

Design

1. Overall System Design

1.1 Short description of the main parts of the system

- Student assessment system
 - Student user interface
 - Teacher user interface
 - Displaying students results
 - Test user interface
 - Adding information
 - Remove information

Teacher user interface:

- The teacher will be presented with a box to type in a password. This password will be used by all of the teachers.
- Once logged into this part of the program, the teacher will be presented with a user interface with a series of options. These are; add information, remove information, view a student's results, or change password.
- Clicking the password button will take them to a user interface, where they can enter their old password, then type in a new one.
- Clicking the view students' results button will take them to a user interface that will allow them to search for a student, and then the program will display the student's results to the screen. They will then be able to click on a test to view a more detailed breakdown of the questions asked.
- Clicking the add information button will present a user interface for adding information for the students, teachers and classes. They will also be able to create questions, group these questions, and create tests and then add these questions to the test.
- Clicking the remove information button will present a user interface for removing information for the students, teachers, classes, tests, question groups and questions.

Add information:

- The system will present the user with a drop down menu, containing all of the options to add information to. After selecting one, the interface will change to a series of boxes to enter information about the option they chose. If one any of the information requires specific information from another table, instead of a box, there will be a drop down menu.
- Once confirmed, the system will then add a unique ID to the information, and save it to the selected database.

Student Interface:

- The student will first be presented with a drop down menu, where they will select their name and press ok.
- This will then take them to a screen where they can choose a topic.
- They will then be presented with the test user interface

Test user interface:

- The first thing that will show up is the option to choose your car colour and design.
- The test user interface will consist of a question displayed at the top and the question number in the top right. Below this there will be a place for the user to input their answer. There are three question types; multiple choice, yes or no, and typing in the answer with the keyboard. If the question type is multiple choice four buttons will be displayed, each with a possible answer on them. If the question type is yes or no, two buttons will be displayed, one with yes, and one with no. If the question type is one where you have to type in the answer, a line edit box will be displayed, and an ok button next to it. Finally below this, there will be a graphic of the car previously created. With each correct answer, the car will have upgrades added to it. After all of the questions have been answered, there will be an animation of a top down, 2D car race. The more questions answered correctly means the more chance of winning the race. If all of the questions were answered correctly, the next screen will be a picture of a smiley sticker, and an ok button to dismiss the screen. This screen will alert a teacher to give the student a sticker. The results of the test will then be saved to a database.

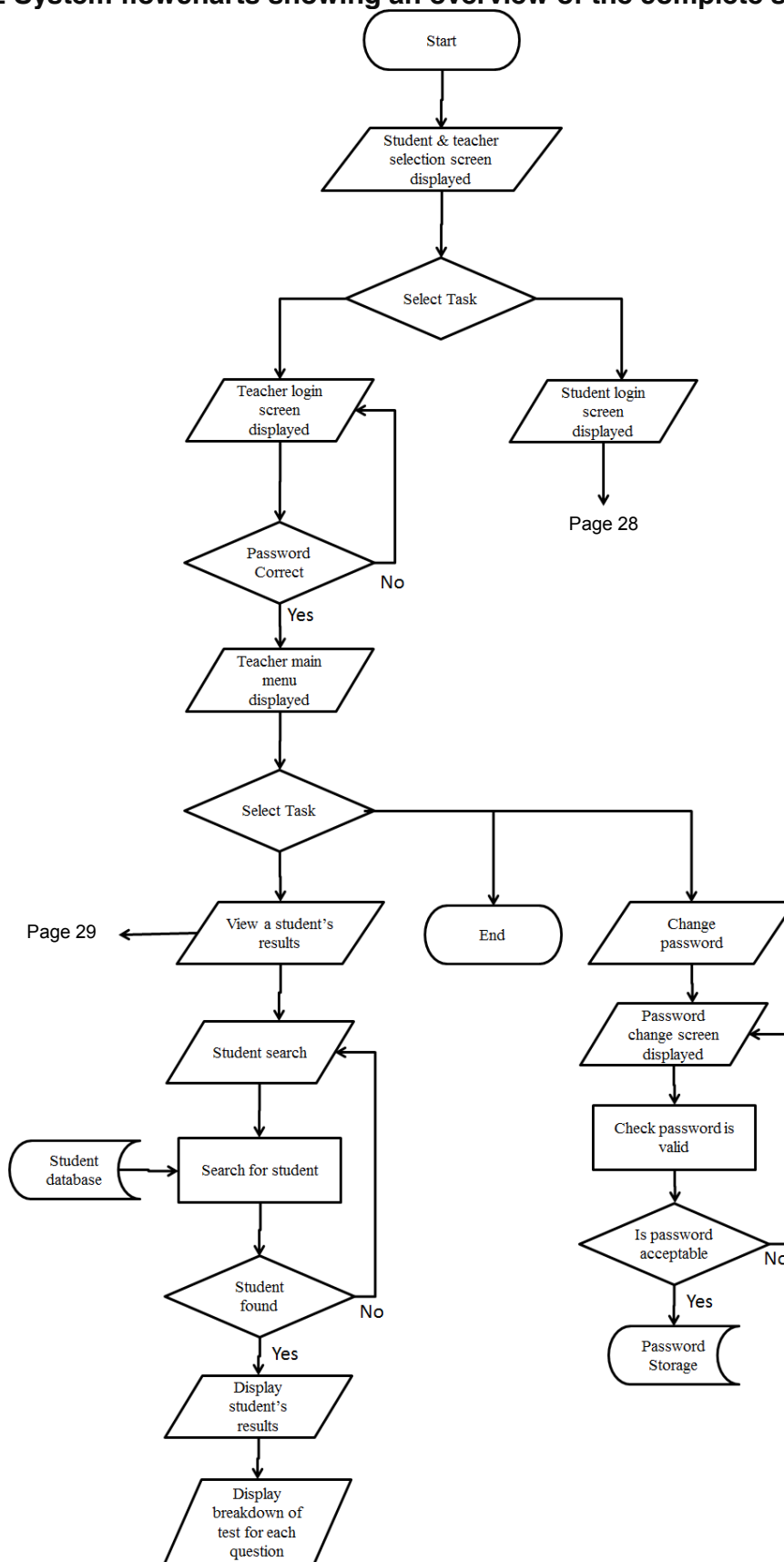
Displaying a student's results:

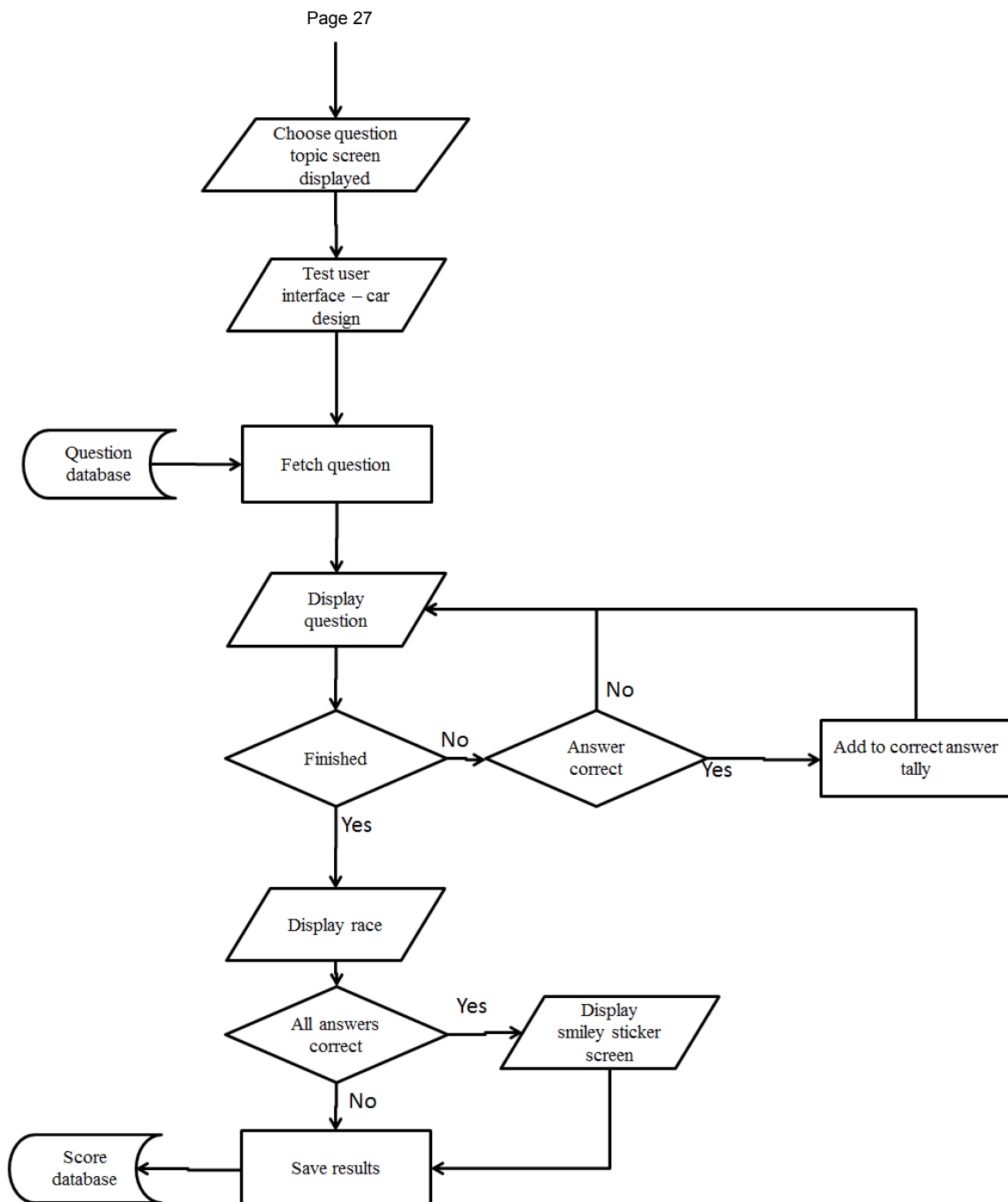
- The system will first present the teacher with a box to fill in the student's name. This will then search the database, and return a list of students with that name to choose from. Once the desired student is selected, all of the test results of that student will be displayed to the screen.

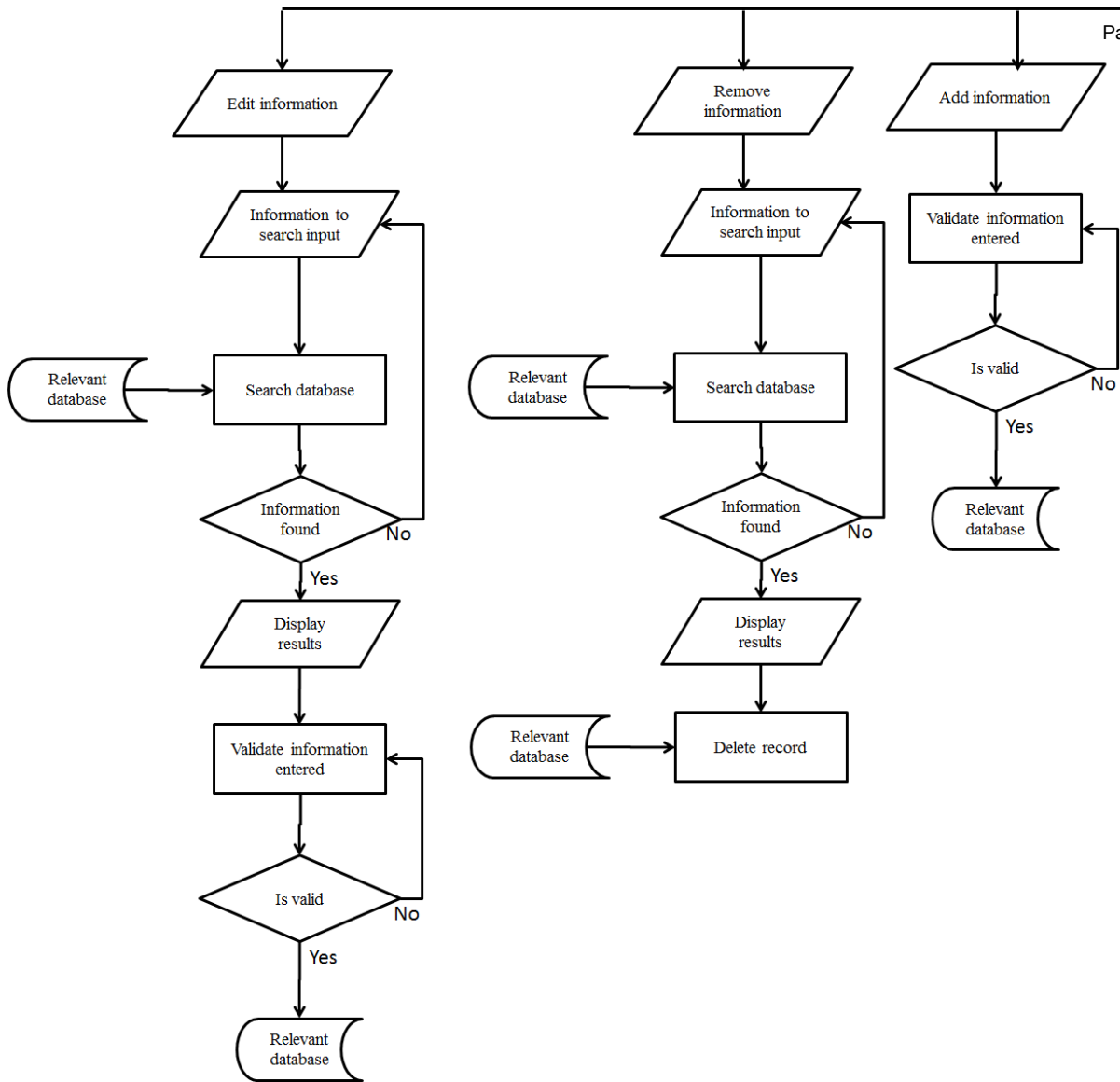
Remove information:

- The system will first present the user with a drop down menu relating to each database table.
- After clicking on one of these, the user will be asked to enter as much information about the record, they wish to delete, as possible.
- The system will then search the database for this record, and return a list of results back to the user. The user will then select one, and after confirmation, the system will delete the record.

