

Project

iPad application to aid learning within primary schools

Author : Victoria Green
Supervisor: Alia I Abdelmoty
Moderator: Stuart M Allen

CM3203 One Semester Project (40 Credits)

Abstract

The project involves developing a system for use in primary schools which will facilitate students learning and enable teachers to assess the students' progress. The system will comprise of an iPad application to obtain student assessment data and a website which will facilitate administration of the system and allow the teacher to view student's data.

The application will be aimed at government run schools within wales and will focus on one of the compulsory subject areas; mathematics. I will also develop this application to specifically target students in year 6. The application will be programmed in Xcode which is the necessary environment for developing iPad applications.

Acknowledgements

I would like to express my very great appreciation to Alia Abdelmoty for her encouragement, suggestions and support throughout the project.

I would also like to offer a special thanks to Lisa Jopson for providing me with useful resources and insight to academic assessments.

In addition I would like to thank Rob Davies for his assistance with setting up an Apple developer account.

Table of Contents

Abstract.....	1
Acknowledgements.....	1
1. Introduction	4
2. Background.....	5
3. Requirements analysis.....	6
3.1. Identifying Users - Personas.....	6
3.2. Defining the problem.....	7
3.3. Website Usability Guidelines.....	10
3.4. Existing system analysis - Incerts.....	11
3.5. Outcome of 'Incerts' Evaluation	20
3.6. Application Usability Guidelines.....	20
3.7. Existing System Analysis – Splash Maths	22
3.8. Outcome of 'Splash Maths' Evaluation.....	32
4. Requirement Specification	34
5. Functional Design.....	37
5.1. Use Case Diagram & flow of events.....	37
5.2. Sequence Diagram	43
5.3. Class Diagram.....	48
5.4. Entity Relationship Diagram	49
5.5. Algorithms	50
6. User Interface Design	52
6.1. Website Interface Design.....	52
6.2. Application Interface Design	62
7. Test Plan	65
7.1. Test Cases	65
7.2. Heuristic Evaluation	66
7.3. User Testing.....	66
8. Phase 1 Testing.....	68
8.1 Heuristic Evaluation	68
8.2 User Testing.....	69
9. Evaluation of design prototype.....	70
10. Implementation.....	71
10.1. Tools Utilised	71

10.2. System Overview	72
10.3. Website Structure	73
10.4. Website implementation.....	76
10.5. Website Implementation problems	78
10.6. Application Structure	78
10.7. Application implementation.....	80
10.8. Application Implementation problems	81
11. Phase 2 testing.....	83
11.1. User Testing.....	83
11.2. Test Cases.....	85
12. Evaluation.....	87
13. Future Work	88
14. Conclusion	90
15. Reflection.....	91
16. References.....	93
Appendices	95
Appendix A – National Numeracy Test.....	95
Appendix B – Level Descriptors	101
Appendix C – Use case flow of events.....	104
Appendix D – Sequence Diagrams.....	109
Appendix E - Algorithms to generate questions	112
Appendix F –Test cases.....	116
Appendix G – Implementation code.....	124

1. Introduction

The following report outlines each stage of the project to develop a system comprising of an iPad application, database and website to aid student learning and teachers' assessment. The system is intended to support the subject of mathematics for year 6 pupils (10-11 year olds) and teachers.

The iPad application will have an assessment section which will assess the student's mathematical ability and generate an ability level which will be stored in a database. The data in the database will then be accessed by the teacher via a website to facilitate the monitoring of students development. The application will also have a practical practise section which will generate random questions for the student to answer. Questions generated in the practise section will be generated based on the student's ability level ensuring the difficulty of the questions is suitable for the student's ability. The assessment section is intended for use in the classroom and the practise section for use by students in their free time.

There are several assessment tools and learning application which exist already however there isn't a system in place which both assesses a student and enables the student to practise questions. The aim of the project is to create a system which both enables interactive assessments and interactive learning. The program I will be developing will be intended for use to aid teacher's classroom assessments and not a replacement for the current national tests.

To develop the system I will follow the user centred design approach. This approach will enable me to include the user at each stage of the development ensuring that the system is suitable for my intended users.

2. Background

In schools assessing learning is an important process for teachers to use in order to establish the level of a student's ability enabling them to determine areas where they need to progress in addition to understanding areas where schools need to improve. At present, schools are currently using a selection of assessment methods which include MaLT tests and national numeracy tests in addition to teacher's individual assessments.

All UK schools legally have to follow the national curriculum. The national curriculum sets out the standards and subjects primary schools and secondary schools must teach (Wales.gov.uk, 2012). In addition a national Literacy and Numeracy Framework (LNF) has been introduced which is a curriculum planning tool for teachers and its use is compulsory in schools in Wales (National Literacy and Numeracy Framework, 2013). The content of the assessments conducted within schools is based on students understanding of the content set out in the National Curriculum and National Literacy and Numeracy Framework.

MaLT tests (Mathematical Assessments for learning and Teaching) are conducted within schools in order to identify areas for improvement within mathematics for the following year and are standardised allowing results to be compared across different schools. The national numeracy test is a combination of numeracy reasoning questions and numeracy procedure questions used to assess student's numerical ability and it is a statutory requirement for schools in Wales to conduct this test annually (Wales.gov.uk, 2014).

Teacher's individual assessments for mathematics in year 6 consist of written tests and observations of student's mathematical ability in the classroom. The teacher creates their own test to assess the students with each test focusing on one of the National Curriculum objectives and skills to be identified in that test being derived from the National Numeracy Framework.

To record each of the students assessment levels a program called Incerts is currently being used (Incerts.org, 2013). Incerts is a software program which allows the teacher to store and view all student achieved levels. Teachers can view students by class, including those they do not teach, and individually. Details of levels for each subject are stored along with the student's name. The teacher of the students is responsible for inputting assessment result into the system.

3. Requirements analysis

3.1. Identifying Users - Personas

In order to fully understand and clarify the audience for my system I have created personas for the various types of user that I would expect to use the system. The personas will be fictional characters representing the goals, attitudes and characteristics of real world users. I have identified students and teachers as the target audience and in order to create personas for these user groups I have spoken to a range of people representing these groups and have combined their common attributes, goals and characteristics to form the following personas.

Key Characteristics

Name: Catherine Matthews

Age: 35-45

Occupation: Primary School Teacher

Hobbies: Reading, Music, Socialising

Beliefs: Every student can achieve if they are given the right support.

She is familiar with popular technology and uses it for social media and browsing the internet. She has no interest in the technical aspects of how technology works but enjoys discovering technology that makes tasks easier for her such as communications and online shopping.

In addition she likes the services that the Incert's software offers as she like to see the statistics it produces through her input. She also likes all her student assessment data to be stored on the system as she feels it's secure and doesn't have to worry about losing the data. She is aware of the importance of backing up data but in reality doesn't back her data up often.

She often uses word processing system to create worksheets and school reports and so is competent with using graphical user interfaces for various applications. A lot of her time is spent on planning for lessons and assessments creating worksheets and developing question sets for the assessments. She also spends a lot of time marking these assessments and the turnaround from assessments to results can take up to two weeks because her time for marking is limited. She often uses her time outside of her working hours to catch up on marking assessments and planning for them and would much prefer to use her free time to enjoy her hobbies and spend time with family.

Goals & Desires:

- She would like to reduce the amount of time she spends planning for assessments.
- She would like to incorporate new technology into her lessons as she can see from student reactions that they are excited by new technology.
- She would like to reduce the content that she has to mark to make time for other things.

Key Characteristics

Name: Robert Evans

Age: 9-13

Occupation: Full-Time Student

Hobbies: Football, Riding his bike, Television

Beliefs: Friends and family are important.

His favourite time in school is dinner hour where he can play games with his friends. He really wants his own iPad after playing on his friend's one. He loves getting a chance to play on the schools iPad's and knows how to use the touchscreen interface. He likes his mathematic lessons and realises that the more questions that he practises the better he gets with his calculations. He enjoys demonstrating his mathematic ability to his parents.

Goals & Desires:

- He wants to get a good school report to show his parents.
- He wants more lessons where he gets to use the schools iPad.

3.2. Defining the problem

In order to determine the requirements for the system I have spoken with a primary school teacher, Mrs Jopson, to establish what the current process is for assessing students' ability. From this discussion I have developed an understanding of the schools current processes and how this could be implemented into the system I will be developing. Based on this discussion I have produced an initial definition of the systems requirements as follows:

- Assessment questions must cover the content set out in the national curriculum. In order for the system to assess a student's level the questions used to calculate the level must correspond to the content which is currently being used to assess a student's ability which is covered by national curriculum for key stage 2.
- Teacher must be able to view the current level a student is achieving. The outcome of each assessment must be calculated and it should be clear to the teacher what the student's current level is. In addition, the teacher should be able to compare one student's achievements to the rest of the class so that the teacher can establish whether a student is achieving what they should at a particular time in the school year.
- Application language must be simple for use by year 6 students. In order to ensure that the students can understand the questions and how to use the application simple language should be used.
- Each assessment should consist of 15 questions and each one should be different.

In order to test a student's ability in mathematics there should be a selection of questions for a student to answer to ensure they are able to correctly answer a range of questions. Having discussed with the teacher a selection of 15 questions has been determined appropriate.

- Students should answer a minimum of 10 questions correctly in order to qualify to have achieved the level of the assessment they have taken.
A teacher determines whether a student is achieving a level if they can "more often than not" correctly answer a question. To ensure that a student would need to correctly answer most questions "more often than not" it has been determined that 10 correctly answered questions would be suitable to determine that a student has achieved the level.
- Practise questions should inform the student what the correct answer is. Whilst practising the question a student should receive feedback to inform them of whether they are correct or incorrect. In addition the correct answer should be displayed to enable them to understand the correct way to answer the questions. This should not be done during an assessment as it should have the same format of a real assessment whereby they do not know how they have done until the end.

In order to establish the requirements for the content of the mathematics I have researched the mathematics section of the Welsh National curriculum (Education.gov.uk, 2000). The national curriculum is sectioned into categories referred to as key stages which correspond to year groups within the school. Key stage 2 covers the subjects and standards for primary school years 3 to 6. The mathematics assessment will therefore need to conform to Key stage 2 of the national curriculum in order for it to apply to year 6 students. Due to the mass amount of content covered in primary school mathematics and the small time given to complete the project I have chosen to focus the application on numeracy skills which will consist of addition, subtraction, multiplication, division and ordering. I have also obtained level specifications and national numeracy tests from the teacher. I have used the level specifications to classify mathematical questions into levels for the assessments and the example national numeracy tests to aid the development of the question sets for the assessment.

For example national numeracy test, see Appendix A.
For level descriptors, see Appendix B.

Using the levels descriptors I have devised the following rules for each level:

- **Level 1** student should be able to count, order, add and subtract positive numbers up to 10.
- **Level 2** student should be able to order add and subtract positive numbers up to 100.

- **Level 3** students should be able to calculate multiplications of, 2, 3, 4, 5, & 10 by numbers up to 10.
- **Level 4** students should be able to divide whole numbers by 10 and 100. Students should know all multiplications up to 10. Students should be competent with addition and subtraction with decimals to two places.
- **Level 5** students should be able to multiply and divide whole numbers and decimals. Students should be able to order add and subtract negative numbers.

Having read through a past national numeracy test I have identified several questions relevant to what the application I design will be testing. I will use the style of these questions within the application. Using the national numeracy past paper as a guide I have derived the following types of questions:

Level 1:

- *Ordering*: Which number is the largest, X or Y? (Where both X and Y are less than or equal to 10 and are not equal to each other)
- *Counting*: Which number is next after X? (Where X is a number between 1 and 9)
- *Addition*: What is $X + Y$? (Where the answer is positive and less than 10)
- *Subtraction*: What is $X - Y$? (Where the answer is a positive number and both X and Y are less than 10)

Level 2:

- *Ordering*: Which number is the largest, X or Y? (Where both X and Y are less than or equal to 100 and are not equal to each other)
- *Counting*: Which number is next after X? (Where X is a number between 1 and 99)
- *Addition*: What is $X + Y$? (Where the answer is positive and less than 100)
- *Subtraction*: What is $X - Y$? (Where the answer is a positive number and both X and Y are less than 100)

Level 3:

- *Multiplication*: What is $X \times Y$? (Where X is 2, 3, 4, 5, or 10 and Y is a number between 0 and 10 inclusive)

Level 4:

- *Multiplication*: What is $X \times Y$? (Where both X and Y is a number between 0 and 10 inclusive)
- *Division*: What is X/Y ? (Where X is a whole number and Y is equal to 10 or 100)
- *Addition*: What is $X + Y$? (Where X and Y are numbers to two decimal places)
- *Subtraction*: What is $X - Y$? (Where X and Y are numbers to two decimal places)

Level 5:

- *Multiplication*: What is $X \times Y$? (Where both X and Y are numbers to two decimal places)
- *Division*: What is X/Y ? (Where both X and Y are numbers to two decimal places)
- *Addition*: What is $X + Y$? (Where X and/or Y is a negative number)
- *Subtraction*: What is $X - Y$? (Where X and/or Y is a negative number)

3.3. Website Usability Guidelines

To determine the requirements of the website interface of the system I have researched guidelines for website usability and will adhere to these guidelines in the design of the interface. I have chosen to use the Usability.gov guidelines as the target audience are the teachers and government employees (Guidelines.usability.gov, 2014). The following set of guidelines is the government standard for optimising the user experience with websites:

1. Do not display unsolicited windows or graphics
Unnecessary content and pop-ups will be avoided in the design.
2. Increase website credibility
The website will be arranged logically and kept up to date. In addition citations and references will be included where necessary.
3. Standardise task sequences
Repeated behaviour will have the same sequence and manor.
4. Reduce users workload
Computer functions will be utilised to limit the tasks the user has to do.
5. Design for working memory limitations
User will not have to remember large amounts of text or content from page to page.
6. Minimize page downloading time
Each page will have a well-defined purpose and I will minimise content on each page to increase the speed of page loading.
7. Warn of timeouts
If a page requires an action be completed within a specified time the user will be informed of this and have the ability to extend the time.
8. Display information in a directly usable format

I will ensure that the format is consistent and that the user does not need to convert values to use them.

9. Format information for reading and printing

The layout of content should ensure that it is easily readable on screen and following printing and therefore I will ensure that the width of content is reasonable so that it is readable on a variety of screen widths without the need to scroll left and right.

10. Provide feedback when users must wait

If the user is required to wait for a process to complete or the result of an action they will be informed.

11. Inform users of long downloading times

If the user is required to wait for a page to download they will be informed.

12. Develop pages that will print properly

I will ensure the width of a page is suitable for portrait and landscape printing.

13. Do not require users to multitask while reading

Users should not need to read large amounts of text whilst performing tasks.

14. Use users terminology in help documentation

Help documentation should not include terms that the user is not familiar with.

15. Provide assistance to users

I will ensure instructions are available for new users to ensure they are able to complete all tasks.

3.4. Existing system analysis - Incerts

To further understand the requirements for the website side of my system I have evaluated the current Incerts software that the school currently uses to monitor assessment levels. I was able to view and interact with a demonstration of the Incerts software through their online demo (Incerts.org, 2013).

I have identified the following task that the 'Incerts' application users can do:

Task for the teacher:

1. View student levels of entire class

A teacher can view a summary of assessment data for all students from one class.

2. Update student assessment data

The teacher uploads data to the system about assessments they have conducted. These can be updated when necessary to ensure that a student's record accurately represents their achievements to date.

3. Analyse student assessment data

A teacher can analyse the data stored for a students and can use this data to determine how they have progressed within a variety of subjects.

Task Analysis

Task 1: View student levels of entire class

Step 1: Login to teacher account

The teacher is added to the Incerts system by their school head teacher which results in them having their own account with a username and password. Using these details they can log in to the system by entering their details in the Login page (Figure 1). The account details that the teacher enters determine what data within the system that they can view.

1 e

On entering the correct login credentials the teacher is redirected to the school view of their account (Figure 2). Within this view they are able to view a list of all the other teachers registered to the schools Incerts system along with a list of all the classes for the school. Incerts also offers the teacher the option to export the schools assessment data to Microsoft excel where they can choose to view the levels in a variety of formats.

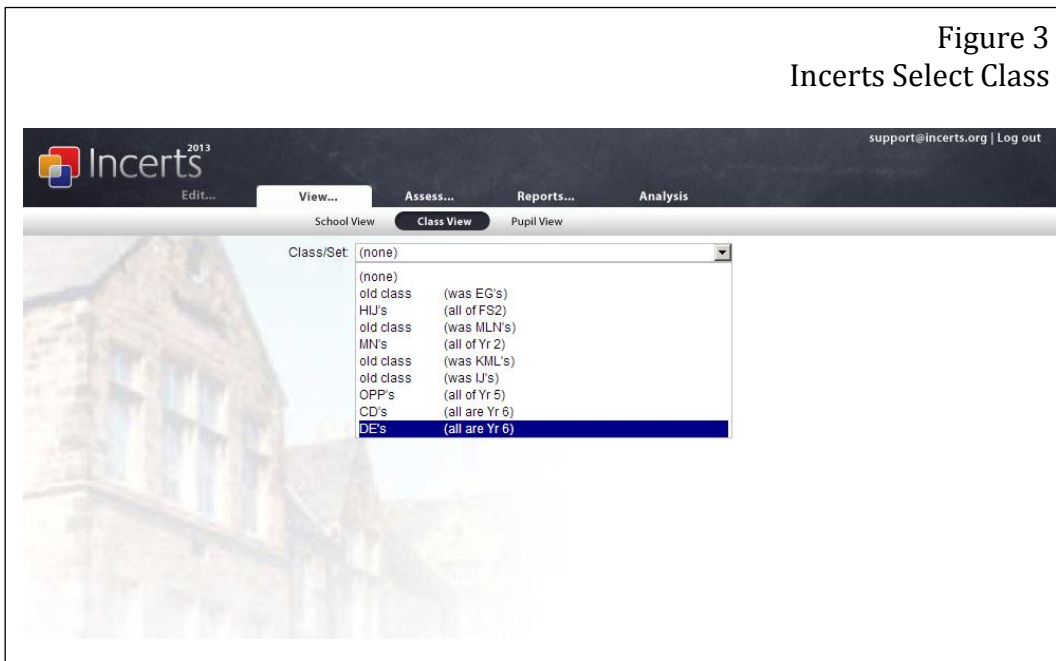
Figure 2
Incerts School View

The screenshot displays the Incerts School View interface. At the top, there is a navigation bar with the Incerts logo and the year 2013. The navigation bar includes links for 'Edit...', 'View...', 'Assess...', 'Reports...', and 'Analysis'. Below the navigation bar, there are tabs for 'School View', 'Class View', and 'Pupil View'. The 'School View' tab is currently selected. The main content area is divided into two sections: 'Active Teachers' and 'Current Classes (to see Groups, [click here](#))'. The 'Active Teachers' section lists teachers with their initials and names, such as AB (Mrs Brown), AC (Miss Carr), CD (Miss Davies), DE (Mrs Evans), EG (Miss Green), HIJ (Mrs Ingleby-Jones), and IJ (Mr Jones). The 'Current Classes' section lists classes with their names and the number of pupils, such as 'old class (was EG's)' with 31 pupils, 'HIJ's (all of FS2)' with 30 pupils, 'old class (was MLN's)' with 31 pupils, 'MN's (all of Yr 2)' with 31 pupils, and 'old class (was KML's)' with 31 pupils. A dropdown menu is open, showing options for 'To output all current' data: 'decimal levels', 'point scores', 'abc sub-levels', 'percentage scores', and 'assessment ratings'. The total pupils are listed as 277.

From this page the teacher can choose from a variety of views, can update assessment results, generate reports or analyse the assessment data using the navigation bar situated at the top of the page. The user that is currently log in is displayed at the top right hand of the page and they can choose to log out at any time using the 'Log out' link to the right of their name.

Step 2: Select 'Class View' option and select the class the teacher wishes to view
Upon selecting the 'Class view' option in the navigation bar, the teacher is asked to select the class they wish to view from the drop down list of the page (Figure 3). The list contains all of the schools classes including those that the teacher does not teach.

Figure 3
Incerts Select Class



Once the teacher has selected a class from the drop down list the assessment data for that class is displayed (Figure 4). The data includes the name of the class teacher, each student in the class along with the level and sub-level they have achieved to date for each of the subjects. The teacher can also choose to hide the names from the display.

Figure 4
Incerts View Class

Class/Set: DE's (all are Yr 6) Taught by: Mrs Evans

These are abc-sublevels. To see point scores, [click here](#)

(Show/Hide Names)

	E	M	S	I		E	M	S	I
Ben Ainslie	--	--	--	--	Ana Edwards	4b	4b	4b	4b
Arlene Booker	2a	2a	2a	2a	Anna Mills	4b	4b	4b	4b
Alyce Koch	2a	2a	2a	2a	Angelica Schwartz	4b	4b	4b	4b
Amelia Smith	2a	2a	2a	2a	Anita Beam	4b	4b	4b	4b
Ariel Atkinson	4c	4b	4b	4b	Anna Booker	4b	4b	4b	4b
Alma Guerrero	4b	4b	4b	4b	Antwan Morse	4b	4b	4b	4b
Allen Ramsey	4b	4b	4b	4b	Archie McKenzie	4b	4b	4b	4b
Arlene Everett	4b	4b	4b	4b	Anita Valentine	4b	4b	4b	4b
Anna Smith	4b	4b	4b	4b	Angelina Walters	4b	4b	4b	4b
Anthony Lara	4b	4b	4b	4b	Anastasia Bowers	4b	4b	4b	4b
Anne Fry	4b	4b	4b	4b	Antoinette Burgess	4b	4b	4b	4b
Arlene Gallegos	4b	4b	4b	4b	Arlie Yates	4b	4b	4b	4b
Amado Long	4b	4b	4b	4b	Andrew Stout	5c	5c	5c	5c
Angeline Luna	4b	4b	4b	4b	Angelia Loka	5c	5c	5c	5c
Alissa Davis	4b	4b	4b	4b	Angela Briggs	5c	5c	5c	5c

To output these abc-sublevels to Microsoft Excel, [click here](#)

The data is presented within a table with the table heading displaying a letter for each subject. When the user hover's their mouse over this letter the full subject name is displayed. The level is the number that is displayed first for each record in the subject column and the sub-level is the letter following this number. There are 3 sub levels for each level, a, b, and c. The meaning for each sub-level is defined on the page where the teacher enters the result. In addition where the level is lower than where the student should be achieving the font is coloured red and where the level is higher than it should currently be the font is coloured green.

Task 2: Analyse student assessment data

Step 1: Login to teacher account

Repeat step 1 of task 1 to login.

Step 2: Select the pupil they wish to view

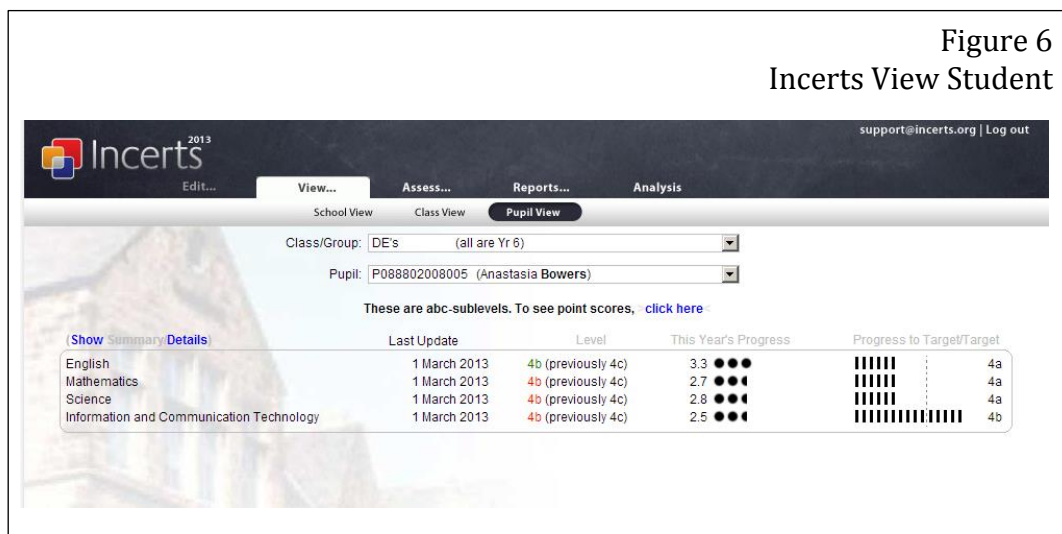
Having selected the 'Pupil View' from the navigation bar under the 'View...' option, the teacher is presented with a selection of drop down list to select the class of the student they wish to view and the student name (Figure 5).

