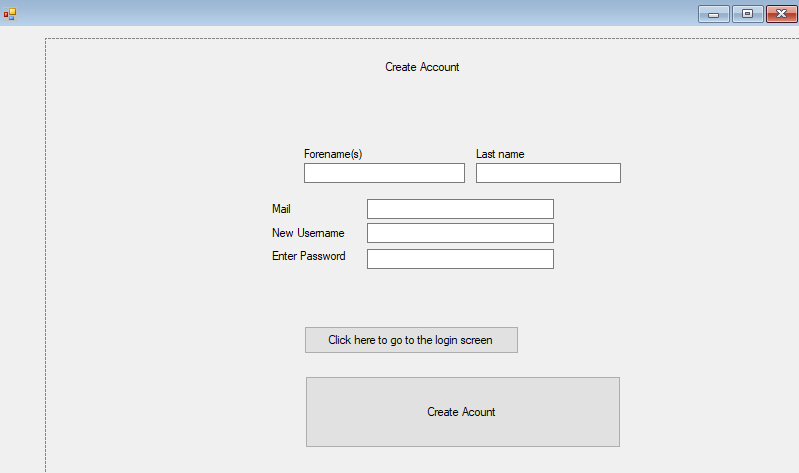
The testing requirements below have been fulfilled because you can enter text and click on the buttons to log in. There is also a button and a textbox for the user.



The register form will be the next form which will be documented and tested

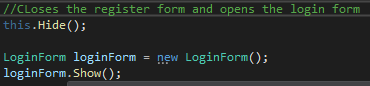
When the user clicks on the “Click here to make an account” button, the register form must open.

The register form has been based off the design in the design section:



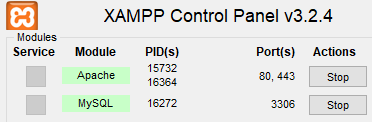
As you can see below, the register form has opened after the user clicks on the button to open the register form. There are text boxes to enter in your details

The button above the “Create account” button will lead you back to the login screen.

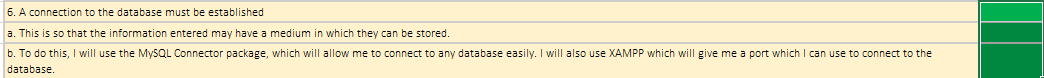


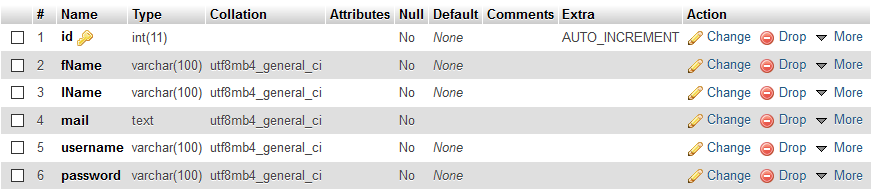
As you can see above, the code is annotated and shows when you click on the login screen button, it will open the login form and close the register form.

Next, I had to download XAMPP in order to connect the program to a database (PhpMyAdmin). After opening XAMPP, I opened the ports to Apache and MySQL in order to use SQL queries and to connect to the PhpMyAdmin database. I also had to download the MySQL connector package to use SQL queries. Also, XAMPP and Microsoft SQL Server Management Studio both use MySQL.

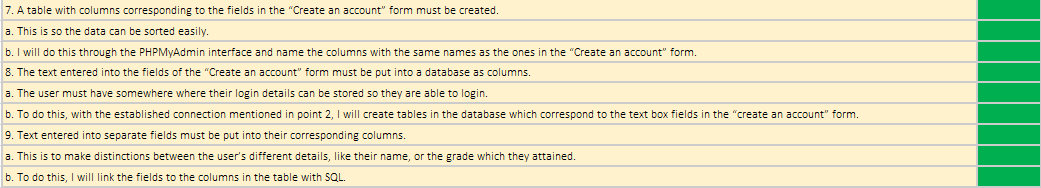
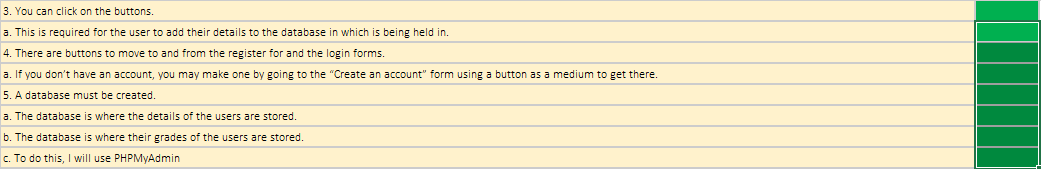
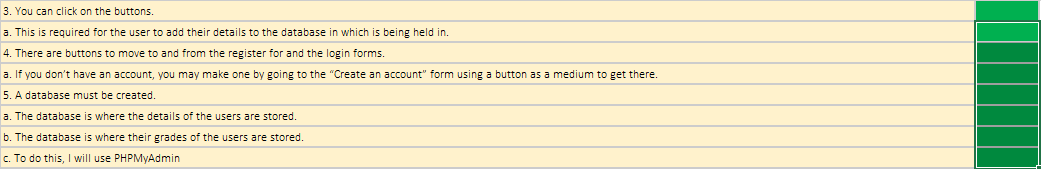


By connecting to the database, I have fulfilled the points below



Next, I created the database in the PhpMyAdmin interface and created the columns where the user data would be held which is illustrated below. 

As you can see below, the success criteria have been filled. I created a database with XAMPP. The users’ details are stored there. I used PhpMyAdmin for the database. The names of the columns in the database correspond to the names of the text box inputs in the register form.



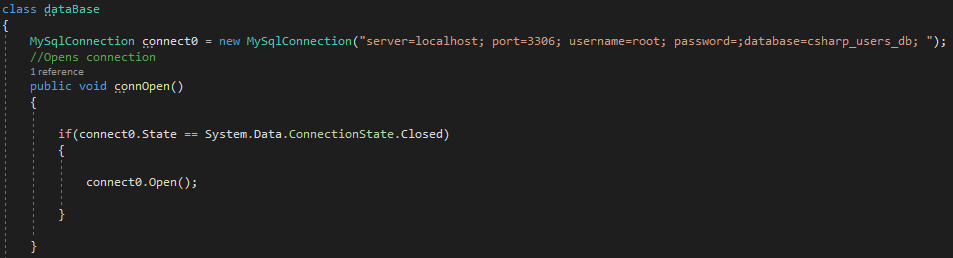
## Review at all key stages

### The development will show review at all key stages in the process.

I downloaded XAMPP to connect to PhpMyAdmin. A PHP database is created using PhpMyAdmin. There are rows which are where the users’ data is to be stored (with input from the register form). The names of the columns correspond to the labels next to the textboxes.

### Database.cs code

To connect the database, I created a new .cs file named “dataBase.cs”. In there, I created a class called “dataBase” which is responsible for opening and closing the connection to the database. Then, I typed some code which connects the database to Visual Studio. I used the ports given to me by XAMPP and made an SQL query in the code to connect to the database.



## Database Review

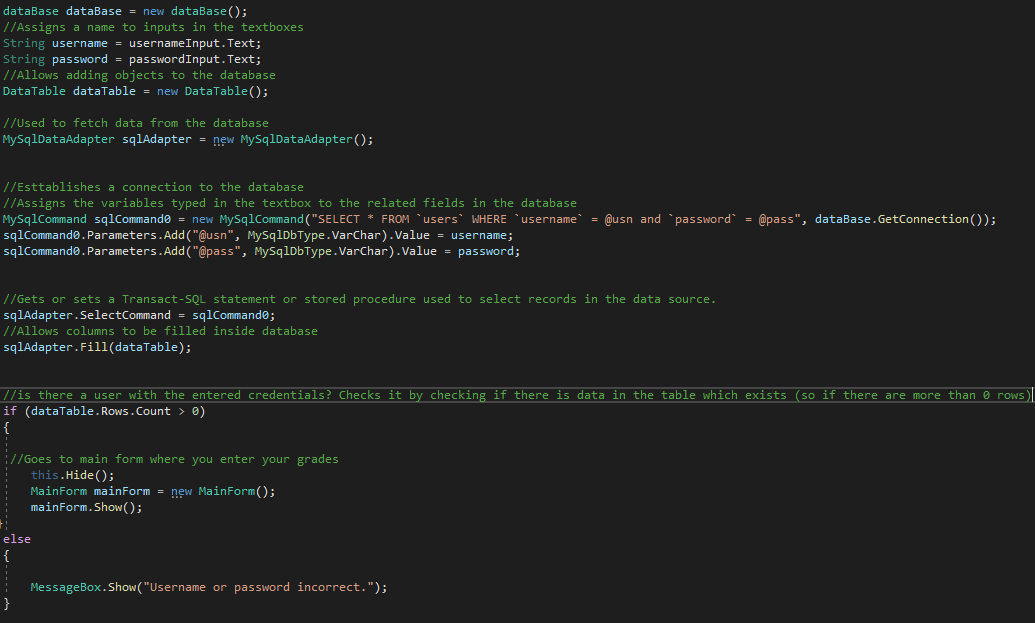
### The development will show review at all key stages in the process.

Opens a connection to the database using XAMPP. It works well.

### Login Form Code

Then, I went into the login form. It’s a new form, so I had to open a connection again.

The next thing I did was make a function which would verify if the inputs in the text boxes in the login form were equal to the ones in the database.



The code assigns the username and password input boxes variables. This is so that the code can use the variables to compare and verify data in the textboxes with the data in the database.

The code selects all of the usernames and passwords and compares them to the ones typed in the login form text boxes. It does this through SQL. The sql adapter allows you to type code in Visual Studio. The sqlcommand uses a variable name to store the username. It assigns the username and password to @usn and @pass. Then, it adds the text box username to the database with the sqlcommand0.parameter, this is only temporary to verify if the database value. Then, it checks if the username is in the table. It does this by setting the entered username and password into the datatable, but only if they match a username and password in the PHP database. Then, if it’s inside the entered username and password match values to the ones in the PHP database, the datatable is updataed with the entered username and password. Then, when entered, the program checks if the datatable has more than 1 row (because the datatable is only filled if the entered username and password match the PHP database username and password). If it does, then the main form where the grades are inputted is opened.

The “MainForm” (which opens if the username and password are correct) is the form which is where the users’ grades are inputted. Currently it is a blank form.



## Login Review

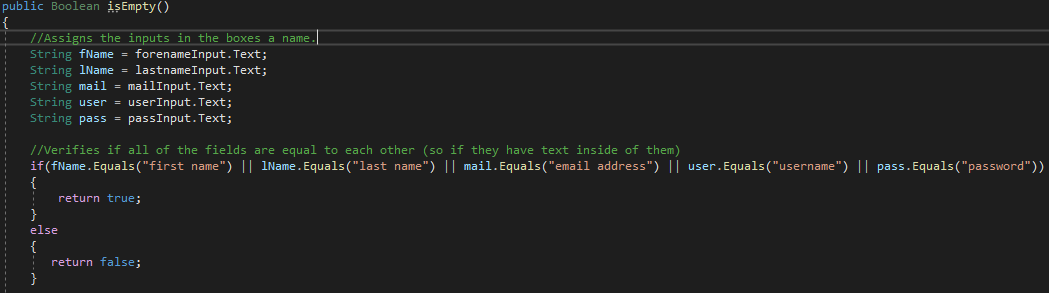
### The development will show review at all key stages in the process.

Here is the code for the login form. It connects to the PHP database and verifies the username and password. It does this by storing all of the usernames and passwords into the database with SELECT \* using the sql adapter. It stores the entered username and password in a datatable and verifies if the values in the datatable are the same as the values in the PHP database. Other key stages are explained above.   
This is checked during the iterative testing.