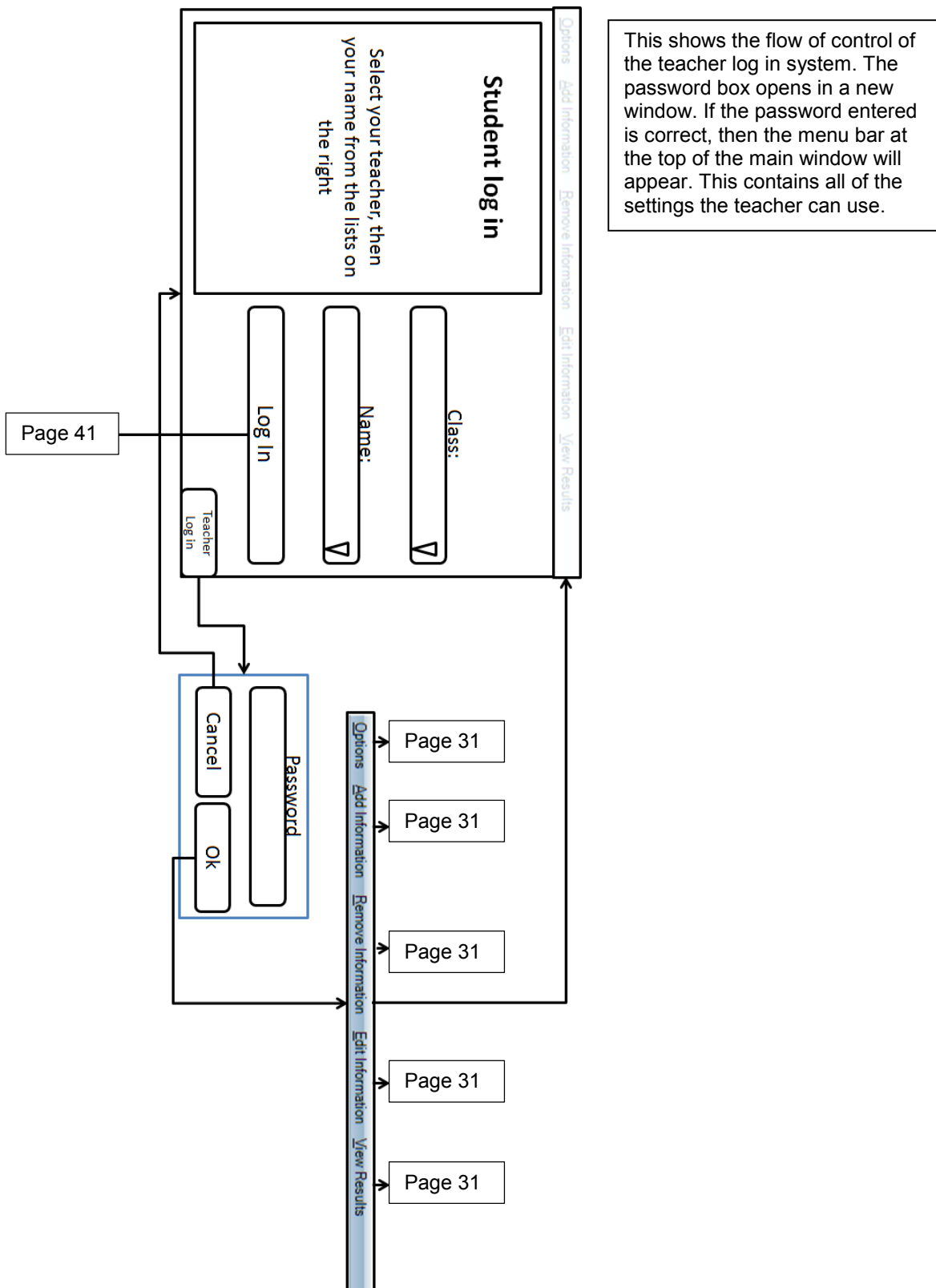
1.2 System flowcharts showing an overview of the complete system

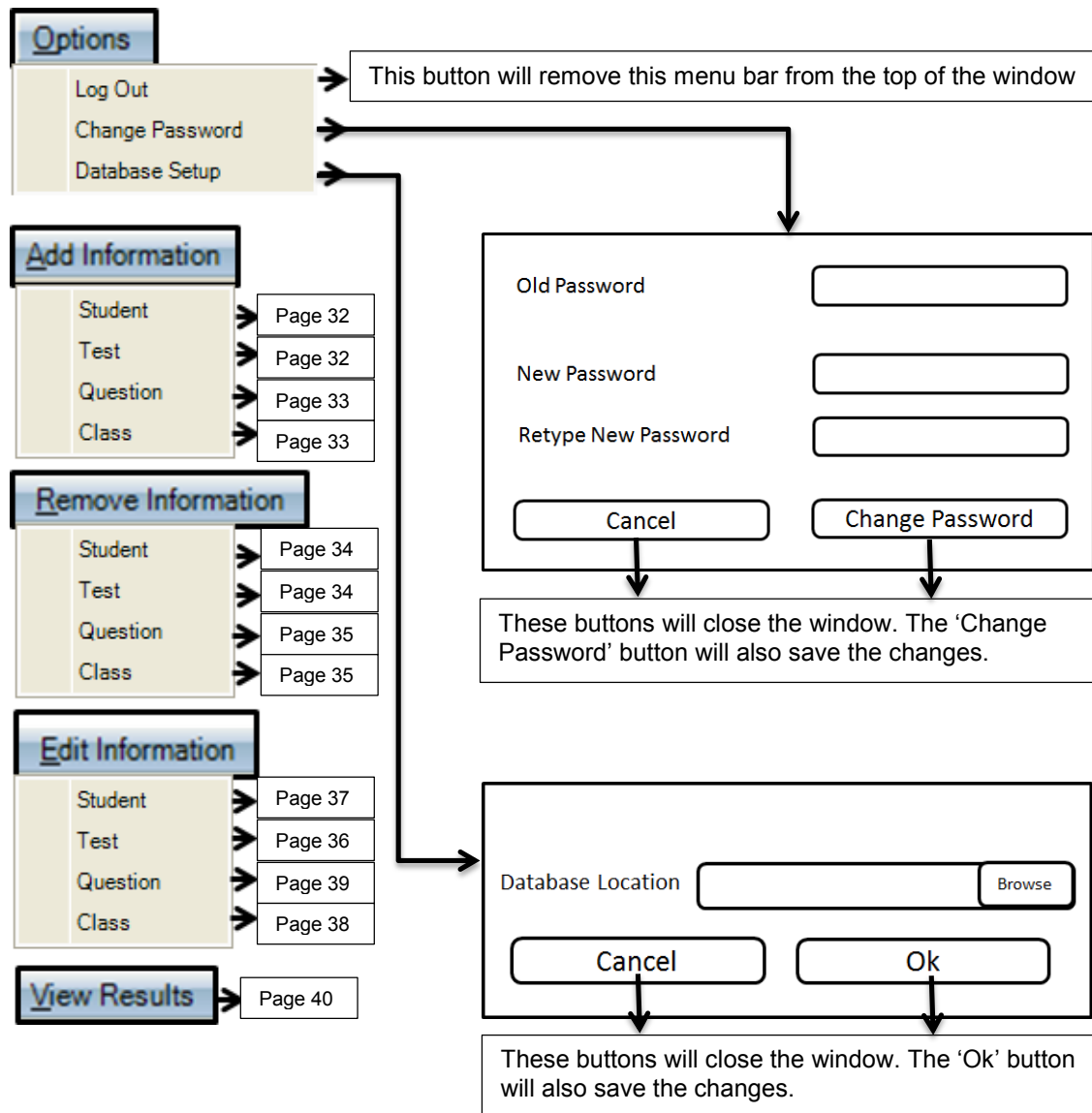






2. User Interface Designs





Test Name

Test Description

Questions:

Delete	Edit	Selected Question 1
Delete	Edit	Selected Question 2
Create New	Existing Questions ▾	

Cancel

Create Test

Page 39

Page 39

These buttons will close the window. The 'Create Test' button will also save the changes.

First Name

Last Name

Class

▾

New Class

Cancel

Add Student

Page 33

These buttons will close the window. The 'Add Student' button will also save the changes.

Class Name

Class Year

These buttons will close the window. The 'Add Class' button will also save the changes.

Answer: ☒ Yes ☐ No

Answer: Choices:

Answer:

Question

Mark

Question type

Answer and Choice selection goes here

These buttons will close the window. The 'Add Question' button will also save the changes.

Search

Example test 1 title This is the description for test 1
Example test 2 title This is the description for test 2
Example test 3 title This is the description for test 3
Example test 4 title This is the description for test 4
Example test 5 title This is the description for test 5

These buttons will close the window. The 'Delete Test' will delete the selected test from the database.

Class

Name

FirstName SecondName Class
FirstName SecondName Class
FirstName SecondName Class
FirstName SecondName Class
FirstName SecondName Class

These buttons will close the window. The 'Delete student' will delete the selected student from the database.

Year

Name

Class Name	Class Year
Class Name	Class Year
Class Name	Class Year
Class Name	Class Year
Class Name	Class Year
Class Name	Class Year

These buttons will close the window. The 'Delete Class' will delete the selected class from the database.

Question

Question 1
Question 2
Question 3
Question 4
Question 5

These buttons will close the window. The 'Delete Question' will delete the selected question from the database.

Search

Example test 1 title This is the description for test 1
Example test 2 title This is the description for test 2
Example test 3 title This is the description for test 3
Example test 4 title This is the description for test 4
Example test 5 title This is the description for test 5

Cancel Edit Test

This button will close the window

Test Name

Test Description

Questions:

Delete	Edit	Existing question 1
Delete	Edit	Existing question 2
Create New	Existing Questions ▾	

Cancel Save Changes

Page 39

Page 33

These buttons will close the window. The 'Save changes' will save all of the changes to the database.

Class

Name

FirstName	SecondName
Class	
Class	
Class	
Class	
Class	

This button will close the window

First Name

Existing First Name

Last Name

Existing First Name

Class

Class

These buttons will close the window. The 'Save changes' will save all of the changes to the database.

Year

Name

Class Name
Class Year

Class Name
Class Year

Class Name
Class Year

Class Name
Class Year

Class Name
Class Year

Cancel

Edit Class

This button will close the window

Class Name

Existing name

Class Year

Existing Class year

Cancel

Save Changes

These buttons will close the window. The 'Save changes' will save all of the changes to the database.

Question

Question 1

Question 2

Question 3

Question 4

Question 5

Cancel Edit Question

This button will close the window

Question

Mark

Question type

Answer and Choice selection goes here

Cancel Save Changes

These buttons will close the window. The 'Save changes' will save all of the changes to the database.

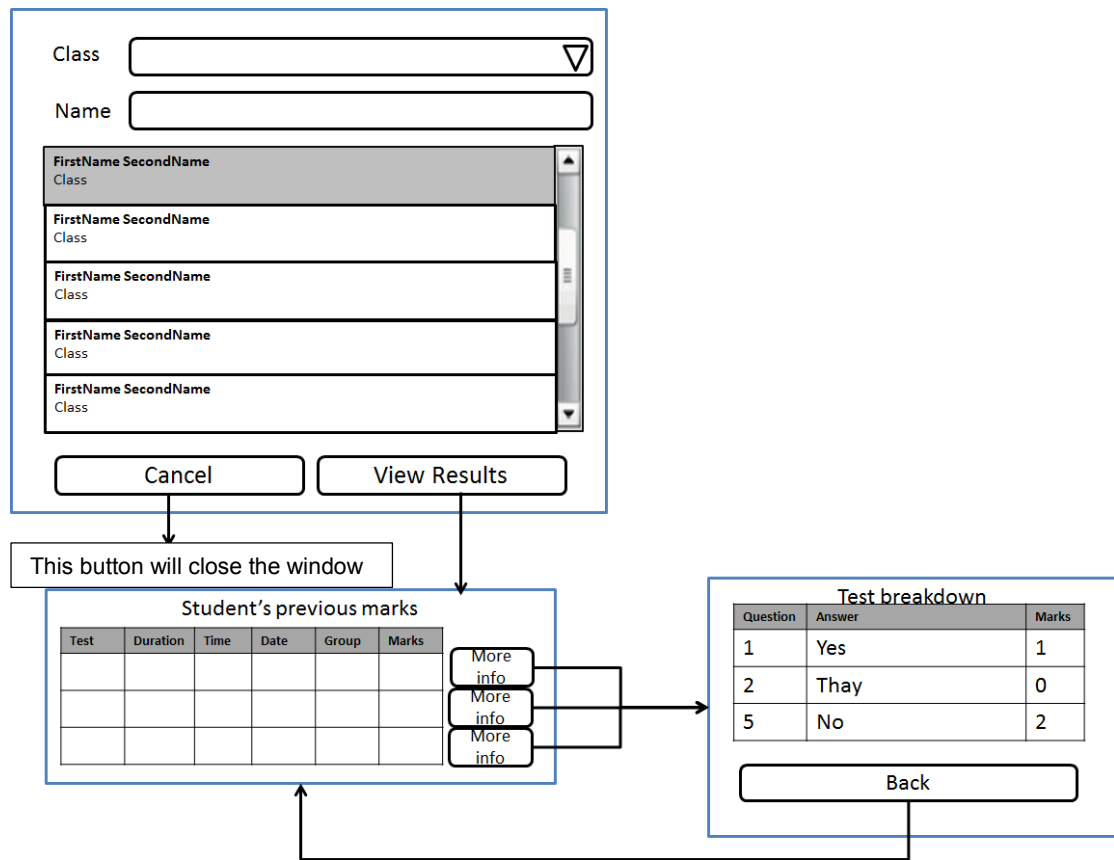
Answer:

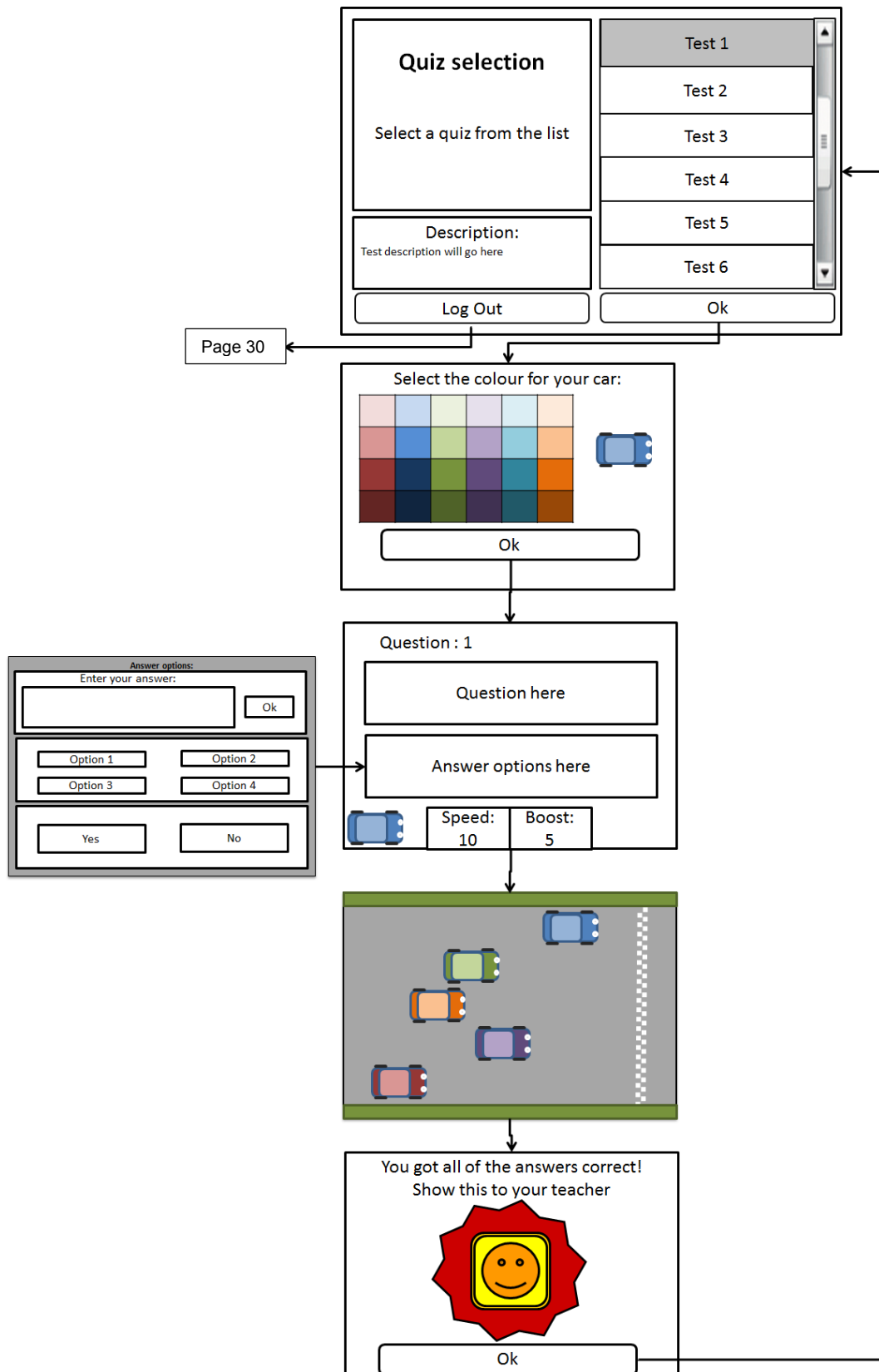
☒ Yes ☐ No

Answer:

Choices:

Answer:





Student log in

Select your teacher, then your name from the lists on the right

Class:

Name:

Log In

Teacher Log in

This is the first screen that will be shown when the program is

There are drop down menus for the student to select their class and then their name

There is then a big log in button to go onto the test selection screen

The box on the right explains to the student what they have to do on this page

Drop down menu keeps the interface easy for the students to understand, and eradicate mistakes

Quiz selection

Select a quiz from the list

Test 1

Test 2

Test 3

Test 4

Test 5

Test 6

Description:

Test description will go here

Log Out

Ok

This is where the student can select the test they want to do.

The description box will change depending on the selected test

The box in the top left tells the student what to do on this page.

The log out box will log the student out, and take them back to the first screen

The 'Ok' button will take them to the car selection screen for the selected test.

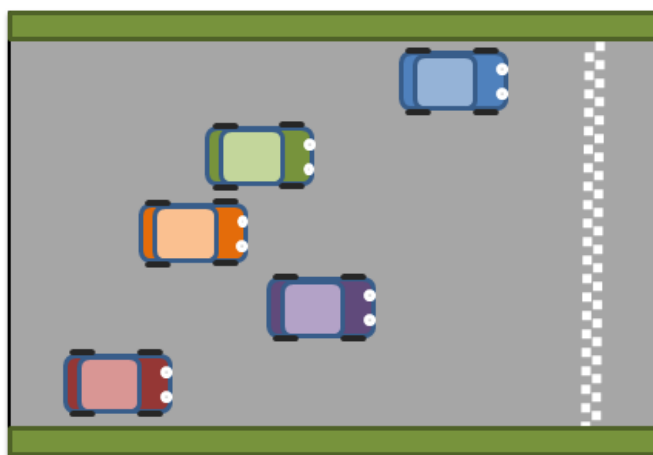
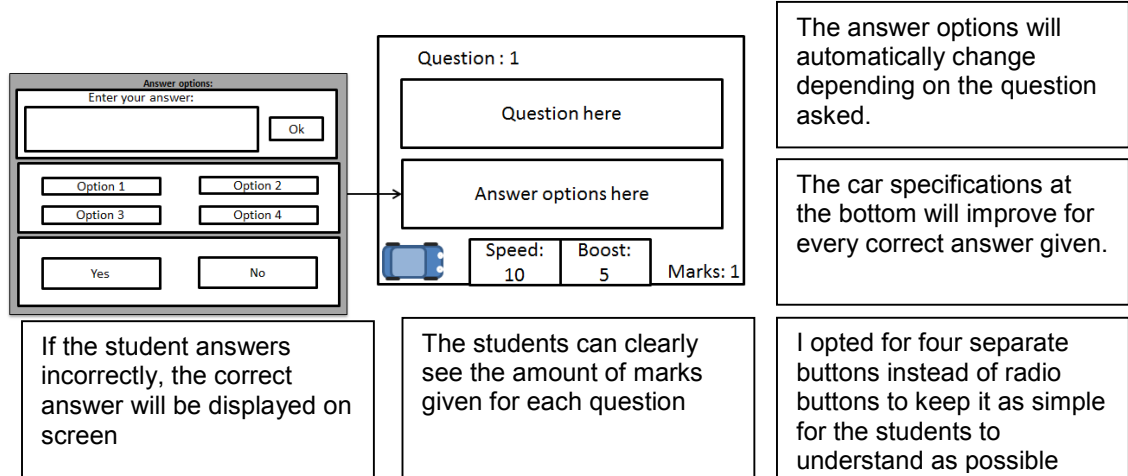
Select the colour for your car:

Ok

This is where the student will select the colour for their car

The car on the right will change colour to reflect the current selection on the left

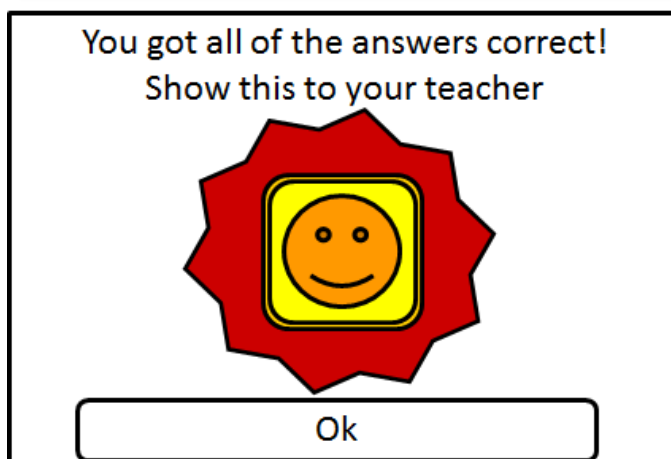
The Ok button will then take them to the Test screen.



Once the test is completed, this screen will be shown.

This will be what the race looks like at the end of the test. The student's car will be shown racing against other randomly advanced cars

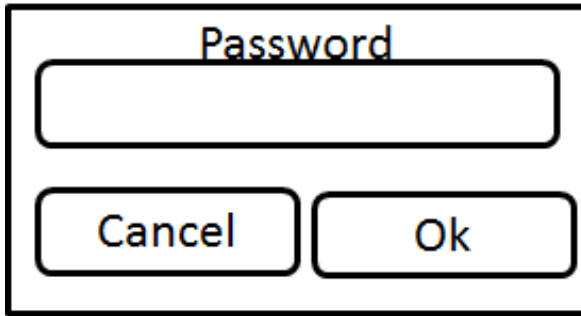
This adds an element of fun to the test, and will make the student's try their hardest to answer each question correctly



This screen will be displayed at the end if the student answers all of the questions correctly.

This is brightly coloured so that it can both alert the teacher, and so that it is admirable to the student.

On the first screen, if the 'Teacher log in' button is clicked, the following window will appear:



A window titled 'Password' with a text input field and two buttons: 'Cancel' and 'Ok'.

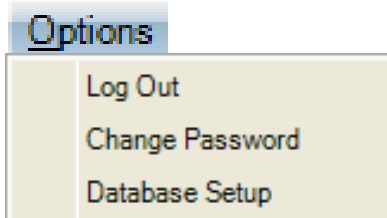
This is where the teacher will have to input the password to gain access to the secure part of the program

If the password entered is incorrect, text in red will appear above the password box stating "The password you entered was incorrect. Please try again."

Once the password is entered correctly, the following menu bar will appear at the top of the program.

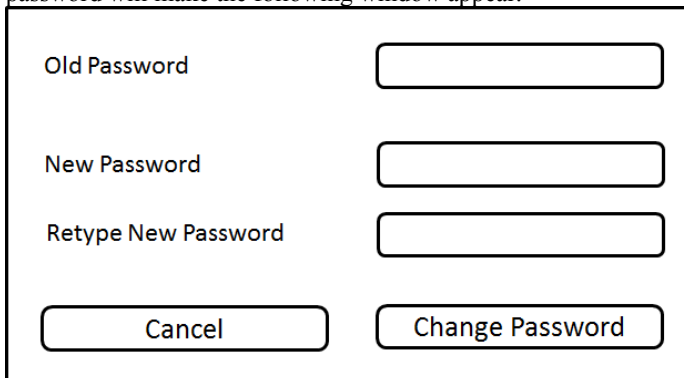
Options Add Information Remove Information Edit Information View Results

Under the options tab, the following options will be available:



A menu bar with 'Options' selected, showing a dropdown list with 'Log Out', 'Change Password', and 'Database Setup'.

Clicking 'Log Out' will log the teacher out, and remove the menu bar from the screen. Clicking change password will make the following window appear.



A window titled 'Change Password' with three text input fields labeled 'Old Password', 'New Password', and 'Retype New Password'. At the bottom are two buttons: 'Cancel' and 'Change Password'.

To keep the program secure when changing passwords, it first asks for the old password.

It asks for the new password twice to eradicate input errors.

The program will also check to make sure the password is valid before continuing. (Contains a letter, a number, and is more than 6 characters)

Clicking 'Change Password' will change the password and close the window if the credentials are correct. Clicking cancel will close the window without saving the password.

If the database setup button is clicked, the following window will appear.

The box next to the browse button will be read only to eradicate input errors; instead, the box will have to be filled with the 'Browse' button

The 'Browse' button will open the default system file selector, so that the path specified exists, and is in the correct format.

Clicking 'Ok' will save the new location and close the window. Clicking cancel will close the window without saving the location.

If the Add information button is selected on the main menu, the following options will be displayed.

Clicking on Test will show the following window.

Questions:	
Delete	Edit
Selected Question 1	
Delete	Edit
Selected Question 2	
Create New	Existing Questions ▼

The test name and description will be entered at the top

Questions will be added to the test by either selecting questions already made, or creating a new question

Clicking delete will remove the corresponding question from the test.

Clicking edit will open up the edit screen for the selected question, as of the one on page 9

Clicking Cancel will close the window and not add the test. Clicking 'Create Test' will save the test and close the window.

Clicking the 'Student' button from the menu will open a new window with the following.

The drop down box will help avoid any input errors.

The 'New class' button will show the add class window as shown below

Clicking cancel will close the window without adding the student. Clicking 'Add Student' will save the student information and then close the window

Clicking the 'Class' button will open a new window with the following.

The spin box means that only numbers can be entered. This reduces input errors.

Clicking cancel will close the window without saving the class. Clicking 'Add Class' will save the class information and then close the window

Clicking the 'Question' button will open a new window with the following.

The spin box means that only numbers can be entered. This reduces input errors.

This box will change depending on the question type.

The drop down menu for the question type helps reduce input errors

Clicking cancel will close the window without adding the question. Clicking 'Add Question' will save the question information and then close the window

If the question type is 'Yes or No' above, the answer selection will be:

If the question type is 'Multiple choice' above, the answer selection will be:

Answer: <input type="text"/>	Choices: <input type="text"/> <input type="text"/> <input type="text"/>
--	---

The Answer box will be for the correct answer. The Choices box will be for the other incorrect answers in the question.