Figure 5
Incerts Select Student

Progress	Progress to Target/Target
	4b
	4b
	4b
	4b

This results in the most recent assessment levels for that student in the core subject's English, Mathematics, Science and Information and Communication Technology to be displayed in a table. As before where the student's level is above what is expected at this stage the font will be green and where they are under achieving font will be red. This view also displays the most recent previous assessment result allowing the teacher to view a comparison between the two. The teacher can also see the date of the last update to these results along with how much the student has progressed in the current year and how much further progress they are required to make to meet the next target.

Figure 6
Incerts View Student



Task 3: View student levels of entire class

Step 1: Login to teacher account

Repeat step 1 of task 1 to login.

Step 2: Select the student whose results the teacher wants to update

To add assessment data to the records the teacher selects the 'Assess...' option in the navigation bar. They then choose the class of the student whose record they wish to update along with the student name and the subject of which the assessment result is for.

This displays the levels for the subject along with sub-levels. The levels are displayed in a table with the table headers being a letter representing the topics covered by each subject. When the user hovers over the letter details about the level descriptors for that topic are displayed (Figure 7). Each paragraph represents a level and so the teacher compares the results of their assessment with the descriptions in order to determine whether the student has achieved that level.

Figure 7
Incerts Subject Topics

2013 Incerts support@incerts.org | Log out

Edit... View... Assess... Reports... Analysis

Assess EYFS Assess NC Assess APP

Class/Group: DE's class

Pupil: P088802008005 (Anastasia Bowers)

Subject: English: Speaking and Listening

Column A

She talks about matters of immediate interest.

She is beginning to show confidence in talking and listening, particularly where the topics interest her.

She talks and listens confidently in different contexts, exploring and communicating ideas.

She talks and listens with confidence in an increasing range of contexts.

She talks and listens confidently in a wide range of contexts, including some that are of a formal nature.

(no statement in this column at this level)

Level 4

Level 5

Level 6

Save and Confirm ...and select next pupil

To see the complete key, [click here](#)

To output this page, with all the statements visible, [click here](#)

Step 3: Update student data

Each level has a selection of tick-boxes which correspond to the sub-levels under each subject topic (Figure 8).

Figure 8
Incerts Insert Assessment Data

2013 Incerts support@incerts.org | Log out

Edit... View... Assess... Reports... Analysis

Assess EYFS Assess NC Assess APP

Class/Group: DE's class

Pupil: P088802008005 (Anastasia Bowers)

Subject: English: Speaking and Listening

Level 1

Level 2

Level 3

Level 4

Level 5

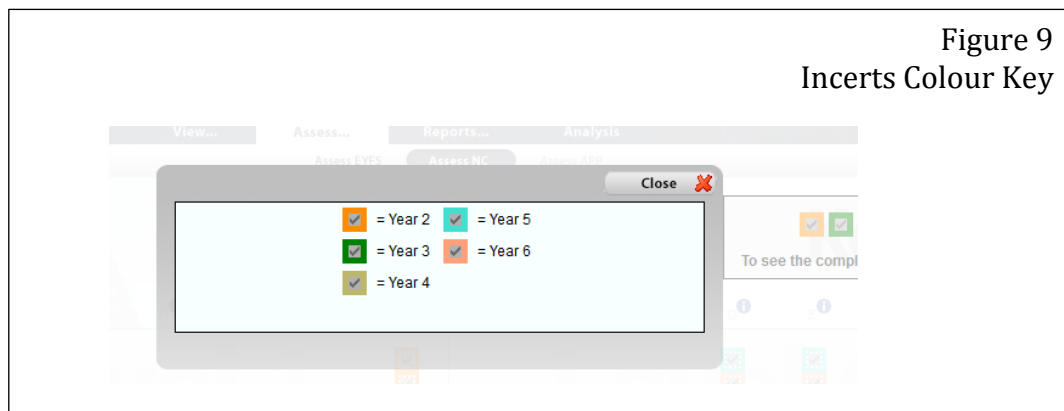
Tick this box when...
She adapts her talk to the demands of different contexts with increasing confidence.
Notes[Add] Guidance[Add]

Save and Confirm ...and select next pupil

To output this page, with all the statements visible, [click here](#)

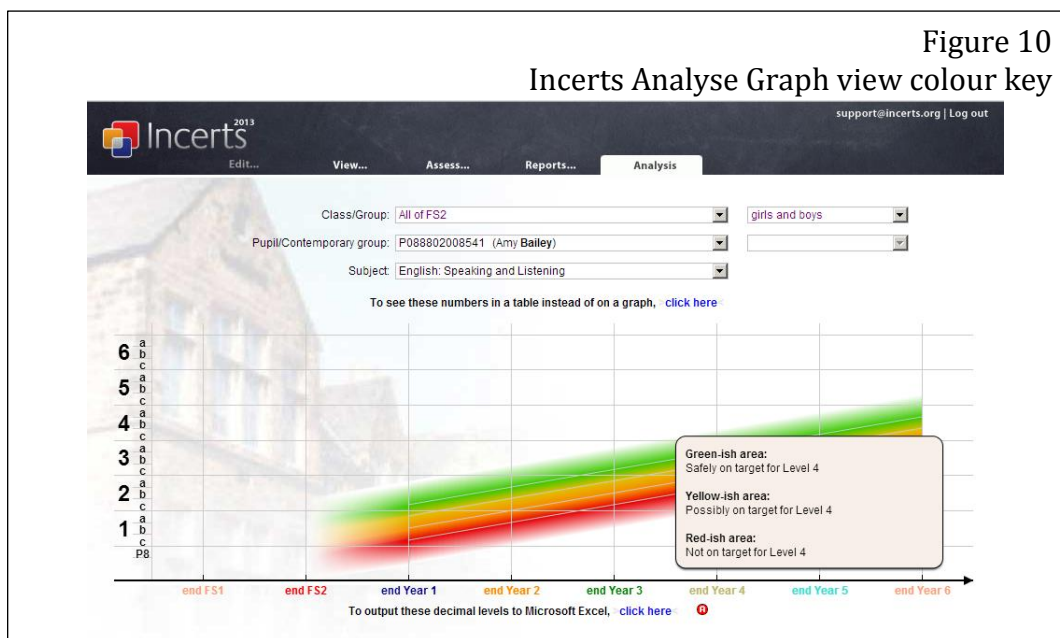
Using the level descriptors as guidance the teacher selects all the boxes that the student has achieved. Hovering over a box displays a description about what the box corresponds to. When the teacher has finished updating the students record they select the 'Save and Confirm' button which will update the record.

The colours are used to represent the year that the student achieved the level. The key for the colours can be seen by selecting the "To see the complete Key. Click here" link (Figure 9).



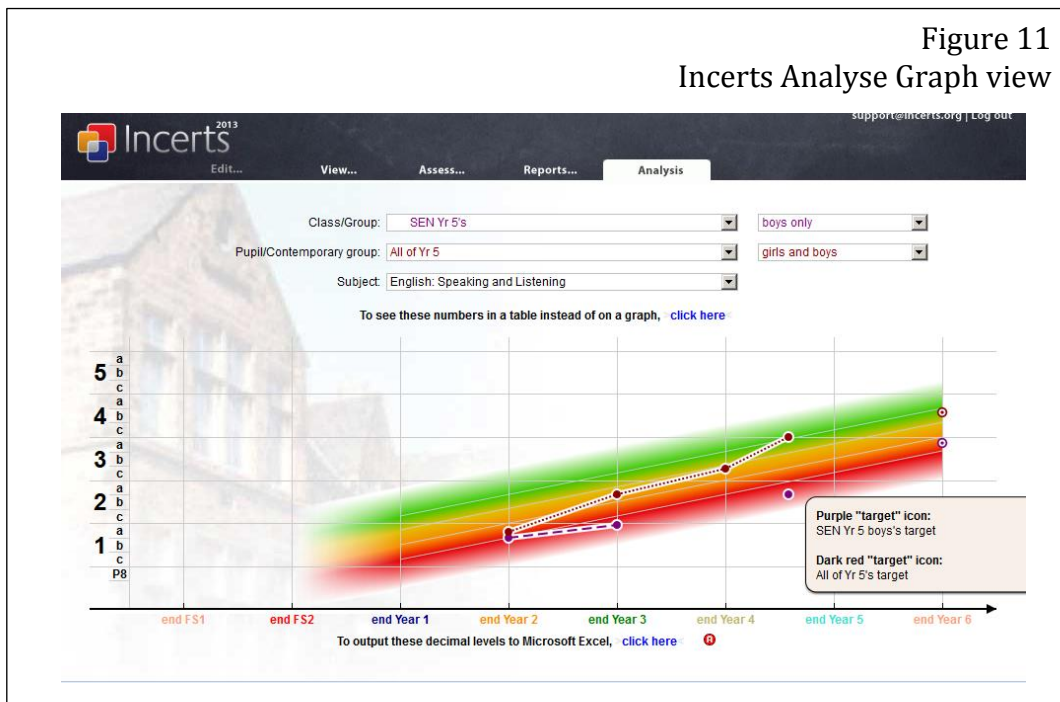
The teacher can also analyse students assessment records by selecting the 'Analyse...' option from the navigation options. This view enables the teacher to view what the student is achieving and compare this with where they should be achieving. The teacher selects the student or class they wish to view and the subject that they wish to view. These can be viewed as a table or as a graph.

In the graph view colours are used to represent whether the level is on track, ahead or behind (Figure 10).



Levels are plotted on the graph using a solid point to denote the achieved level and a circle with a centred dot to represent the target (Figure 11). The teacher can choose to view only boys, only girls, summer born or all children of a class.

Figure 11
Incerts Analyse Graph view



In table view the teacher can view the target level data for each year in a table (figure 12).

Figure 12
Incerts Analyse Table view

Class/Group: All of FS2 | girls and boys | girls and boys

Pupil/Contemporary group: HLJ's class | girls and boys

Subject: English: Speaking and Listening

To see these numbers on a graph instead of in a table, [click here](#)

To output these decimal levels to Microsoft Excel, [click here](#)

	end of FS1	current (FS2)	end of FS2	end of Year 1	end of Year 2	end of Year 3	end of Year 4	end of Year 5	end of Year 6
All of FS2									
national targets			0.34 / P4	1.34 / 1b	2.34 / 2b	2.84 / 2a	3.34 / 3b	3.84 / 3a	4.34 / 4b
HLJ's class									
national targets			0.34 / P4	1.34 / 1b	2.34 / 2b	2.84 / 2a	3.34 / 3b	3.84 / 3a	4.34 / 4b

3.5. Outcome of 'Incerts' Evaluation

After experimenting with the Incerts software I discussed its uses with the primary school teacher in order to understand what the main advantages and disadvantages of the system were from a teacher's perspective. Having spoken to Ms Jopson the main advantage was that all students' data is in one place which makes it very accessible and easy to find. In addition using 'Class view' the teacher can clearly identify the students who are under achieving. One of the main disadvantages was that the teacher couldn't always guarantee that the data was accurate as there was no guarantee that it is up to date and so comparing with other classes can be a problem. Ms Jopson also explained how updating every students record can be extremely time consuming due to the vast amount of subjects and topics within that subject that need to be assessed.

From my analysis of the Incerts application I have identified functional aspects that I will also be ensuring my website can complete. Having evaluated the Incerts application using the government design principles I was able to determine several bad usability aspects which I will avoid in my implementation as well as good usability aspects which I will utilise.

From a functional perspective I will intend to ensure teachers can view the levels that students have achieved. This will require the teacher to be able to add students to the system. This will also require that there be access restrictions to the teachers account so that the students cannot see each other's levels. Response time on selecting different students and classes was very quick which enables the user to efficiently view records. In order to eliminate the need for teachers to update records manually I will ensure that the application I develop will automatically update the students achieved levels.

From a usability perspective the use of the drop down lists and consistent navigation options works really well in this application, and enables the user to familiarise themselves with where they can find things. The use of colours to describe the data was also very effective enabling the teacher to understand the level without having to compare it with the targets. One poor usability aspect I found was the use of single letters to represent topics. I did not understand which subjects the letters represented before speaking to Ms Jopson, and also the light grey font used on the headings of the "Assess..." page made it difficult to see. I will therefore avoid using unexplained abbreviations and use dark colours for the font.

3.6. Application Usability Guidelines

In order to establish the requirements for the user interface of the application I have researched the Apple developer usability guideline's which are the set of guidelines Apple recommends developers adhere to in order to ensure the application is easy and efficient to use. In order to ensure the application I develop will be easily usable

by the target audience I intend to develop an interface for the application that meets the following Apple design principles (Developer.apple.com, 2014)

- **Aesthetic Integrity** – Appearance and behaviour of application should suit the task intended. To ensure that the interface adheres to this I will ensure question layouts are similar to that of the current tests used to assess student's levels. In addition I will avoid any distracting content where a student is required to concentrate.
- **Consistency** – I will ensure that the application has uniform terminology and styling throughout to enable users to predict the behaviour of unseen navigation tools where the appearance is similar to one they have used before.
- **Direct Manipulation** – Application should make use of the touchscreen to engage with objects to help users understand the result of actions and simplify navigation. I will make use of the touchscreen throughout the interface.
- **Feedback** – Application should acknowledge user actions and be clear on system status. I will ensure all screens are clearly identifiable with the use of labels and ensure that the student knows the remaining questions they have to answer. In addition, I will ensure that the user receives feedback where necessary for example with an incorrect login.

In addition to the above general usability guidelines I will also incorporate the following principles identified by Chiasson and Gutwin which are designed for child users (Chiasson and Gutwin, 2014):

- **Appropriate Literacy** – Child user often do not read content when it is displayed in large sections of text. As a result the interface should consist of mostly visual content such as images and avoid excessive amounts of text. This will ensure that steps to complete tasks are memorable and easy to understand.
- **Feedback & guidance** – A child user is more impatient than an adult user and will therefore require instant feedback. In addition icons should be meaningful to ensure the user understands its purpose.
- **Mental development** – A child user's lack of experience with using computer devices means they may not understand abstract concepts as well as an adult user. As a result of this they may often use a trial and error technique to complete a task. To compensate for this the interface should avoid the use of overpopulated menus and submenus and stick to a small number of interaction styles.

- Motor skills – A child will find it more difficult to manipulate complex controls than an adult user. As a result the user interface should make use of one click actions rather than actions which requires the user to click and hold/drag. In addition inaccuracy of targeting is more likely with a child and so buttons should be larger and more spaced out.
- Motivation & encouragement – To ensure that a child user will continue to use the application it should adopt supportive reward structures and the interface should appear interesting and appealing to keep the user engaged.

3.7. Existing System Analysis – Splash Maths

I have identified an application that is currently being used by students to aid learning in order to identify features that improve the usability of the interface. A mathematical application 'Splash Maths' developed by 'StudyPad inc.' is an iPad application which allows students to practice mathematical questions. The developers have a range of Splash Maths applications aimed at students in years 1 to 5 and I have evaluated the year 5 version which has similar features to the application I will be developing.

The intended audience of the 'Splash Maths' application is parents and their children with the intention of aiding the children's learning of mathematics and enabling the parent to monitor this progress over a period of time. I have identified the following task that the 'Splash Maths' application users can do:

Task for the child:

1. Practise Mathematics.
The child can choose from a range of mathematical subjects and questions are generated for that subject. For each practise session the user is given 20 questions to answer and for each correctly answered question they receive a UFO.
2. Play a game.
The child can use the UFO rewards they received through practising mathematical subjects to play games within the application. Each game will cost the user some of their UFO's and if they do not have enough to play the game they can earn more by answering more mathematical questions.
3. Complete homework.
Users can complete homework which has been set by a parent user. Homework questions presented the same as practise questions with 20 questions to answer. When a user has completed this homework the result is stored and accessible through the parental view.

Tasks for the parent:

1. View child's homework results
The parent can view the result from the child's homework in both graph representation and statistics. They can view details about which mathematical subjects have been completed and a comparison of how well they are performing across the subjects.
2. Set Homework
Parents can choose a mathematical subject and assign a system generated worksheet for this subject as homework for the child. In addition they can view what assignments have already been set.

Task Analysis

Child Task 1: Practise Mathematics

Step 1: Select the correct profile.

The user is presented with the home screen on launch of the application (Figure 1).



The navigation has been slightly over complicated by the use of unfamiliar images used to represent buttons in the lower corners of the main menu. As a result I struggled to understand what the buttons did before selecting them. The icon on the lower right corner opens a link to Apples App Store which was not clearly represented by the icon. To access the profile menu the user must select the icon in the lower right corner of the screen.

In the profile menu the user selects their profile by selecting the image of the character with their name below it (Figure 2). Users can also create a new profile using this menu. The user can also customise their profile by choosing a character to represent them, change their display name and select preferences for sound options. In addition the child user can delete their account, via the lower left button on the profile menu, and all data for this user is deleted.

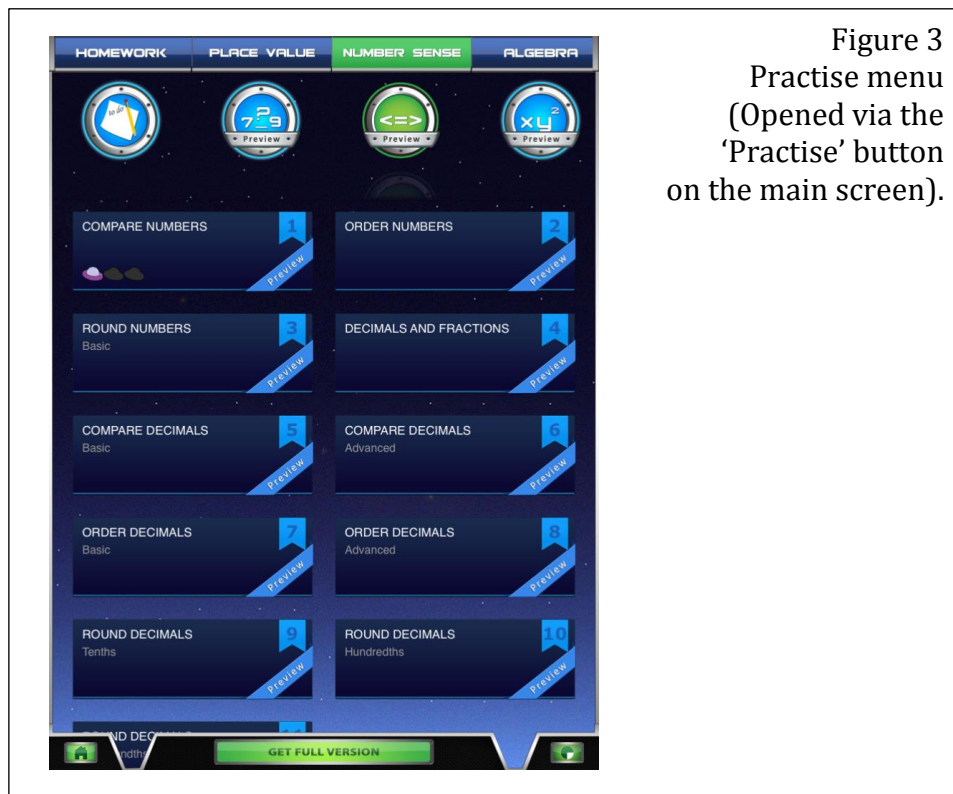


Step 2: Select the 'practise' option from the main menu.

After selecting their profile the profile menu closes and the user is presented with the main menu (Figure 1). From this window the user selects the 'Practise' button which opens the subjects menu.

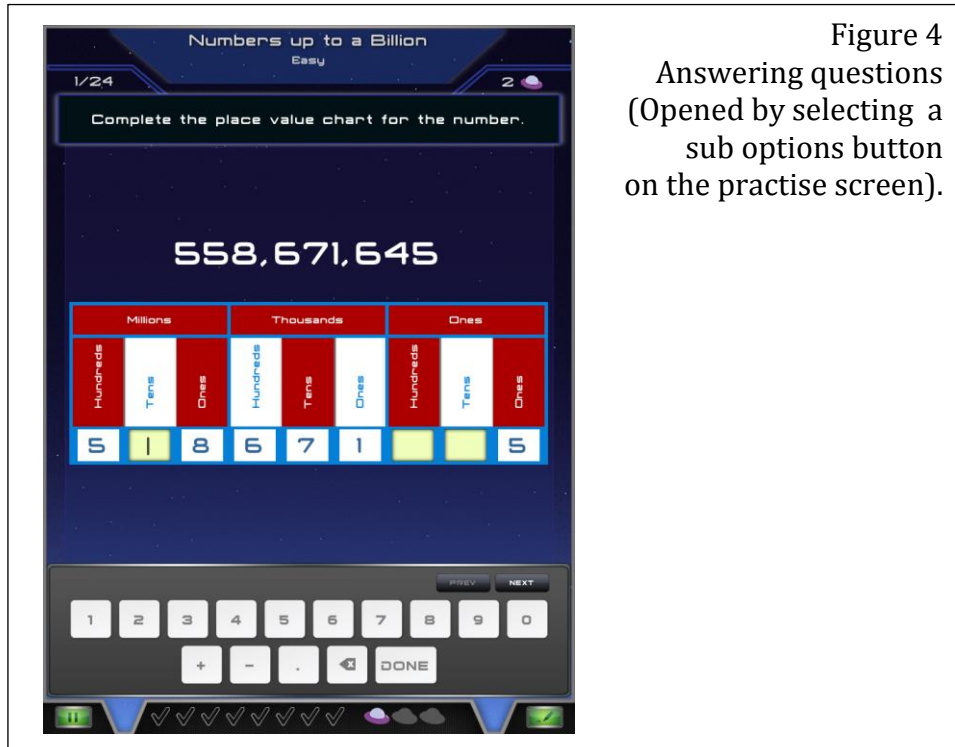
Step 3: Select the subject to practise

The user will be presented with a list of different mathematical questions they can complete (Figure 3). Within the practise section of the application there is an overwhelming amount of buttons for the variety of mathematical questions types. Due to this the buttons do not fit the screen size and the user is required to scroll top to bottom and left to right of the screen in order to see all of the options available to them There are 10 options in total in the horizontal list with each of those options having a sub list of between 3 and 12 options.



Step 4: Answer questions

On selecting one of the mathematical subjects from the lists and one of their subcategories the child is presented with a set of 20 questions which they answer one at a time by using the keyboard to input the answer or selecting from a predefined list of answers (Figure 4).



A useful feature implemented by 'Splash Maths' is the scratchpad which allows users to draw on an area of the screen to make calculations in order to work out the answer (Figure 5). This scratchpad can be accessed by dragging from the bottom of the screen to the centre whilst a question is displayed.