### Register form code

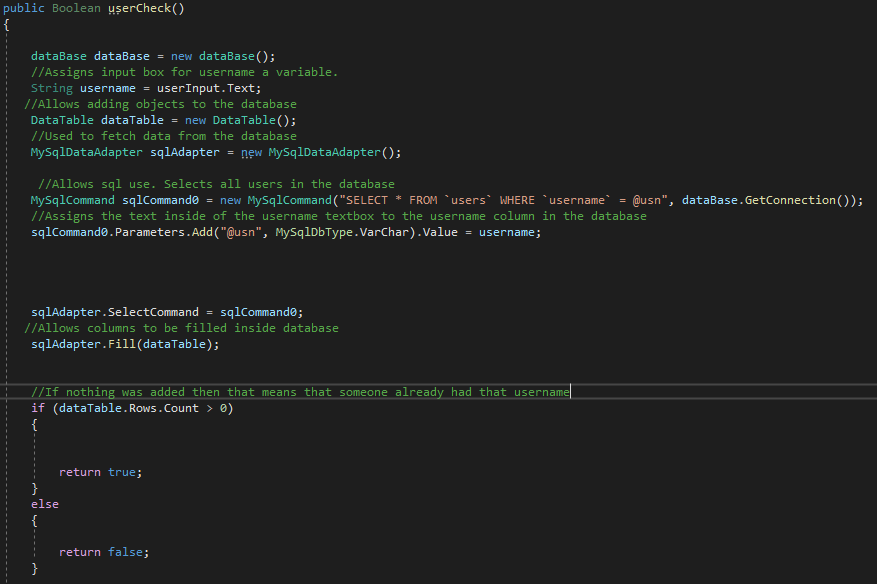
Next, I moved onto the register form.

After that, I typed some code which checks if the values in the textboxes are empty or full for the fields in the register form. It is a public Boolean called “isEmpty”. I assigned what the user inputs in the textboxes to string variables. If the fields are empty, then it should return false. As a shortcut, I used the “short-circuiting or operator” which only goes to the next input if the previous is false. In this case, the previous input would be false if it was empty. I did this instead of typing multiple “if statements” which would have wasted a lot of time. It also saves CPU resources because if the first field is empty then it does not check if the other fields are empty.

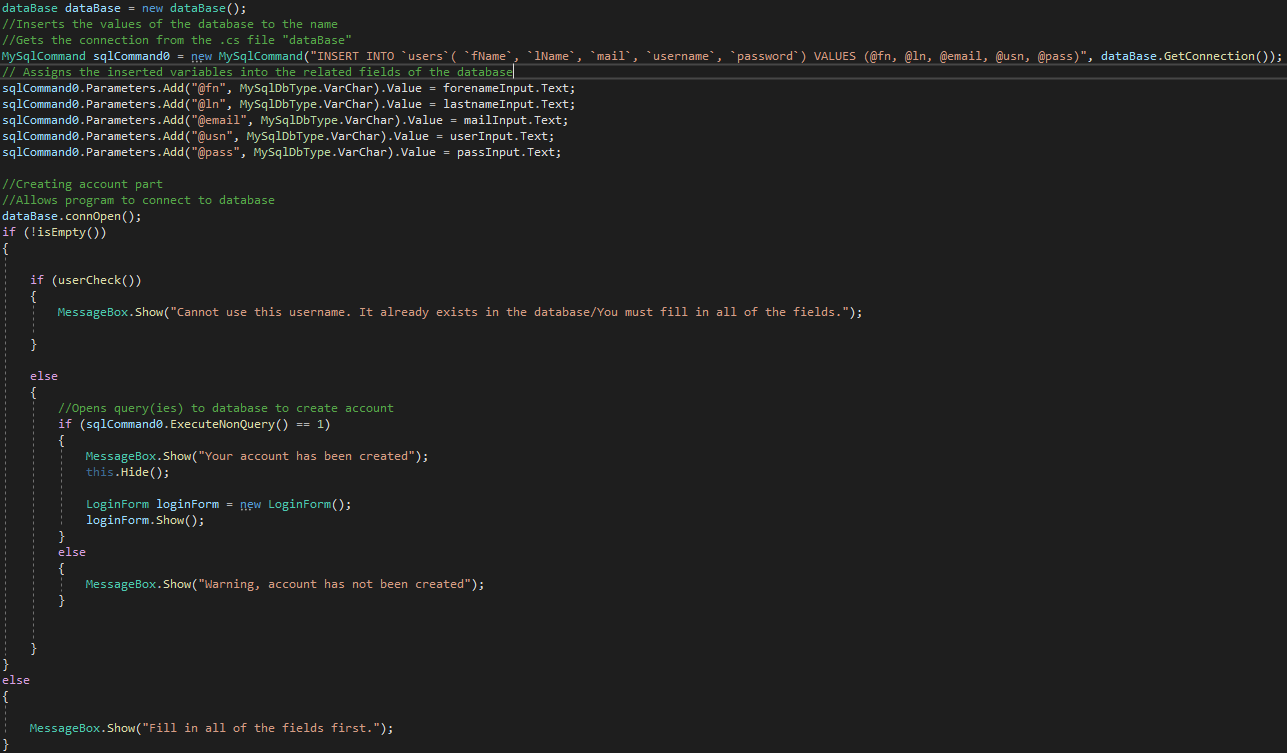


The next function I typed was to prevent duplicate usernames. If two people had the same username then they could use both of their passwords. This is a security flaw because someone can access someone else’s account by accident.

To do this, I assigned the username text box a variable name, which was “username”. There some reused code from the login form. The comments explain the datatable, the sqlAdapter and the .fill code. I commented the code here so I won’t have to comment it anywhere else where I reuse the code for concurrency.



Next, I typed the code for the button which happen when you press the “Create Account” button, which is in the register form. Firstly, all the fields would have to be full because there is a condition where your information will be saved into the database only if there aren’t any empty fields (!isEmpty). Also, you can’t have the same username as someone else otherwise an error message will appear. This is illustrated in the else statement where the if (usercheck) statement appears. ExecuteNonQuery is an SQL code which won’t return any data. It is made for editing. It only works for update, insert and delete statements.



If all of the requirements are met (no duplicate usernames and no empty fields) then the user is added to the database and the login form is opened.

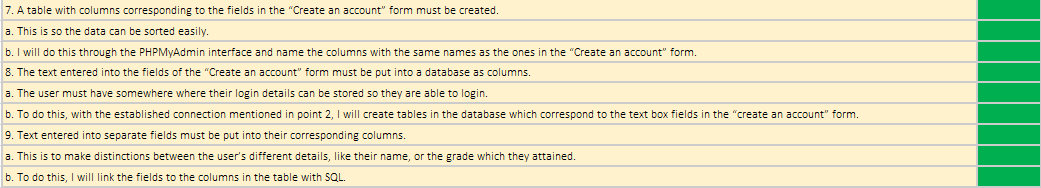
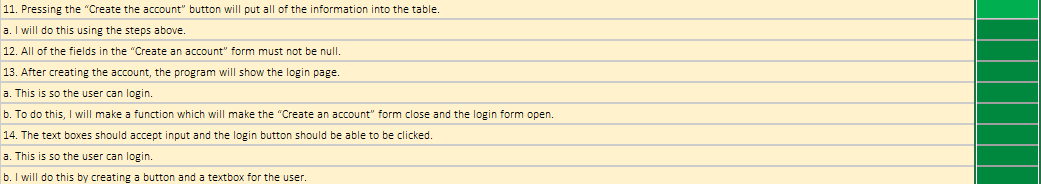
## Register review

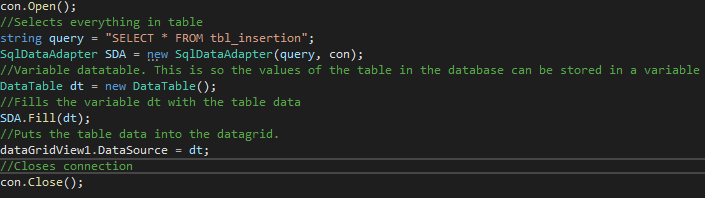
### The development will show review at all key stages in the process.

### 

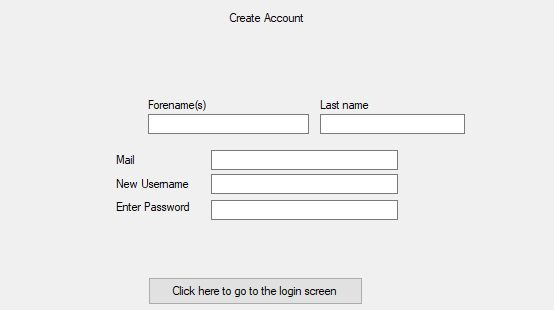
This part of the program (the register form code) took the longest time to implement. This is because there were lots of tasks which needed to be fulfilled.   
I have created the register button. I also had to learn how to use the code which was inside of the sql adapter package/library. The sql adapter functions are explained either in the annotations/comments above, or in the screenshot below with the comments, or in other parts of the iterative development. All of the screenshots of commented code above show that a user can be created and added into a phpMyAdmin database.

The points below are satisfied by the code above. Pressing the “create an account” button will add the values which correspond to the rows in the PhpMyAdmin database. You can also see the SQL which connects the database columns to the input boxes by assigning the input boxes to variables.

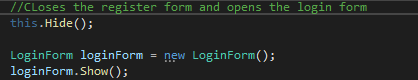




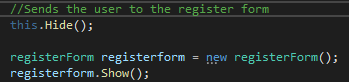
### Testing:

You can move between the login form and the register from.

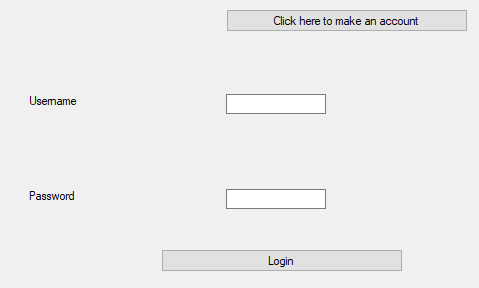
When the “go to the login screen” button is pressed, this code executes:

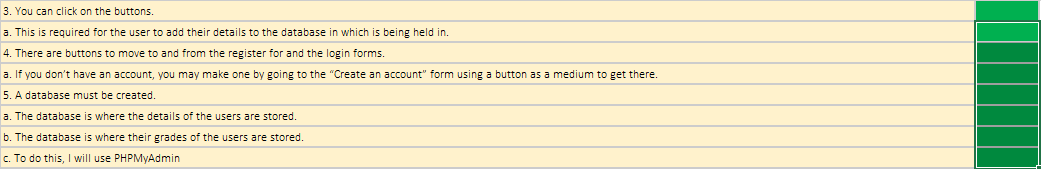


In the login form, if you click on the “make an account button”, this code executes:



Below is a picture of the button in the login form.



This satisfies the points below.

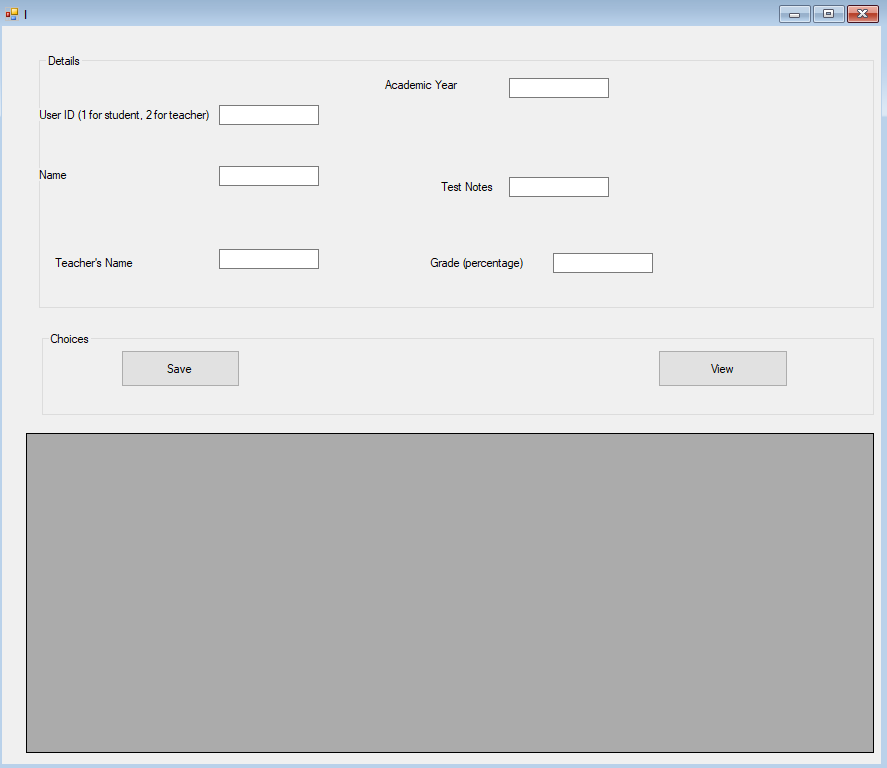
## Testing Review

### The development will show review at all key stages in the process.

Here you can see that you can move between the register and login form by clicking on the buttons which are on both of the forms.