If the question type is 'Typed answer' above, the answer selection will be:

Answer: <input type="text"/>
--

If the remove button is clicked on the main menu, the following options will be displayed

Remove Information	
	Student
	Test
	Question
	Class

If Test is selected from the options above, the following window will appear.

Search <input type="text"/>
<div> <div>Example test 1 title This is the description for test 1</div> <div>Example test 2 title This is the description for test 2</div> <div>Example test 3 title This is the description for test 3</div> <div>Example test 4 title This is the description for test 4</div> <div>Example test 5 title This is the description for test 5</div> </div>
<input type="button" value="Cancel"/> <input type="button" value="Delete Test"/>

The search box will search the names and descriptions of all the tests for the text in the search box. The search results will automatically update to show these results

If the delete button is clicked, the currently selected test will be removed from the database, and the window will close. If the cancel button is clicked, the window will close without deleting from the database.

If Student is selected from the options, the following window will appear.

Class <input type="text"/>
Name <input type="text"/>
<div> <div>FirstName SecondName Class</div> <div>FirstName SecondName Class</div> <div>FirstName SecondName Class</div> <div>FirstName SecondName Class</div> <div>FirstName SecondName Class</div> </div>
<input type="button" value="Cancel"/> <input type="button" value="Delete Student"/>

If the class field isn't blank, the name box will search the first name and last name of all of the students in that class. If the class field is blank, the name will search all of the students. The search results will- update automatically to reflect the changes in the boxes above.

If the delete button is clicked, the currently selected test will be removed from the database, and the window will close. If the cancel button is clicked, the window will close without deleting from the database.

If Question is selected from the list above, the following window will appear.

The question field will search all of the questions. The results at the bottom will update automatically.

If the delete button is clicked, the currently selected question will be removed from the database, and the window will close. If the cancel button is clicked, the window will close without deleting from the database.

If Class is selected from the list above, the following window will appear.

If the year field isn't blank, the name box will search the names of all of the classes in that year. If the year field is blank, the name will search all of the classes. The search results will update automatically to reflect the changes in the boxes above.

If the delete button is clicked, the currently selected class will be removed from the database, and the window will close. If the cancel button is clicked, the window will close without deleting from the database.

A spin box is used for the year to reduce on input errors.

On the main menu bar at the top, if the edit button is clicked, the following menu will appear.

If 'Test' is clicked from this menu, the following window will appear.

The search box will search the names and descriptions of all the tests for the text in the search box. The search results will automatically update to show these results

If the edit button is clicked, the currently selected test's information will be loaded into the next screen for editing. If the cancel button is clicked, the window will close.

Test Name	<i>Existing Name</i>
Test Description	<i>Existing Description</i>
Questions:	
Delete	Edit
<i>Existing question 1</i>	
Delete	Edit
<i>Existing question 2</i>	
Create New	Existing Questions ▾
Cancel	Save Changes

The selected test's attributes will be loaded into the boxes for editing.

The Save Changes button will update the database with the new values.

The cancel button will close the window.

Clicking edit will open up the edit screen for the selected question, as of the one on page 9.

Clicking 'Student' will load the following in a new window.

Class	▾																
Name																	
<table border="1"> <tr> <td>FirstName</td> <td>SecondName</td> </tr> <tr> <td colspan="2">Class</td> </tr> <tr> <td>FirstName</td> <td>SecondName</td> </tr> <tr> <td colspan="2">Class</td> </tr> <tr> <td>FirstName</td> <td>SecondName</td> </tr> <tr> <td colspan="2">Class</td> </tr> <tr> <td>FirstName</td> <td>SecondName</td> </tr> <tr> <td colspan="2">Class</td> </tr> </table>		FirstName	SecondName	Class		FirstName	SecondName	Class		FirstName	SecondName	Class		FirstName	SecondName	Class	
FirstName	SecondName																
Class																	
FirstName	SecondName																
Class																	
FirstName	SecondName																
Class																	
FirstName	SecondName																
Class																	
Cancel	Edit Student																

First Name	<i>Existing First Name</i>
Last Name	<i>Existing First Name</i>
Class	<i>Class</i> ▾ New Class
Cancel	Save Changes

If the class field isn't blank, the name box will search the first name and last name of all of the students in that class. If the class field is blank, the name will search all of the students. The search results will update automatically to reflect the changes in the boxes above.

If the edit button is clicked, the currently selected test will be loaded into the next screen. If the cancel button is clicked, the window will close.

The selected Student's attributes will be loaded into the boxes for editing.

The Save Changes button will update the database with the new values.

The cancel button will close the window.

Clicking new class will open the new class window as on page 9

If class is selected from the menu, the following window will appear.

If the year field isn't blank, the name box will search the names of all of the classes in that year. If the year field is blank, the name will search all of the classes. The search results will update automatically to reflect the changes in the boxes above.

If the edit button is clicked, the currently selected Class will be loaded into the next screen. If the cancel button is clicked, the window will close.

A spin box is used for the year to reduce on input errors.

The selected Class's attributes will be loaded into the boxes for editing.

The Save Changes button will update the database with the new values.

The cancel button will close the window.

If Question is selected from the menu, the following window will appear.

The question field will search all of the questions. The results at the bottom will update automatically.

If the edit button is clicked, the currently selected Question will be loaded into the next screen. If the cancel button is clicked, the window will close.

The selected Question's attributes will be loaded into the boxes for editing.

The Save Changes button will update the database with the new values.

The cancel button will close the window.

This box will change depending on the Question type from one of the following

If the view results button is clicked, the following window will appear.

Class

Name

FirstName	SecondName	Class
FirstName	SecondName	Class
FirstName	SecondName	Class
FirstName	SecondName	Class
FirstName	SecondName	Class

If the class field isn't blank, the name box will search the first name and last name of all of the students in that class. If the class field is blank, the name will search all of the students. The search results will update automatically to reflect the changes in the boxes above.

If the view results button is clicked, the currently selected student's results will be displayed in a new 'Student's previous marks' window as below. If the cancel button is clicked, the

Student's previous marks

Test	Duration	Time	Date	Group	Marks

Clicking the more info button will load the information for the corresponding test into the Test breakdown window.

Test breakdown

Question	Answer	Marks
1	Yes	1
2	Thay	0
5	No	2

This window will break down the student's answers for the current selected test.

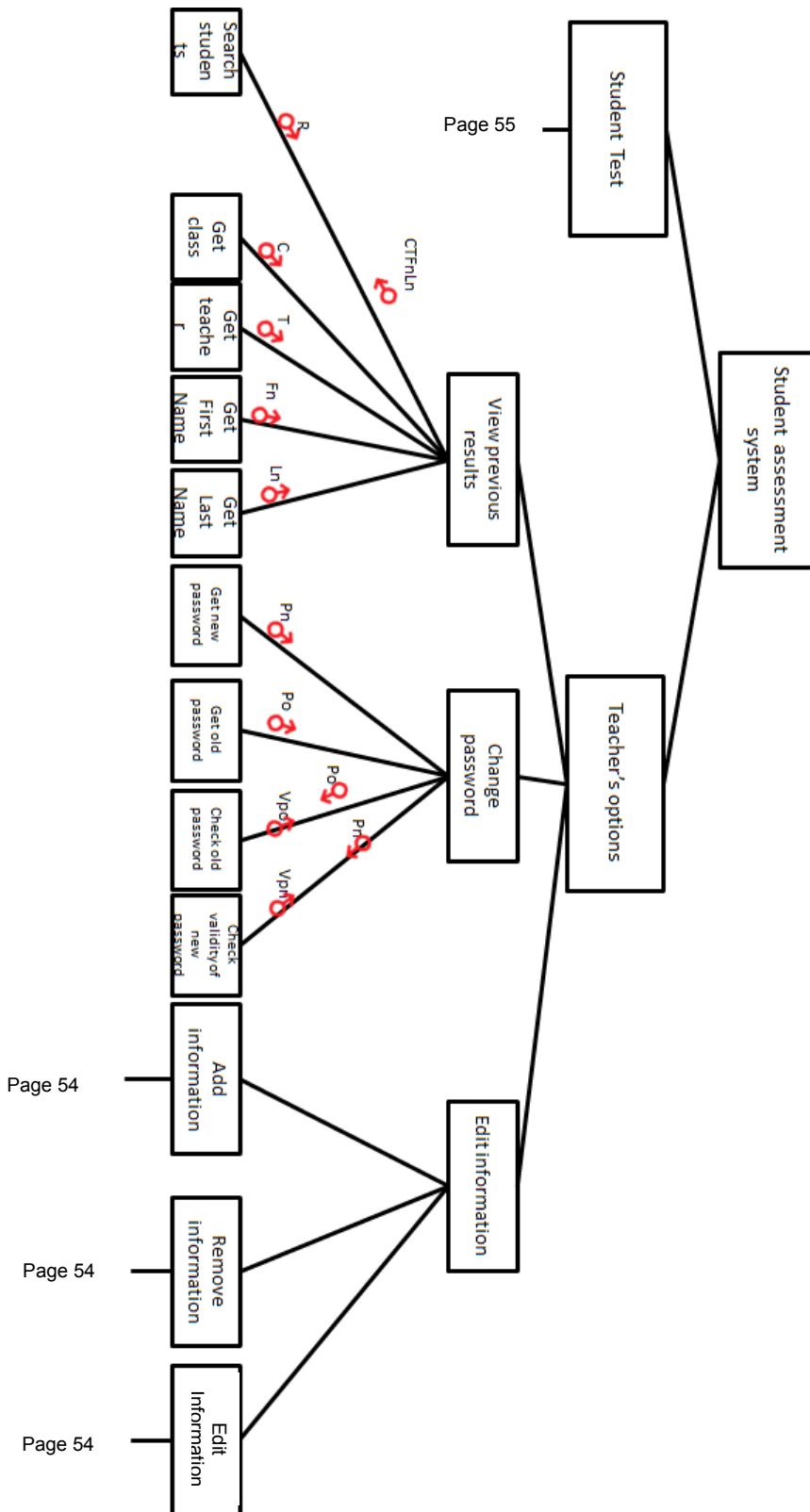
Clicking the back button will load the student's previous marks screen again

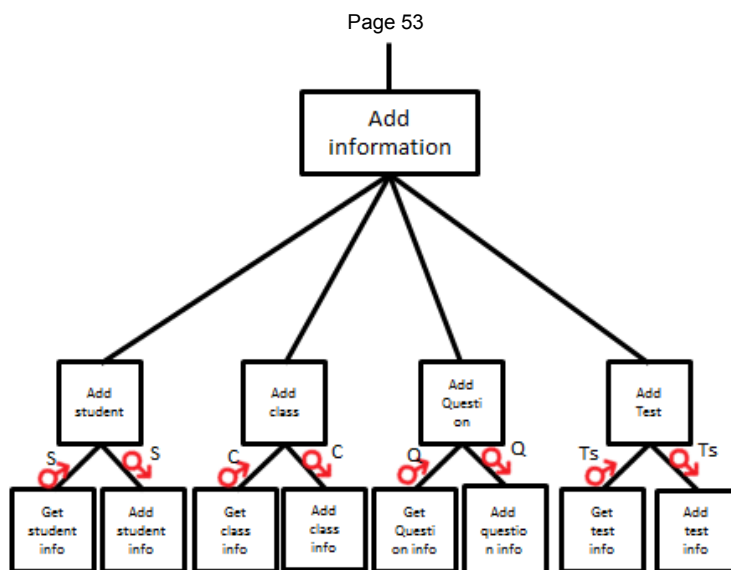
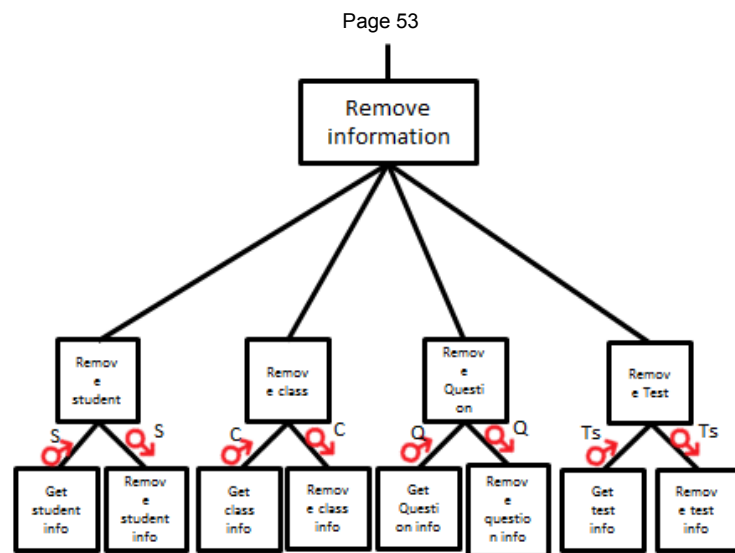
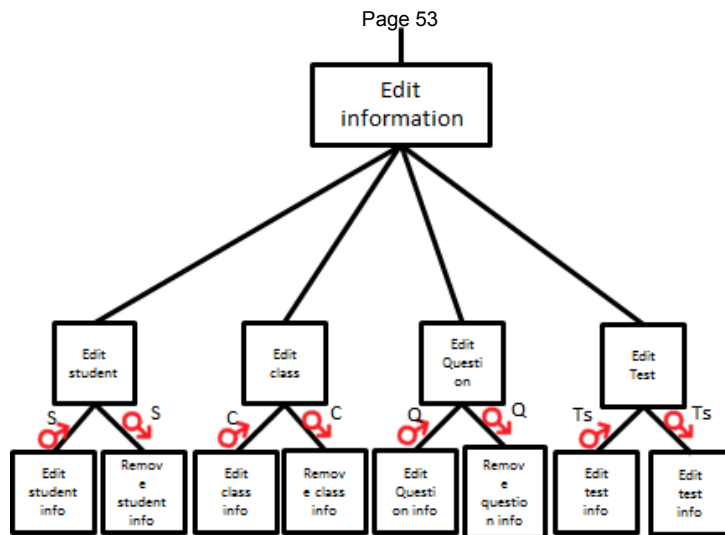
3. Hardware Specification