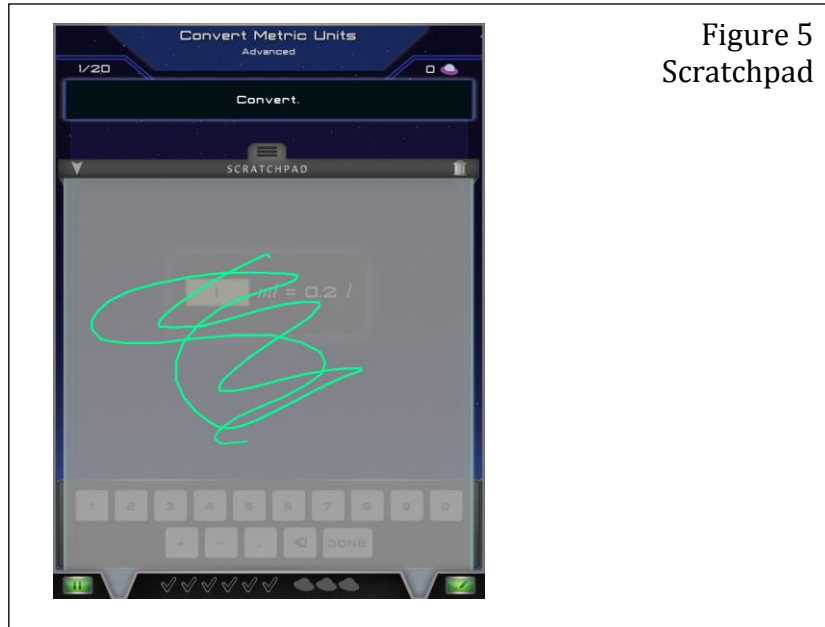


Figure 5
Scratchpad

Step 5: Check answer

On inputting an answer to a question the user is then presented with feedback on whether they answered the question correctly or incorrectly (Figure 6 and 7). In addition if the user gets the question wrong they are informed of what the correct answer was (Figure 6).

The user repeats step 4 and 5 until they have completed all questions or the user can exit the practise questions at any time by selecting the bottom left hand icon of the question screen. The new view contains two options; to quit the practise questions or to resume with the questions they were doing (Figure 8). If the user chooses the quit option they are taken back to the main menu.

Step 6: End practise session

Having completed all questions the user is redirected back to the main menu (Figure 1). No feedback is given on how they performed.

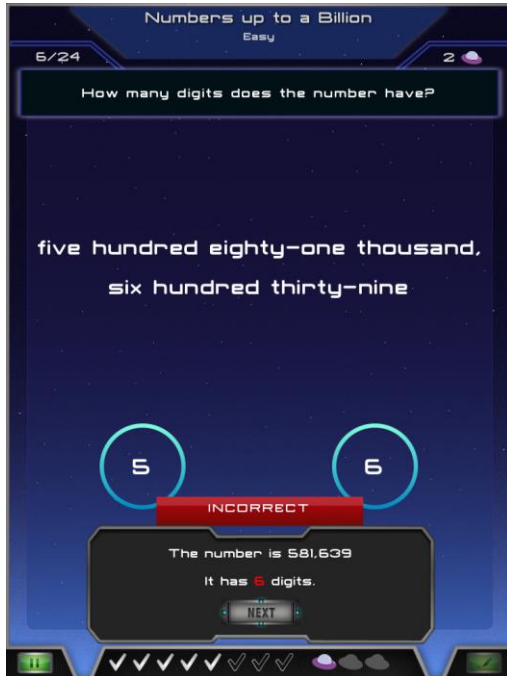


Figure 6
Incorrect answer Feedback

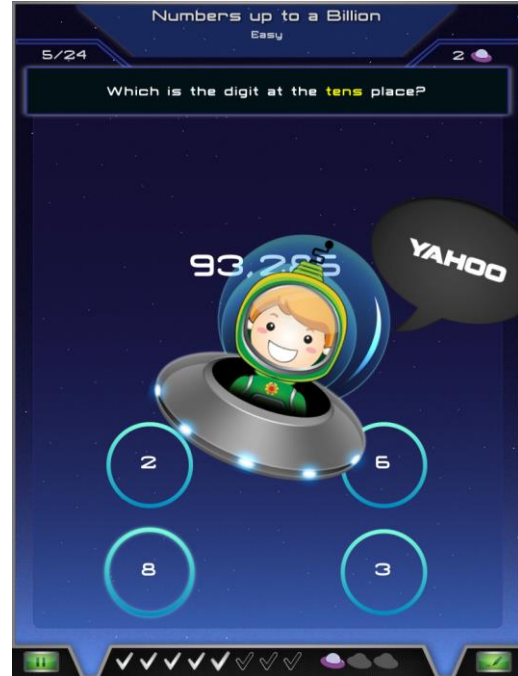


Figure 7
Correct answer feedback

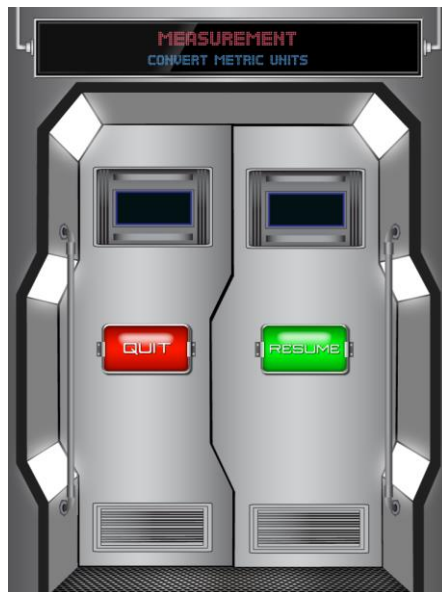


Figure 8
Exit practise questions

Child Task 2: Play games

The user selects the 'Games' option from the main menu (Figure 1), which will take them to a menu of a list of games they can play. Users are required to win spaceship tokens through completing practise questions correctly in order to play the game which provides a good incentive for the users to practice mathematics rather than go straight to the games.

On selecting the game they wish to play from the menu the corresponding game will begin.

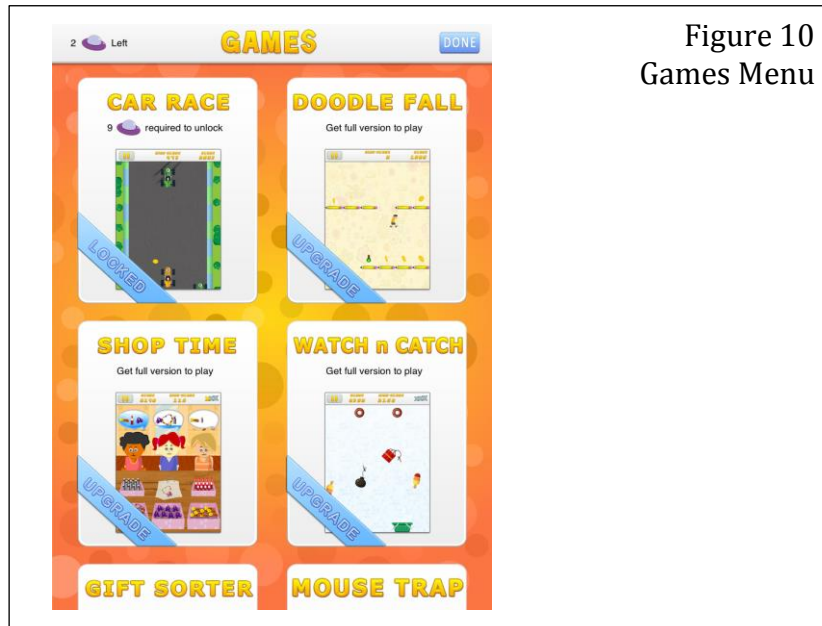


Figure 10
Games Menu

Child Task 3: Complete homework

The child selects 'Practise' from the main menu (Figure 1). When the mathematics subject menu loads the user selects the 'Homework' tab from the navigation options (Figure 2).

If no homework is set the user is presented with the message "No homework for me!" on the menu (Figure 11). If homework has been set the user selects the homework they wish to complete from the menu (Figure 12). This then launches the selected homework in the same way as the practise questions.

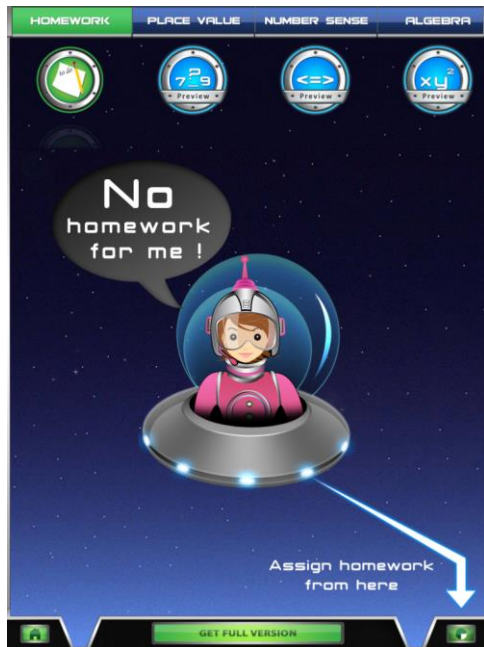


Figure 11
No homework set

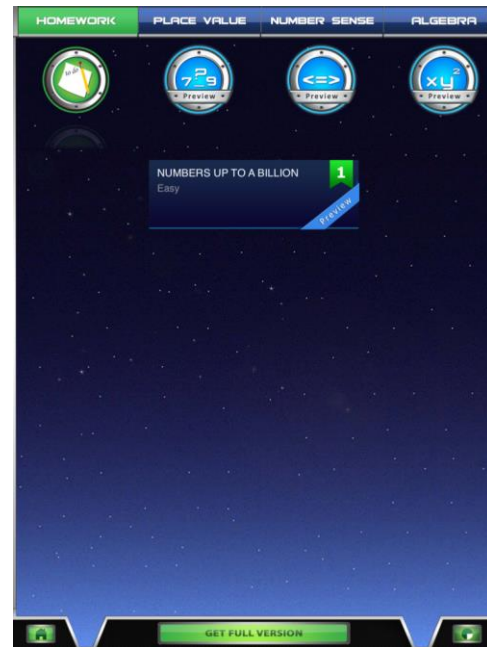


Figure 12
Homework set

Parent Task 1: View Homework result

The application has built the parental view into the same application as the child's and as a result the interface colour scheme and images are targeted at children. To access the parents view the user is required to select the 'Parents' button from the main menu (Figure 1).

On doing this the user is presented with an access restriction in the form of number input verification where the user is required to enter the numbers which are written on the screen as text (Figure 14). At year 5, which is the target audience, children can read and therefore access the parental view enabling them to view other user's scores and assign and remove homework for other users. The parental view has been developed for parents and the restrictions in place to lock this view from the children will not limit access to children who can read numbers written in text.

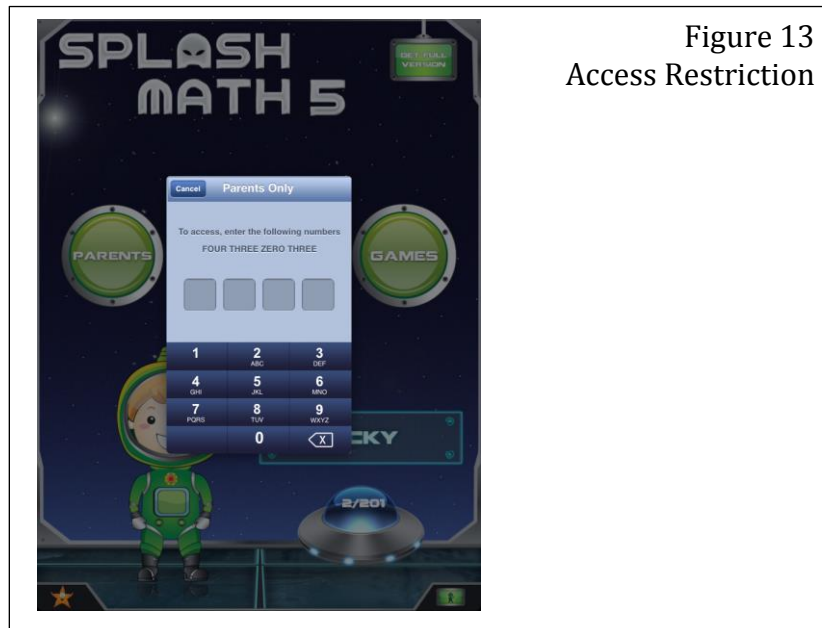
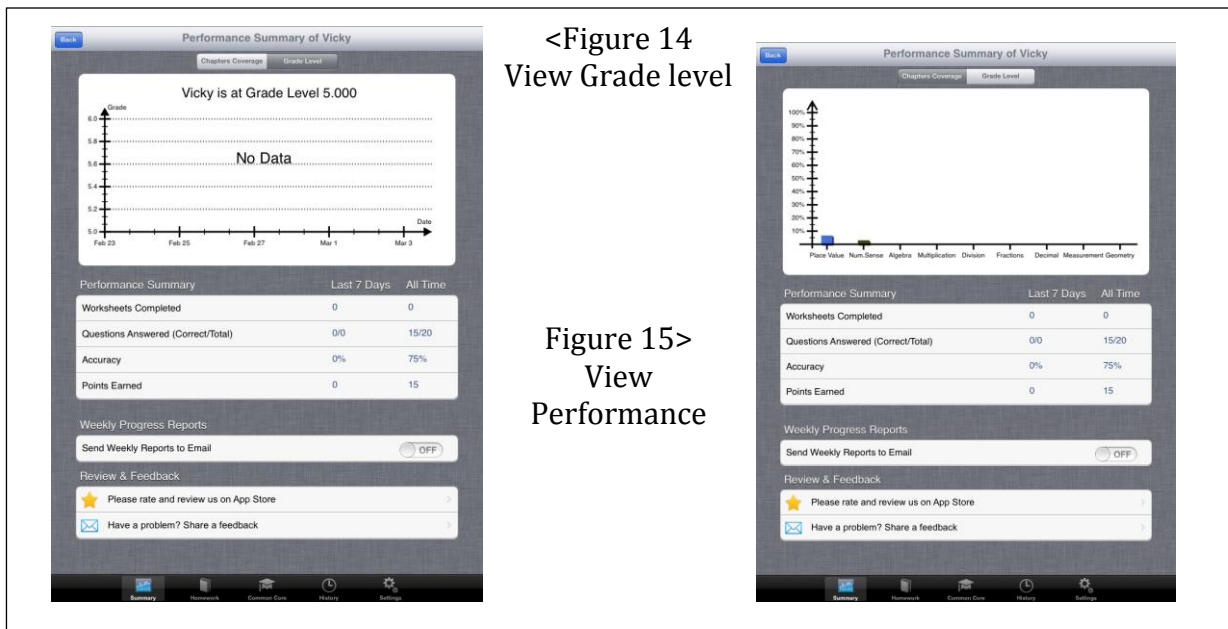


Figure 13
Access Restriction

Once the user has entered the correct details they will be presented with the summary of chapters that have been completed by the user whose profile is currently active (Figure 14). Users can also view details of the child's grade level by selecting the tab 'Grade Level' at the top of the screen (Figure 15). Results are displayed in the form of a graph which is effective at simply displaying the information to the user. In addition, further details on how many worksheets have been completed by the current profile are displayed along with the accuracy, and number of questions answered in total.



Parent Task 2: Set homework

The parent can assign homework for the child by selecting the 'Homework' icon in the bottom navigation bar. The assignment will be made for the child whose profile is active at the time that the parent uses the assignment feature. The parent selects the subject from the list on the left which activates a list on the right of homework options for that subject (figure 14). To set this homework the parent selects the 'Assign' button which is located in the same box as the homework they wish to set.

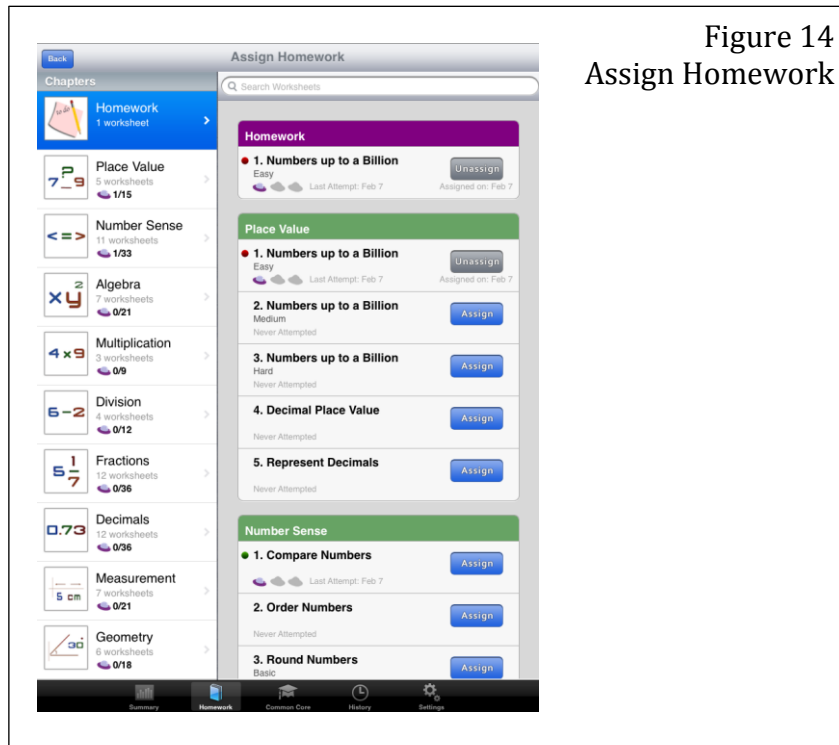


Figure 14
Assign Homework

3.8. Outcome of 'Splash Maths' Evaluation

From my analysis of the 'Splash Maths' application I have identified functional aspects that I will also be ensuring my application can complete:

- Access Restrictions
The application I develop will require further access restrictions than the number verification that 'splash math' uses such as the use of passwords to ensure that the records can only be seen by the authorised persons. In addition, in the 'Splash math' application a child can delete their account with no parental authorisation and given that a child may not understand the implications of this and that everything they have done so far is removed I will restrict the child from being able to do this.
- Teacher can view all students data concurrently

I will also add the extra functionality of the teacher being able to view students' data without them having to log in to each account so that the teacher can compare assessment results across students and allocate assessments without needing every student to log in first.

- **Response time**
Response time on selecting an option was very quick which enables the user to navigate efficiently through the options. Also users can complete questions and get feedback for these questions instantly. I will ensure response time in my application is quick enough to also meet this requirement.

Having evaluated the splash maths application using the design principles produced by Apple I was able to determine several bad usability aspects which I will avoid in my implementation as well as good usability aspects which I will utilise. From a usability perspective for the system I develop I will ensure:

- **Grouping of related items**
The 'Splash Math' overwhelming choice of menu options has highlighted the importance of limiting the menu options by grouping related items and ensuring that the interface is fixed to the screen size to simplify navigation. In addition similarly to the splash maths application I will utilise buttons and tabs in the menus to simplify navigation.
- **Maintain consistency of design**
The colour scheme fonts and layouts are consistent within the three main sections of the application however the different sections of the application have an entirely different appearance, such as different buttons, fonts and colour schemes, which makes them look like entirely different applications and therefore doesn't maintain consistency across the interface. I will separate the student view from the teachers view to ensure that the interface is suitable for each user and that consistency can be maintained within the application and within the website.
- **Design which attracts a young user**
The 'Splash Math' application has colourful menus and backgrounds along with minimal text to capture the attention of a young user which I will also incorporate in my design.

4. Requirement Specification

Following the research, discussion with intended users and evaluations of similar applications I have developed the following set of requirements for the system I will create.

Functional Requirements

1. Teachers should be able to set up a new teacher account.
This is required to facilitate access restrictions and to enable each teacher to have a unique set of student.
2. Teacher must be able to add and remove students.
In order for the teacher to be able to set assessments for their students they will need to be added to the system. Student will also need to be added to the system to use the application. The teacher may also wish to remove students when they move school or to their next class.
3. Users will need to log-in so that the system can identify the current user.
To ensure that a teacher can access data about their students they will need to log in to the website to differentiate themselves from students and other teachers. In addition, students will be required to log in to the application to enable the system to derive the correct level of practise questions or assessments for them.
4. The student should have the ability to practise mathematic questions.
A student must be able to complete a set of practise questions at the appropriate level for that student.
5. The application should generate practise questions and answers for the user.
The level of the questions generated should correspond to the level a student is achieving currently incremented by one for practise questions. In addition the answers to these questions must be calculated to facilitate feedback to the user and to enable the system to calculate the users score at the end.
6. Students should be able to do an assessment.
If an assessment has been set for a student they should be able to complete this assessment. The assessment must be completed between the dates it was set and with questions difficulty conforming to the level set.
7. The application should generate questions and answers for the assessment.

The level of the questions generated for the assessment should correspond to the level set by the teacher. In addition the score must be calculated to enable the system to determine whether the user has achieved the assessment level.

8. Students should be able to see if they have answered a question correctly or incorrectly.
To enable students to learn from the questions they are practising it is important that they understand whether they are answering them correctly or incorrectly.
9. The application should calculate the level a student has achieved in assessment.
The application should calculate this level based on their score from the assessment. This result should then be stored so that a teacher can access them.
10. Teachers should be able to view student's assessment results individually and as a group.
To enable the teacher to establish how a student's results compares to the rest of the class the teacher should be able to view all results at one time. In addition a teacher should be able to view the result of single student's achievements which will enable them to see more detailed data about a student's assessment.
11. Ensure appropriate content in the assessments and practise questions.
Questions generated by the application should consist of content that is relevant to the programme of study and the level of the student. Also, levels generated from the assessments outcome should comply with the level descriptions.
12. Feedback about performance should be given to students when using the application.
Upon a student answering a practise question the application should state whether the student's answer is correct or incorrect. In addition, the application should display the number of questions remaining in both the assessment questions and practise questions.

Non- functional requirements

1. Ensure security of user data
The system will need to ensure that one student cannot access other student's records. Only the teacher should be able to access student data and account details.