### Main Form Design:

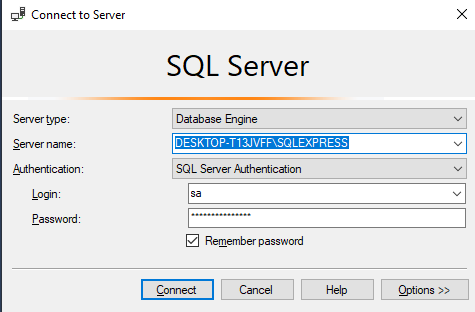
I created this form using the design which I made in the design section.

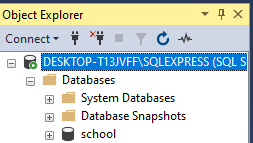


The grey box is the dataTable box where the table of data will be displayed from the database. The textboxes are where the user puts all of their information. They will press the “Save” button to save the data into the table. The “View” button displays the data in the table.

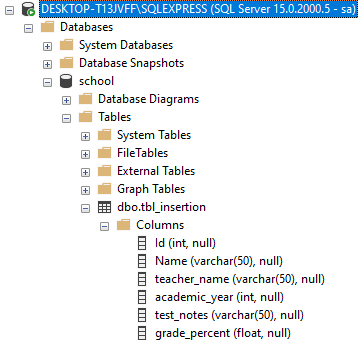
### SQL Server 2019

I downloaded SQL Server and set up a local server. I set a password for security. After that, I opened Microsoft SQL Server Management Studio (SSMS) and created new database called school after logging into the local server which I created. The pictures below show this.

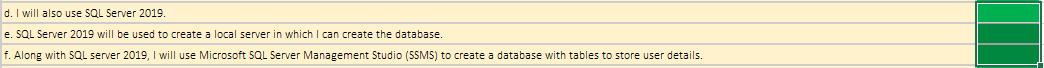
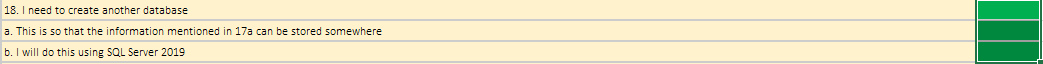




After that, I created a table with columns. They all had appropriate names which are similar to the labels of the text boxes where you enter your information.



As you can see, there are columns with relevant names.

As that is done, I have fulfilled points 5d, 5e and 5f and the other points below.

## Server Review

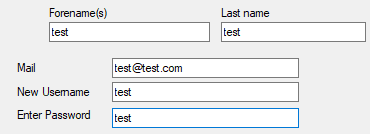
### The development will show review at all key stages in the process.

Here, you can see that I have created the database using SSMS and created relevant table names. Next, I will connect the database to Visual Studio. The server which I created is a local one. If you want to use a cloud on, you can connect the cloud server to your computer, and you can set it up with SQL Server 2019. I have given the database, the tables and the columns relevant names. The main form design created with relevant names.

### Testing:

### Creating an account and logging in:

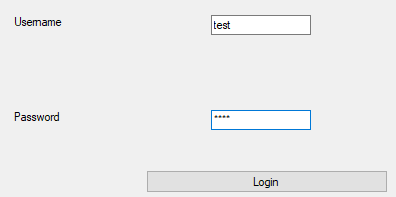
I have used this data to create an account:



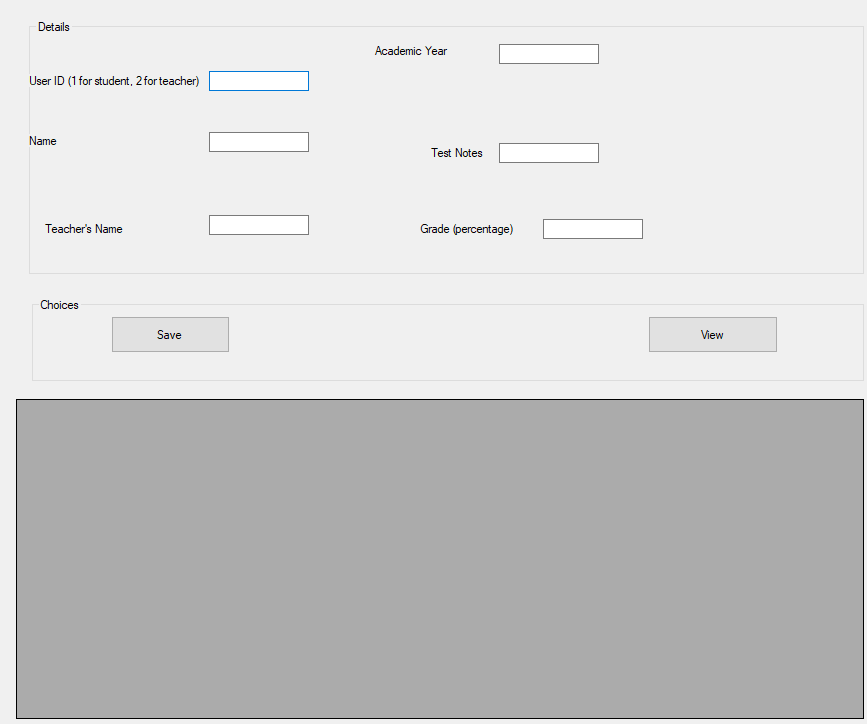
It was added to the database.



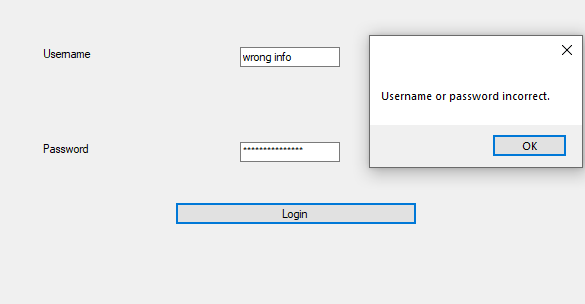
After signing up, the login form opened where I entered the test values and logged in.

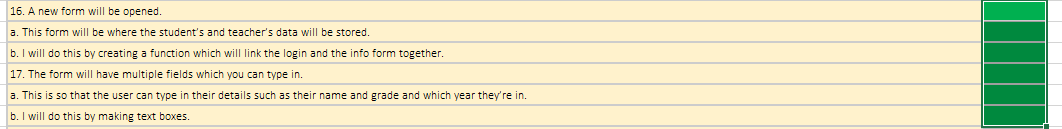


It moved me to this form:



If you type an incorrect username which isn’t in the database:



With that, these points have been fulfilled below:

16b has been fulfilled in the login section of the iterative development section.

## Login testing review

### The development will show review at all key stages in the process.

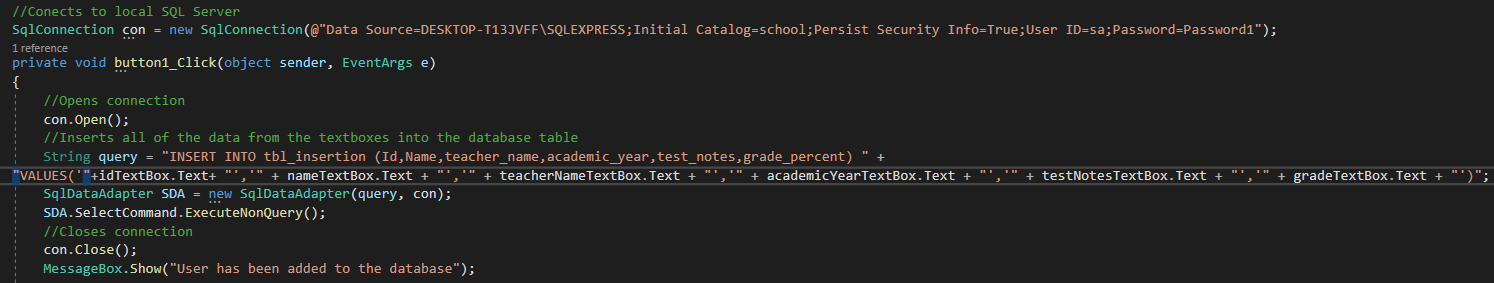
As you can see, the testing of the login form works, and the login disallows you from logging in if you enter incorrect information. It shows you an error box for usability purposes. If the inputs are correct and full then the form should lead you to the main form where you can enter your grades and your teachers’ name and information. I have also illustrated that the data which was entered in the register form has been added into the PHP database.

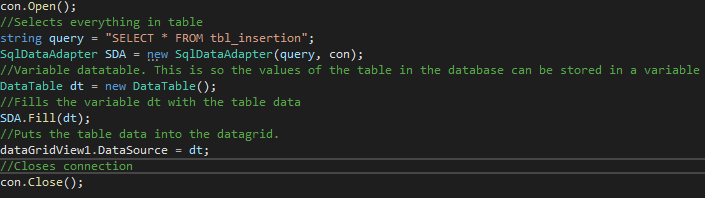
### Developing the gradebook:

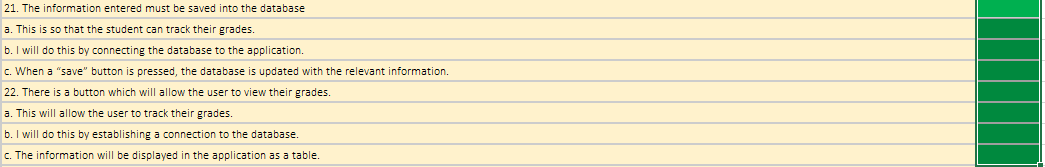
To develop the gradebook, I created and SqlConnection variable to connect Visual Studio to the local database



I opened the connection to the local database. Then, I used the INSERT function to insert the grades from the textboxes into the database. This happens when you click on the “Save” button.

The next part I developed is explained by the comments in the code.



The code above fulfils the points below.

I have used the SQLClient libraries in order to connect the databases to Visual Studio. You can see this below.

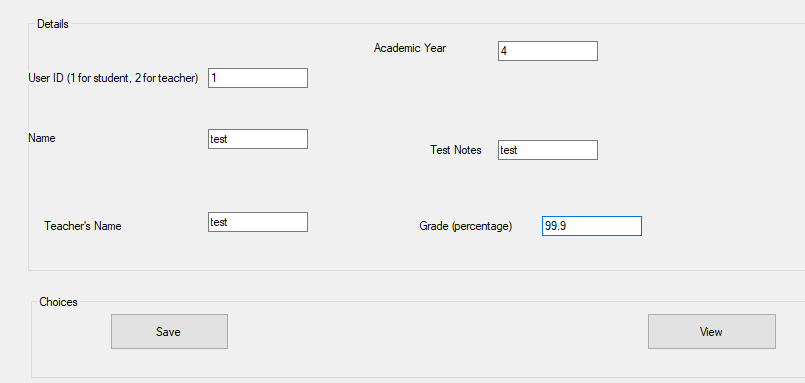


This fulfils these points below.