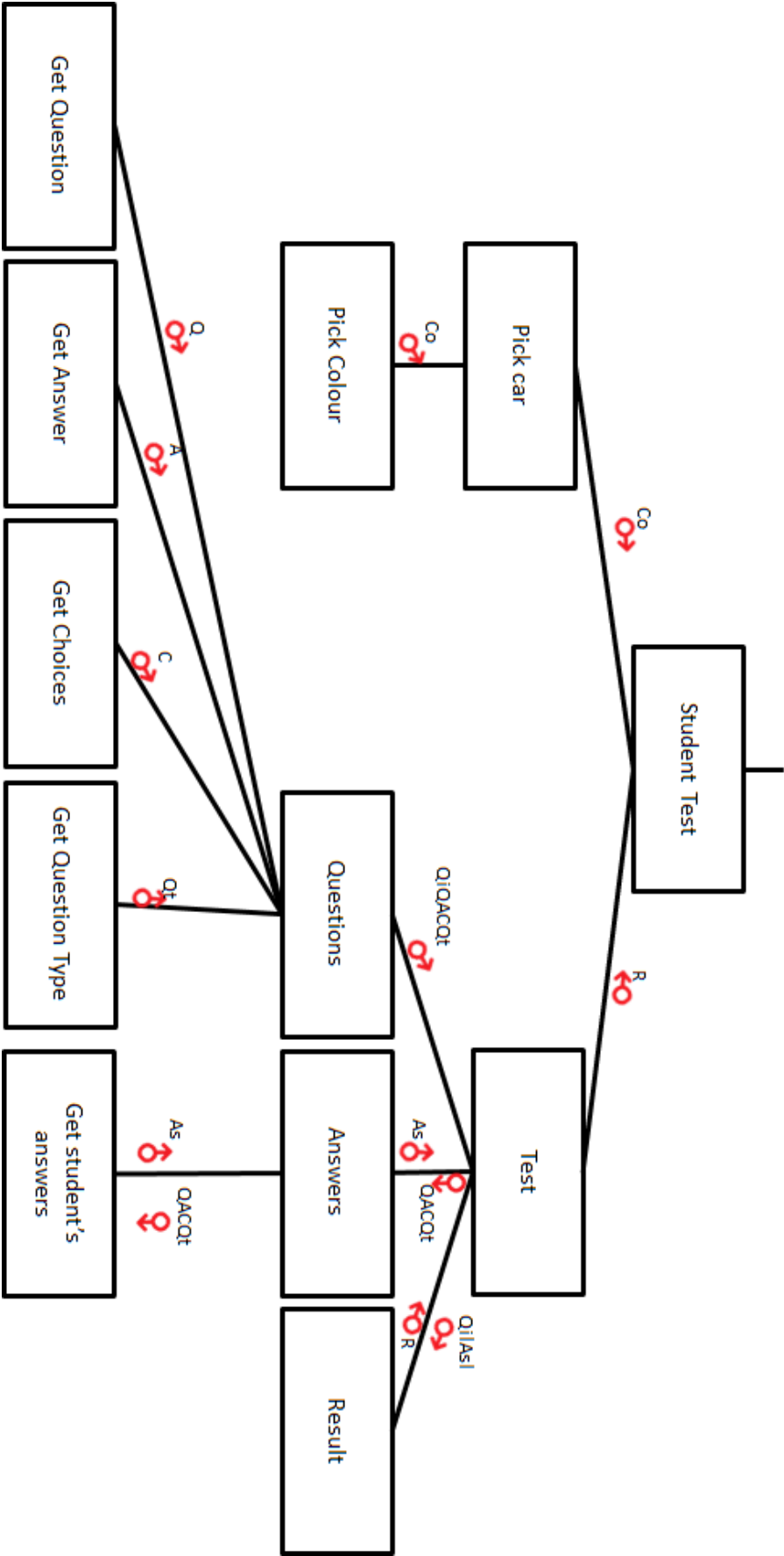
The system will need to run on a desktop computer and a laptop, both with a 1360x768, 16:9 aspect ratio screen and running Windows XP. This is important as I have to make sure my program will fit on this screen size. This is especially important for the race aspect of my program as that will not be resizable. A keyboard will be needed for inputting the information to the program, as well as if the student needs to give an answer with the keyboard. A mouse will be used to navigate around the program, as well as selecting the answer in multiple-choice. A display will be used for the output of the program. The data for the program will be held on a hard drive.

4. Program structure

4.1. Top down design structure charts







```
FUNCTION SearchStudent (Class: CLASS, FirstName: FIRSTNAME, LastName: LASTNAME)
    Count: Integer
    Count ← 0
    ARRAY StudentList: Integer [∞]
    CONNECT to Student Database
    Students ← SEARCH Student Database for Student
    FOR Student in Students DO
        IF Student.FirstName = FirstName AND Student.LastName = LastName AND
Student.Class = Class DO
            StudentList [Count] ← Student
        END IF
        Count ← Count + 1
    END FOR
    RETURN StudentList
END FUNCTION
```

```
FUNCTION CheckOldPassword (OldPassword: OLDPASSWORD, EnteredOld: ENTEREDOLDPASSWORD)
    PasswordValid: Boolean
    PasswordValid ← FALSE
    IF OldPassword = EnteredOld THEN
        PasswordValid ← TRUE
    END IF
    RETURN PasswordValid
END FUNCTION
```

```
FUNCTION NewPasswordValid (NewPassword: NEWPASSWORD)
    LetterFound: Boolean
    NumberFound: Boolean
    FOR Character in NewPassword DO
        TRY:
            Int (CHARACTER)
            NumberFound ← TRUE
        EXCEPT:
            LetterFound ← TRUE
    END FOR
END FUNCTION
```

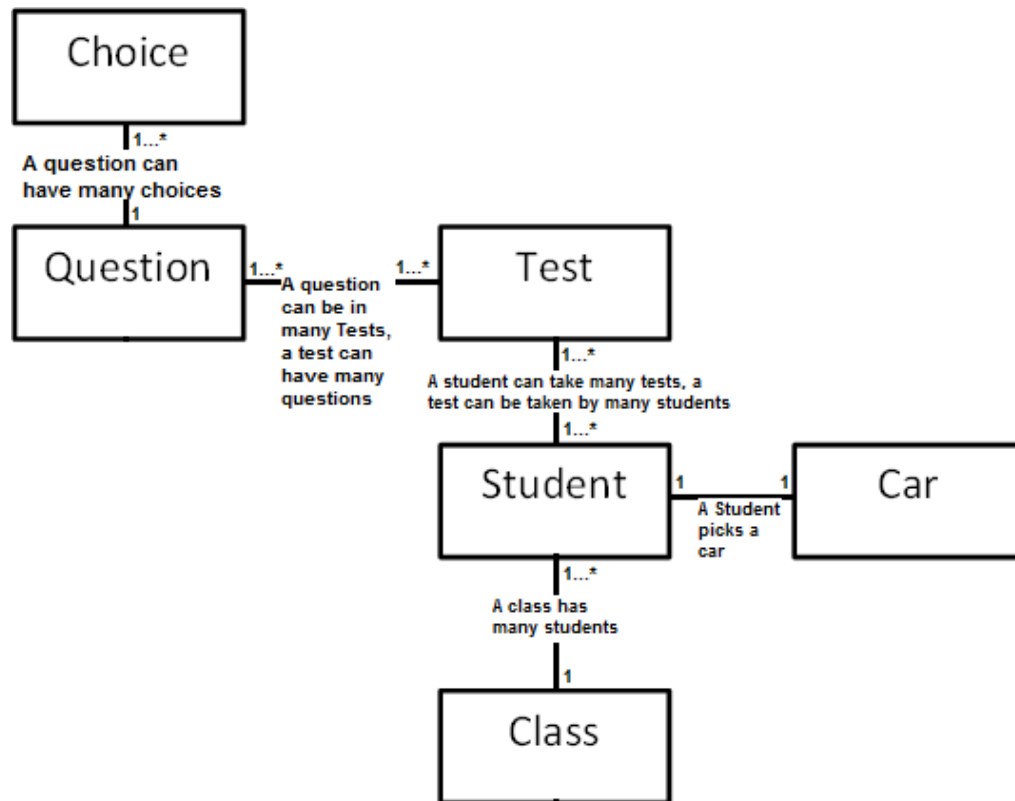
```
FUNCTION MarkAnswer (QuestionID: QUESTION, StudentAnswer: ANSWER)
    CONNECT to Question Database
    Mark: Integer
    Mark ← 0
    QuestionRecord ← SEARCH Question Database for QuestionID
    Answer = QuestionRecord.Answer
    IF StudentAnswer = Answer THEN
        Mark ← AnswerMark
    END IF
    RETURN Mark
END FUNCTION
```



```
FUNCTION AddStudent (FirstName: FNAME, LastName: LNAME, ClassID: CLASSID)
    ClassIDFound: Boolean
    CONNECT to Class Database
    CONNECT to Student Database
    ARRAY ClassList: Integer [∞]
    ClassList ← SEARCH Class Database for ClassID
    FOR Class in ClassList DO
        IF ClassID = Class THEN
            IDFound ← TRUE
        END IF
    END FOR
    IF IDFound THEN
        ADD Student Database FirstName = FNAME, LastName = LastName, ClassID
        =ClassID
    END IF
END FUNCTION
```

```
FUNCTION GetTestInfo (TestID: TESTID)
    ARRAY QuestionList: String [∞]
    CONNECT to Test Database
    CONNECT to Question database
    CONNECT to Choice Database
    CONNECT to QuestionInTest database
    Test ← SEARCH Test database WHERE TestID = TestID
    QuestionInTestList ← SEARCH QuestionInTest Database WHERE TestID = Test.ID
    FOR QuestionInTest in QuestionInTestList DO
        Question ← SEARCH Question Database WHERE QuestionID =
QuestionInTest.QuestionID
        QuestionList.Append(Question)
        ChoiceList ← SEARCH Choice Database WHERE QuestionID =
QuestionInTest.QuestionID
        QuestionList.Append(ChoiceList)
    END FOR
    RETURN Test, QuestionList
END FUJUNCTION
```

4.3 Object diagrams



4.4 Class definitions

<table> <tr> <th>Car</th> <th>Question</th> <th>Student</th> </tr> <tr> <td> CarColour CarUpgradeState CarCurrentSpeed CarDistance CarTopSpeed CarAcceleration SetCarColour SetCarUpgradeState GetCarColour GetCarUpgradeState StepForward SetCarTopSpeed SetCarAcceleration GetCarStatus IncreaseAcceleration IncreaseTopSpeed </td> <td> QuestionID QuestionType Question Answer Mark QuestionGroupID SetQuestionInfo AddToDatabase GetQuestionID GetQuestionType GetQuestion GetAnswer GetMark LoadToObject </td> <td> StudentID FirstName LastName ClassID SetStudentInfo AddToDatabase GetStudentID GetFirstName GetLastName GetClassID LoadToObject </td> </tr> </table>		Car	Question	Student	CarColour CarUpgradeState CarCurrentSpeed CarDistance CarTopSpeed CarAcceleration SetCarColour SetCarUpgradeState GetCarColour GetCarUpgradeState StepForward SetCarTopSpeed SetCarAcceleration GetCarStatus IncreaseAcceleration IncreaseTopSpeed	QuestionID QuestionType Question Answer Mark QuestionGroupID SetQuestionInfo AddToDatabase GetQuestionID GetQuestionType GetQuestion GetAnswer GetMark LoadToObject	StudentID FirstName LastName ClassID SetStudentInfo AddToDatabase GetStudentID GetFirstName GetLastName GetClassID LoadToObject	<table> <tr> <th>Class</th> <th>Choice</th> <th>Test</th> </tr> <tr> <td> ClassID Name Year GetClassID GetName GetYear SetClassInfo LoadToObject AddToDatabase </td><td> ChoiceID QuestionID Choice GetChoice SetChoice GetQuestionID SetQuestionID AddToDatabase LoadToObject </td><td> TestID Name Description GetTestID GetName GetDescription SetTestInfo AddToDatabase LoadToObject </td></tr> </table>	Class	Choice	Test	ClassID Name Year GetClassID GetName GetYear SetClassInfo LoadToObject AddToDatabase	ChoiceID QuestionID Choice GetChoice SetChoice GetQuestionID SetQuestionID AddToDatabase LoadToObject	TestID Name Description GetTestID GetName GetDescription SetTestInfo AddToDatabase LoadToObject
Car	Question	Student												
CarColour CarUpgradeState CarCurrentSpeed CarDistance CarTopSpeed CarAcceleration SetCarColour SetCarUpgradeState GetCarColour GetCarUpgradeState StepForward SetCarTopSpeed SetCarAcceleration GetCarStatus IncreaseAcceleration IncreaseTopSpeed	QuestionID QuestionType Question Answer Mark QuestionGroupID SetQuestionInfo AddToDatabase GetQuestionID GetQuestionType GetQuestion GetAnswer GetMark LoadToObject	StudentID FirstName LastName ClassID SetStudentInfo AddToDatabase GetStudentID GetFirstName GetLastName GetClassID LoadToObject												
Class	Choice	Test												
ClassID Name Year GetClassID GetName GetYear SetClassInfo LoadToObject AddToDatabase	ChoiceID QuestionID Choice GetChoice SetChoice GetQuestionID SetQuestionID AddToDatabase LoadToObject	TestID Name Description GetTestID GetName GetDescription SetTestInfo AddToDatabase LoadToObject												

The load to object method will take the raw information from the database, and add it to the attributes in the object. The save to database will save the information from the object to the database.

5. Prototyping

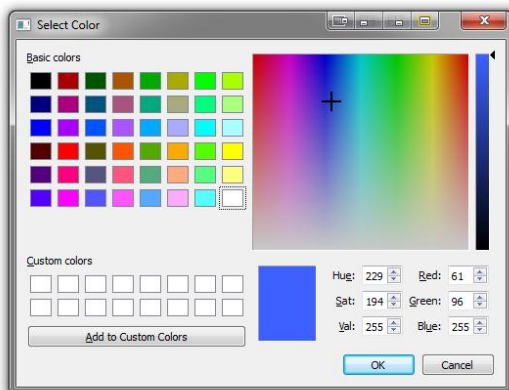
Prototyping will allow me to test out certain aspects of the program, to make sure that are feasible, and fit the user's specification.