Interface Design requirements

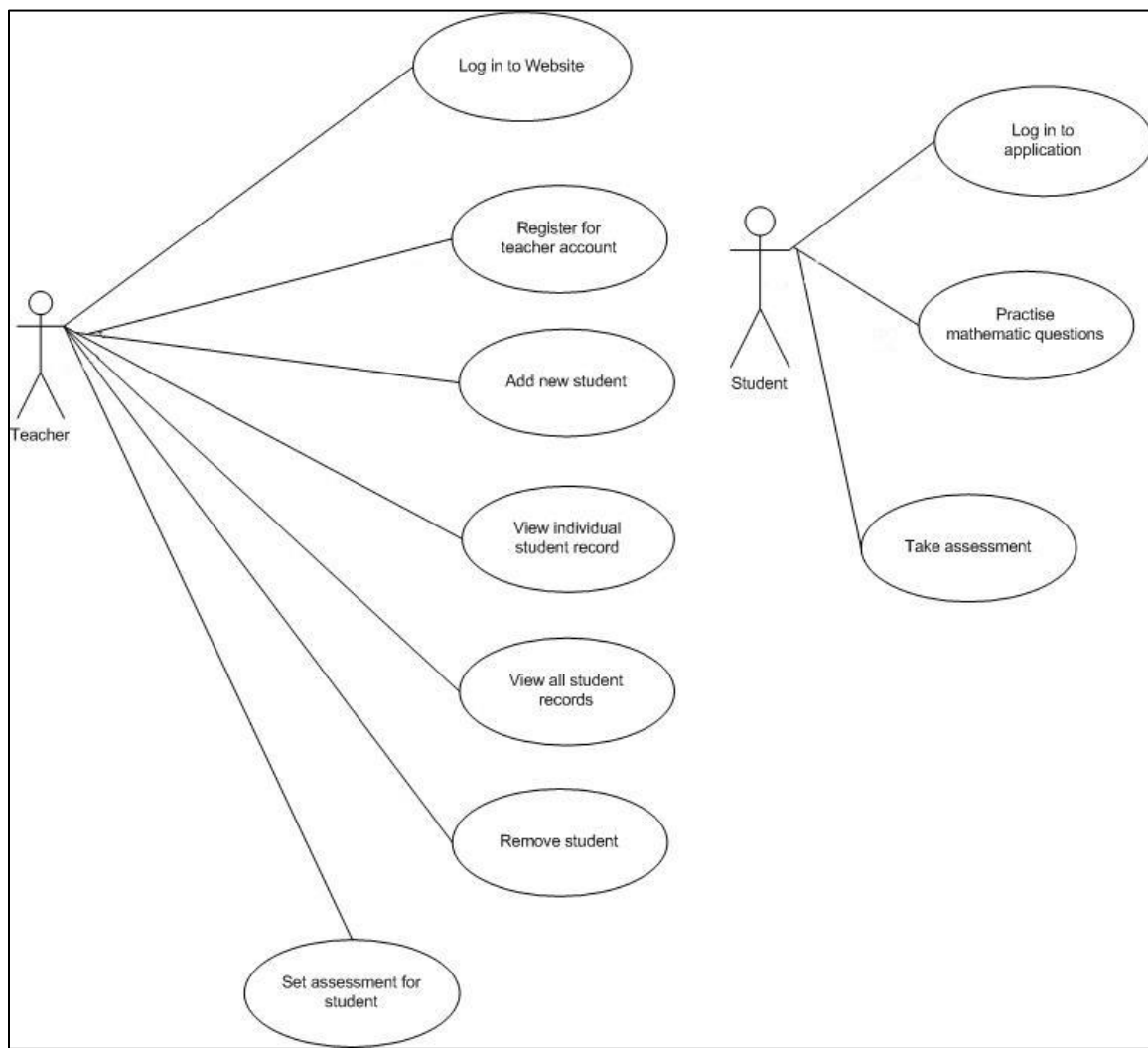
1. A graphical user interface should be implemented for the website which conforms to the website guidelines by Usability.Gov (outlined above in requirements analysis, section 3.3. Website Usability Guidelines).
2. Graphical user interface should be implemented for the application which adheres to the Apple developers design principles (outlined above in requirements analysis, section 3.6. Application Usability Guidelines).
3. The application interface should also adhere to the interface guidelines by Chiasson and Gutwin for children (outlined above in requirements analysis, section 3.6. Application Usability Guidelines).
4. The application should be appealing, interesting and fun to use to ensure the continuous and effective use of the system.

5. Functional Design

5.1. Use Case Diagram & flow of events

The use case diagram models the interactions the users will have with the system. I have identified the following actors which will be interacting with the system:

- Teacher: A teacher is a person with a teacher ID who has the authorisation to add and remove students from the system. This person also has access to view student assessment results.
- Student: A student is a person who is registered by a teacher and has a valid student ID. This person knows their student ID and password.



The flow of events outlines the steps to achieve each use case and identifies the possible alternative actions that may occur in each use case. I have included the flow of events for the main use cases:

- Set Assessment for student
- View individual student record
- View all student records
- Take assessment
- Practice mathematic questions

For flow of events for all other use cases, see Appendix C.

View individual Student record

This use case allows the teacher to view the level an individual student has achieved from the assessments they have completed. They will see the date an assessment was completed and the result for that assessment.

1. Flow of events

1.1 Basic flow of events

1.1.1 Select 'view individual student record'.

The website displays the options available to the teacher. The teacher selects 'view individual student record'.

1.1.2. Enter student ID.

The teacher enters the students ID of the student whose record they wish to view.

1.1.3. View student data.

The teacher selects the 'view record' button. The student ID entered is compared with Student ID records in the database. If the Student ID exists in the database details of student assessments level and assessment date are retrieved from the database and displayed.

1.2 Alternative flow of events

1.2.1 User doesn't exists

If in step 2 of the use case the teacher enters a student ID which doesn't exists in the database an error message will be displayed. The use case ends.

2. Pre-requisites

The teacher will need to complete the 'Log in to website' use case. The student the teacher wishes to view results of must have been added to the database, use case 'add new student', and the account of that student must have not been deleted.

3. Post-condition

Following successful completion of the use case the teacher will be presented with the specified student's assessment results.

View all Student records

This use case allows the teacher to view the highest level each student has achieved from the assessments they have completed.

1. Flow of events

1.1 Basic flow of events

1.1.1. Select 'view all student records'.

The website displays the options available to the teacher. The teacher selects 'view individual student record'.

1.1.2. View student data.

Details of all students' assessments levels are retrieved from the database and displayed.

1.2 Alternative flow of events

1.2.1 Log out

A teacher can log out of the website at any time. The use case ends.

2. Pre-requisites

The teacher will need to complete the 'Log in to website' use case. The student the teacher wishes to view results of must have been added to the database, use case 'add new student', and the account of those students must have not been deleted.

3. Post-condition

Following successful completion of the use case the teacher will be presented with all students' assessment results.

Set Assessment for student

This use case allows the teacher to set a mathematic assessment date for a particular student. The student can then complete the mathematics assessment anytime between these dates.

1. Flow of events

1.1 Basic flow of events

1.1.1. Select 'Set Assessment'.

The application displays the options available to the teacher. The teacher selects 'Set Assessment'.

1.1.2. Enter student ID and assessment date

The teacher enters the student ID of the student they wish to set the assessment date for. They then select the start date for the assessment which is the earliest date the student can do the assessment. The

teacher then enters the end date for the assessment which is the latest date the student can do the assessment. The level of assessment is chosen from a list.

1.1.3. Store assessment date.

The teacher selects the 'set assessment' button. The student ID will be checked against the database. If the student exists the assessment will be stored in the database.

1.2 Alternative flow of events

1.2.1 Log out

A teacher can log out of the website at any time. The use case ends.

1.2.2 Student entered doesn't exist

If in step 2 of the use case the teacher enters a student ID that does not exist in the database an error message will be displayed. The use case returns to step 2.

1.2.3 End-date before start-date

If in step 2 of the use case the teacher enters a 'end-date' which is before the 'start-date' entered an error message will be displayed. Use case returns to step 2.

2. Pre-requisites

The teacher will need to complete the 'Log in to website' use case. The student the teacher wishes to add an assessment date for must have been added to the database, use case 'add new student', and the account of that student must have not been deleted.

3. Post-condition

Following successful completion of the use case the student specified will be able to access the assessment between the dates specified.

Practise mathematic questions

This use case allows the student to practise mathematic questions where the difficulties of the questions are based on their current achieved level.

1. Flow of events

1.1 Basic flow of events

1. Select 'Practise Mathematics'.

The application displays the options available to the student. The student selects 'Practise Mathematics'.

2. Retrieve student level.

The Student ID used to log in is compared to the database and the most recent achievement level obtained is retrieved.

3. Generate questions.

The application generates questions with a difficulty suitable for the level retrieved and presents these questions to the user.

4. Answer questions

The student selects the answer to each mathematical problem from a list.

5. Check Answer

The user selects the 'check answer' button. The application calculates the answer to the question and compares this answer to the student's selection and determines whether they are the same.

6. Display result

The application displays a message to confirm the answer is correct where the input matches the applications answer. The applications displays a message stating the answer is incorrect and displays the correct answer where the student input is different from the answer generated by the application.

1.2 Alternative flow of events

1.2.1 Log out

A student can log out of the website at any time. The use case ends.

1.2.2 No answer entered

If in step 4 of the use case the student does not enter an answer to the question an error message will be displayed. The use case returns to step 4.

2. Pre-requisites

The student will need to have logged in, use case 'Log in to application'.

3. Post-condition

Following successful completion of the use case the student will have practised mathematical questions.

Take Assessment

This use case allows the student to begin a mathematic assessment where the difficulties of the questions are based on the next level they are trying to achieve.

1. Flow of events

1.1 Basic flow of events

1.1.1. Select 'Start Assessment'.

The application displays the options available to the student. The student selects 'Start Assessment'.

1.1.2. Retrieve student level.

The Student ID used to log in is compared to the database and the level set for the assessment is retrieved.

1.1.3. Generate questions.

The application generates 15 questions with a difficulty suitable for the level retrieved and presents these questions to the user.

1.1.4. Answer questions

The first unanswered question is displayed to the student. The student inputs the answer to the current mathematical problem.

1.1.5. Check Answer

The user selects the 'Next Question' button. The application calculates the answer to the question and compares this answer to the student's selection and determines whether they are the same. The application adds 1 to the correct question count if the answer was answered correctly.

1.1.6. Generate next question

Steps 4 and 5 are repeated for each of the 15 questions.

1.1.7. Calculate student level.

The count of questions correctly answers is used to determine whether the student has achieved the level of the assessment. If the count is 10 or higher the application sends the new level and the date the assessment was completed to the database. If the correct count is less than 10 the application sends the students current level to the database along with the date of assessment.

1.2 Alternative flow of events

1.2.1 Log out

A student can log out of the website at any time. The use case ends and no assessment data is sent to the database.

1.2.2 No answer entered

If in step 4 of the use case the student does not enter an answer to the question an error message will be displayed. The use case returns to step 4.

2. Pre-requisites

The student will need to have logged in, use case 'Log in to application'. The teacher will need to have set an assessment date for the student which covers the date the user attempts to complete this use case.

3. Post-condition

Following successful completion of the use case the student will have completed the assessment and the assessment results will be stored in the database.

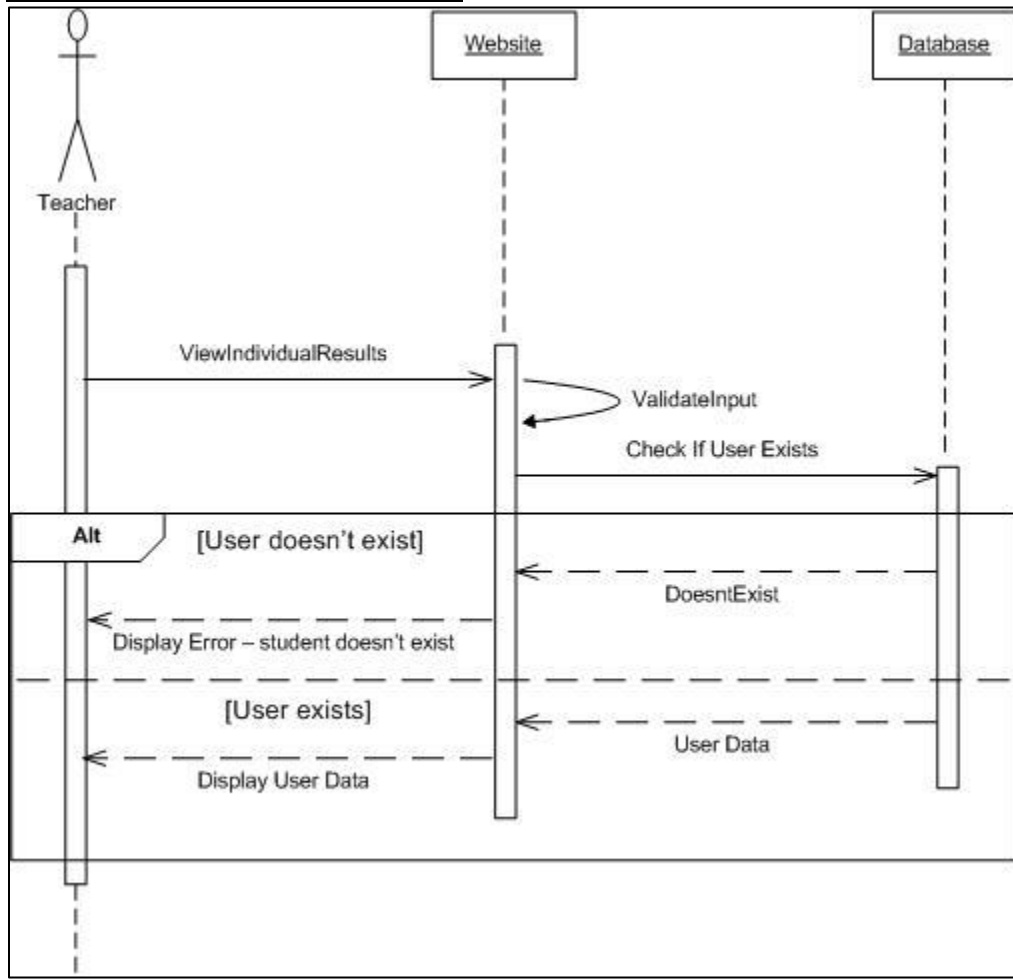
5.2. Sequence Diagram

The sequence diagrams models how the different components of the system collaborate to execute the functions necessary for the use cases to be developed. I have included the sequence diagrams for the main use cases:

- View individual student record
- View all student records
- Take assessment
- Practice mathematic questions

For sequence diagram for all other use cases, see Appendix D.

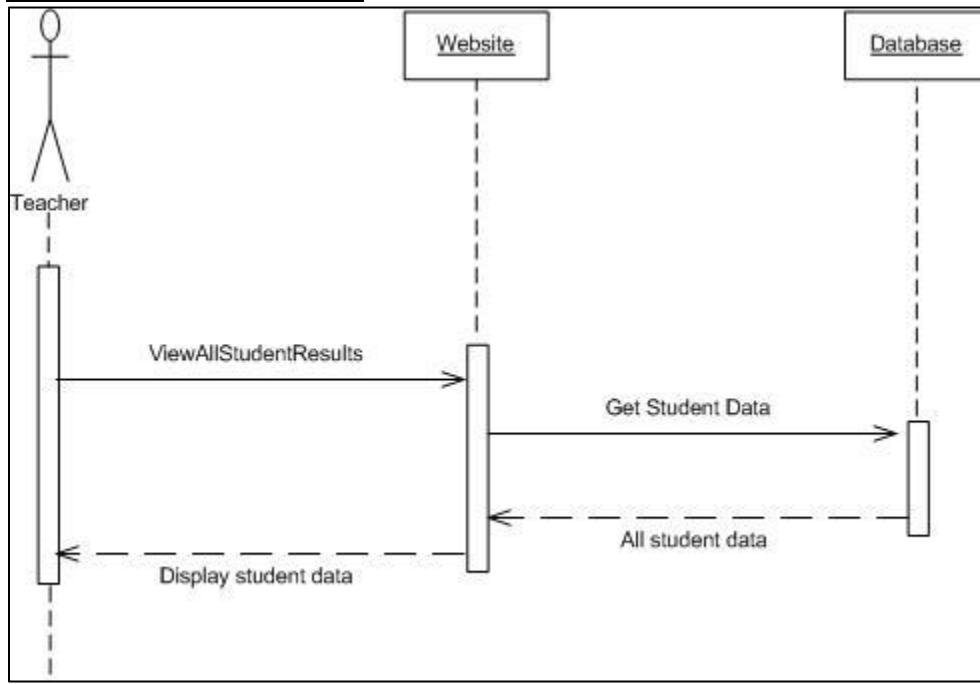
View individual student record



The teacher will choose the option to view an individual student result and through this will input the student details of which they wish to view. The website will validate this input ensuring that a value has been given for the student ID. The website will then use this student ID to check if a record exists in the database for this user.

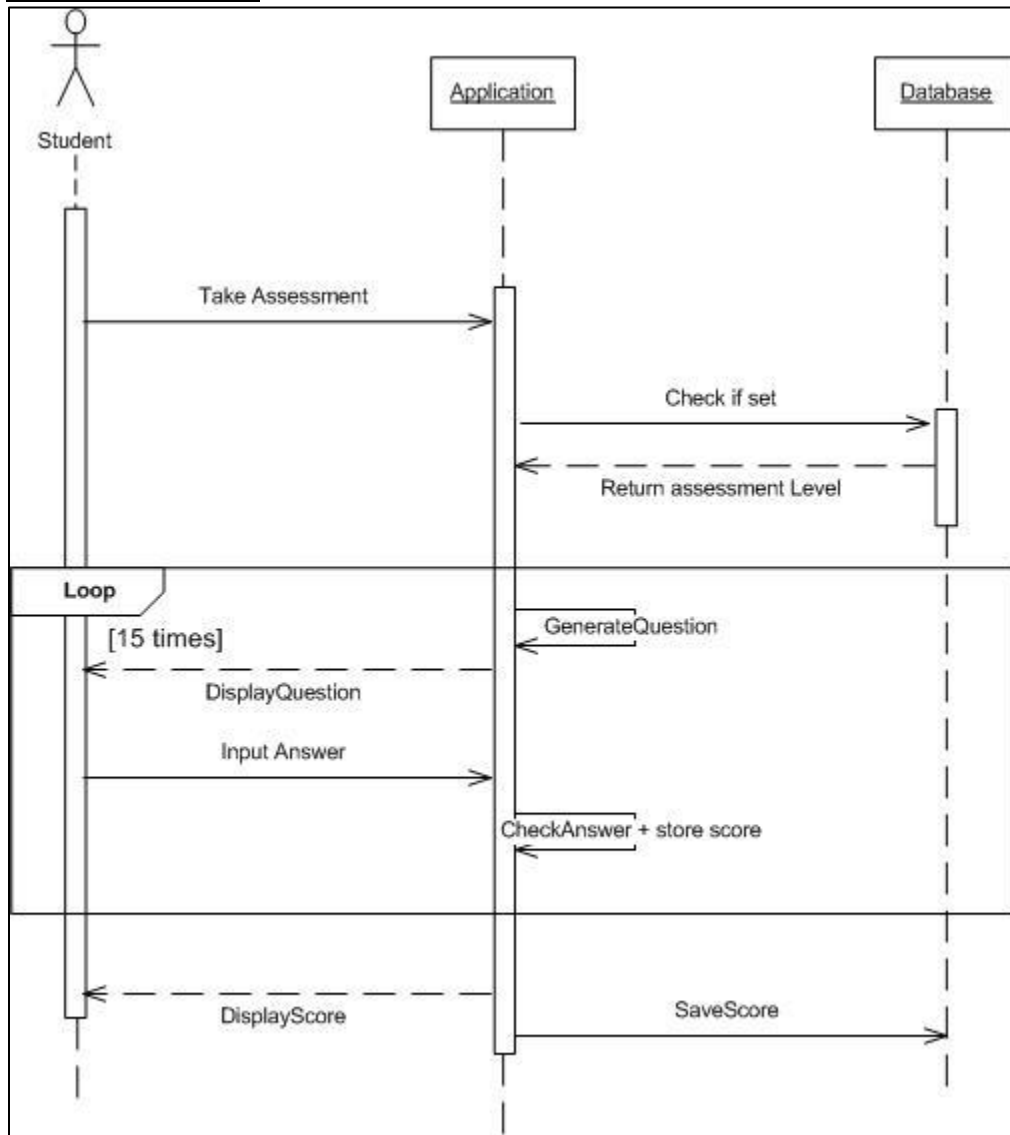
If the user exists the student's data will be retrieved from the database and presented to the teacher. If the student does not exist in the database an error message will be displayed to the teacher explaining that no record exists for the student they entered.

View all student records



The teacher will choose the option to view all student result on the website. The website will retrieve all students' records from the database and present the data to the teacher.

Take Assessment

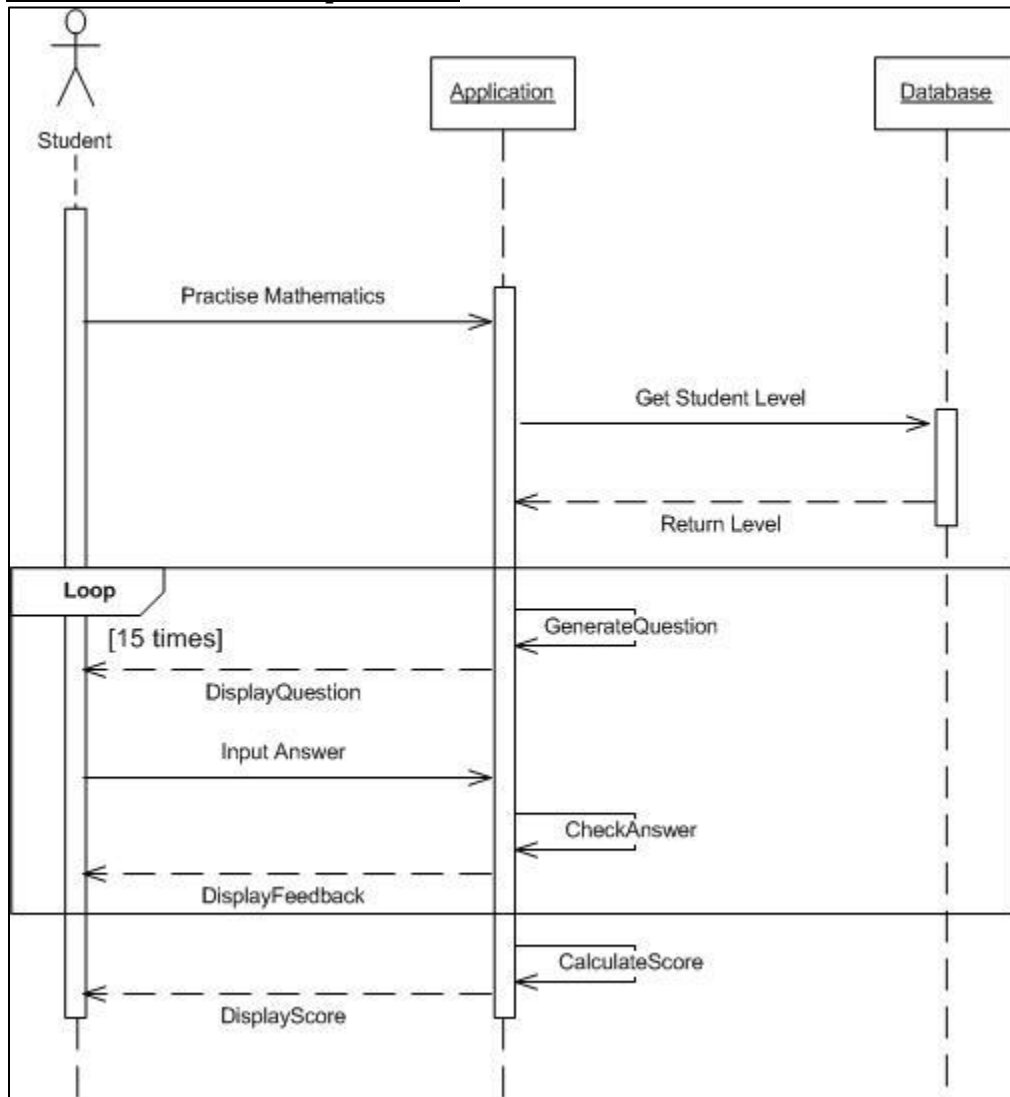


The student will choose the option to take an assessment through the application. The application will check if an assessment has been set through checking if an assessment record exists in the database for the student.

If an assessment has been set the application will generate 15 questions for the student based on the level that was set for the assessment. The student will provide an answer for each of the questions and the application will check if the answer given was correct. When the user has completed all 15 questions the application will display the users score to the student and the score and student ID will be sent to and stored in the database.

If an assessment has not been set for the student the option to take an assessment will not be presented to the student.