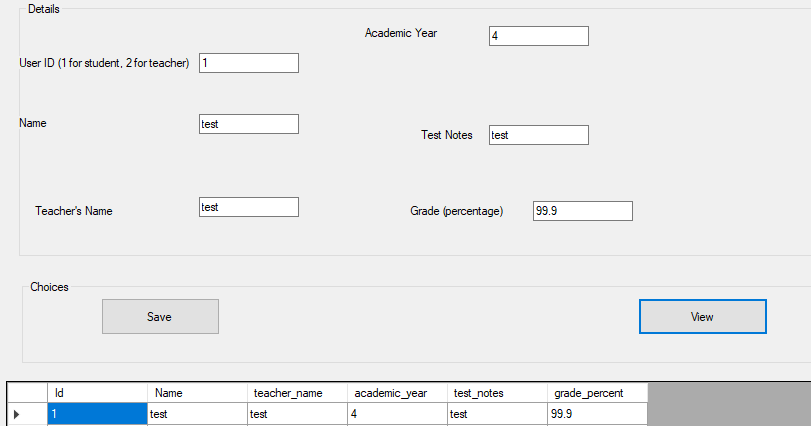
### Testing:

### Saving info to the database:

After logging in, I have saved info into the database. You can see this below.



Now, I am going to view it.



## Database test review

### The development will show review at all key stages in the process.

This is a key stage of the process. This is because it’s the end of the iterative development and it is the end of the main form testing. So, as you can see here, the purpose of the userid is to choose if the user is a student or a teacher. I have set this to an integer value, so any value can be taken. I can change the variable to a Boolean and add a combo box with 2 options, teacher and student. One which will return true. That will happen if you select student. If you select teacher, the box output should be false. To avoid confusion, I will change the variable name from id to isStudent.

Viewing the software works. It accesses the information using the SQL queries, data tables and the fill function. Most of the code is explained inside Visual Studio. You can see the comments in the screenshots which I have taken which explain the data tables and fill function and so on.

## Failed tests

### Provided evidence of any failed tests and the remedial actions taken with full justification for any actions taken.

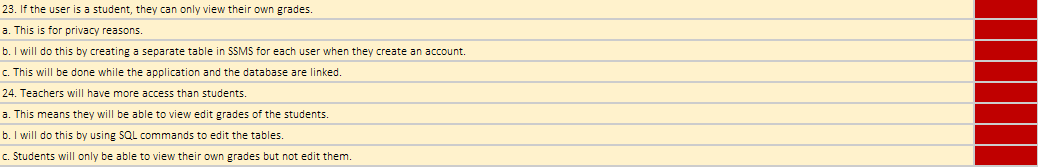
The failed tests are shown below

14c



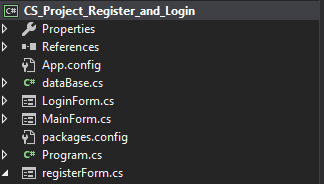
18c





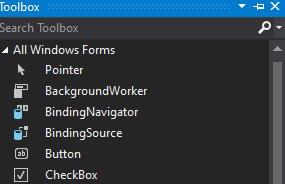
I could not complete these due to the complexity and time constraints required for these tasks. I will explain why I could not do this is the testing part of the evaluation.

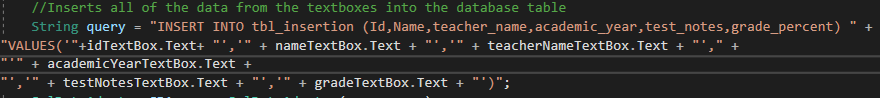
### The solution will be well structured and modular in nature.

The created solution is well structured. This is because there are different forms for different tasks as I did. Modularity is when you separate the program into independent modules. The different forms and subroutines are all modules of the program. 

There are different .cs files which are all different parts of the solution.

Modularity can also describe something that can be easily changed in the future, which leads to the next point of the future maintenance of the system. Since I have used Visual Studio C#, I can add features easily using the form designs with the provided toolbox.

  
Another example of modularity is how the code has been structured. An example of this is in the main form (where you can add your grades). In the SQL statement, if I want to add more fields to type in (e.g. parents’ name, or contact info), then I can add it easily by adding it to this part of the code.



The code has been designed so that you can add more features by adding a new column in the database and then creating another textbox (or any kind of input box) using the toolbox then going to the code and adding the name of the input box to the end of that last line of the picture.

## Code annotation

### Code will be annotated to aid future maintenance of the system.

An annotation is when you write comments on something to explain it. I have commented the code in order to tell which part of the code is which. If I want to change a feature of the solution for future maintenance and cannot find what I want to change in the code, then I can use the commented code to guide myself because the annotations explain the code. I can also use the comments to aid the future maintenance by adding features (such as the failed tests above) easily as I can tell where everything is because of the commented of the code, so I know exactly in which part I can type my code in.   
I can also refer back to this document where I have explained the code in more detail with annotations.

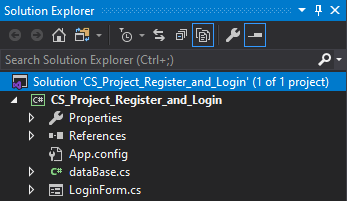
You can see the commented code during the iterative development section.

## Variable naming and structure

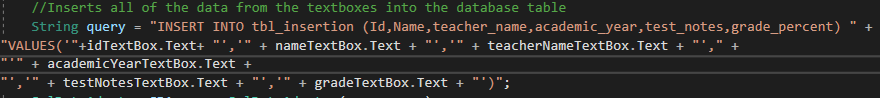
### All variables and structures will be appropriately named.

In the iterative development part, you can see that the database column names are appropriately named. This also aids modularity and future maintenance as I can understand what each variable does when looking at the variable name.

As you can see in the solution explorer, all of my forms (or .cs files) are named appropriately, so if I want to maintain the system in the future (for example the login form), I can just open the LoginForm.cs file in Visual Studio. If I want to edit the dataBase.cs file to change how the program connects to the database, I can click on that icon in the solution explorer in Visual Studio. You can see this in the photo below.



In this photo, you can see that each variable name corresponds to the names of the columns in the table.



Those are examples of appropriately names variables (in this example, the textbox names) and structures (the .cs files).

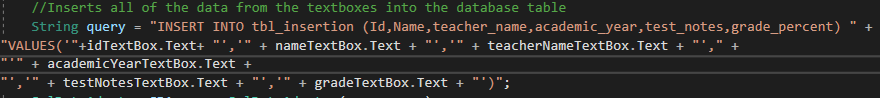
## Key validation

### There will be evidence of validation for all key elements of the solution.

The three key elements in the solution are the main form, the register form and the login form. The validations of these key elements have been carried out by assessing the success criteria during the iterative development with the testing diagrams coloured green for validating that the key elements are functional and red for if those key elements are not functional in the solution.

You can see the validation of the username and the password in the annotated and commented code which I did during the iterative development, with a picture of what happens if you type incorrect or correct details in the login fields. It shows you an error box if you type in incorrect details and moves you to the next form if you typed in correct details.

You can also see the validation of the SQL to the inputs of the data in the textboxes if you look at the picture below. When you type in the textboxes and click save, you can click view to validate the if the data is correct.



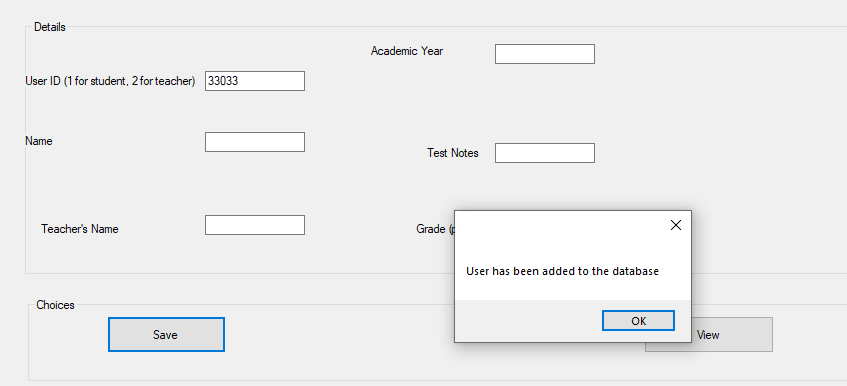
Another validation is when you register. I have shown in the commented code that the details you typed are validated using SQL. The data goes into the PhpMyAdmin database. I have shown this in the testing part of the iterative development above.

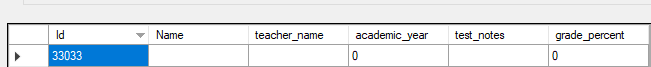
## Annotated validation test:

Most of the validation has been carried out in the video.

This annotated validation section will be part of the testing.

As you can see below, any value can be input into the User ID, if the user selects something other than teacher or student teachers will not know if the user is a student or teacher. To fix this, I will turn the textbox into a combobox.

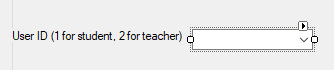




To fix this, I will force the user to only select 2 values.

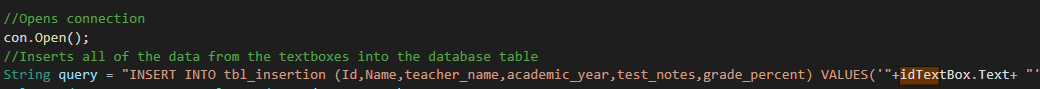
### Failed test fix (for validation):

The UserID text box was replaced with a comboBox.

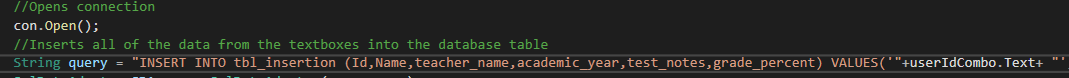


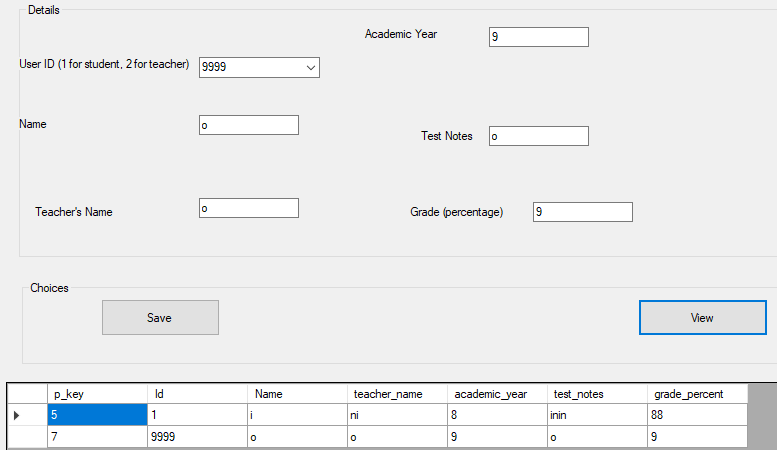
I created a comboBox which would only have 2 values, 1 for student and 2 for teacher. Now, the user can only select 1 or 2 (for student (1) or teacher (2)).  


I updated the code by deleting the name of the idTextBox.



Then I replaced it with the combobox name, “userIdCombo”.



This didn’t work as I could still type in the combobox, and the value was still added. 

Then, I went to the properties of the combobox and changed the type of dropdown box from “DropDown” to “DropDownList”. DropDownList prevents manual input.

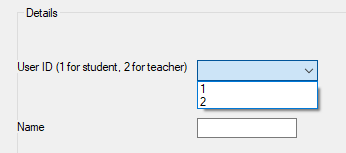




As you can see above, I changed it. Now, it is greyed out which signifies that manual input is not allowed.

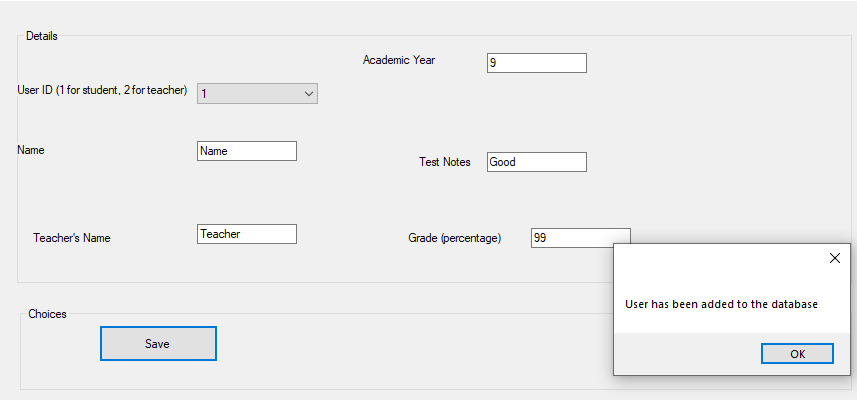


Now, you can only select 2 options.