Race prototyping

I made a prototype of the race section of the game. It allows you to pick the colours of each car, and the design, and then race 4 cars at a time. The program also works out which car will win the race.

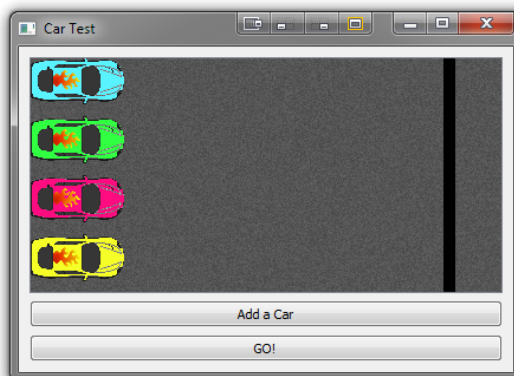
Colour selection:

This is the screen displayed for selecting the colour of the car. It uses the default colour picker for the operating system.

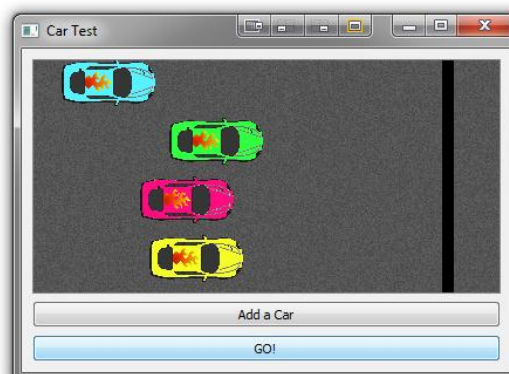


Race Screen

This is what the beginning of a race looks like. In the prototype there are buttons for add car and to start the race. In the main program, these will not be here.



This is what it looks like while the race is in progress. Each car has its own random acceleration and top speed. The program calculates where the car should be on each frame.



The program also calculates which car will get to the end first. Originally, the program detected when the car collided with the finish line, however I found this method to be inaccurate when two or more cars crossed the line at the same time. The current way it detects which car finished first is it calculates how long it will take for each car to get to the finish line. The screen shot bellow shows the output of the system for the race above. The cars each have their own number, the top most being car one. In the actual program, this data will not be printed into the console, but the user's personal position will be displayed to the screen.

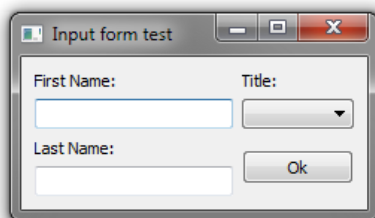
```
2
4
3
1
```

Database input form

I made a prototype of an input form with validation. If any of the fields are left blank, the program will alert the user to the problem, and prompt them to add the required information.

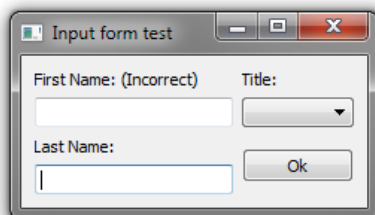
Initial screen

This is the add information form for a teacher. The user must enter something into all of the fields before the program will continue.

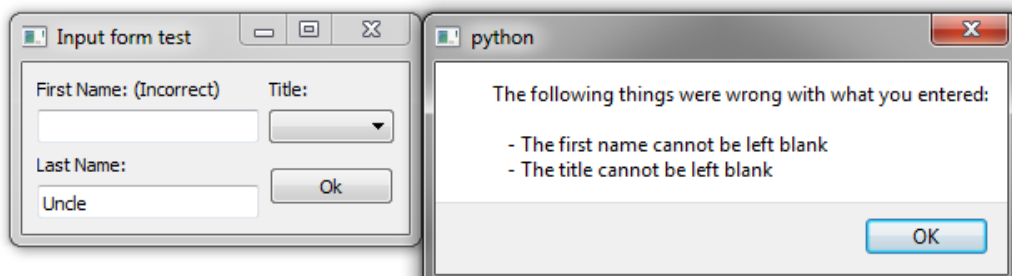
A screenshot of a Windows-style window titled "Input form test". It contains three input fields: "First Name:" (empty), "Last Name:" (empty), and "Title:" (a dropdown menu). There is an "Ok" button at the bottom right.

Validation

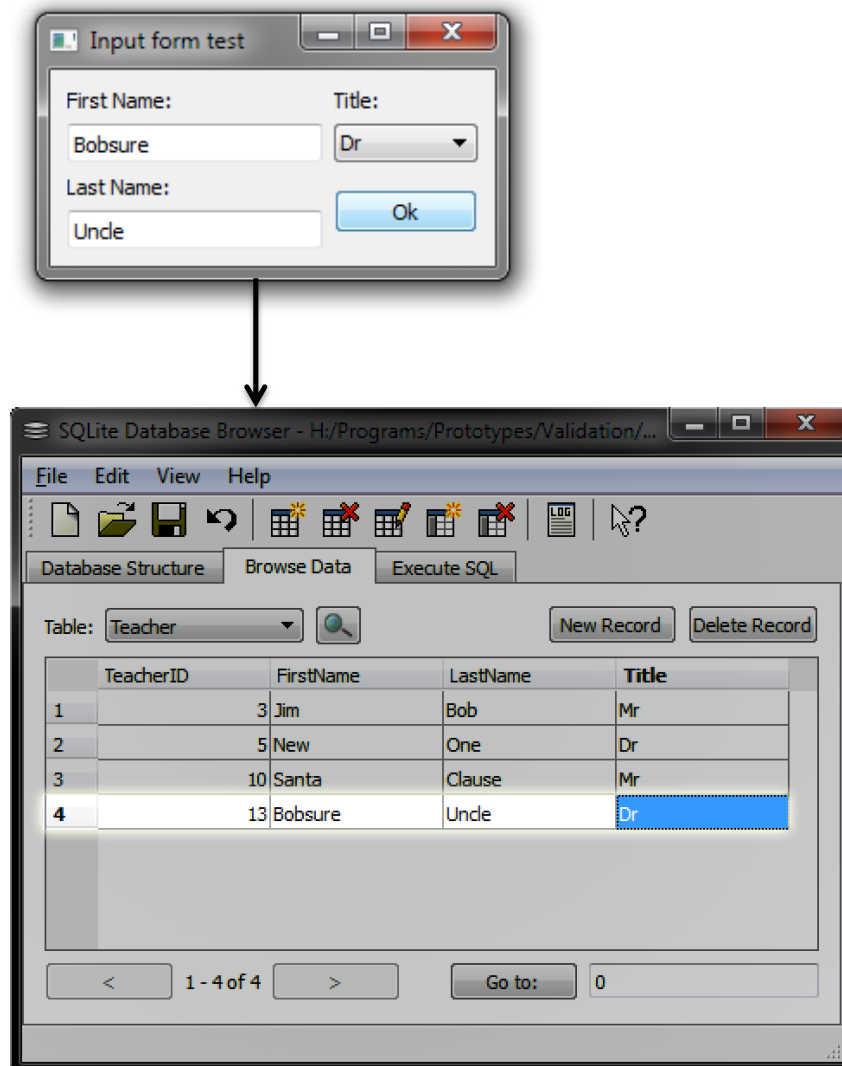
If the user clicks in a box, then without entering anything clicks in another box, a notice appears next to the label for the box.

A screenshot of the "Input form test" window. The "First Name:" label is now followed by "(Incorrect)". The "Title:" dropdown is empty. The "Last Name:" field contains the text "Uncle". The "Ok" button is still present.

If the user leaves any of the inputs empty and proceeds to click the 'OK' button, a dialogue window will appear, prompting the user to enter information into the missing boxes.

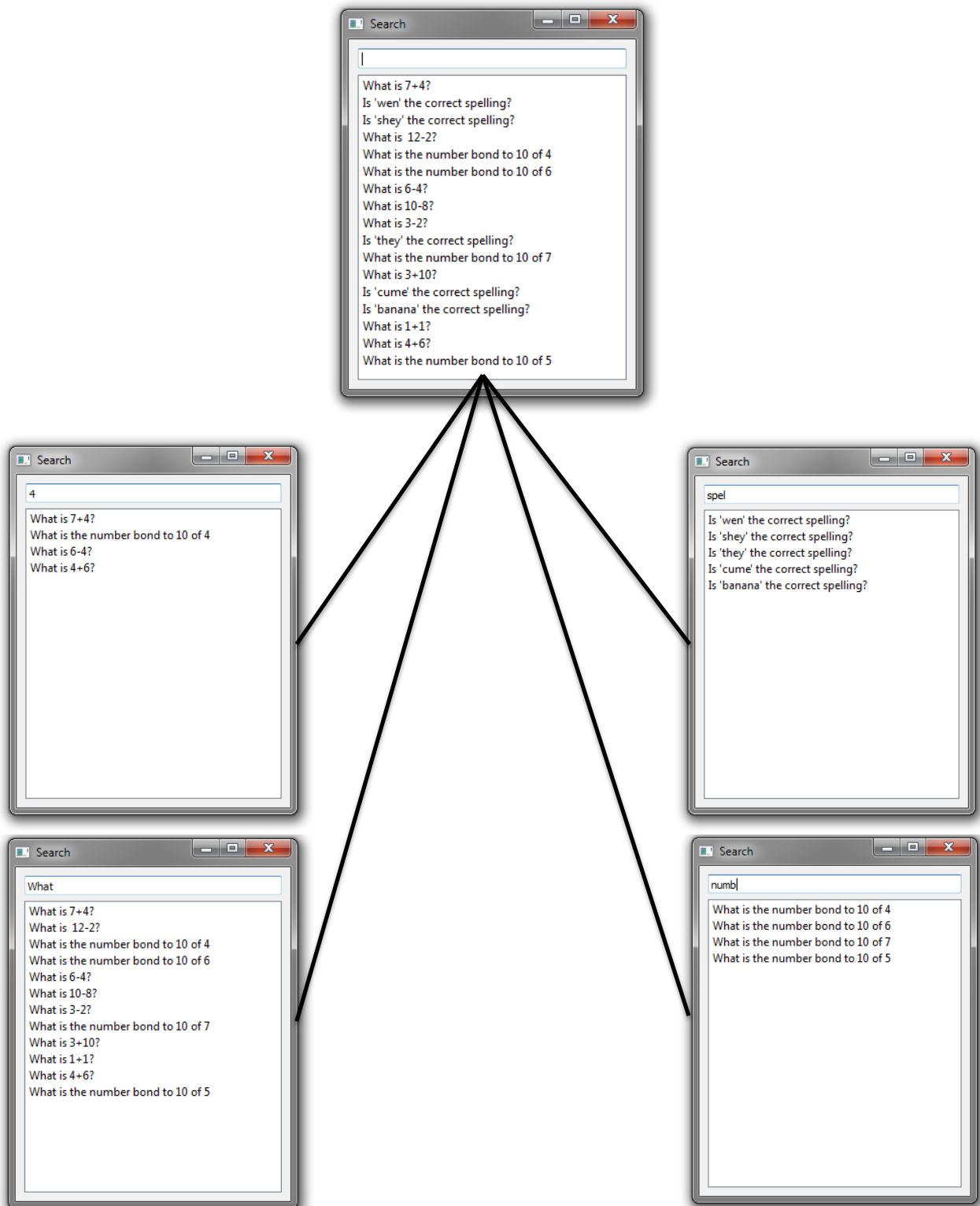
A screenshot showing two windows. On the left is the "Input form test" window, which now has "First Name: (Incorrect)", "Last Name: Uncle", and an empty "Title:" dropdown. On the right is a "python" error dialog box with the text: "The following things were wrong with what you entered:", "- The first name cannot be left blank", and "- The title cannot be left blank". There is an "OK" button at the bottom right of the dialog box.

If all of the information is entered, once the 'Ok' button is pressed, the information will be added to the database.



Search

This prototype concentrates on making a search that is instant and fast to use. I thought about having a search where the user would input some information, then click next and then the program would display the results of the search. With this method, if what the user is looking for doesn't show up with the information entered, the user then has to go back a screen, and change the information entered, and repeat this process until the item that the user is looking for shows up. The moment the search in this prototype shows up, the user is presented with all of the possible choices available. By typing in the box at the top, this list will refine instantly. This allows the user to quickly find what they are looking for.