Practise mathematic questions

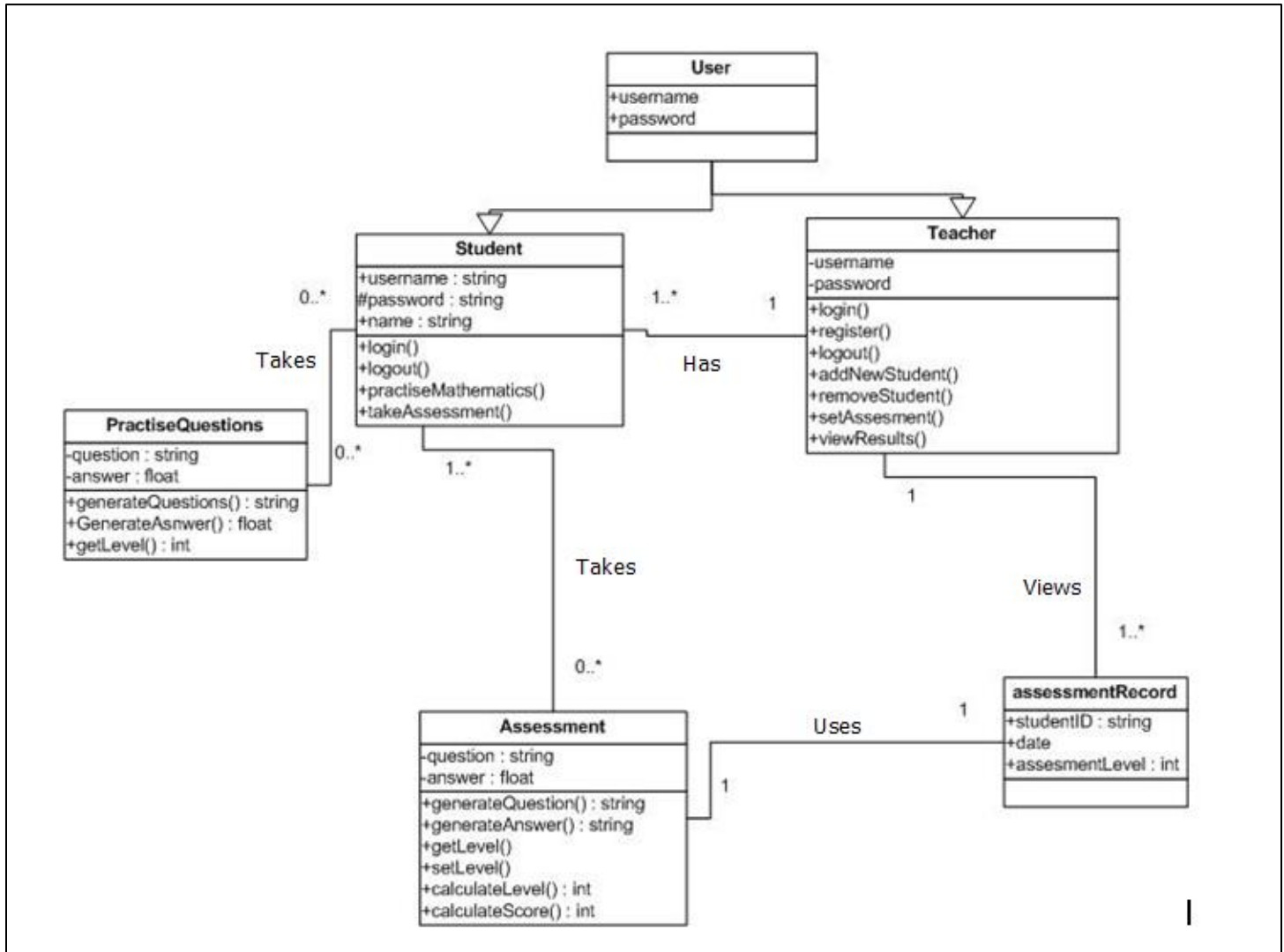


The student will choose the option to practise mathematics through the application. The application will check the level of the student through checking the student's record in the database.

The application will generate 15 questions for the student based on the level that was retrieved from the database. The student will provide an answer for each of the questions and the application will check if the answer given was correct. The applications will then feedback to the student whether they had correctly answered each question. When the user has completed all 15 questions the application will display the users score to the student.

5.3. Class Diagram

The following class diagram has been derived from the use cases outlined above and outlines the structure of the system:



Student

The student class contains the details relevant to the student which are used to identify a specific student. Students can log in and out of the application, take an assessment and practise questions. Each student has one teacher.

Teacher

Contains the details used to identify a teacher and allows a teacher to perform the following functions; login, logout, register for a teacher account, add a new student,

remove a student, set an assessment and view the results of an assessment. A teacher can have one or more students.

Practise Questions

This class contains a question and an answer which will be presented to the user when they chose the practise questions option.

Assessments

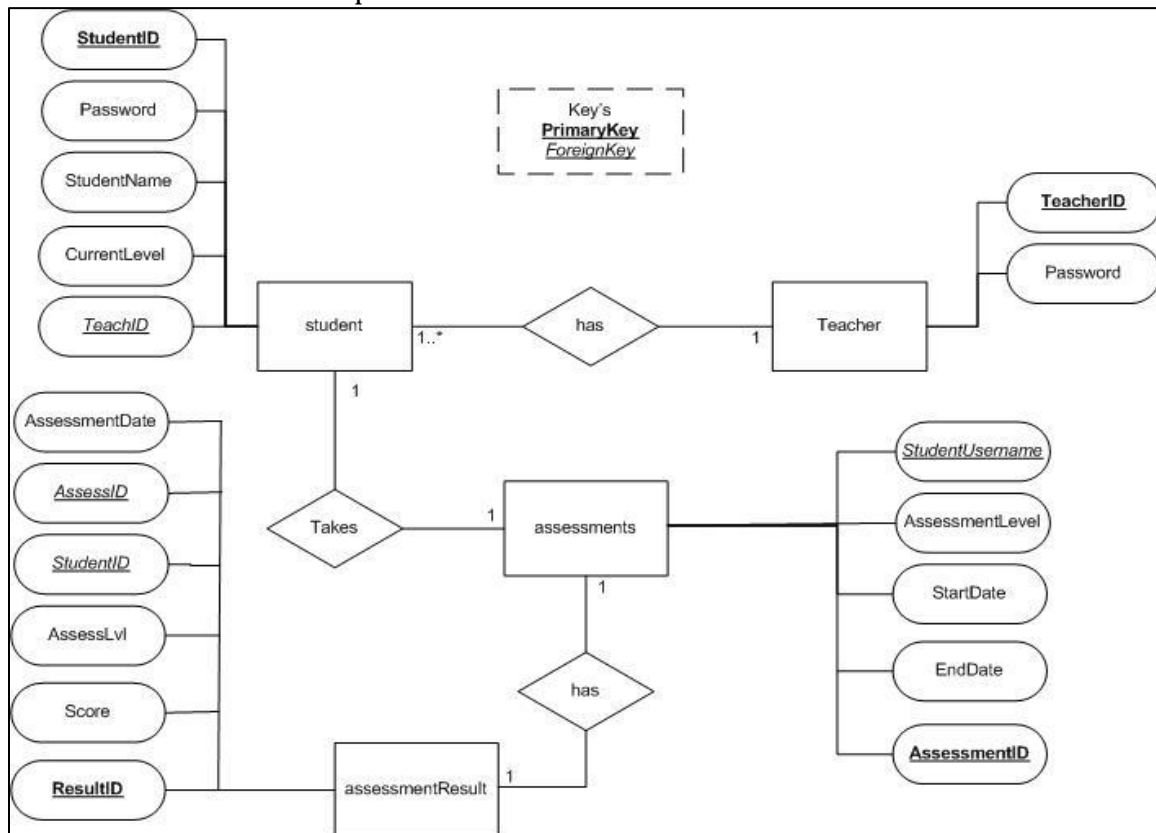
This class contains the questions and answers for an assessment. There are additional functions associated with this class which calculate the students score and will update the student's level in the database.

AssessmentRecords

This is the class that enables a teacher to view details of the results of a student's record. It stores details of the assessment including the score the students achieved, the level of the assessment and the date that the assessment was completed.

5.4. Entity Relationship Diagram

I have created an entity relationship diagram to model the database design. The diagram outlines each of the tables that the database will consist of along with their attributes and relationships to other tables.



I have identified the following tables which will be implemented:

Student

The student table will contain all data about the student and will be required for storing account details and the students latest achieved level. Such data will be required for the login system and to establish what level of questions the system should generate for the practise questions.

Teacher

The teacher table will store the teachers account details and will be used to assign students to each teacher account and enable the teacher to log in to the website.

Assessments

The assessments table will store details of any assessments that have been scheduled and used to determine if a student has an assessment available to them and if so will determine the level of the questions that will be generated for the assessment.

Assessment Results

The assessments results table will store all details of the result of each student's assessments. Data stored in this table will be used to inform the teacher of how well a student had performed in each of their assessments.

5.5. Algorithms

Each question set will compromise of 15 questions. To generate different questions and answers based on the student's level I have derived a set of algorithms which will generate these questions and their correct answer.

For algorithms to generate questions for levels 2 to 5, see Appendix E.

Level1(){

Question number +=1;

int X = random number between 0 and 10;

for QuestionNumber 1 to 3 do

int y = random number between 0 and 10;

Question = What is larger X or Y?;

If X > Y

Answer = X;

End if

If Y > X

Answer = Y;

End if

end for

```
for QuestionNumber 4 to 7 do
    Question = What is number next after X?;
    Answer = X + 1;
end for
```

```
for QuestionNumber 8 to 11 do
    int y = 11 - x;
    int z = random number between 0 and y;
    Question = What is X + Z?;
    Answer = X + Z;
end for
```

```
for QuestionNumber 12 to 15 do
    int y = X + 1
    int z = random number between 0 and y;
    Question = What is X - Z?;
    Answer = X - Z;
end for
```

```
}
```

The algorithm will be called one time for each question. QuestionNumber will be initialised as 0 and will be used to determine the style of question that the user will be presented with. The question text will be displayed to the user and an input box will be presented for the user to enter an answer. The users input will then be compared to the answer to establish whether they have answered the question correctly.

6. User Interface Design

To establish whether I have understood the user's needs for the interface I have created a prototype of the systems design. I will then test this prototype on intended users to establish whether my design is suitable for what they will use the system for in addition to ensuring that its ease of use is appropriate for both users.

6.1. Website Interface Design

In order to create a prototype for the user interface I used Balsamiq which is a wireframing tool allowing me to build mock-ups of the interfaces for my system (Balsamiq.com, 2014). Balsamiq enabled me to use pre-defined user interface elements in my designs in addition to linking the pages to mimic navigations through buttons. Balsamiq doesn't have the capability to demonstrate the functional aspect of my system and so I have simply focused on how the display will look in this prototype.

For the interface design of the website I have followed the guidelines from 'Usability.gov'. The layout of each page is consistent throughout with the navigation on the left hand side, content occupying the majority of the page to the right, and the system name as a heading to the top of the page. On each page the content area has its own heading which identifies the purpose of the page. I have labelled all buttons with names that describe the main task of the page that the button opens to ensure users understand the purpose of navigating to that page. Separation of navigation controls ensures the user can quickly identify the pages they wish to access. I have chosen to keep background colour of the area for the main content white and the contrast in colour draws the users focus to this area.

Welcome Page

I have created the welcome page to inform non-registered members about what the system does. The purpose of this page is to give an overview of the features that registered members have access to and entice new users to sign up. From this page users can navigate to the register page and the login page. I have listed the options in the navigation bar in the order that a first-time user should complete the tasks.



Register Page

The purpose of the register page is to enable new users to register their account details to the system in order to use these details to login. The page consists of text input boxes with headings to clearly state the information that the user is required to enter in these fields. In order to avoid input errors the user is required to enter their password and email details twice to enable double entry validation on the input which should help to avoid errors with the user entering the wrong details. The purpose of using the red font and '*' symbol is to ensure that the user is aware

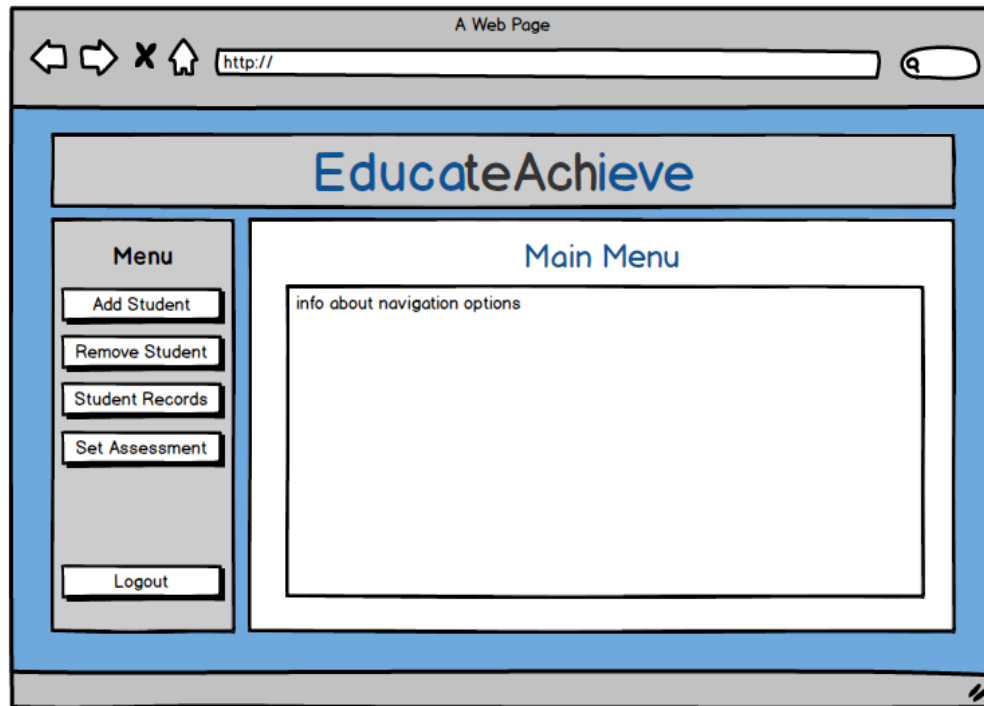
that all fields are mandatory. Upon entering their details the user will select the 'Register' button which will validate the input and enter these details into the database. This functionality could not be achieved through the prototype and so the button in the prototype simply redirects to the 'Login' page mimicking what would happen on successful entry of registration details.

Login page

The image shows a web browser window with the title "A Web Page". The address bar contains "http://". The main content area has a blue border and a header "EducateAchieve". On the left, there is a "Menu" sidebar with two buttons: "Register" and "Login". The "Login" button is highlighted in blue. The main content area is titled "Login" and contains two text input fields labeled "Username" and "Password", followed by a "Login" button.

The user can access this page through selecting the 'Login' button in the navigation bar or through being re-directed from the 'Register' page. The purpose of the page is to enable the user to enter their pre-defined login details in order to access data stored in their account. This will require the user to enter valid user credentials that have previously been registered and therefore added to the database. Again, I have used text input boxes to enable the user to enter these details. The boxes are labelled by the default text that is displayed in the input boxes. I have used the same terminology to describe the users details as the terminology used for the registration page to ensure users know which data they are required to enter. The 'Login' button within the content section is used to submit these details to be checked by the database. As this functionality cannot be demonstrated through the prototype, the button in the prototype redirects to the 'Main Menu' that a user would be redirected to on successful login.

Main Menu



The 'Main Menu' is the first page that the user will see when they have logged in. This page contains information about the options available to the user whilst they are logged in. In addition the navigation bar contains a selection of buttons to the pages that only a logged in user can access. Users can Logout at any time through selecting the 'Logout' button in the navigation menu.

Add Student

To access this page the user will select 'Add Student' from the left hand navigation bar. The user is required to enter the following details for the student they wish to add; the students ID, the students name and the student's password. These will be decided by the user and the user will be required to enter unique student ID's for each student. The details entered here by the teacher will be used by the student to login to the iPad application.

To avoid user input error on the password the teacher is required to enter the password twice which will be validated against each other to ensure they match when the user selects the 'Add Student' button. This functionality could not be demonstrated in the prototype and so the button simply navigated the user back to the 'Main menu' which is where they would be redirected after successfully adding a student.

A Web Page

http://

EducateAchieve

Menu

Add Student

Remove Student

Student Records

Set Assessment

Logout

Add Student

**All Fields Required*

StudentID *

Student Name *

Password*

Confirm Password*

Add Student

Remove Student

A Web Page

http://

EducateAchieve

Menu

Add Student

Remove Student

Student Records

Set Assessment

Logout

Remove Student

**All Fields Required*

StudentID *

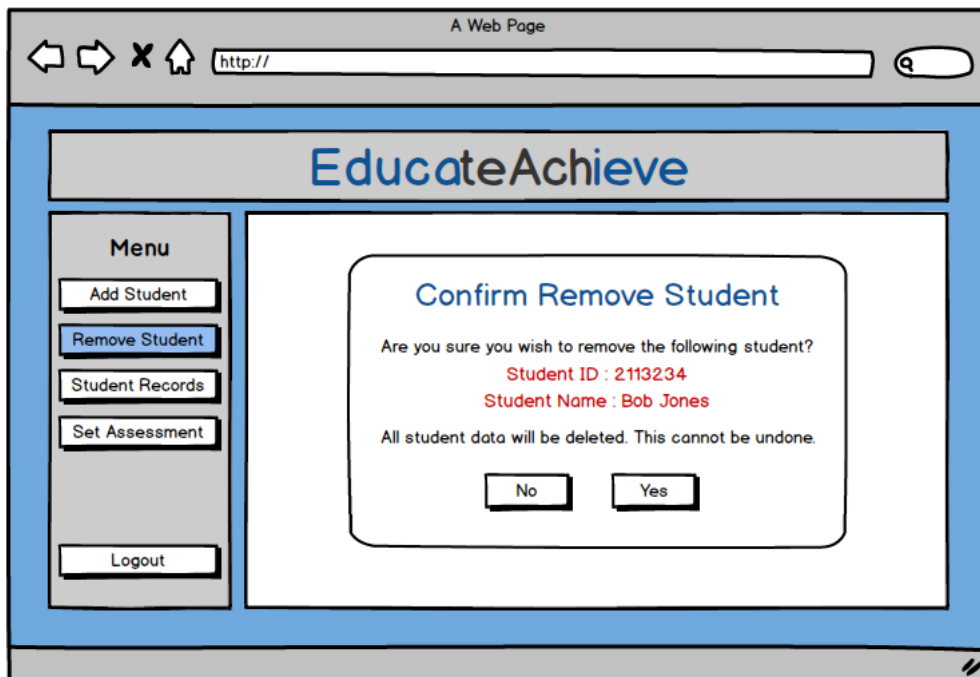
Remove Student

The remove student page can be accessed via the 'Remove Student' button in the navigation bar located on the left hand side of the page. The purpose of the page is to allow a user to remove students details from the database and therefore from the

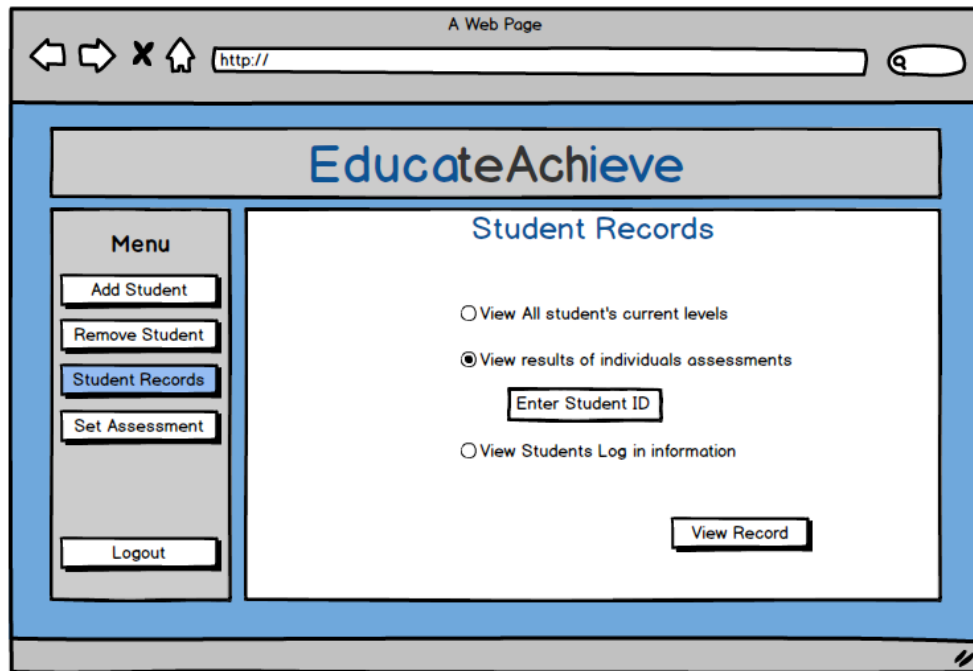
system. The user is required to enter the StudentID of the student they wish to remove. The user must already exist in the system to be removed. The user will then select the 'Remove Student' button. This will then redirect them to the 'Confirm Remove Student' page.

Confirm Remove Student

The purpose of this page is to ensure that the user does not accidentally remove the wrong student. The page will use the StudentID that was entered to identify the students name and will display a message asking if the user is sure they wish to remove the student. The message also explains that the action cannot be undone and as a result all data held about that student will be removed. This ensures that the user is aware of the impact that deleting a student will have on the data stored about them. The user can then use the 'yes' and 'no' buttons to confirm or cancel the action.

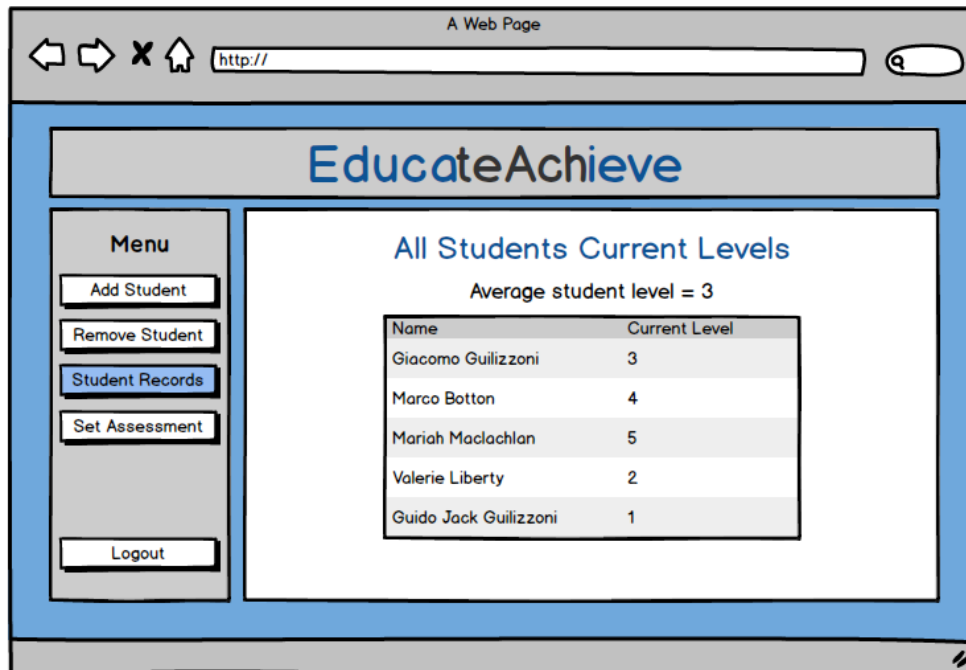


View Student Records



The student records page contains a list of the data that the user can view about the students. The page can be accessed by selecting the 'Student Records' button in the navigation menu. I have used radio buttons to select the records the user wishes to view. The user can choose between viewing the current levels of all students, viewing the result of a specific students assessments and viewing the log in details for students. If the user selects the radio button labelled "View results of individual's assessments" they are required to provide the student ID for the student whose record they wish to view. The user will then need to select the 'View Record' button to see their chosen data. The Balsamiq software doesn't allow the functionality of adding a button to be linked to radio buttons and so for the prototype the radio buttons themselves are linked to their page and the 'View Record' button is for demonstration purposes only.