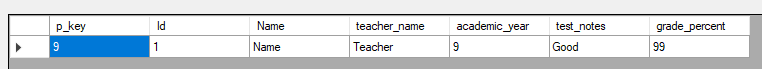


The verification has been complete.

You can now see the user being added.



Here it is



## Annotated validation test:

For the academic year, you can also enter any value. This means that you can enter “year 100” when there is no such thing as year 100. The teachers and students will get confused, so to fix this, I will create a combobox with set values (years 1 to 13).

I have entered a large value into the academic year box.



It has been added to the database.



### Failed text fix (for validation):

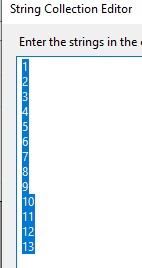
Now, I need to make a similar combobox as before for the academic year.



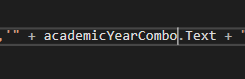
I’ve set it to a DropDownList



These are the values which you can select in the drop down box (up to year 13).



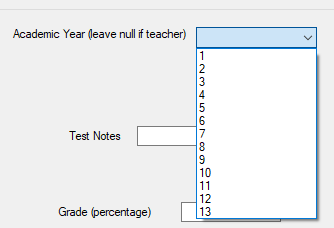
I changed the variable name in the code as well.



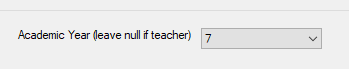
I also told the user that if they are a teacher, they should leave this field null.



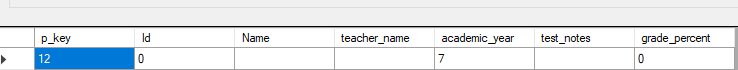
Now, when you open the program and click on the drop down box, it shows years 1-13



When I add it to the database, it works.



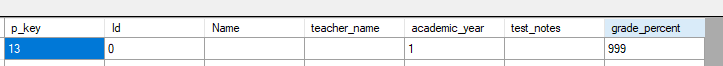
Here is the value in the table.



As you can see, it now works. Now the teachers won’t get confused anymore.

## Annotated validation test:

The next problem is that any float value can be set for the grade percentage (even above 100). This is bad because if a teacher is trying to calculate the average grade of the student then students can add values greater than 100 which will inflate their grade average.



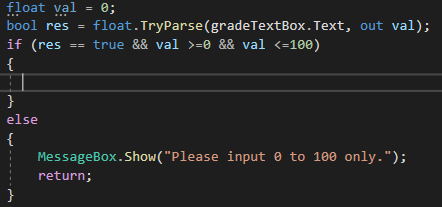
As you can see, I have added the value 999 to the table and updated it.

To fix the above, I will add some C# code which will limit the values from 0 to 100.

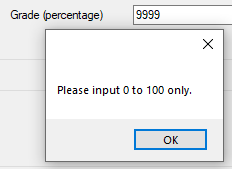
### Failed text fix (for validation):

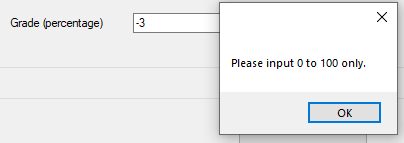
#### Annotation of code:

To fix the failed test, I have parsed the value of the textbox from a string to a float with this code and set it to a variable. Now that it has been parsed, I entered an if statement which would limit the values from 0 to 100.

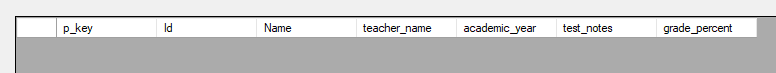


Now, when you enter in something bigger than 100 or less than 0, a warning message appears and it doesn’t let you add it to the database.





When I click on “View”, the table is empty.



## Validation review

### The development will show review at all key stages in the process.

I have shown most of the validations in the video, and the outstanding validations have been shown and carried out above. Where the validations didn’t work, I found a solution for them. Integer or float limits were common problems. The first two problems above were fixed by adding a drop down box without manual input (so users can’t add their own data).

The next issue regarding float limits was a little more difficult. I couldn’t resort to using simple controls like using the toolbox. I had to learn about parsing and apply it to the program using Boolean variables and an if statement with a float variable. In the if statement I also had to use the && (and) operator.

## Final Testing Review

### The development will show review at all key stages in the process.

The testing and the review have been joined together with the green checkboxes. The red checkboxes are tests which have failed and forms part of the review of key stages. The problems with the red checkboxes will be rectified in the evaluation section and in the video.

Ultimately, the review at all key stages are split up into the login section, the register section and the main form section. This is through the explanations of the code in between the screenshots of the code, which is how the review at all key stages are split up.

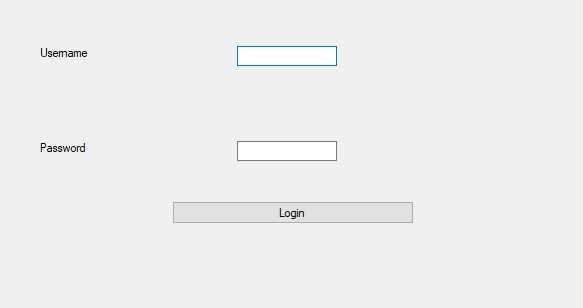
The iterative testing shows if the code in the program works, which is also part of the review in the key stages.

### Additional Testing

### Provided evidence of testing at each stage of the iterative development process

The program opens

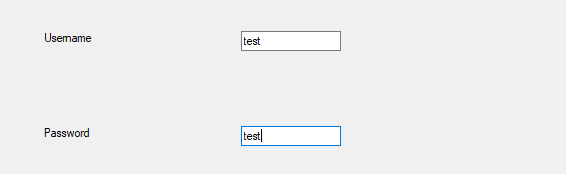
1.



As you can see, the program opens.

Testing for if the textboxes accept input for the username and password

2.



The program checking if the username or password are in the database and if they are correct.

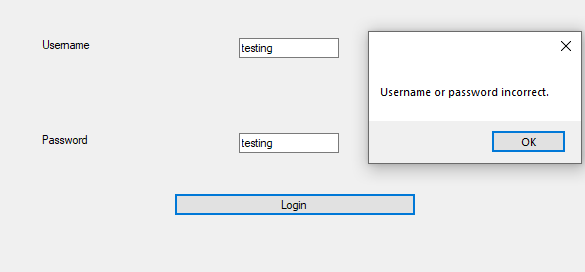
Picture of database:

3.



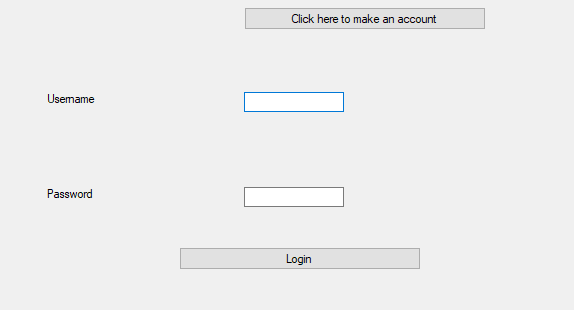
The error which appears if you enter something else other than the username and password in the database and if the login button works

4.



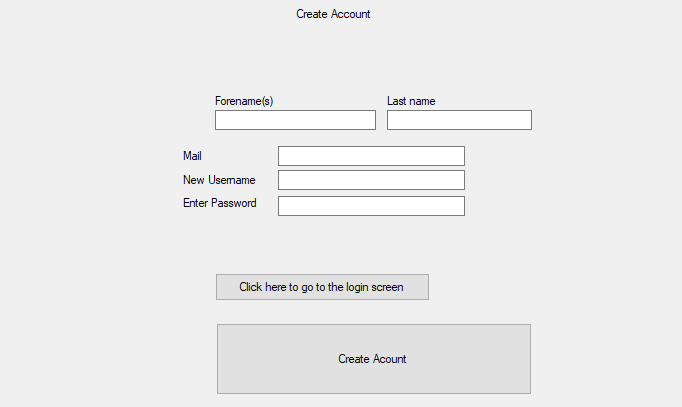
Testing if “Click here to make an account” button works or not:

5.



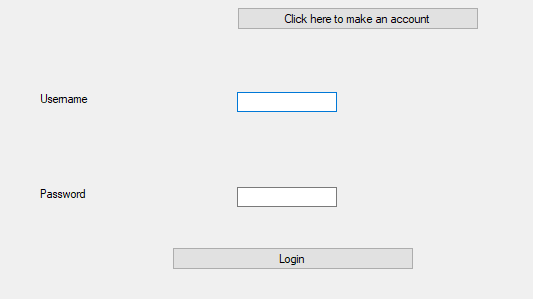
Once clicked, the program is prompted to open the “Create Account” form. The button functions correctly and the form below appears.

6.



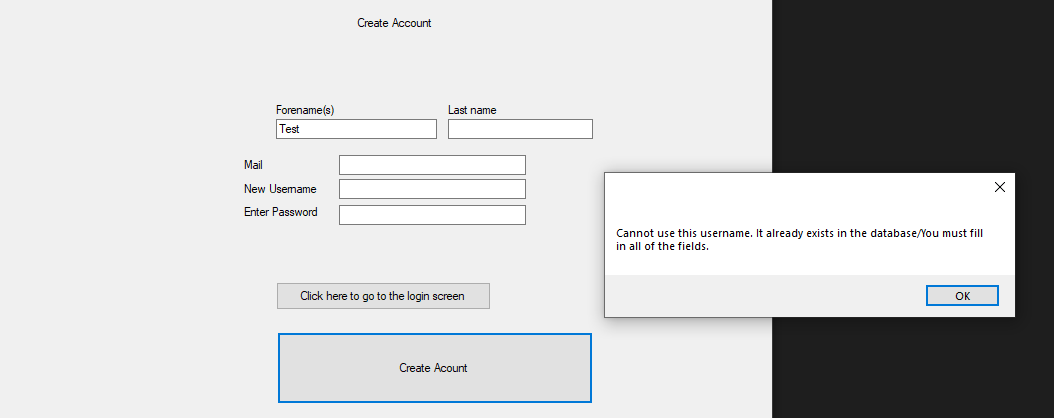
Clicking the “Click here to go to the login screen” button will prompt the program to go to the login screen shown below.

7.



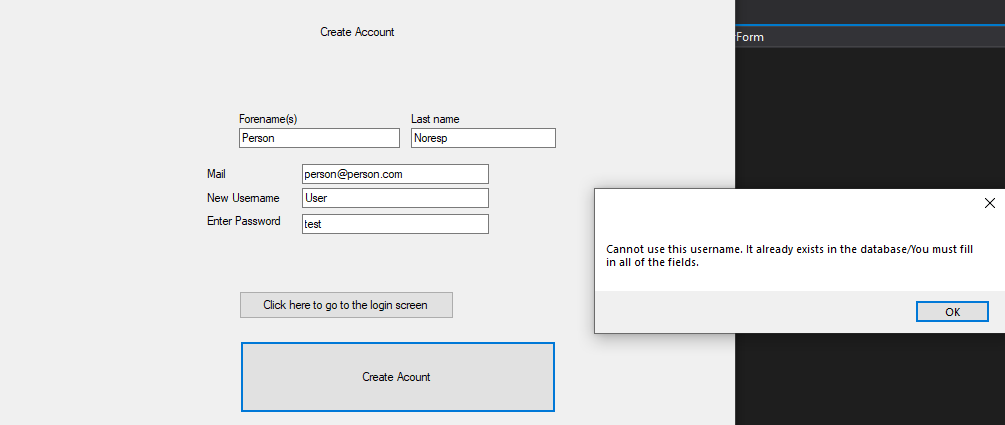
To create an account, you must fill in all of the fields. Failure to do so will result in an error message.

8.



When a user already exists, and you try to use their username to sign up, the same error message appears.

9.