6. Definition of data requirements

6.1. Identification of all data input items

- Student first name
- Student last name
- Student class
- Question
- Question type
- Question answer
- Question choices
- Question mark
- Test name
- Test description
- Class name
- Class year
- Teacher password
- Student's answer

6.2 Identification of all data output items

- Students previous test results
- Students attempts at particular questions

Output to database:

- Student first name
- Student last name
- Student class
- Question
- Question type
- Question answer
- Question choices
- Test name
- Test description
- Class name
- Class year

- Teacher password
- Student's answer
- Total test mark
- TestDuration
- Time of test completion
- Date of test completion
- StudentDateAdded
- TeacherDateAdded

6.3 Explanation of how Data Output Items are generated

Output	How the output is generated
Students previous test results	Fetches from TestTaken database.
Students attempts at particular questions	Fetches from Attempts database
Student first name	Teacher inputs information
Student last name	Teacher inputs information
Student class	Teacher inputs information
Question	Teacher inputs information
Question type	Teacher inputs information
Question answer	Teacher inputs information
Question choices	Teacher inputs information
Test name	Teacher inputs information
Test description	Teacher inputs information
Class name	Teacher inputs information
Class year	Teacher inputs information
Teacher password	Teacher inputs information
Student's answer	Student inputs information
Total test mark	Calculated by comparing student's answers with answers given in the question database. Marks for these questions are then tallied
Test duration	Time for test to be completed. From displaying of first question to answering of last.
Time of test completion	Taken from system time
Date of test completion	Taken from system time
StudentDateAdded	Taken from system date

6.4 Data Dictionary

Name	Data Type	Length	Validation	Example Data	Comment
StudentID	Integer	2 bytes	Number, not used before	0142	Holds Unique ID for student
StudentFirstName	String	20 chars	Not more than 20 chars, must not be left empty.	Johnson	Holds Student's first name
StudentLastName	String	20 chars	Not more than 20 chars, must not be left empty.	Boris	Holds Student's last name
TestTakenID	Integer	2 bytes	Must be a number, not used before	5478	Holds Unique ID for the current score
QuestionMark	Integer	1 byte	Must be a number, between 0 and 20	15	Holds current score out of 20 for student
Duration	Integer	2 bytes	None	612	Holds the duration of the previous test in seconds
TimeOfCompletion	String	5 chars	Must be in form HH:SS	12:34	Holds the time of completion for the previous test
DateOfCompletion	String	11 chars	Must be in form YYYY/MM/DD	2012/12/25	Holds the date of completion for the previous test
ClassID	Integer	2 bytes	Number, not used before	0142	Holds Class ID for student
ClassName	String	20 chars	Not more than 20 chars, must not be left empty.	Class A1	Holds name for the class currently loaded.
ClassYear	String	2 chars	Not more than 2 chars, must not be left empty.	1	Holds the year of the class currently loaded. Has to be text, because there is a reception year.

Question ID	Integer	2 bytes	Number, not used before	0012	Holds Question ID for a Question
Question	String	200 chars	Can't be more than 200 chars	Is 'thay' the correct spelling?	Holds the current Question
QuestionType	Integer	1 byte	None	2	Holds the question type. 1 = Multiple choice 2 = Yes/No 3 = Enter the answer with the keyboard
Answer	String	40 chars	None	They	Holds the correct answer
Choice	String	40 chars	None	Thay	Holds one of the other answers for the multiple choice questions.
TestMark	Integer	1 byte	Must be a number, between 0 and 20	15	Holds the number of marks for a question
Password	String	20 chars	Must not be blank, between 6 and 20 chars, and must contain a number, and letter	Pa\$sW0Rd	Holds the password for the teacher account
AttemptID	Integer	2 bytes	Number, not used before	1231	Holds Attempt ID for an attempt
StudentAnswer	String	40 Chars	None	Yes	Holds answer given by student on attempt.
TestID	Integer	2 bytes	Number, not used before	7154	Holds Test ID for a test
TestName	String	100 chars	None	Tricky Words Test 1	Holds the name for a test
TestDescription	String	200 chars	None	Final tricky words test of spring half term	Holds the description of a test.
QuestionInTestID	Integer	2 bytes	Number, not used before	0171	Holds Test ID for a test
StudentDateAdded	String	11 chars	Must be in form YYYY/MM/DD	2012/12/25	Holds the date the

					student was added
--	--	--	--	--	-------------------

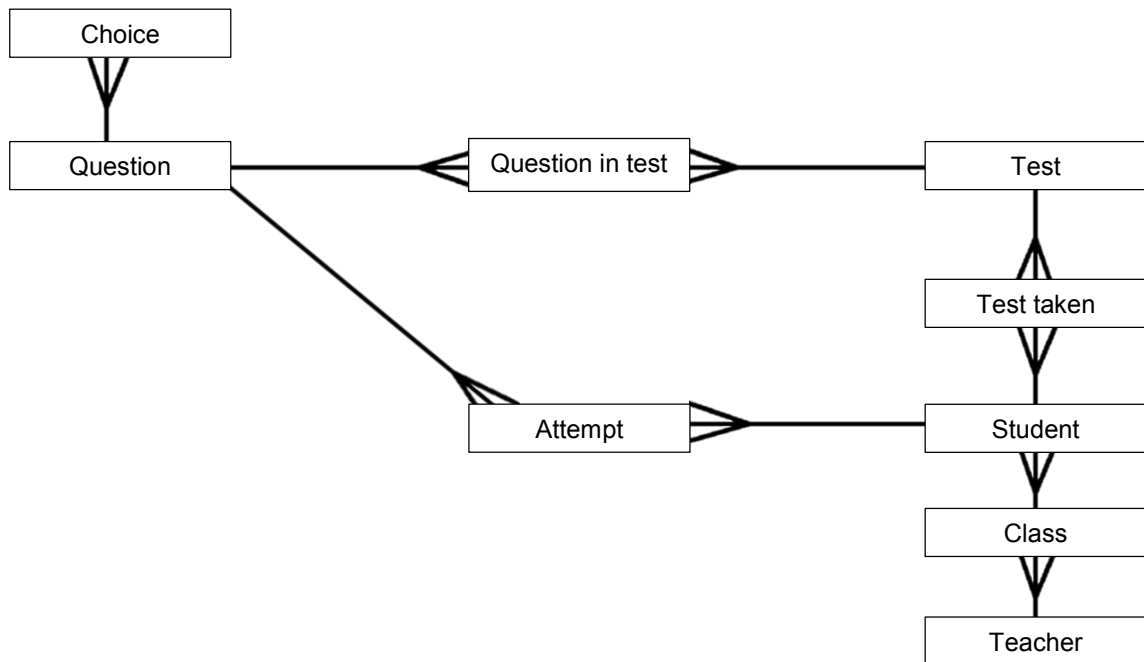
6.5 Identification of appropriate storage media

Because my system will need to be accessed by many computers at once, storing the database file separately on each computer would not be feasible, as in order to collect any information, the teacher would have to both enter all of the details on every computer, and then check every computer to see the results of the tests individually. Therefore the best place for the files of my system to be stored would be the schools server. Every computer has access to the files on this server. This way the teacher can go on any computer and enter the information needed, and view the results of tests for every student. This data could also be backed up onto the computers connected to the network, or onto a memory stick.

7. Database Design

7.1. Normalisation

7.1.1. ER Diagrams



7.1.2. UNF to 3NF

<u>Un-normalised</u>
ClassYear
ChoiceID
TeacherTitle
QuestionType
StudentID
FirstName
LastName
QuestionID
QuestionInTestID
TestTakenID
ClassID
AttemptID
TestID
Question
Choice
Answer
Mark
TestMark
Duration
TimeOfCompletion
DateOfCompletion

ClassName
StudentAnswer
QuestionGroupName
QuestionGroupDescription
TestName
TestDescription
StudentDateAdded

<u>1NF</u>	
<u>Repeating</u>	<u>Non-Repeating</u>
AttemptID	AttemptID
StudentID	StudentAnswer
FirstName	TestTakenID
LastName	
QuestionID	
ClassID	
TestID	
Question	
ChoiceID	
Choice	
Answer	
TestMark	
StudentDateAdded	
TeacherDateAdded	
ClassName	
TeacherFirstName	
TeacherLastName	
TestName	
TestDescription	
Duration	
ClassYear	
QuestionType	
Mark	
TimeOfCompletion	
DateOfCompletion	
QuestionInTestID	

<u>2NF</u>	
<u>Repeating</u>	<u>Non-Repeating</u>
AttemptID StudentID TestID QuestionID Question Answer TestMark Mark TimeOfCompletion DateOfCompletion AttemptID TestName TestDescription Duration QuestionType Choice ChoiceID QuestionInTestID StudentID FirstName LastName ClassID ClassName ClassYear StudentDate	AttemptID StudentAnswer TestTakenID

<u>3NF</u>			
<div> <div>QuestionInTestID</div> <div> <i>QuestionID</i> <i>TestID</i> <i>StudentID</i> </div> </div>	<div> <div>StudentID</div> <div> FirstName LastName <i>ClassID</i> DateAdded </div> </div>	<div> <div>QuestionID</div> <div> QuestionType Question Answer Mark <i>QuestionGroupID</i> </div> </div>	<div> <div>TestTakenID</div> <div> <i>StudentID</i> TestMark Duration TimeOfCompletion DateOfCompletion </div> </div>
<div> <div>AttemptID</div> <div> <i>StudentID</i> <i>QuestionID</i> StudentAnswer </div> </div>	<div> <div>ClassID</div> <div> ClassYear ClassName </div> </div>	<div> <div>TestID</div> <div> TestName TestDescription </div> </div>	<div> <div>ChoiceID</div> <div> <i>QuestionID</i> Choice Correct </div> </div>

Entity relationshipsQuestion(QuestionID, QuestionType, Question , Mark)QuestionInTest(QuestionInTestID, *QuestionID*, *TestID*)Test(TestID, Name, Description)TestTaken(TestTakenID, *TestID*, *StudentID*, Mark, Duration, TimeOfCompletion, DateOfCompletion)Student(StudentID, FirstName, LastName, *ClassID*)Class(ClassID, Name, Year)Attempt(AttemptID, *QuestionID*, *StudentID*, Answer)Choice(ChoiceID)**7.2. SQL Queries**

For these SQL statements, I have formatted them with the python .format() function to format the SQL strings.

SQL	Description
<pre>"""insert into Student(FirstName, LastName, ClassID) values ('{0}', '{1}', '{2}')</pre> <pre>"".format(FirstName, LastName, ClassID)</pre>	This is an example of an SQL statement to add records to the database. In this case, it is entering a new student record with the attributes <i>FirstName</i> , <i>LastName</i> and <i>ClassID</i> .
<pre>"""create table student(StudentID INTEGER, FirstName TEXT, LastName TEXT, ClassID INTEGER, PRIMARY KEY(StudentID)) FOREIGN KEY(ClassID) REFERENCES Class(ClassID) """</pre>	This is an example of an SQL statement that creates a new table called student, with the attributes <i>FirstName</i> , <i>LastName</i> and <i>ClassID</i> . The primary key is set to <i>StudentID</i> , and there is a foreign key that is <i>ClassID</i> .
<pre>"""select * from Student where FirstName = '{0}' and LastName = '{1}' """</pre> <pre>"".format(FirstName, LastName)</pre>	This statement will return all of the records from the <i>Student</i> table that have the first name <i>FirstName</i> , and the last name <i>LastName</i>
<pre>"""delete from Student where StudentID = '{0}' "".format(StudentID)</pre>	This will delete the student from the <i>Student</i> table with the ID of <i>StudentID</i> .
<pre>"""select Answer</pre>	This will fetch the answers a

<pre> From Attempt, Student Where StudentID = '{0}' and QuestionID = '{1}' """.format(StudentID, QuestionID) </pre>	<p>student with the ID of <i>StudentID</i> gave for the question with the ID of <i>QuestionID</i></p>
<pre> """update Student set FirstName = '{0}' LastName = '{1}' ClassID = '{2}' where StudentID = '{3}' """.format(StudentFN, StudentLN, StudentClass, ID) </pre>	<p>This will change the information in the Student table for the student with the ID <i>StudentID</i>. It will set their First Name to <i>StudentFN</i>, their Last Name to <i>StudentLN</i>, and their Class to <i>StudentClass</i>.</p>

8. Security and Integrity of the System and Data

8.1. Security and Integrity of Data

This system will store personal information about the students that fall under the data protection act. This means that the data must be kept up to date, and so there will be a way to edit information in the program. The data stored also shouldn't be older than 10 years old, so when the program starts up, it will check for old data and automatically delete it. This also means that all of the information in the database must be encrypted to securely store this data so that the only way anyone can gain access to the database is via my program, where all of the data will be stored behind a password. To make sure that the data that is stored is also valid, at the input stage, as many drop down menus will be used as possible, and wherever the user types in the information via the keyboard, the data will be checked to make sure that it is feasible. I will also need to make sure that I keep referential integrity in my database. This means I need to add checks when adding and removing data to the database to make sure there are never any records with missing key data.

8.2. System Security

It is important that the system is protected from data theft, corruption and tampering. This will be done via the use of a password to gain access to the information. This will limit the number of people who have access to the system. If the password is entered incorrect, the user will not be able to gain access to the information. The database will also be encrypted to avoid people accessing the information without the use of the system. Therefore I can determine in the program exactly how I want people to be able to access the data. All of the data stored by this system will also have to go through validation on input, and so be correct; however, if any of it is wrong or needs to be updated, this can be done via the system.

Because this data falls under the data protection act, I need to make sure that it is followed:

- The data will not be transferred to other countries.
- The data will be secured securely, to ensure that it is only accessed by authorised people
- The data that is stored will only be used by the school and not passed on to anyone else
- The data will be destroyed after 11 years of collection
- The data will be updated when necessary so that the data is up to date and accurate.
- Only data that is necessary will be collected and stored.

9. Validation

To avoid any incorrect data entries are added to the database, the system will check to make sure each field is acceptable.