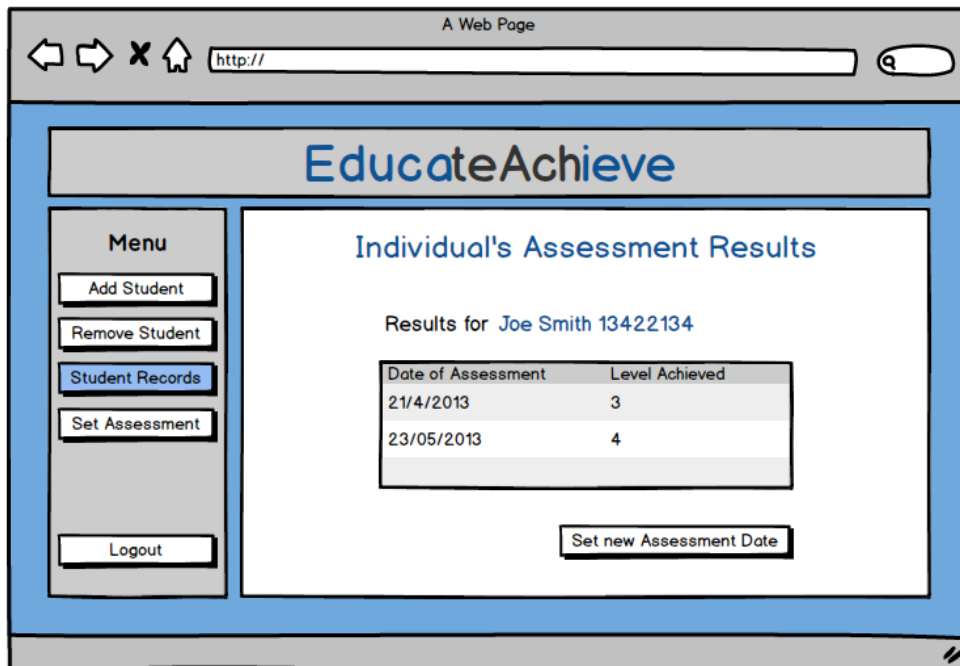
View all students' current levels



This page will be displayed on selecting the “view All students current Levels” radio button on the ‘Student Records’ page. Information about each user’s current level is displayed in a table. I have chosen to place the information in a table to ensure that the data is organised and easy to interpret. In addition above the table I have added the average of all students’ levels so that the user can compare individual student levels to this to see if they are achieving above, below, or on average compared to the rest of the class.

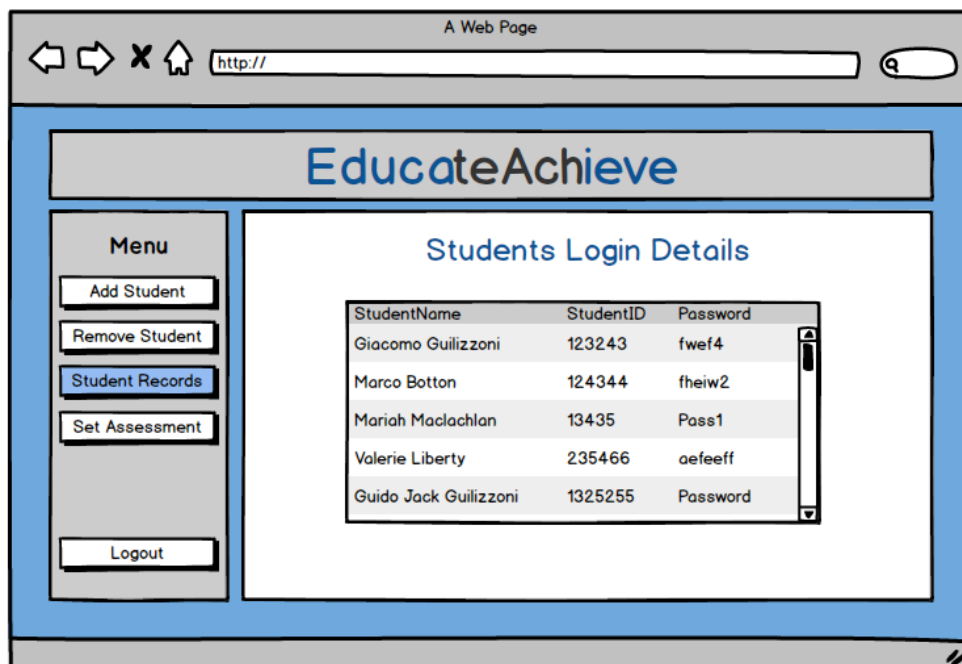
Individual’s assessment results

This page is accessed through selecting the ‘view individual’s assessment record’ radio button on the ‘Student Records’ page. The Student ID entered in the input box on the ‘Student Record’ page is used to collect all the relevant data about that student from the database to be displayed. Information about all the assessments the student has completed will be displayed in a table to ensure that it is easily readable. In addition the name of the student is displayed on top to confirm to the user which student’s details they are viewing.



Students Log in Details

The 'Students Login Details' page can be accessed by selecting the "view students log in information" radio button on the 'Students Records' page. The purpose of the page is to display the student account details for all the students that the teacher has registered. This will enable the user to identify the log in details of students if they have forgotten their login details. In addition the user can identify if any students have not been added to the system. Again I have placed these details in a table to ensure that they are displayed in an organised and easily readable format.



Set Assessment

The 'Set Assessment' page allows the user to schedule an assessment for one or all students. This will allow the student to complete an assessment between the start and end dates specified by the teacher. The user can also select the difficulty level of the assessment. I have used date input fields to enter the dates which enable the user to view the dates in calendar view. This enables the user to easily determine which dates to enter for a particular duration of time. In addition I have utilised the drop down list to display all the available assessment levels that the teacher can use. This will ensure that the teacher doesn't enter a level that doesn't exist and it also enables simple and quick selection. The radio buttons are used to select whether these assessments should be set for all students or a specific student. If a specific student is chosen then the user must enter the Student Id for the student who they wish to set the assessment for. When all details of the assessment have been entered/selected the user will select the 'Set Assessment' button which will schedule the assessment for the student(s).

The screenshot shows a web browser window titled 'A Web Page' with a URL bar containing 'http://'. The main content area is titled 'EducateAchieve' and 'Set Assessment'. On the left is a 'Menu' sidebar with buttons for 'Add Student', 'Remove Student', 'Student Records', 'Set Assessment' (highlighted), and 'Logout'. The main form area contains the following fields and controls:

- Start Date:** Two input fields for day and month, followed by a calendar icon.
- End Date:** Two input fields for day and month, followed by a calendar icon.
- Assessment Level:** A dropdown menu currently showing 'Level 1', with a list of options: Level 1, Level 2, Level 3, Level 4, and Level 5.
- Apply to:** Two radio buttons: 'All Students' and 'Individual Student' (which is selected).
- Enter Student ID:** A text input field, visible only when 'Individual Student' is selected.
- Set Assessment:** A large button at the bottom right of the form.

6.2. Application Interface Design

The iPad application interface design will be based on Apples Usability Principles and will also take into account the reduced screen size of the tablet and so will limit the amount of content on each page. In addition I will take into account the touchscreen interface in the design and ensure that objects are both large in size and not too close to other objects to avoid accidental selections.

Login Page

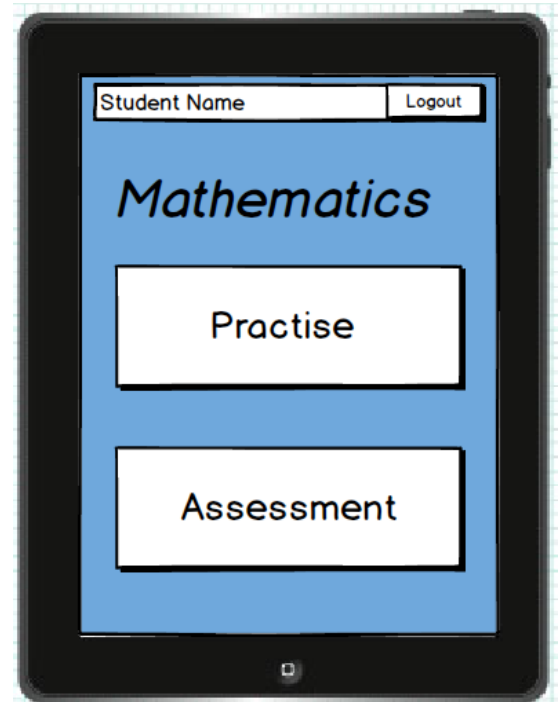
The 'Login Page' will be the first screen the user is presented with when they open the application. The user will be required to enter a username and password which will have been created by their teacher. I have included the application name on the first page so that users know what application they are using. I have also used labelled text input boxes so that the user knows which box to put which detail into. The user will proceed to check the details they entered are correct by selecting the 'Login' button.



Homepage

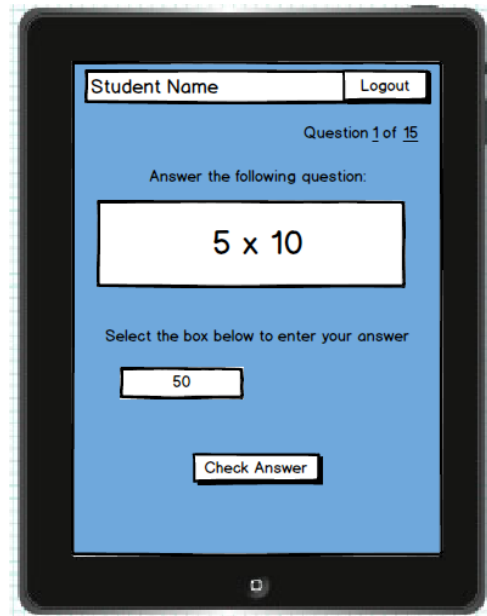
This page will be the one displayed when a user logs in with the correct user credentials. As each iPad may have several users I have chosen to display the name of the user that is logged in at the top along with a 'Logout' button so that users can check that they are logged in to the correct account and if they are not they can 'Logout'. Users can logout at any time throughout the application by selecting this 'Logout' button which will remain in the same place throughout and on selecting will redirect the user to the 'Login' page.

The page contains a button to access practice questions and a button to access the assessment when an assessment has been scheduled. I have used the label "Mathematics" to inform the user of the content that the 'Practise' and 'Assessment' material will be based on.



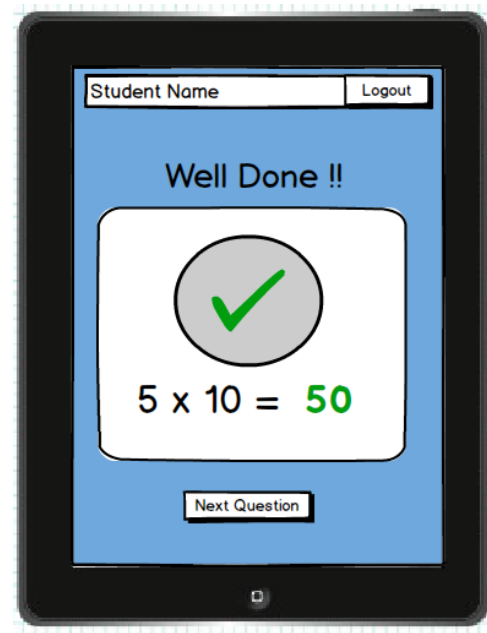
Generated question – correct answer

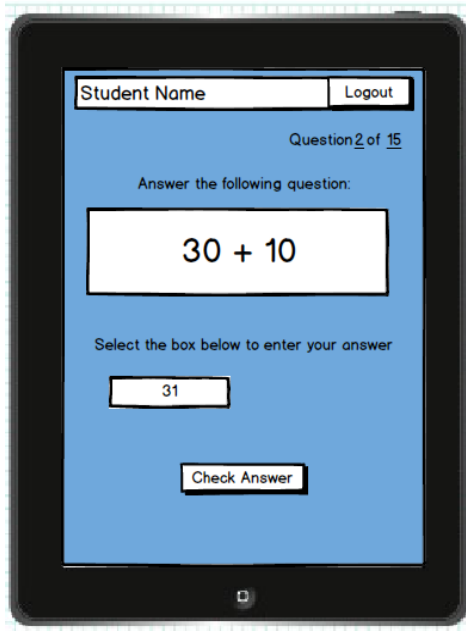
Both the 'Practise' and 'Assessment' buttons will direct the user to a mathematical question page. There will be 15 questions generated each being displayed the same way as demonstrated in the prototype. There will be a label in the top right corner to display the number of questions so far and a label with a white background displaying the question. The question is placed over a white background and has a larger font than the rest of the content to ensure that it is clearly visible and easy to identify. I have also used an input box where the user would be required to enter the answer to the question. The Balsamiq software doesn't enable the user to enter the text themselves in the prototype demonstration and so I have included the correct answer to display what would happen when the correct answer is entered. The user will then select the 'Check Answer' button to determine if the answer they entered is correct.



Check correct Answer

If the answer entered was correct a message and picture will be displayed to the user explaining this. The message will include the question along with the correct answer which I have highlighted by using a green font. The use of a tick image is used to visually represent that the answer is correct. This is to enable the user to quickly identify whether their answer was correct or not. Having read the feedback the user would then select the 'Next Question' button to generate the next mathematical problem.



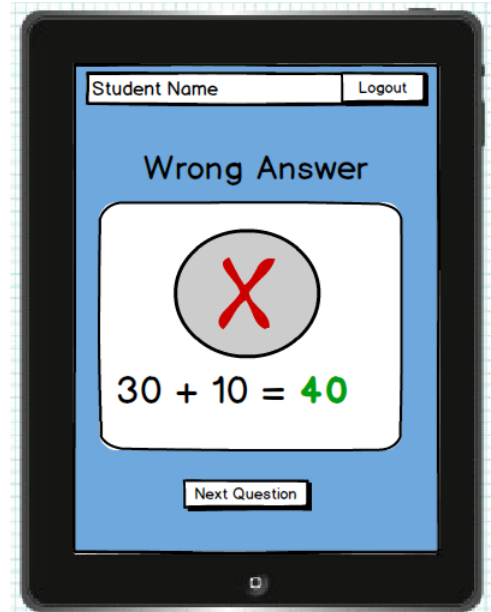


Generate question – Incorrect Answer

The following page of the prototype is used to display the result of an incorrect answer. In the same way as the first question the user would enter they're answer in the input box and select 'Check Answer' button.

Check Answer

If the answer entered was incorrect a message and picture will be displayed to the user



explaining this. The message will include the question along with the correct answer which I have highlighted by using a green font. The use of a red cross image is used to visually represent that the answer is incorrect. This is to enable the user to quickly identify that the answer they gave was incorrect or not. Having read the feedback the user would then select the 'Next Question' button to generate the next mathematical problem. This would be repeated until all 15 questions had been answered.

Final score



After completing all 15 questions the user would be presented with a message displaying how many questions they correctly answered out of the 15. They would also be informed that the practise session or the assessment had finished. If the questions were generated as a result of selecting the 'Practice' button from the main menu no data would be stored. If the user selected the 'Assessment' button the result would be stored in the database to be viewed by the user's teacher. I have used green font for the score to make it stand out on the page to enable the users to quickly identify their score. From this page the user can navigate back to the 'Main Menu' using the 'Main Menu' button.

7. Test Plan

This section of the report outlines the testing that will be done on the system to ensure it meets both the functional requirements of the system and the interface design requirements.

7.1. Test Cases

In order to test the functionality of the system I have identified test cases which will be used to determine whether the functionality of the system meets the requirements. For all detailed test cases, see Appendix F.

This test will be conducted on the system implementation only as the interface prototype does not have the functionality that the test cases will be testing.

Prior to conducting these tests I will populate the database with the following records:

assessresults						
resultID	assessID	studentID	AssessmentLvl	Mark	achievedLVL	assessmentDate
1	74	31	1	5	0	2014-04-02
2	75	31	2	15	2	2014-04-02
4	76	31	3	13	3	2014-04-02
5	77	31	3	2	0	2014-04-02
6	78	33	1	15	1	2014-04-05

teachaccount			
TeachID	TeachUsername	TeachPass	TeachEmail
1	greenVR	Rach1410	greenvr@gmail.com
3	Bob123	Pass123	Bob@cardiff.ac.uk

studaccount						
StudID	StudUsername	StudName	StudPass	TeachID	CurrentLvl	Stars
1	Jfields	Jim Fields	Pet3	1	3	15
2	ZDavis	Zak Davis	cardiff	3	3	0
8	LJones	Lisa Jones	calon	3	1	0
31	RJops	Rhys Jopson	Alffie123??	1	3	4
33	LBishop	Lily Sue	lily	1	2	0

assessments				
assessID	studUsername	startDate	endDate	assessLvl
1	Jfields	2014-04-13	2014-04-20	4

7.2. Heuristic Evaluation

I will conduct a heuristic evaluation of the design prototype in order to evaluate whether the system's interface meets the usability requirements of the system.

The heuristic evaluation will involve evaluating the websites user interface with Nielsen's ten heuristics (Nielsen, 1995).

1. Visibility of system status
2. Match between system and the real world.
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use
8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

In addition I will use Nielsen's severity rating scale to determine how much of an impact the heuristic violation will have on the system (Nielsen, 1995):

Severity	Description
0	I don't agree that this is a usability problem at all
1	Cosmetic problem only: need not be fixed unless extra time is available on project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, so should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released

7.3. User Testing

In addition to the test case testing and heuristic evaluation I will conduct user testing on both the interface designs and final implementation with its intended users.

The purpose of the test is to establish whether the users understand how to complete tasks using the interface. As the developer, I am always able to accomplish these tasks as I have developed the interface designs and therefore conducting the user testing allows me to determine whether users without any additional knowledge of the system can use the interface effectively.

Participants will be given a set of scenarios which outline the tasks that they should complete. In addition they will be advised to think out loud about the steps they are taking to complete the tasks. Participants are also advised that the designs do not

yet have functionality and therefore where text input is required the user is to assume this has already been entered.

The user testing of the interface designs for the website will be conducted with 3 users using the following 'Scenarios for teachers':

Scenarios for teacher

Scenario 1

You would like to schedule a level one assessment for 'Peter James' and would like this assessment to be completed between the 10th of April and the 17th of April.

Scenario 2

The student 'Lucy Jones' has left the school and you would like to remove her from your list of students on the 'EducateAchieve' system.

Scenario 3

You have previously set an assessment for the student 'Rhys Thomas' to be completed between the dates 5th of April 2014 and 9th of April 2014. You now wish to view the result of this assessment to answer the following questions:

- What score did Rhys Thomas achieve?
- Did he achieve the level you set out?
- What is the level he is currently achieving?

Scenario 4

You have added the student 'Janet Evans' as a student of yours on the 'EducateAchieve' system. She has forgotten the account details that you set up for her. She will need her username and password to login. You now wish to view these details.

The interface designs for the application have been tested with 2 users using the 'Scenarios for students'.

Tasks for student

Scenario 1

Your teacher has created an account and has provided you with the following details:

Your username is 'JBennet' and your password is 'Bob08'.

Using these details you now want to log in to the application and practise mathematical questions.

Scenario 2

An assessment has been set for you by your teacher. You would now like to complete this assessment.

8. Phase 1 Testing

8.1 Heuristic Evaluation

During the heuristic evaluation of the website prototype the following heuristic flaws were identified:

Heuristic	Observation	Severity Rating
Visibility of system Status	No feedback has been implemented to explain the response to the user following setting an assessment.	4
Error Prevention	System should state that all fields in set assessment are required to avoid the user submitting the details with missing information.	3
Help users recognise, diagnose, and recover from errors	No error messages have been designed throughout the prototype.	3
Help and documentation	No help and documentation exists.	3

During the heuristic evaluation of the Application prototype the following heuristic flaws were identified:

Heuristic	Observation	Severity Rating
Visibility of system Status	No feedback has been implemented to explain the response to the user following an incorrect login.	4
Error Prevention	System should state that all fields are required to avoid the user submitting the details with missing information.	3
Help users recognise, diagnose, and recover from errors	No error messages have been designed.	3
Help and documentation	No help and documentation exists.	0
Match between system and the real world.	Students would refer to an assessment as a test. Button title for assessment page should be renamed to test to avoid possible confusion.	3

8.2 User Testing

Website

From observing the teachers attempts understand how to complete the scenarios of the 'Tasks for teacher', outlined in the test plan section, I found that they were able to complete all tasks set out. They had no problems understanding how they would interact with the interface.

As the interface designs lacked functionality the user was unable to input their own data and fully test the interface and so the information was already input into the designs. As a result, the main purpose of the testing was to establish whether the style of input was appropriate. The participants felt that the website was simple enough to use and they could explain to me how they would complete each task if the system was a functioning prototype.

One participant explained that the current system would not enable them to identify students who had used the application to practise mathematics. The teacher was keen for the system to provide them with data about students practise attempt so that they could identify students who had put the effort in to developing their mathematics however may not be a high achiever with regards to levels. Having discussed this it was decided that the students should receive points each time they practised mathematics so that the teacher could identify students who were making the effort to learn. The teacher could then implement a reward system of their own and also utilise this information when writing a student's report.

Application

The user testing with students and the application enabled me to establish whether the interface of the system was easy to use. Again, the system was deemed to be easy to use by the students and they could explain how they would complete the scenarios for the 'tasks for students' (outlined above in 'Test Plan' section).

To establish whether the user enjoyed using the system I provided them with questions about the aesthetics of the system. From these questions I found that the students enjoyed mathematics and would enjoy using the application as they liked the instant feedback in the practise section. However, both participants explained that the aesthetics of the application were not appealing to a young user and recommended changing the background as it was 'very plain'.

9. Evaluation of design prototype

Following the heuristic evaluation the main changes to both the application interface and the website interface will be to implement error prevention through validation and error messages where appropriate.

I will implement the following changes to the application design:

- Rename the assessment button label from 'Assessment' to 'Take Test'.
- Provide feedback for incorrect login.
- Provide error message where login input fields are empty.
- Add a background image to all mathematic screens to make the application look more appealing to a young audience.
- Add a reward system to motivate students to practise questions and allow the teacher to identify students that are regularly using the application to practise mathematical questions. Students will earn a star each time they answer a question correctly in the practise section.

I will implement the following changes to the website design:

- Provide feedback for incorrect login.
- Provide error message where login input fields are empty.
- Provide error message where set assessment fields are empty.
- Display the number of questions that a student correctly answered in an assessment.
- Provide feedback after setting an assessment.
- Display the number of stars a student has earned in student records allowing the teacher to identify students that regularly use the application to practise mathematics.

10. Implementation

10.1. Tools Utilised

The system I have implemented has made use of the following tools and languages:

Xcode

Xcode is an integrated development environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. I have used this IDE to ensure that the program I developed was suitable for iOS which is required for compatibility with the iPad. To develop the application I have used the latest version of Xcode; Xcode 5. Xcode facilitates the building of the application and also enabled me to test the application on a simulated iPad. In addition, with an Apple developer account which I acquired, I was able to run the application on my iPad device which has also been registered for testing.

XAMPP

XAMPP is a development tool which allowed me to develop and test the website and database element of the system on my computer locally. The use of XAMPP is not necessary for the final implementation as I have utilised my account on the Cardiff University server to host my website and host the database however during development it was quicker to develop the system locally avoiding having to transfer files onto the university's server after each change.

PHP

I have utilised PHP to implement the webpages for the system. PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. I have used PHP to process the teachers input in the website and determine the output to be displayed. I have also utilised PHP to facilitate the connection between the website and the application to the database.

CSS & HTML

I have utilised a CSS throughout the website to format the display of the web pages that will be displayed to users. HTML has been used to implement the content of the webpages and to implement the forms which are used to collect the users input. During development I have used the browser Google Chrome Version 34.0.1847.116 m to test the display of the website and it is therefore currently the recommended browser to use for the system.