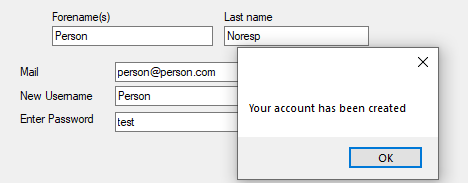


10.



After you type in valid data, the PHP database will be updated with the credentials.

11.

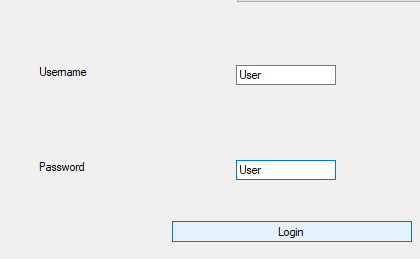


12.

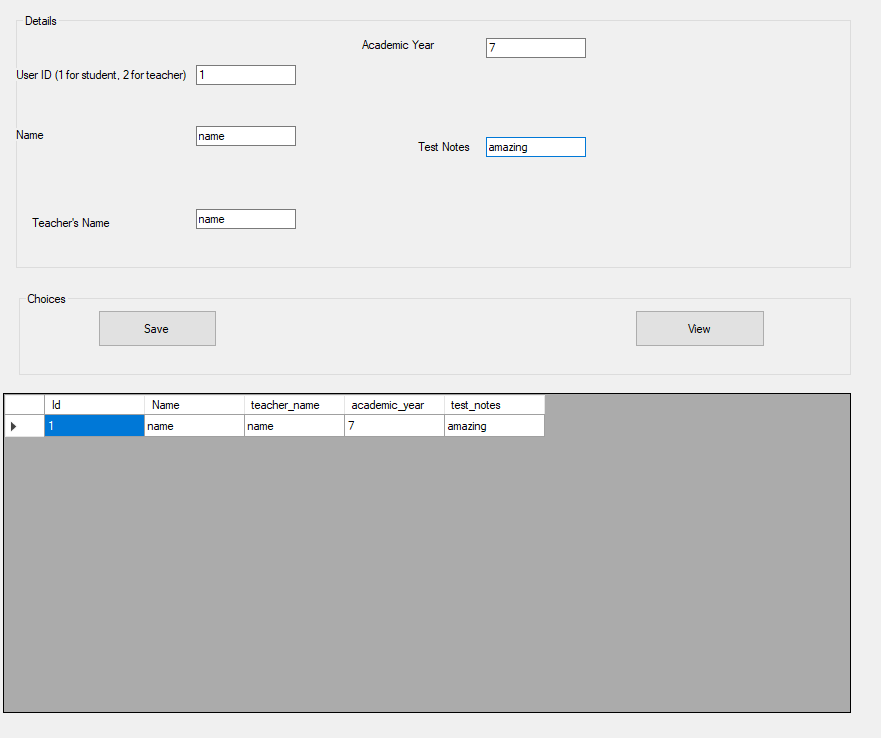


After that, the registration form will close. Then the login form will be shown on screen. Typing a valid username and password will lead you to another form where you can enter the student’s or teacher’s details. The details will be saved into a new database by pressing the “Save” button.

13.

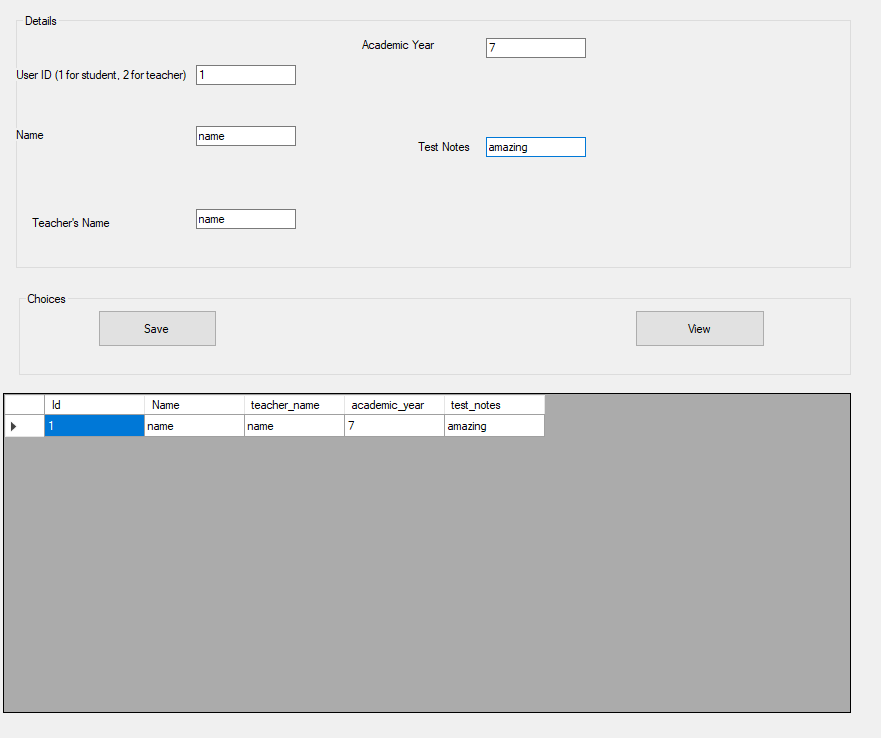


14.



When you press “view”, a list of all the students and teachers will appear.

15.

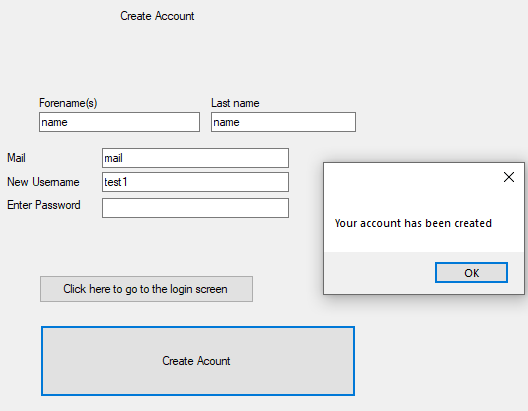


## FAILED TESTS

### Provided evidence of any failed tests and the remedial actions taken with full justification taken.

One failed test which I had was where I had not developed the program well enough which led to the program accepting null values which would be written into the database.

16.

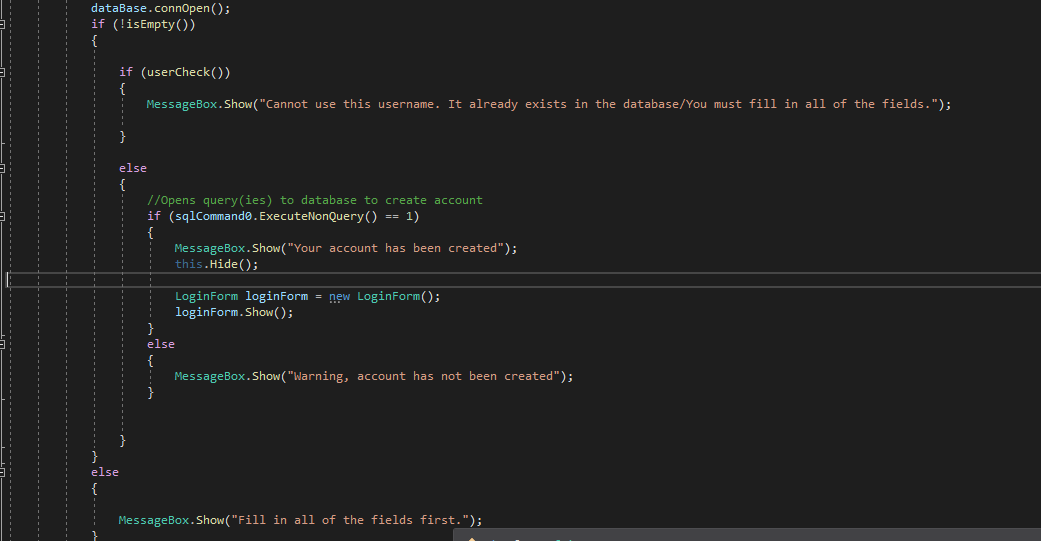


17.

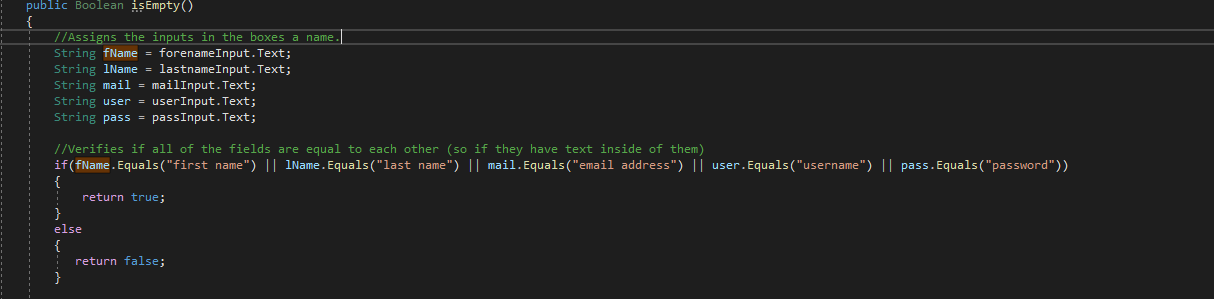


As you can see, the password field is empty, and the user had still been entered into the database. To combat this, I added this code, so that the user would be forced to fill in all the fields.

18.



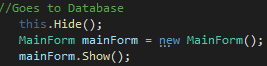
19.



The second picture is the code which checks if the fields aren’t populated. The first picture of the code is executed. If the fields are empty, then an error message appears, and the program doesn’t try to connect. However, if the all the fields are full, then the user will be added to the database.

Another issue was that pressing the login button on the main form would lead to the last form where you can view your grades without verifying if the username and password fields matched the data in the database.

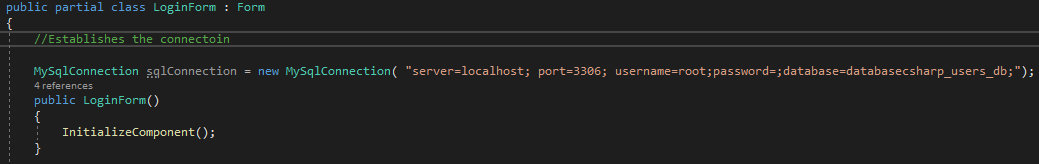
20.



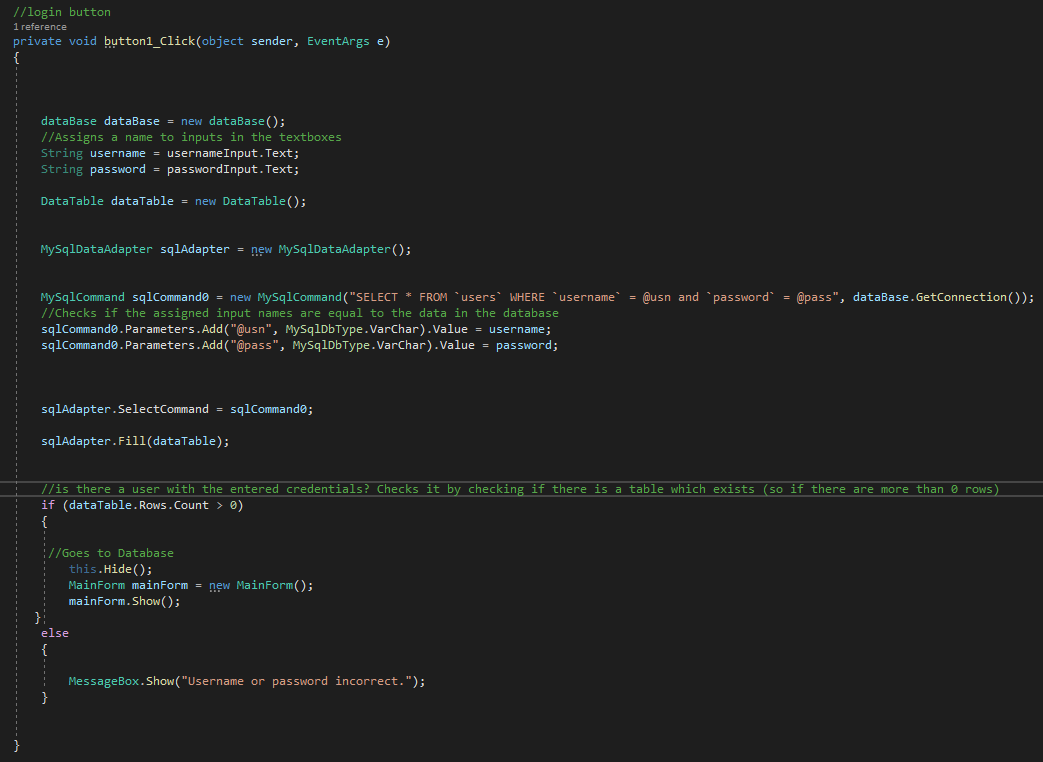
“MainForm” is the form in which you can view and add grades to the database.