Item	Example	Validation/Verification applied	Comments
Question	Is 'thay' a word?	Presence check	To ensure that a question is entered. No other validation checks other than presence as a question could be any length, and doesn't have a common structure.
Question Type	Multiple choice	Lookup check Presence check	There are only three question types, so a lookup table will be used to avoid errors.
Choice	Thee	Presence check	One choice for the question will be held in this variable
Correct	True	Presence check	This will be a Boolean and will correspond to a choice as to whether it is the correct answer or not.
TestName	Spelling	Presence check	To ensure that a test name is entered.
TestDescription	Spelling test for year 3. Level 2 section A	Presence check	To ensure that a test description is entered.
StudentFirstName	Bob	Presence check	To ensure that a name is entered.
StudentLastName	Smith	Presence check	To ensure that a name is entered.
StudentClass	A	LookupCheck Presence check	To ensure the class has already been created
ClassName	A	Presence check	To ensure that a name is entered.
ClassYear	4	Presence check Type check	To ensure a year has been entered and is in numerical form
StudentAnswer	They	Presence check	To ensure that the student enters an answer
Password	Pa\$sw0r12	Presence check Length check Type check	To ensure a password is entered, more than 6 characters, and contains a number and letter

10. Testing

10.1. Outline Plan

10.1.1. Identification and explanation of suitable test strategies

Test series	Purpose of test series	Testing strategy	Strategy Rationale
1	Test the flow of control between the user interfaces	Top-down testing	
2	Test validation of data input is detected	Bottom-up testing	Each component will be tested once it is developed
3	Test information input is stored in the correct place	Black box testing	
4	Test algorithms to make sure that the output is correct	White box testing	
5	Test that the system fulfils the specification	Acceptance testing	

10.2 Detailed Plan

Test Series and number	Purpose	Description	Test Data	Test data type	Expected Result	Actual result	Evidence in appendix
1.01	Test the student log in button on the main menu functions correctly	This should link to a screen where the student selects their name	Click the Student button	Normal	The quiz selection screen should be displayed		
1.02	Test the Teacher log in button on the main screen functions correctly	This should open a new window where the teacher can enter their password	Click the exit button	- Normal	A window should appear asking for the teacher password.		
1.03	Test the 'OK' button on the quiz selection screen functions correctly	This should link to a screen where the student can select the colour of their car	Click the Teacher button	- Normal	The car colour selection screen should be displayed		
1.04	Test the 'Log Out' button on the student select screen functions correctly	This should link to a back to the main screen	Click the 'Log out' button	- Normal	The Main screen should be displayed		
1.05	Test the 'OK' button on the car colour selection screen functions correctly	This should link back to the main menu	Click the 'Back' button	- Normal	The		
1.06	Test the 'Ok' button on the test topic select screen functions correctly	This should link to a the test screen	Click the 'Ok' button	- Normal	The test screen should be displayed		
1.07	Test the Answer buttons function correctly	This should link to the next question. If there are no more	Click the buttons for each answer type button	- Normal	The next question should be displayed. If there are no		

		questions, the race screen should be displayed.			more questions, the race screen should be displayed.		
1.08	Test the 'Ok' button on the smiley sticker reward screen dismisses the window	This should close the pop up window	Click the 'Ok' button	- Normal	The pop up window should close		
1.09	Test the 'Cancel' button on the teacher password screen functions correctly	This should link back to the main menu	Click the 'Cancel' button	- Normal	The Main menu screen should be displayed.		
1.10	Test the 'Ok' button on the teacher password screen functions correctly	This should link to the Teacher main menu screen.	Click the 'Ok' button	- Normal	The Main menu bar should appear at the top if the password is correct.		
1.11	Test the 'Log out button' under options in the menu at the top functions correctly	This should cause the teacher menu bar to go away.	Click the 'Options' button, then 'Log Out'	- Normal	The teacher menu bar should disappear		
1.12	Test the 'Change Password' under options in the menu at the top functions correctly	This should cause the change password screen to appear.	Click the 'Options' button, then 'Change Password'	- Normal	The change password window should appear		
1.13	Test the 'Cancel' button on the teacher password screen functions correctly	This should close the window.	Click the 'Cancel' button	- Normal	The change password screen should close.		
1.14	Test the 'Change Password'	This should close the window. If	-Click the 'Change Password'	- Normal	The change password		

	button on the teacher password screen functions correctly	any of the boxes are incorrect, the window should stay open, and a alert window should appear.	button with correct information -Click the change password button with incorrect information	-erroneous	screen should close. -An alert screen should appear telling the user what is wrong with the information input.		
1.15	Test the 'Database Setup' under options in the teacher menu functions correctly.	This should show a window where database settings can be applied/	-Click Database setup	- Normal	A screen where the user can change the database settings should appear.		
1.16	Test the 'Browse' button on the database setup screen functions correctly.	The default file browser for the system should open, allowing the user to select the location or their database to be stored in.	Click the 'Browse' button	- Normal	The default file browser should appear		
1.17	Test the 'Cancel' button on the database setup screen functions correctly.	This should close the window	Click the 'Cancel' button	- Normal	The window should close		
1.18	Test the 'Ok' button on the database setup screen functions correctly.	This should close the window and move the database save folder to the current location in the box	Click the 'OK' button	- Normal	The window should close		
1.19	Test the 'Student' button under the	This should open the Add student	Click the 'Student' button under add	- Normal	The add student information screen		

	add information menu functions correctly.	information window.	information		should be displayed.		
1.20	Test the 'Test' button under the add information menu functions correctly.	This should open the Add test information window.	Click the 'test' button under add information	- Normal	The add test information screen should be displayed.		
1.21	Test the 'Class' button under the add information menu functions correctly.	This should open the Add class information window.	Click the 'Class' button under add information	- Normal	The add class information screen should be displayed.		
1.22	Test the 'Question' button under the add information menu functions correctly.	This should open the Add question information window.	Click the 'Question' button under add information	- Normal	The add question information screen should be displayed.		
1.23	Test the 'Student' button under the remove information menu functions correctly.	This should open the remove student information window.	Click the 'Student' button under remove information	- Normal	The remove student information screen should be displayed.		
1.24	Test the 'Test' button under the remove information menu functions correctly.	This should open the Remove test information window.	Click the 'test' button under remove information	- Normal	The remove test information screen should be displayed.		
1.25	Test the 'Class' button under the remove information menu functions correctly.	This should open the Remove class information window.	Click the 'Class' button under remove information	- Normal	The remove class information screen should be displayed.		
1.26	Test the	This should	Click the	- Normal	The		

	'Question' button under the remove information menu functions correctly.	open the Remove question information window.	'Question' button under remove information		remove question information screen should be displayed.		
1.27	Test the 'Test' button under the edit information menu functions correctly.	This should open the Edit test information window.	Click the 'test' button under edit information	- Normal	The edit test information screen should be displayed.		
1.28	Test the 'Class' button under the edit information menu functions correctly.	This should open the Edit class information window.	Click the 'Class' button under edit information	- Normal	The edit class information screen should be displayed.		
1.29	Test the 'Question' button under the edit information menu functions correctly.	This should open the Edit question information window.	Click the 'Question' button under edit information	- Normal	The edit question information screen should be displayed.		
1.30	Test the 'Test' button under the edit information menu functions correctly.	This should open the Edit test information window.	Click the 'test' button under edit information	- Normal	The edit test information screen should be displayed.		
1.31	Test the Create test button in the Add Test window works	The window should close, and the data entered should be saved to the database.	Click 'Create Test'	Normal	The window should close		
1.32	Test the Create student button in the Add Student	The window should close, and the data entered	Click 'Create Student'	Normal	The window should close		

	window works	should be saved to the database.					
1.34	Test the Create question button in the Add Question window works	The window should close, and the data entered should be saved to the database.	Click 'Create Question'	Normal	The window should close		
1.35	Test the Create class button in the Add Class window works	The window should close, and the data entered should be saved to the database.	Click 'Create Class'	Normal	The window should close		
1.36	Test the Cancel button in the Add Test window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.37	Test the Cancel button in the Add Student window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.38	Test the Cancel button in the Add Question window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.39	Test the Cancel button in the Add Class window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.40	Test the Save Changes button in the Edit Test window works	The window should close, and the data entered should be saved to	Click 'Save Changes'	Normal	The window should close		

		the database.					
1.41	Test the Save Changes button in the Edit Student window works	The window should close, and the data entered should be saved to the database.	Click 'Save Changes'	Normal	The window should close		
1.42	Test the Save Changes button in the Edit Question window works	The window should close, and the data entered should be saved to the database.	Click 'Save Changes'	Normal	The window should close		
1.43	Test the Save Changes button in the Edit Class window works	The window should close, and the data entered should be saved to the database.	Click 'Save Changes'	Normal	The window should close		
1.44	Test the Cancel button in the Edit Test window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.45	Test the Cancel button in the Edit Student window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.46	Test the Cancel button in the Edit Question window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.47	Test the Cancel button in the Edit Class window works	The window should close	Click 'Cancel'	Normal	The window should close		

1.48	Test the Remove button in the Remove Test window works	The window should close	Click 'Remove'	Normal	The window should close		
1.49	Test the Remove button in the Remove Student window works	The window should close	Click 'Remove'	Normal	The window should close		
1.50	Test the Remove button in the Remove Question window works	The window should close	Click 'Remove'	Normal	The window should close		
1.51	Test the Remove button in the Remove Class window works	The window should close	Click 'Remove'	Normal	The window should close		
1.52	Test the Cancel button in the Remove Test window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.53	Test the Cancel button in the Remove Student window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.54	Test the Cancel button in the Remove Question window works	The window should close	Click 'Cancel'	Normal	The window should close		
1.55	Test the Cancel button in the	The window should close	Click 'Cancel'	Normal	The window should close		

	Remove Class window works						
1.56	Test the 'New Class' button works in the add and edit student windows.	A new add class window should appear.	Click the New Class Button.	Normal	A new add class window should appear.		
1.57	Test the 'Create New' button in the add and edit Test window.	A new add Question window should appear.	Click the Create New button	Normal	The add question window should appear.		
1.58	Test the 'Edit' button in the add and edit Test window.	A new edit question window should appear.	Click the Create edit	Normal	A new edit question window should appear.		
1.59	Test the view results button functions correctly	The view results search window should appear.	Click the view results on the teachers main menu.	Normal	The view results search window should appear		
1.6	Test the cancel button in the in the results search window works	The window should close	Click the 'Cancel' button	Normal	The window should close		
1.61	Test the 'View results button' works in the search window works.	The selected student's pervious marks window should appear.	Click the view results button.	Normal	The students previous results screens should appear		
1.62	Test the More info button works. In the students previous marks screen	The test breakdown screen should be displayed	Click the More info button	Normal	The test breakdow n screen should be displayed.		

1.63	Test the back button on the Test breakdown screen	Should go back to the students previous marks screen	Click the 'back' button	Normal	The Student's previous marks screen should be displayed		
2.01	Verify a question was entered	The input box should show an error if the box is left empty	- What is 1+1? - Nothing	Normal erroneous	Accept Error		
2.02	Verify that the type is entered and correct	The input box should show an error if the box is left empty and isn't either Multiple choice, Yes/No or Entered with keyboard.	- Multiple Choice - Nothing - Random	Normal erroneous erroneous	Accept Error Error		
2.03	Verify all three choices are entered	All three choices should be entered	-a,b,c -a,b -Nothing	Normal erroneous erroneous	Accepted Error Error		
2.04	Verify an answer is entered	The field should not be left blank	-Nothing -Banana	erroneous Normal	Error Accepted		
2.05							
2.06	Verify the test name is entered	The field cannot be left blank	-Nothing -Spelling	erroneous Normal	Error Accepted		
2.07	Verify the test description is entered	The field cannot be left blank	-Nothing -Spelling test for 4 th Jan 2013.	erroneous Normal	Error Accepted		
2.08	Verify the student first name is entered	The field cannot be left blank	-Nothing -Bobsure	erroneous Normal	Error Accepted		
2.09	Verify the student last name is entered	The field cannot be left blank	-Nothing -Uncle	erroneous Normal	Error Accepted		
2.1	Verify the student class is entered	The field cannot be left blank and must be one of the previously defined classes	-Nothing -Class A -Classg2(if this class wasn't pre-defined)	erroneous Normal erroneous	Error Accepted Error		
2.11	Verify the	The field	-Nothing	erroneous	Error		

	class name is entered	cannot be left blank	-Class A	Normal	Accepted		
2.12	Verify the class year is entered	The field cannot be left blank	-Nothing -3	erroneous Normal	Error Accepted		
2.13	Verify the student entered an answer	The field cannot be left blank.	-Nothing -They	erroneous Normal	Error Accepted		
2.14	Verify a password is entered, and contains at least 6 characters, a number, and a letter	The field cannot be left blank and must contain at least 6 characters, a number, and a letter	-Nothing -passw0rd -pas5 -password -123456	erroneous Normal erroneous erroneous erroneous	Error Accepted Error Error Error		
3.1	Verify that all the student details entered are added to the student database	All of the information should be added to the correct fields in the student table.	Student information	Normal	Added to the student table		
3.2	Verify that all the teacher details entered are added to the student database	All of the information should be added to the correct fields in the teacher table.	Teacher information	Normal	Added to the teacher table		
3.3	Verify that all the class details entered are added to the student database	All of the information should be added to the correct fields in the class table.	class information	Normal	Added to the class table		
3.4	Verify that all the question details entered are added to the student database	All of the information should be added to the correct fields in the question table.	Question information	Normal	Added to the question table		
3.5	Verify that all the test details entered are added to the student database	All of the information should be added to the correct fields in the test table.	Test information	Normal	Added to the test table		
3.6	Verify that the student's	All of the information should be	Attempt information	Normal	Added to the attempt		

	attempt information is stored in the attempt database	added to the correct fields in the test attempt.			table		
3.7	Verify that the student's test result information is stored in the attempt database	All of the information should be added to the correct fields in the test results attempt.	Attempt information	Normal	Added to the test results table		
3.8	Verify that the when adding a question to a test, it is added to the question in test database	All of the information should be added to the correct fields in the question in test attempt.	Questions in a test information	Normal	Added to the question in test table		
4.1	Verify that the questions are marked correctly	If the answer given by the student matches the answer stored in the question database, then a correct answer is given.	-Answer all of the answers correct -Answer all of the answers wrong -Answer half of the answers correct	Normal	-Full marks should be given -No marks should be given -Half of the marks should be given		
4.2	Verify the total marks is calculated correctly	If the answer is correct, then the marks allocated to that question should be added to the total	Do a test, answering some correctly and some incorrectly, calculate the marks manually, then see if these calculated marks match up	Normal	The total given by the program should match the calculated value.		
4.3	Verify that when changing the password, the old password is correctly	If the old password entered does not match the old password stored, the	-Try changing the password with an incorrect old password	Normal	-Error -Accepted		

	checked against the one entered by the user.	change to the new password should not be accepted	-Try changing the password with the correct old password				
4.4	Verify that the search function works correctly	Entering details about a student should search the database for this student and return all of the information back.	-Enter some information of a student in the database	Normal	Return all of the information about student's with that information back to the screen		
5	Verify the program fulfils the specification	Run through the program, testing the different aspects to make sure they fit the objectives in the specification	Add some information to the program, start a student test, and view the results of this test	Normal	Program fulfils the specification		