Adobe Photoshop

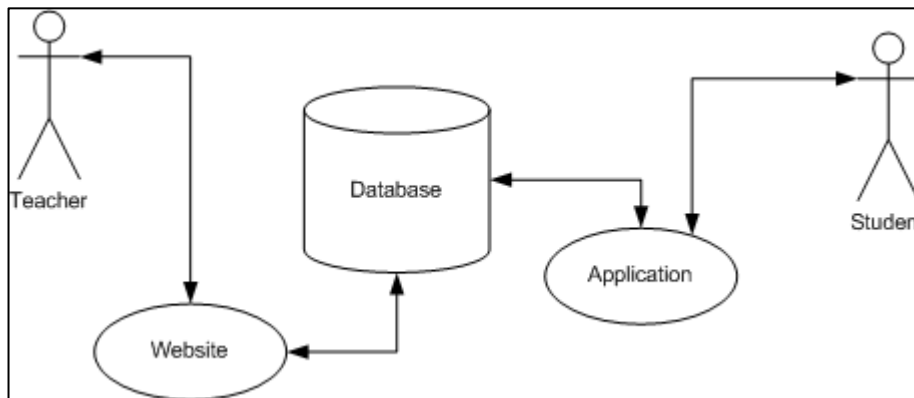
To create the images for the application and the website I have used Adobe Photoshop. It was used to create the systems logo and the applications feedback and background images.

mySQL

I have used MySQL within the PHP pages to access, delete, add and modify the data held in the database.

10.2. System Overview

The system I have implemented is made up of the components identified in the following diagram:



The teacher will interact with the website to:

- Input log in details
- Input registration details
- Input details of the student whose record they wish to view
- Input details of assessments to be scheduled

The website then queries the database with the teachers input to:

- Verify log in details
- Retrieve records
- Insert new records
- Delete records

Response from the database is then presented to the user through the website.

The student will interact with the application to:

- Input log in details
- Begin an assessment
- Practise mathematics

The application then queries the database with the students input to:

- Retrieve students level
- Check to see if an assessment has been scheduled
- Update assessment records

Response from the database is then used to determine the level of questions that should be presented to the user through the application.

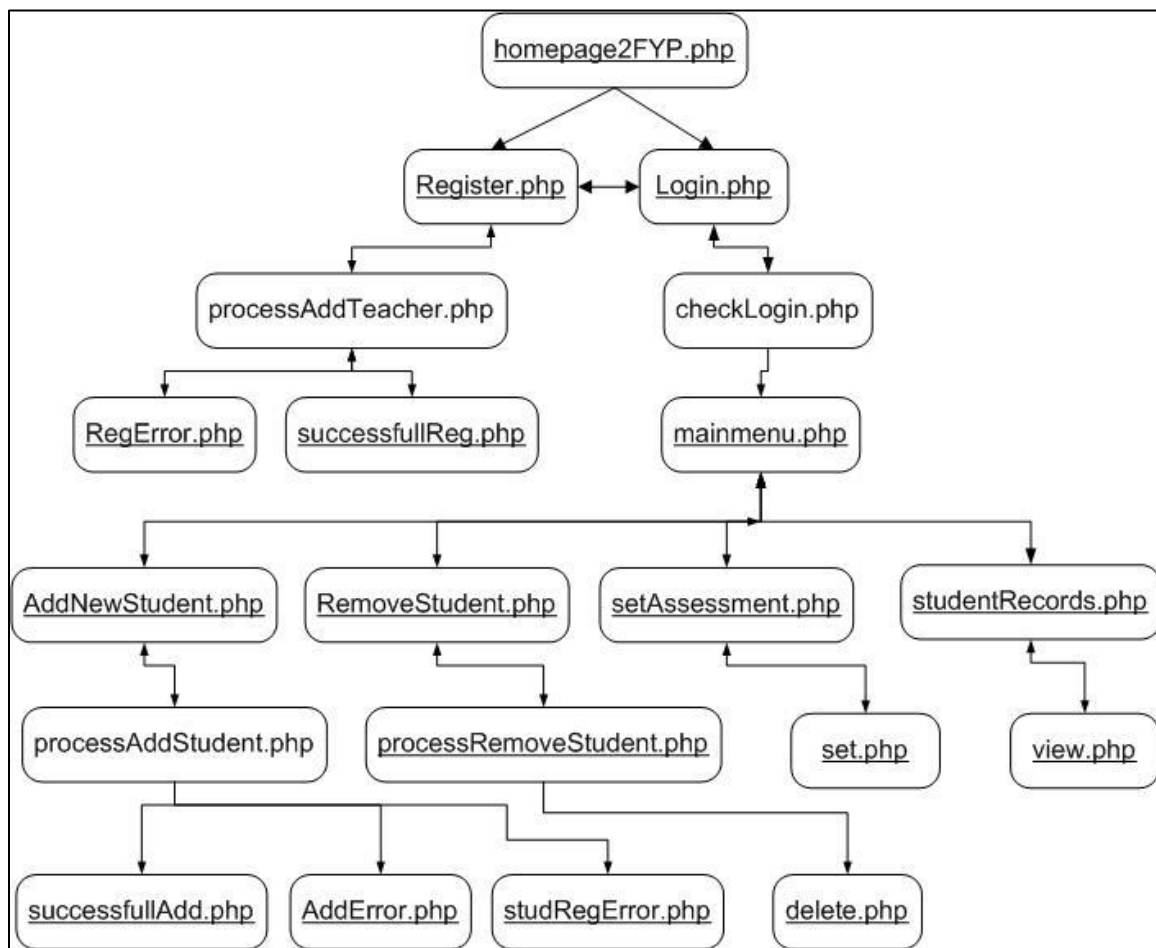
For information on where to locate full source code for the implementation, see Appendix G.

10.3. Website Structure

The website implementation consists of several PHP pages; and can be accessed through the following link:

<https://project.cs.cf.ac.uk/V.R.Green/LatestWeb/WebendFYP/homepage2FYP.php>

The hierarchal structure of these pages is outlined in the following diagram:



The arrow indicates that the page can be called from the current page. In addition pages whose file names are underlined contain the EducateAchieve.jpg file which is the logo used and displayed as the systems name, and the stylesheetFYP.css file which is the style sheet used to format the display.

Homepage2FYP.php

This page is the first page the user will be presented with when accessing the website. Its purpose is to explain what the systems intended use is and to guide new users on how to create an account.

Register.php

This page is the page the user will be presented with when they choose to register as a teacher to use the system. Its purpose is to retrieve the user's details through form input. This page validates the form input checking that the user has filled in all fields of the form with the correct syntax. When all input fields have been entered the input is posted to the processAddTeacher.php page.

processAddTeacher.php

Double entry validation has been implemented for the password and email fields. This page compares the two password inputs and the two email fields and if both are matches queries the database to determine if the username that was entered in the Register.php file already exists. If the username exists then the system cannot register the user as the usernames must be unique and RegError.php page is called. If the username doesn't already exist in the database a new record is added which contains the information from the Register.php form. The successfulReg.php page is then called. If either the passwords or the emails do not match the RegError.php page is called.

RegError.php

This page provides the user with feedback that the registration request they made could not be completed due to the passwords or emails not matching or due to the username being in use.

successfullReg.php

This page provides the user with feedback that the registration request they made has been completed and that they can now use the details entered to log into their account.

Login.php

This page is the page the user will be presented with when they choose to login to the system from the menu or if they try to access any page below Login in the hierarchy without initially logging in. Its purpose is to ensure authorised users are accessing the data within the system and to establish which of the students in the database are student of the teacher. The user's login credentials are obtained through form input. This page validates the form input checking that the user has filled in all fields of the form with the correct syntax. When all input fields have been entered the input is posted to the checkLogin.php page.

checkLogin.php

This page uses the details entered in the Login.php page to query the database and establish whether a record exists which matches the input provided. If a match

exists the user is redirected to mainmenu.php. If a match doesn't exist they are redirected back to Login.php and an error message is activated explaining that the wrong username or password was entered.

mainmenu.php

This page is the first page the user will be presented with when they have successfully logged in to the website. Its purpose is to explain how the user can use the systems features. This page and all pages below in the hierarchy can be used to navigate to AddNewStudent.php, RemoveStudent.php, setAssessment.php, and studentRecords.php.

AddNewStudent.php

This page obtains information about the student the teacher wishes to add to the system through form input. The form data is validated to ensure all fields have been entered and are then posted to the processAddStudent.php page.

processAddStudent.php

Double entry validation has been implemented for the password field. This page compares the two password and if both are matches queries the database to determine if the username that was entered in the AddnewStudent.php file already exists. If the username exists then the system cannot register the user as the usernames must be unique and AddError.php page is called. If the username doesn't already exist in the database a new record is added which contains the information from the AddNewStudent.php form. The successfulAdd.php page is then called. If the passwords do not match the studRegError.php page is called.

successfullAdd.php

This page provides the user with feedback that the add new student request they made has been completed and that the student can now use the details entered to log into the application.

AddError.php

This page provides the user with feedback that the student could not be added to the system because the username is in use.

studRegError.php

This page provides the user with feedback that the student could not be added to the system because the passwords entered did not match.

RemoveStudent.php

This page obtains the username of the student the teacher wishes to remove from the system. It's is obtained through a form and the value is then posted to the processRemoveStudent.php page.

processRemoveStudent.php

This page uses the username entered in the RemoveStudent.php form to query the database and obtain the full name of that username's user. The page presents this username and the student's full name to the teacher to confirm the deletion. The teacher's response is posted to the delete.php page.

delete.php

This page uses the values posted from the processRemoveStudent.php page. If the value 'Yes' was selected, indicating the user confirms they wish to delete the student, the record which matches the student username input given in the RemoveStudent.php form is deleted. If the value 'No' is selected the user is redirected back to the RemoveStudent.php page.

setAssessment.php

This page obtains information about the assessment the teacher wishes to schedule on the system through form input. The form data is validated to ensure all fields have been entered and are then posted to the set.php page.

set.php

The start and end date are validated to ensure that the start date is before the end date and that no assessment scheduled overlaps with these dates. If a clash exists then an error message is displayed to the teacher. If not it checks if the teacher wishes to add an assessment for all students or an individual student using the input from the setAssessment.php form. If the assessment is being scheduled for an individual student the database is queried to establish whether the user exists. If the user exists a new assessment record for that student is added to the database. If the student doesn't exist an error message is displayed to the user explaining that the assessment could not be scheduled for the student as the user doesn't exist. If the assessment is scheduled for all students the database is queried for the name of all students whose teacher is the active teacher and an assessment record is added for each of them.

studentRecords.php

This page obtains the type of records the user wishes to view. It's obtained through a form which utilises radio buttons. The selected radio button value is then posted to the view.php page.

view.php

This page checks the type of record the user wishes to view by using the value set in the studentRecords.php form. The page then queries the database for the records which match the user's selection and presents this data in a table to the user.

10.4. Website implementation

To ensure that a user can not directly access any page below checkLogin.php in the structure hierarchy without being logged in I have utilised PHP session variables.

```
<?PHP

session_start();

if (!(isset($_SESSION['login']) && $_SESSION['login'] != '')) {

header ("Location: login.php");

}

?>
```

All pages which require the user to be logged in contain this code. If a session variable has not been set for the login the user will be redirected to the login page. The logout button re-directs the user to the homepage2FYP.php page which contains code to clear this session variable and therefore ends the user's session.

To obtain user input I have utilised HTML forms.

```
<form name = "getDetails" action ="processAddStudent.php" method="post">

    <input type ="text" name="StudentID" required>

    <input type="password" name="Password" required>

    <input type = "text" name="StudentName" required>

    <input type = "submit" value = "Add Student"></center>

</form>
```

I have used the post method to submit the input data to the action page as this method is more secure for handling data such as password variables as it will not be stored in the browser history or cache of the user's computer.

HTML forms have a range of defined input types which I have utilised in all forms. HTML will validate the users input checking that it is of the right format before submitting the input to the action page and for inputs of type password the browser will hide the input from display which ensures that other people can't see what the users password is when typing it in. In addition I have also utilised the forms required attribute to ensure the user cannot submit the form data without completing all fields.

10.5. Website Implementation problems

When I developed the website I was hosting the webpages locally through XAMPP. On uploading the webpages to be hosted on to the university's server a MySQL deprecation error was displayed. This was due to my computer having an old version of PHP and the universities server being a later version of PHP. To resolve this issue it is advised to change the mysql_connect and mysql_query methods to the latest extensions MySQLi or PDO_MySQL. Due to time constraints and unfamiliarity with these extensions I have currently suppressed the error however once the mySql extension has been removed the website will no longer function correctly and it therefore needs to be amended.

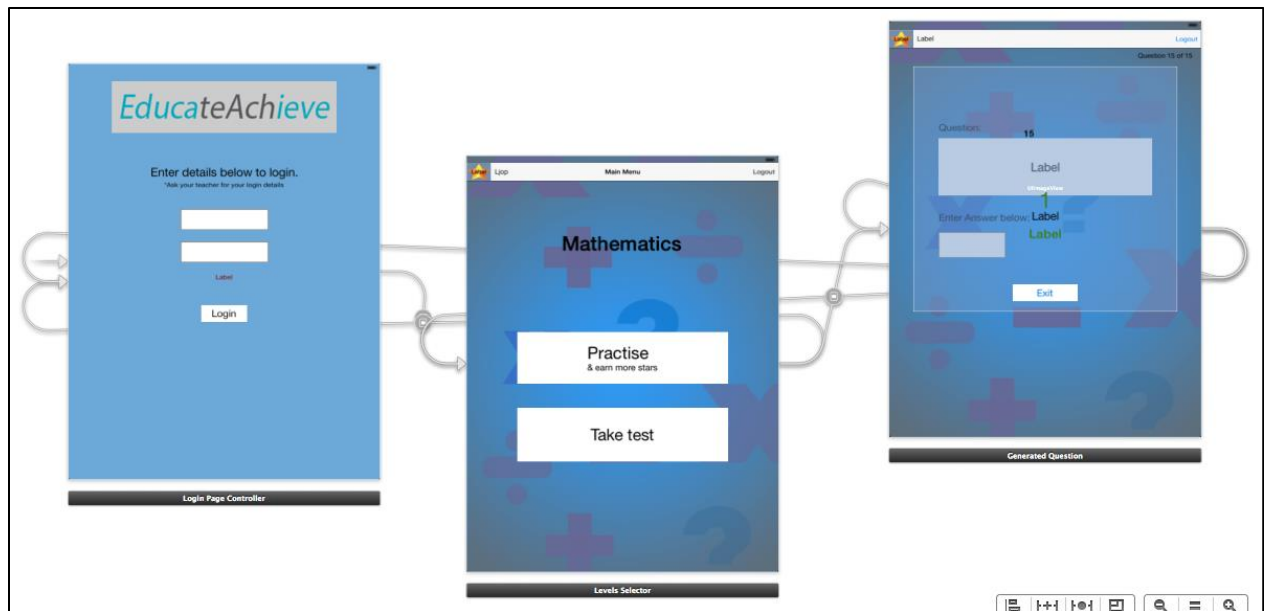
```
<?php
error_reporting(E_ALL ^ E_DEPRECATED);
?>
```

Another problem I experienced when implementing the websites was that the web browser was pulling the pages from cache and as a result was not always showing updates when I had modified the code. Once I had realised what was causing the old version of code to be displayed it was no longer a problem as I continued to test the code modifications after clearing the browsers cache.

Another problem that I have experienced is that the webpages input fields are displayed differently in different browsers. For example the inputs of type date do not display in Safari. At present these problems have not been rectified due to time constraints and it is therefore recommended that the website be run on Google Chromes browser.

10.6. Application Structure

To implement the interface of the application I have utilised Xcode storyboards which holds a collection of views and the flow between them with the use of segues. There are three views in the application:



Each view has a view controller class which contains the code to be executed when the user interacts with objects within the view. I have implemented the following classes:

LoginPageController

The LoginPageController class checks that the user has provided input for both the password and username field. If they have not then the error labels text is set to explain that both fields are required.

If both fields have been entered the input is sent to a connection.php page which compares the data with records in the database and returns the value 1 to indicate that there is a match. The system then adds the user name to an UserDefaults variable which is used throughout the application to determine the user. A segue is then activated calling the LevelsSelector view.

If there is no match in the database the connection.php file returns the value // and the LoginPageController class sets the error label text as incorrect username and/or password to inform the user that the wrong credentials have been entered.

LevelsSelector

The username of the current logged in user is sent to the connection.php which checks the database to see if an assessment has been set for the specific student for the date they have accessed the page. If an assessment has been set the 'Take Test' button is made visible to the user and the level of the assessment is stored as an UserDefaults variable.

The LevelSelctor class also checks how many stars the users has achieved so far and displays them to the user. In addition, when either the 'Practice' or 'Take Test'

button is clicked by the user an UserDefaults variable is used to store the type of button that was selected. This is used in the GeneratedQuestion class to determine whether the user is taking a test or practising mathematic questions. A segue is also activated calling the GeneratedQuestion view when the user selects the button.

GeneratedQuestion

The GeneratedQuestion class contains the algorithms to generate the mathematic questions to be displayed to the user. It uses the UserDefaults variable storing the level to determine the level of questions that should be generated.

The class sets the question label to the correct text and checks if the users answer input is the correct answer for the question. If the UserDefaults variable for the type of button selected in LevelsSelector class was set to practise then the GeneratedQuestion class also provides feedback to the user explaining whether they answered the question correctly or not. This class also keeps track of the users score throughout the 15 generated questions and then stores this score in the database through the connection.php after all 15 questions have been attempted.

10.7. Application implementation

To connect to the database I have used the following code:

```
urlString = [NSString
stringWithFormat:@"http://users.cs.cf.ac.uk/V.R.Green/connection.php?use
rname1=%@&password1=%@", usernames, passwords];

NSData *data = [NSData dataWithContentsOfURL:[NSURL
URLWithString:urlString]];

NSString *result = [[NSString alloc] initWithData:data encoding:
NSUTF8StringEncoding];

result = [result stringByTrimmingCharactersInSet:[NSCharacterSet
whitespaceAndNewlineCharacterSet]];
```

The urlString stores the usernames and passwords variables along with the web address of the connection.php page. The data object accesses this URL and stores the URL's response as an NSData object. To capture the response I have created a NSString result which uses the using [NSString initWithData] function to convert NSData to NSString. The connection.php page determines what should be retrieved from the database or stored through checking which variables have been set in the URL parameters.

I chose to use this method having failed to understand how to implement a POST function through Xcode within the time constraints of my implementation plan. A POST method would have been more appropriate as it is more secure, ensuring that the user details are not stored in browser history and in plain text for those monitoring network traffic.

I have utilised segues within the application to control the flow of the views. These can be generated in the storyboard view by control clicking from an action button to the page that the button should open. In addition they can be called in the code itself.

```
[self performSegueWithIdentifier:@"LoginSuccess" sender:self];
```

To store data which is accessible across all classes I have utilised NSUserDefaults which stores key value pairs.

```
[[NSUserDefaults standardUserDefaults] setObject:username forKey:@"username"];
```

10.8. Application Implementation problems

When developing the application I encountered problems with the connecting to the database through the connection.php page. When the connection.php page was stored locally on my computer the page returned the expected response value however when I moved this to the universities server the NSData object was retrieving unexpected values. The problem appeared to be that the university server was returning the values in XML format and was causing the NSString to contain a whitespace character. To resolve this I added a line of code to remove all the whitespace characters.

```
result = [result  
stringByTrimmingCharactersInSet:[NSCharacterSetwhitespaceAndNewlineCharacterSet]];
```

Another problem that I encountered while implementing the LevelsSelector class was that the computational speed of performing a segue to initialise the GeneratedQuestion view was quicker than retrieving the users level from the NSUserDefaults object. Initially the segue was triggered when the user selected the 'Practise' or 'Take Test' button which meant that the GeneratedQuestion view was loaded before the users level had been set. As a result, the first of the 15 questions remained blank but all others were generated as expected. Through the use of a series of Xcode NSLog commands I was able to establish the series of events and discovered the cause of the problem. To resolve the problem I deleted the direct segue from the button and hard coded it below the code to retrieve the level from the NSUserDefaults object.

```
[[NSUserDefaults standardUserDefaults] setValue:@"test" forKey:@"type"];  
[[NSUserDefaults standardUserDefaults] setInteger:AssessLevel forKey:@"level"];  
[[NSUserDefaults standardUserDefaults] setInteger:AssessID forKey:@"AssessID"];  
NSLog(@"LS type: test ");  
NSLog(@"LS level: %d", AssessLevel);  
[self performSegueWithIdentifier:@"getQuestions" sender:self];
```

11. Phase 2 testing

11.1. User Testing

To test the system with its intended user I re-tested the participant's ability to complete the scenarios set out in the test plan section with the functional implementation of the system.

Website

From observing the teachers attempts to complete the scenarios of the 'Tasks for teacher', outlined in the test plan section, I found that they were able to complete all tasks set out. They had no substantial problems with using the interface and performing the intended functions. Since the design prototype participants were happy with the implementation of the star rewards which enabled them to view how often the students were using the application to practise mathematics.

However some design concerns were identified:

1. Student results view

Teachers with a large amount of students on the system may find it difficult and time consuming to compare all students records as there would be a considerable amount of records to scroll through in the table. For example the participants were concerned that some teachers may have multiple classes and wish only to compare students within the same class. It was suggested that the system would benefit from having the ability to group sets of students based on their class group. In addition the average performance of each group would be of interest to the teacher to compare overall performance across different classes. In addition, participants suggested implementing the ability to order records by level or by the number of stars a student had achieved. They explained that it would be useful for them to be able to quickly recognise a student's ranking with regards to both effort and achievements in comparison to other students.

2. Viewing scheduled assessments

In the initial requirements analysis the ability to view scheduled assessments had been overlooked. When participants scheduled an assessment for a student they would like to have seen what assessments had been scheduled. Without this functionality it was difficult for the teacher to establish which assessments had already been scheduled and when these assessments were due for completion.

3. Labelling of input fields

On the login page of the system the input fields for password and username fields had not been labelled. In the initial design they had been designed to be labelled however this was overlooked during implementation. It did not prove to be a substantial problem for the participants who used their experience with similar login systems to predict what the inputs where. However, inexperienced

users may struggle to understand where to enter their credentials without the labels.

4. Removing a student

Participants advised it would be beneficial to include a list of all registered students that the teacher could select from to remove the student instead of having to know their username.

Application

Participants testing the application completed the scenarios for the 'tasks for students', (outlined above in 'Test Plan' section). Participants for testing the application included two school teachers and two year 6 students. From observing the participants interaction with the application I found that all participants could complete the tasks set out with ease.

From a design perspective participants identified a number of areas which could be improved to increase the overall appeal of the system:

Teacher's suggestions:

1. Labelling of input fields

Similarly to the website implementation, the login page of the system's input fields for password and username fields had not been labelled. These should be labelled to avoid confusion of where students should enter their details.

2. Implement timer for the assessment

To ensure that all assessments are completed within a reasonable amount of time and ensure all students have the same amount of time completed the application would benefit from the inclusion of a time limit to complete the assessment questions.

3. Restrictions on leaving the application

Teacher would like the application to impose restriction which would stop users from accessing the iPad's calculator whilst using the application. They suggested that if a student should leave the application whilst completing the practise or test questions then their score should be submitted and the question set should end. Due to the nature of the question in the application a student could use the iPads built in calculator to gain a higher level than they are capable of and it would be difficult for the teacher to identify students using the calculator without fully observing all students complete each test individually.

4. Application button label

Participant evaluating the website implementation were also given the opportunity to evaluate the application. A participant identified a spelling mistake on the 'Practice' button which needs to be renamed 'Practise'.