To fix this, I compared the data in the database to the input in the text boxes. If they were the same, then you would only be moved to the grade book form only if one’s credentials matched the ones in the text boxes.

21.

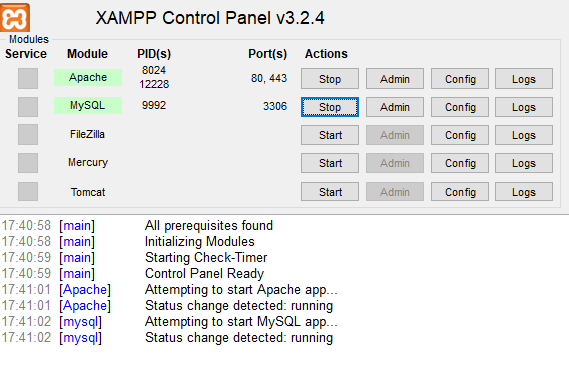


22.



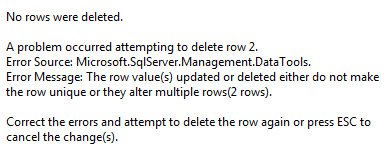
I chose the ports given to me by XAMPP.

23.



Another failed test was that I couldn’t delete any rows in the SSMS. If the teacher needs to delete rows. An error would appear every time I tried to delete a row.  
To fix this, I had to create a primary key, but it still didn’t let me delete the rows, so I had to use a custom SQL query to delete them in the SSMS interface.

The error:



The above error shows that the entries could not deleted because the entries were not unique. I identified that the primary key was missing, which didn’t keep the entries unique, so I decided to create a primary key.

The primary key:

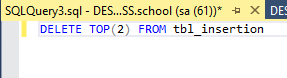


It didn’t let me save the primary key because of this error.



This error means that I had to delete the whole table. To prevent this, I used an SQL query which would force the rows to be deleted.

The custom query:

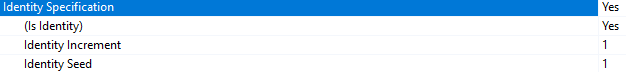


The top 2 rows needed to be deleted from the database, so the code above was used to delete them which refers to “TOP(2)” in the code above.

After that, only this placeholder row remained below, showing that the all the required rows had been deleted:

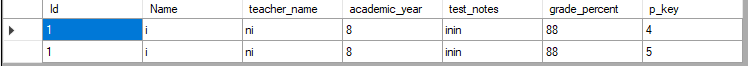


The primary key was saved. Now, I had to find a way to increment the primary key every time an account was created. In order to save time instead of using an SQL query, or updating the C# code, I selected an option in the SSMS interface providing an inbuilt solution which increments the primary key by “1” every time a new account is created.

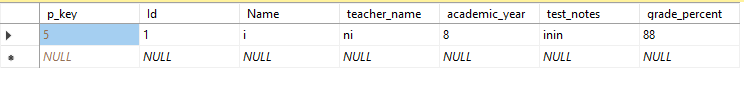


The identity increment shows how much the primary key should increase by when a new entry is added, this is set to 1 increment for each new entry.

Now, I can delete anything I want without any errors.



In the picture below the entry has been deleted. Even though the data is the same, the primary key has changed.



# Evaluation

### Using test evidence to cross evidence with the success criteria to evaluate the solution. Explain how the evidence shows that the criteria have been fully, partially or not met in each case.

### Provided comments on how any partially or not met criteria could be addressed in further development.

The program video presentation shows how partially or not met criteria could be addressed in further development.

I have found a different approach for the parts highlighted in red and yellow which are shown in the video presentation and also listed below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Key:** | **Works** | **Does not work. Will add in further development** | **Partially met** |
|  |  |  |  |
| **TESTING** |  | **Does it work?** | **Evaluation** |
| 1. The program opens. | |  | The program opens |
| a. The login form will be the first part to open in the program. | |  | It opens first |
| b. This is so the user can start to use the program. | |  |  |
| 2. You can enter text into the text boxes. | |  | You can enter text |
| a. This is required so that the user will be able to enter their details. | |  | That is true |
| b. I will do this by creating textboxes to enter text. | |  | Already done in the development |
| 3. You can click on the buttons. | |  | You can click on the buttons and they work |
| a. This is required for the user to add their details to the database in which is being held in. | |  | That is true |
| 4. There are buttons to move to and from the register for and the login forms. | |  | The buttons go to and from both forms |
| a. If you don’t have an account, you may make one by going to the “Create an account” form using a button as a medium to get there. | |  | Filling in all of the fields allows you to do it. Filling in some of the fields yields an error. I have created an account after opening XAMPP |
| 5. A database must be created. | |  | Already done in the development. Open the PHP server |
| a. The database is where the details of the users are stored. | |  | Check the PHP server to see the details of the creating account |
| b. The database is where their grades of the users are stored. | |  | It is written that the grades are stored in the PHP database. They are actually stored in the SSMS database |
| c. To do this, I will use PHPMyAdmin | |  | It is written that the grades are stored in the PHP database. They are actually stored in the SSMS database. Other details such as name and email are stored in the PHP database. |
| d. I will also use SQL Server 2019. | |  | I used it to create the database which uses SSMS for database managemnet |
| e. SQL Server 2019 will be used to create a local server in which I can create the database. | |  | I used it to create the database which uses SSMS for database managemnet |
| f. Along with SQL server 2019, I will use Microsoft SQL Server Management Studio (SSMS) to create a database with tables to store user details. | |  | I have opened SSMS and there is a database and there are tables there where the users' details are stored |
| 6. A connection to the database must be established | |  | This happens when the program opens in the dataBase.cs file. It also happens when I open XAMPP |
| a. This is so that the information entered may have a medium in which they can be stored. | |  | The information has a medium in which it can be stored |
| b. To do this, I will use the MySQL Connector package, which will allow me to connect to any database easily. I will also use XAMPP which will give me a port which I can use to connect to the database. | |  | XAMPP has been used before to show the user creation |
| 7. A table with columns corresponding to the fields in the “Create an account” form must be created. | |  | I have done that in the development |
| a. This is so the data can be sorted easily. | |  | That is true |
| b. I will do this through the PHPMyAdmin interface and name the columns with the same names as the ones in the “Create an account” form. | |  | I have shown the PHP names corresponding with the create a user labels |
| 8. The text entered into the fields of the “Create an account” form must be put into a database as columns. | |  | Create another account and show that the field has been added |
| a. The user must have somewhere where their login details can be stored so they are able to login. | |  | That is true |
| b. To do this, with the established connection mentioned in point 2, I will create tables in the database which correspond to the text box fields in the “create an account” form. | |  | I have shown the PHP names corresponding with the create a user labels |
| 9. Text entered into separate fields must be put into their corresponding columns. | |  | I have shown the PHP names corresponding with the create a user labels |
| a. This is to make distinctions between the user’s different details, like their name, or the grade which they attained. | |  | That is true |
| b. To do this, I will link the fields to the columns in the table with SQL. | |  | That has been done in the development |
| 10. The user must have a chance to select if they are a teacher or a student. | |  | You cannot do this in the PHP database but you can do this in the SSMS database |
| a. This is so that they can have different priorities. The student will only be able to view their grades while the teacher will be able to add grades and edit their grades. | |  | The student will only be able to access the program, while the teachers will have access to the SSMS database where they can make changes. The SSMS database will only be on the teachers' computers when connected to their own cloud database using SSMS |
| b. I will do this by creating a comboBox which have a teacher selection and a student selection. | |  | I have not done that as you can see it is not useful to create a separate student and teacher database because it is a security issue. Any student could choose to create an account by selecting the teacher option, then they could edit their grades to higher than they actually are. If only the teachers install SSMS on their computers, then the student will need physical access to change their grades. This is very hard to do so it is much more secure. |
| 11. Pressing the “Create the account” button will put all of the information into the table. | |  | I have shown the PHP names corresponding with the create a user labels |
| a. I will do this using the steps above. | |  | That is true |
| 12. All of the fields in the “Create an account” form must not be null. | |  | I have shown that all of the fields have to be full |
| 13. After creating the account, the program will show the login page. | |  | The program opens the login page so the user can login with their details after they register |
| a. This is so the user can login. | |  | This is the same as the point one cell above |
| b. To do this, I will make a function which will make the “Create an account” form close and the login form open. | |  | When I create the account the login form opens and the create an account from closes. |
| 14. The text boxes should accept input and the login button should be able to be clicked. | |  | I have shown the login form accepting input |
| a. This is so the user can login. | |  | I have shown the user logging in with the test data |
| b. I will do this by creating a button and textboxes for the user to login | |  | I have shown the user logging in with the test data |
| c. The user must be able to select if they are a teacher or a student. | |  | I have already mentioned in the evaluation of point 10a why you shouldn't be able to choose if you're a teacher |
| 15. The username and password will be verified. | |  | I have shown the username and the password in the login form and the PHP database. |
| a. This is so that only authorised people will be able to access the gradebook | |  | This is true as you can only access the gradebook with a valid username and password. |
| b. I will do this by using SQL to verify if the inputs in the database and the login input boxes in the program are the same. | |  | This has been shown in the development |
| 16. A new form will be opened. | |  | The grade book has been opened (the main form) |
| a. This form will be where the student’s and teacher’s data will be stored. | |  | I have explained the purpose of the text boxes and that they will store data into the SSMS database |
| b. I will do this by creating a function which will link the login and the info form together. | |  | This has been shown in the development |
| 17. The form will have multiple fields which you can type in. | |  | I have explained the purpose of the text boxes and that they will store data into the SSMS database |
| a. This is so that the user can type in their details such as their name and grade and which year they’re in. | |  | That is true |
| b. I will do this by making text boxes. | |  | I have explained the purpose of the text boxes and that they will store data into the SSMS database |
| 18. I need to create another database | |  | I will show the SSMS database |
| a. This is so that the information mentioned in 17a can be stored somewhere | |  | That is true |
| b. I will do this using SQL Server 2019 | |  | I have done that in the development |
| c. Each user should have their own table for their own information. | |  | This has not been implemented, so that all students can view each others' grades and information. This is a security issue because people can see others' private information and grades. To fix this, I will create a function which using the mysqladapter which will insert a whole new table into the database. This will be done in the mainform.cs file of the VIsual Studio C# code. |
| 19. The database needs to have a table | |  | I have shown the table in SSMS |
| a. These are required so that the information mentioned in 17a can be sorted appropriately | |  | Show that the data can be sorted by clicking on edit top 200 rows and sorting the data by clicking on the column headings |
| b. I will do this in SQL server by creating a table in the database | |  | I have done that in the development |
| 20. The database must be connected to the application. | |  | This is already done after login. It is shown in the development |
| a. I will use the SQLClient libraries in C# to do this. | |  | Shown in development |
| b. This is so the entered information will be able to go the database. | |  | I have entered the data in the main form and shown it being updated in the SSMS interface |
| 21. The information entered must be saved into the database | |  | I have entered the data in the main form and shown it being updated in the SSMS interface |
| a. This is so that the student can track their grades. | |  | That is true |
| b. I will do this by connecting the database to the application. | |  | Done in development section |
| c. When a “save” button is pressed, the database is updated with the relevant information. | |  | I have entered the data in the main form and shown it being updated in the SSMS interface |
| 22. There is a button which will allow the user to view their grades. | |  | I have clicked the view button and compared the data in the program to the data in the SSMS database |
| a. This will allow the user to track their grades. | |  | That is true |
| b. I will do this by establishing a connection to the database. | |  | Done in development section |
| c. The information will be displayed in the application as a table. | |  | I have clicked the view button and compared the data in the program to the data in the SSMS database |
| 23. If the user is a student, they can only view their own grades. | |  | See point 18a |
| a. This is for privacy reasons. | |  | That is true (see point 18a) |
| b. I will do this by creating a separate table in SSMS for each user when they create an account. | |  | See point 18a |
| c. This will be done while the application and the database are linked. | |  | The database and the application are already linked, but it cannot allow the student to view their own grades |
| 24. Teachers will have more access than students. | |  | That is partly true because it has not been implemented programmatically, but only the teachers will have access to the SSMS database so they will have more access than the students. |
| a. This means they will be able to view edit grades of the students. | |  | See point 24 and 10a |
| b. I will do this by using SQL commands to edit the tables. | |  | See point 24 and 10a to see why I haven't made a separate user and teacher login and the workaround I have made so that teachers will be able to edit and delete grades while students will only be able to view and add them. If students have typed in incorrect grades, then teachers can correct it |
| c. Students will only be able to view their own grades but not edit them. | |  | See point 24b, 24 and 10a |

The success criteria have mostly been met, areas not met are listed below or explained in the video.

With regard to point 1 (in the success criteria), that has been met as you can see in the first picture of the testing.  
Points 2 and 14 has been fulfilled as you can see, there is text inside the box in picture 2.   
Points 3 and 4 have been fulfilled because you are able to see the “Create an account” screen, which can only be accessed by clicking on the button which moves you from the login to the create an account screen. This is shown in picture 5 and 6. Picture 3 in the testing shows the PhpMyAdmin interface which has been created where the login details are stored. This fulfils point 5a and c. You can see the interface of the SMSS in picture 15, which fulfils 5f.

Points 6, 7, 8, 9 and 11 are fulfilled in the testing section. You can see this in pictures 9, 10 ,11 and 23 where the account is created and added to the database.

Point 12 has been fulfilled with picture 8 in the testing section. Some of the boxes are empty so the program doesn’t let you create an account.

Point 13 has been fulfilled in picture 13.

Point 15 has been fulfilled in picture 4. This shows that a user which isn’t in the database cannot be entered.

Points 16 and 17 have been fulfilled in picture 14, where you can see a new form open and text entered into multiple fields.

Points 18, 19, 20, 21 and 22 have been fulfilled in picture 15. As you can see the tables of the database which have appeared after you click on the “View” button after you enter the data and save it.

Points 10, 23 and 24 have not been fulfilled. I will add these points in future editions of the program.

To fulfil point 10, I will carry out point 10b, which is to create a comboBox in C# which will allow the user to select if they’re a student or a teacher. Since point 10 is related to the other unfulfilled points (23 and 24), this is what I will do next: Create a new column in the database with the name “studentYesNo”, which will be an integer value. If the user picks student, then the value “1” will be stored into the database. The value “2” will be set for the teacher. Next, I will have to link the two databases together. Since they both use MySQL, it should be possible. I must download the ODBC driver from the MySQL website. After that, I will connect them with a linked server which I will create in SMSS.   
Now, I will create a function which will create a new table for each student using SQL. A student will only be able to have one table assigned to them to prevent confusion.   
Next, I will create a function which will test to see if the user is a student or a teacher once the user logs in by confirming their integer value, if it’s “2”, then the user will have access to all of the grades in one master table. If it’s “1”, then the user will have their own personal table.

If the user is a student, their grades should be entered into their own table as well as the master table. I should generate a primary key for each user. The master table should have a same primary key as the student’s table to keep referential integrity. The primary key will also be easy for tracking.

In order for the teacher to edit the master table, I would have to implement the UPDATE and the SET commands in my code. This is only accessible for users who have permission to use the master table.

I plan to add these features which haven’t been fulfilled in the future. They are close to being added.

## Success of the usability features

### Provided evidence of the usability features justifying their success, partial success or failure as effective usability features.

**Table showing Success and Failure**

|  |  |  |
| --- | --- | --- |
| Usability feature | Success? | Evaluation |
| Error message when you type an incorrect username or password in the login | Full success | This is to inform the user that their username input hasn’t worked. If this message wouldn’t appear then the users may be confused. This is a success because through testing, typing intentionally incorrect text, a warning message appears. Then, you have to try again. |
| Blinking cursor shifts between text boxes in order when pressing “tab” | Partial success | The blinking cursor is supposed to move from textbox to textbox in the order they’re listed on the form. In all the forms apart from the “create an account” form, the blinking cursor shifts in order. The blinking cursor is supposed to move between textboxes so the user doesn’t have to constantly switch between their keyboard and mouse. In the create an account form, when pressing tab, the blinking cursor shifts un-ordered between text boxes. This is bad because the user will have to go back to using their mouse which defeats the purpose of this design feature. |
| Error message when you miss out some fields in the register form | Full success | The purpose of this is to tell the user that they have missed out some things such as their username. This is good because it prevents the confusion of some users when they miss out entering text in some text boxes. If the user misses one of the boxes and the error message doesn’t appear, then that means that their account would be created without the correct criteria, so they wouldn’t be able to log in. This is a “full success” because an error message appears when intentionally missing out some fields, which is what this usability feature was made for. |
| Design features 1 | Partial success | The design features in the design section have been fulfilled apart from the part where you choose if the user is a student or a teacher. This will be rectified in the next table below. |
| Design features 2 | Failure | Adding asterisks next to the labels in the forms to show that the fields are mandatory. This is so that the users will be able to tell that the fields in the register form are mandatory. If they leave some parts out thinking that it isn’t mandatory, then an error message will appear asking them to try again. This wastes the users’ time as they could have filled in all the fields if the asterisk were there. |

## How partially or unmet usability features could be addressed in further development

### Provided comments on how any issues with partially or unmet usability features could be addressed in further development.

**Table addressing Failures and how to fix in further development.**

|  |  |
| --- | --- |
| Usability feature | How could be addressed in further development |
| Blinking cursor shifts between text boxes in order when pressing “tab” | By creating new textboxes and putting the earlier created ones on top of the later created ones. The old textboxes in the register form were created and they were not put in order of which they were created. This is because pressing shift moves blinking cursor between the textboxes which were created in chronological order. |
| Design features 1 | This was a partial success because all of the features were fulfilled apart from some of them. The fulfilled features are:  The placement of the buttons and textboxes and labels of the forms. If you look at the testing of the project, you will be able to see that the username textbox in the login form is places on top of the password textbox. You can also see that the login button is placed at the button, so the user will type in their username and password before clicking it. If the user wants to create an account, they can click on the button on the top. That button is there first because first time users will see it first and they will know immediately where to go.  For the register form, your username and password are at the top. The username textbox is on top of the password textbox as that is how it is laid out in the login form. The email is placed in the middle so the user can differentiate between the username textbox and the first name textbox (as some users use their real names as usernames).  The student and teacher checkboxes have been left out because the teachers will only have access to the SSMS database where the grades are stored. Originally, the purpose of that box was to allow students and teachers to have different database tables, but I have found a solution where only the teachers will have access to SSMS where they can edit and delete the students’ grades.  There is one other unfulfilled part of the register form. That is that the link to the login form in my program is near the bottom of the form but the link to the login form in the usability features is at the top, so if the user accidentally clicked on the register form button, then they would know immediately where to click to go back.  A successful usability feature is that after registering, the user is redirected back to the login form. This is good because the user will not have to close the form or click unnecessarily to go back to the login form.  For the gradebook form, the design has mostly been followed, except for some minor changes, like the student and user checkboxes being merged into a drop-down box due to validity issues. The rest of the design has been followed. The student gradebook has been removed and has been replaced with the teacher’s design gradebook only. This is because teachers and students will use the same gradebook so I have merged the features. The are using the same gradebook because teachers will have access to the SSMS interface where they can edit and delete grades instead of having their own form. |
| Design features 2 | I will open the forms designer in C# forms and edit the textboxes so that they have an asterisk next to them. This way, the student or teacher will know that the textbox entry is mandatory and will not waste time with the error message if they don’t fill in the boxes. Upon viewing the asterisks, the user will know to fill in all of the forms. |

## Maintenance

### Considered maintenance issues and limitations of the solution.

There are 2 parts of the program. One is the login and register system and the other is gradebook. I have used different systems (PhpMyAdmin and SMSS) for each because I wanted to test to see which system would work better. I found out that they work equally as well and are both maintainable. This also allows different people to work on each part of the code. However, in the future, the state of the maintenance may change once the servers will be linked. It would also allow developers with different skillsets to work on different parts of the program. There is also a common ground. If one person working with PhpMyAdmin requires assistance from one person using SMSS, they can help each other because of the mutual renditions of the SQL that they are using, MySQL.

New features such as multiple tables or even additional databases can be added by reusing code. This can be done because the code is commented and is easy to read, so the developers know what each part of the code does.

When the code is released, features will constantly be added (see limitation / improvement section below) to deter the school from using competitor’s software. Future features may include online grading, online storage, graphing of the grades and prediction of grades. These features are essential to make the system more efficient. The online grading and storage could benefit teachers by allowing them to collaborate and send messages. The graphs may indicate to the teacher in which part of the year the students are underperforming and so they can teach the topics in more depth or amend their teaching style for those topics to suit the students’ needs.

Current features which require maintenance is the style of the program. Currently, the program looks simple and may require additional colours to make it look pleasant. I am planning on keeping the modern simplistic style in the future but addition pastel colours.

Other future features include a shared table between teachers and separate tables for students, as written in the evaluation of the success criteria. This one step goes towards the teacher collaborating and keeping privacy for each student’s grades.

## Limitations and improvements

### Described how the program could be developed to deal with limitations and potential improvements / changes.

One limitation which is quite significant is that this software isn’t suitable for alternative school, such as school with people who have behavioural issues or schools which are suited for people with disabilities. In the case of people who have behavioural issues, the gradebook doesn’t contain a section to record how students are behaving, or their attendance levels. This is a limitation because teachers cannot tell if the student’s behaviour is improving or getting worse. The teachers also cannot tell if they are skipping school either due to their behaviour. I may have to create an entirely different section for this because of its difference to tracking grades. This will require more text with a key search function or a drop-down list.

In the case of a disabled school, or for disabled students in general, this program has limitations because there are no accessibility features. There is no text-to-speech support if the student is blind, or speech-to-text support, if the student wants to input text using their voice, if they do not know how to type. There is no colour-blind support currently. Currently, the software uses colours like grey and black and white which do not affect colour-blind people, so it is not an issue now. However, that will only be an issue once I implement a better design with an array of colours or graphical presentations.

Another issue is the look and feel of the software. Currently, it looks simple and may deter people from using it. The users may spend less time using it because of its simplicity, this has advantages as it is focused on the key point. If I make it look more attractive, then the user will be more inclined to use it.

Another limitation is that because the master table for teachers hasn’t been implemented, all students can view everyone else’s grades which is bad for privacy. I have highlighted this limitation in the video.

Another limitation is that there is a basic sorting system implemented. This will be improved in future version with graphical presentations. There is a basic sorting system which is implemented in the SSMS software which allows you to sort the tables.

Another limitation is that there is no timetable, so the student cannot view what time they have their lessons, or what time they need to study by themselves. When implemented, the student can increase their studying time., this is also another development being considered.

I have already noted this in the stakeholders’ section of this document, but there is no support for mobile phones currently. I am planning on adding mobile support post development to support unrepresented opinions in stakeholders’ survey.

I have developed the code so that it is user friendly so that the future students of the school can update it (mentioned in the analysis). Other future additions by the students will include graph databases, machine learning and augmented Data Management. This will be possible because I have commented and annotated the code for easy readability for the future students. This will save money for the school and will help further the students’ coding skills.