Student's suggestions:

1. More appealing design.

Participant explained that they would like to see animation and characters implemented in the design to give the application a more 'Fun' appeal. In addition, participants would like to be able to personalise their profile and have the ability to customise the colour scheme of the system as they are able to do so with other applications they use.

2. Explanation of star rewards.

Participant did not understand the notion of the star rewards and queried what they could be used for. This feature was implemented to enable the teacher to identify students who used the application to practise mathematics regularly. It was suggested that another feature be implemented enabling the users to use the stars or an explanation that the teacher would use the stars to implement their own reward structure.

3. Interactive feature

Participants suggested that the application could implement a feature to challenge their friends. Participants advised that they would be more likely to use the application to practise mathematics if they were able to do so with their friends.

11.2. Test Cases

The functional implementation was tested using the test cases to establish whether all intended functionality had been implemented and whether the system behaved as expected. The outcome of testing the test cases is outlined below:

Test No	Test Name	Pass/Fail	Comments
1	Test whether the teacher can register for an account	Pass	Expected outcome
2	Test whether the teacher can login to the website	Pass	Unexpected outcome – Username & Password not case sensitive
3	Test whether the teacher can add a new student	Pass	Expected outcome
4	Test whether the teacher can remove a student	Pass	Expected outcome
5	Test whether the teacher can schedule an assessment for a single student.	Pass	Expected outcome
6	Test whether the teacher can schedule an assessment for all students.	Pass	Expected outcome
7	Test whether the teacher can view details of all students log in	Pass	Expected outcome

	credentials.		
8	Test whether the teacher can view details of all students' current levels.	Pass	Expected outcome
9	Test whether the teacher can view results of a single student's assessment data.	Pass	Expected outcome
10	Test whether a teacher can log out of the website.	Pass	Expected outcome
11	Test whether a student can log in to the application.	Pass	Unexpected outcome – Username & Password not case sensitive
12	Test whether questions are generated for the practise function and that the answers generated for these questions are correct.	Pass	Expected outcome
13	Test whether questions are generated for the assessment and that the answers generated for these questions are correct.	Pass	Expected outcome
14	Test whether the student can log out of the application.	Pass	Expected outcome
15	Test application feedback for wrong username or password	Pass	Unexpected outcome – This feedback is also displayed when no internet connection exists.
16	Test to ensure a student account can't be used to log in to the website	Pass	Expected outcome

All of the tests for the test cases passed as the systems functionality had been implemented and worked as expected. However, from these test cases I have identified unexpected behaviour in the systems functionality:

1. Test Case 15

The feedback should only be displayed when a user enters the incorrect login credentials. At present this error message is displayed when the device is not connected to the internet. To ensure no confusion for the user this should be amended and another error message should be implemented to explain that the system needs to be connected to the internet to function.

2. Test Case 2 & Test Case 11

Username and Passwords fields are not case sensitive. This should be rectified to ensure the secureness of the users chosen usernames and passwords.

12. Evaluation

This section outlines whether the system has achieved what was set out in the requirements and if it is therefore suitable for its intended use. I have used the results from the testing conducted to determine whether these requirements were met.

From both the user testing and use case testing I have established that the system has met the functional requirements set out. The user testing outlined that users found the system easy to use and that they were able to perform the tasks that the system was designed to facilitate. The implementation has been successful at enabling a teacher to automate assessments and view the results which was the main goal. In addition, teachers are able estimate a student's effort level by how much they practise mathematics using the application. However, a functional aspect which was overlooked, the ability to view scheduled assessments, means that it may be difficult for teachers to utilise the test scheduling feature in advance of the test. Without this feature teachers would need to keep their own records for when assessments have been set which reduces the appeal to use the system.

From a security perspective restrictions have been implemented to avoid unauthorised users obtaining students results which are a core aim of the system. However, a security weakness of the system is that the student's usernames and passwords are sent in plain text from the application via a websites URL to check if the credentials are authorised. This method should be rethought to ensure user credentials are not obtained by unauthorised users.

From a usability perspective, the interface has been deemed easy to use by participants of the testing which was an important requirement for the system. However, some design features should be rethought to increase the appeal to a younger user. In addition, minor amendments should be made to the interface such as rectifying spelling mistakes within the labels to ensure the look is professional and add labels to all input fields to ensure no problems arise with user's ability to login.

Additional requirements have been identified during phase 2 testing which when implemented would further improve the system appeal and functionality.

13. Future Work

I have identified areas of improvement for future versions of the system.

Security

From the testing and evaluation of the current system some weaknesses with security were identified. To rectify this in future version I would change the connection method from the application to the database to a POST method. This would ensure any user details passed to the connection.php page from the application would not be visible to people monitoring network traffic or those who have access to the users browsing history.

In addition I would implement the code to ensure that the password fields for both the website and application login pages are case sensitive. This would increase the security of users' passwords by increasing the number of combinations a user would need to try to access a user's account.

Design

In order to improve the websites design I would implement ordering functionality so that the teachers could order records by the level, stars or date. This would aid teacher's use of the system enabling them to identify higher and lower achieving students easily.

To further improve the website records will be displayed on pages in groups of ten. This will ensure that teacher with a large group of students will not have to wait a considerable amount of time for the page to load.

To improve the application design the identified spelling mistake will be corrected and all login input fields will be labelled.

Core Functionality

To ensure no problems arise with future use of the system future versions of the system will require changing the extension for the connection method from PHP files to the database from MySQL to the later extension MySQLi. This will ensure that the website will function correctly with the latest versions of PHP.

In addition I would add the functionality to view scheduled assessments so that teachers are clear on the scheduled they have already set. This will ensure that they are able to schedule any additional assessments whilst avoiding clashes.

Additional Features

I have identified further functional aspects that could be implemented in the system in future to increase the appeal of the system for its intended users.

The application could include a wider range of subjects and additional content for each subject. In addition question types could be adapted and expanded to be used

with students from all year groups. This would increase the appeal to use the system as the application could be used to assist students learning and assessments in all subjects and for primary school students of all ages.

Implementing a feature where students could spend the stars that they earn will remove the responsibility of implementing a reward system from the teacher. In addition it would encourage users to utilise the application to practise mathematics.

From the user testing students were keen for a feature which they could customise and therefore the system would benefit from implementing a feature where student have their own customisable profile. In addition, a multiplayer feature could be implemented where students can compete against their friends to answer the most questions correctly to include a social aspect to the system and increase its appeal to users.

14. Conclusion

The main goal of the project, as set out in the project plan, was to implement a system which enables year 6 students to practise mathematical questions and be assessed on these questions. The goal included that the system should automatically assign students to their correct ability level in accordance to levels defined by the UK national curriculum. In addition the system should enable the teacher to monitor student's levels and results over a period of time.

In conclusion, I have deemed the project a success as the original goals of the system have been met and the implementation is a good base for which the system can expand on.

The implementation I have implemented a can be used to minimise teachers workload and speed up assessment process of mathematics. In addition, the application can be used to enable users to develop their mathematic ability. The design has been successful in that the system has been deemed easy to use by the people who would use it. However, the design appeal to younger users is minimal and will require further work as they may become bored with the interface and not utilise the application to develop their mathematics. Overall the functionality requirements have been achieved and the system in its current state could be utilised to support teaching and learning.

15. Reflection

Throughout the project I have had an opportunity to develop and learn new skills which I can utilise in future. This section of the report identifies these skills and outlines aspects of the projects approach that I can improve.

One of the main skills that I have developed through this project is learning Xcode. I was able to utilise my knowledge of other programming languages and adapt this to develop an application in iOS. Initially I struggled with the Xcode developer interface and the features specific to Xcode. To overcome this I followed several of Apple's developer tutorials but found these took longer to complete than I had anticipated and were consuming more time than I had allowed for. I then chose to follow a variety of video tutorial's which contained less content which was relevant to the system I was creating. On reflection, it would have been more appropriate to follow the Apple developer tutorials as the sources may not have used the best programming practises and this may be reflected in the code I have developed.

Throughout the project I have struggled with time management due to lack of experience of working on a project of this scale and often underestimating the time it would take to complete each tasks. Initially I hoped to avoid this by ensuring that all of the tasks to be completed were identified at an early stage and in the project plan I had ensured that there were two weeks which were to use if the plan did not go to schedule. On many occasion my project did deviate from the time plan due to underestimation of the tasks, other commitments and a coursework from another module. I feel the deviations from the time plan were managed well as I re-assed the time plan each time a deviation occurred and in the end I had an adequate amount of time left to complete the project.

One aspect of the project that I had struggled with was working on multiple tasks concurrently. Throughout I aimed to complete each task before beginning the next. As a result of this learning the basics of Xcode was left later than planned and on the beginning of implementation phase I struggled to begin any coding as I had yet to learn it. On reflection, if I were to complete a similar project I would follow a structured plan to learn the language alongside the additional task to complete.

The project has given me an opportunity to utilise aspects of the BSc Computer Science content which I hadn't used recently. Before beginning the implementation I felt that I already had the knowledge required in PHP and HTML to implement the system. In reality as I had not used any of these languages since the first year of university I struggled for a short period while I familiarised myself with the syntax and structure. What I have learnt from this is that it is important to exercise my knowledge often to ensure that I do not forget the skills I have developed.

With regards to the approach I used for the testing phase I feel that I should have planned who the participants would be earlier on in the project. This could possibly have enabled me to arranged a visit to a school and assess the system in its intended

environment. The process to include students in testing would require the school to contact students' parents for parental consent and the time I had remaining meant that this approach was not suitable. Instead, I organised the testing with siblings of friends.

In addition, I would also aim to create a functional prototype to test prior to implementation as I found participant gave more detailed and constructive feedback with a functional prototype as they appeared to understand the goals of the system better. In addition, I would try to develop the functional aspects earlier on to ensure the design is suitable which in this project may have brought the lack of a 'view scheduled assessments' function to my attention.

Overall the project has enabled me to further develop my skills in software development and I can use what I have learnt throughout the process to guide my decisions in future projects.

16. References

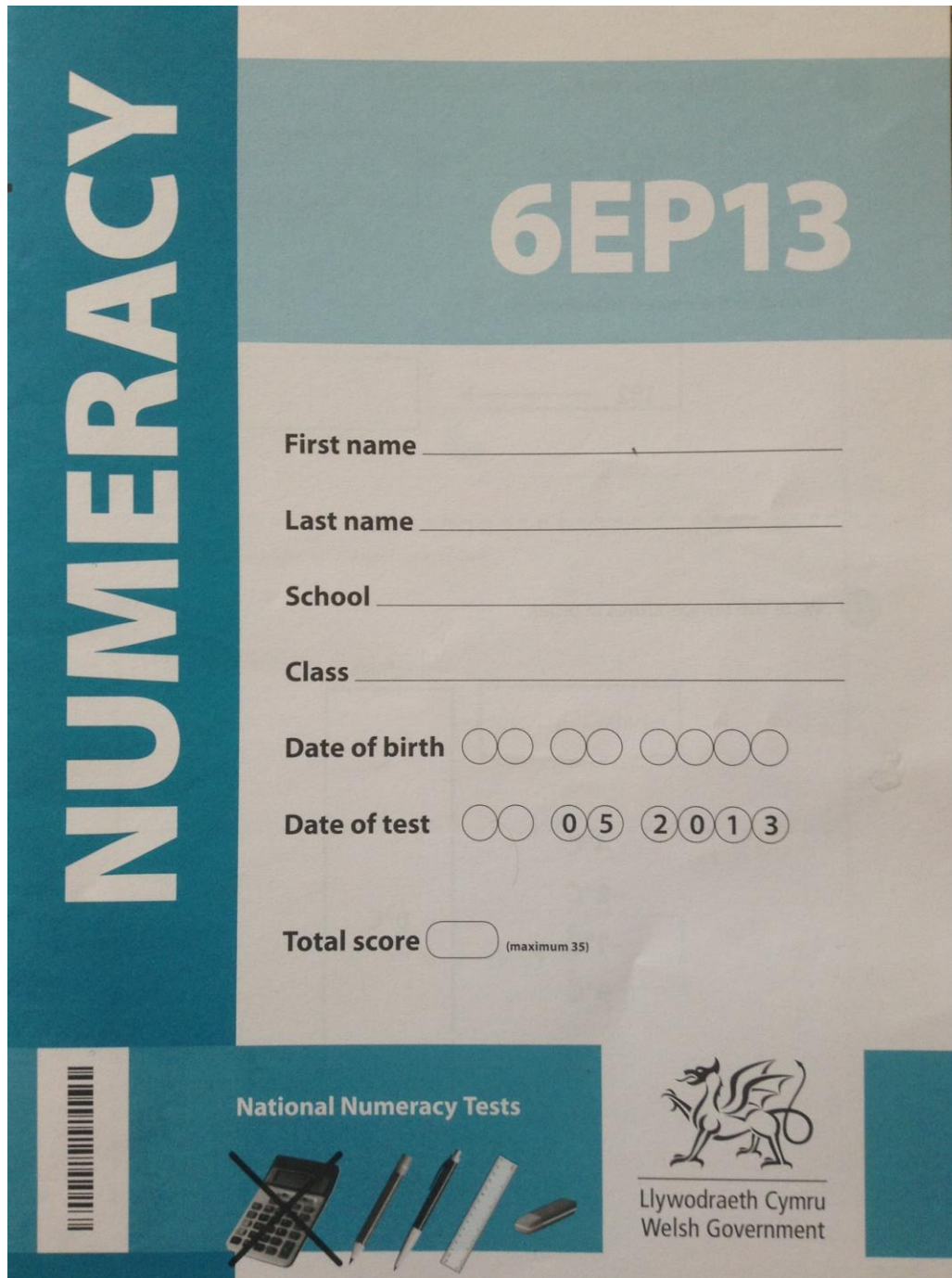
- Balsamiq.com, (2014). *Balsamiq. Rapid, effective and fun wireframing software..* [online] Available at: <http://balsamiq.com/> [Accessed 12 Mar. 2014].
- Chiasson, S. and Gutwin, C. (2014). *Design Principles for Children's Technology*. 1st ed. [PDF] Department of Computer Science, University of Saskatchewan. Available at: http://www.hci.usask.ca/publications/2005/HCI_TR_2005_02_Design.pdf [Accessed 15 Mar. 2014].
- Developer.apple.com, (2014). *iOS Human Interface Guidelines: Design Principles*. [online] Available at: https://developer.apple.com/library/ios/documentation/userexperience/conceptual/MobileHIG/Principles.html#//apple_ref/doc/uid/TP40006556-CH4-SW1 [Accessed 12 Mar. 2014].
- Education.gov.uk, (2000). *Maths programmes of study for key stage 1 and 2 until July 2015 - Publications - GOV.UK*. [online] Available at: <http://www.education.gov.uk/schools/teachingandlearning/curriculum/primary/b00199044/mathematics/ks1/breadth> [Accessed 16 Feb. 2014].
- Guidelines.usability.gov, (2014). *Guidelines / Usability.gov*. [online] Available at: <http://guidelines.usability.gov/> [Accessed 10 Mar. 2014].
- Incerts.org, (2013). *Incerts / Home*. [online] Available at: <http://www.incerts.org> [Accessed 6 Feb. 2014].
- Incerts.org, (2013). *Incerts / Make A Demo*. [online] Available at: <http://www.incerts.org/demo/step1> [Accessed 10 Mar. 2014].
- National Literacy and Numeracy Framework. (2013). 1st ed. [PDF] Welsh Government. Available at: <http://learning.wales.gov.uk/docs/learningwales/publications/130415-lnf-guidance-en.pdf> [Accessed 5 Feb. 2014].
- Nielsen, J. (1995). *10 Heuristics for User Interface Design: Article by Jakob Nielsen*. [online] Nngroup.com. Available at: <http://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed 2 May. 2014].
- Nielsen, J. (1995). *Severity Ratings for Usability Problems: Article by Jakob Nielsen*. [online] Nngroup.com. Available at: <http://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/> [Accessed 17 Apr. 2014].
- Wales.gov.uk, (2012). *Welsh Government/National curriculum - Key Stages 2, 3 and 4*. [online] Available at:

<http://wales.gov.uk/topics/educationandskills/schoolshome/curriculuminwales/a-revisedcurriculumforwales/nationalcurriculum/?lang=en> [Accessed 6 Feb. 2014].

Wales.gov.uk, (2014). *Welsh Government/National Reading and Numeracy Testing*. [online] Available at: <http://wales.gov.uk/topics/educationandskills/schoolshome/literacynumeracy/testing/?lang=en> [Accessed 6 Feb. 2014].

Appendices

Appendix A – National Numeracy Test



The image shows the front cover of the National Numeracy Test 6EP13 form. The cover has a teal and light blue color scheme. On the left, the word "NUMERACY" is written vertically in large white letters. In the top right, "6EP13" is printed in large white letters. Below this, there are several fields for personal information: "First name", "Last name", "School", and "Class", each followed by a horizontal line for writing. Below these are "Date of birth" and "Date of test", each followed by a series of circles for digits. The "Date of test" circles are pre-filled with "052013". At the bottom left, there is a "Total score" field with a circle for the score and the text "(maximum 35)". At the bottom, there is a teal banner with a barcode on the left, the text "National Numeracy Tests" in the center, and the Welsh Government logo and name on the right. The logo is a white dragon on a teal background, and the text "Llywodraeth Cymru" and "Welsh Government" is below it. In the center of the banner, there is an illustration of a calculator, two pens, a ruler, and a mobile phone, all crossed out with a large 'X'.

NUMERACY

6EP13

First name _____

Last name _____

School _____


Class _____

Date of birth ○ ○ ○ ○ ○ ○ ○ ○

Date of test ○ ○ 0 5 2 0 1 3

Total score ○ (maximum 35)

National Numeracy Tests


Llywodraeth Cymru
Welsh Government

1 Round to the nearest **ten**.

182 →

Round to the nearest **hundred**.

182 →

2 Write the temperatures in order.

4°C

-8°C

-1°C

9°C

highest

0°C

lowest

2

3 Vehicles in a car park

Key: ● stands for 4 vehicles

Bus	●
Lorry	● ●
Car	<div style="border: 1px solid #00a0e3; width: 100px; height: 20px;"></div>
Bicycle	●

There are **14 cars**. Complete the chart.

How many vehicles altogether?

vehicles

4

12 - 3 = 5 +

3

7 $90\text{mm} =$ cm

$400\text{cm} =$ m

1m


1m

1m

TOTAL 7m

8 How many envelopes in 32 packs?

One pack




200 envelopes

envelopes

9 Work out **half** of 6.3kg.

kg

5




Bethan eats **2** pieces of this pizza.
Owen eats **5** pieces.

What **percentage** is left?

%

1m

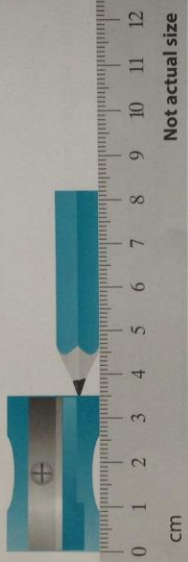
6



Work out the length of the **pencil**.

cm

1m

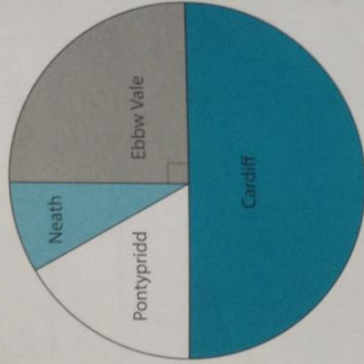


cm

Not actual size

- 10 Alun asked his friends which of four rugby teams they liked best.

The pie chart shows the results.



6 people chose **Ebbw Vale**.

How many chose Pontypridd?

How many people altogether?

11 $5009 \times 8 \div 8 =$

12 Write five hundred thousand and twenty in figures.

What number is **100 more** than forty thousand?

What number is **1000 less** than 1 million?

13 Write in order.

2.5 2 1.83 2.15

smallest

largest

TOTAL

14 Train timetable:

Swansea	06:47	07:45	08:57	09:50
Cardiff	07:50	08:50	09:55	10:55
Hereford	08:51	09:51	10:56	11:56
Shrewsbury	09:43	10:48	11:48	12:53
Stockport	10:58	11:58	12:58	13:58
Manchester	11:15	12:15	13:15	14:15

Bev needs to get from Hereford to Manchester **before 1pm**.

What time is the latest train she can get from **Hereford**?

How long does it take to get from Swansea to Cardiff on the **08:57** train?

15 Gwen spent £19.67 to make biscuits.

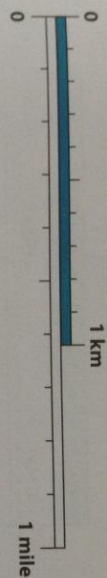
She made £34.25 selling the biscuits.

How much **profit** did she make?

£

8

16



Use the scale to estimate:

0.5 km is about

miles

17

Work out 22×43

18

1000g of grapes cost £4

How much do 300g of grapes cost?

£

9

- 19 Harri's bank account:

Date	Paid out	Paid in	Balance
10 May		150.00	150.00
12 May	8.79 25.50 5.99		141.21 115.71 109.72
13 May		50.00	159.72

Altogether, how much money was **paid out** on 12 May?

£

20 $4600 \div 20 =$

$275 \div 25 =$

10

- 21 Write $\frac{2}{5}$ as a decimal.

- 22 Here are the heights of five sunflowers (in cm).

23.2 22.7 25.5 24.8 22.5

What is the **range** of the heights?

cm

- 23 Work out 5% of £120



£

10% of £80 = 20% of

£

11

Appendix B – Level Descriptors

 Llanedeyrn Primary School Mathematics Assessment 											
Name:	<table border="1" style="display: inline-table;"><tr><td>Y1</td><td>Y2</td><td>Y3</td><td>Y4</td><td>Y5</td><td>Y6</td></tr></table>	Y1	Y2	Y3	Y4	Y5	Y6	<table border="1" style="display: inline-table;"><tr><td>Autumn</td><td>Spring</td><td>Summer</td></tr></table>	Autumn	Spring	Summer
Y1	Y2	Y3	Y4	Y5	Y6						
Autumn	Spring	Summer									
<div style="display: flex; justify-content: space-between;"><div><input type="checkbox"/> Formative Assessment (Best Fit Level =)</div><div><input type="checkbox"/> Focused Assessment Activity: Number (Abacus) (Best Fit Level =)</div></div>											
<div style="border: 1px solid black; padding: 5px;"><p><u>Level 1</u></p><p>Pupils use mathematics as an integral part of classroom activities. They represent their work with objects or pictures and discuss it. They count, order, add and subtract numbers when solving problems involving up to 10 objects, and can read and write the numbers involved. They count on and back in steps of different sizes and from different numbers. They measure and order objects using direct comparison, and order events. They are aware of the value of different coins. They use everyday language to compare and to describe positions and properties of regular shapes. They recognise, use and make repeating patterns. They sort and classify objects, demonstrating the criterion they have used.</p><div style="border: 1px solid black; padding: 5px; min-height: 40px;"><u>Comments:</u></div></div>											
<div style="border: 1px solid black; padding: 5px;"><p><u>Level 2</u></p><p>Pupils talk about their work using familiar mathematical language, and represent it using symbols and simple diagrams. They count sets of objects reliably, and use mental recall of number facts to 10 to add or subtract larger numbers. They order numbers up to 100. They choose the appropriate number operation when solving addition or subtraction problems. They identify and use halves quarters in practical situations. They recognise sequences of numbers. They use everyday non-standard and standard units to measure length and mass. They distinguish between straight and turning movements; recognise half-turns and quarter-turns and right angles in turns. They sort objects and classify them using more than one criterion. When they have gathered information, they record their results in simple lists, tables, diagrams and block graphs.</p><div style="border: 1px solid black; padding: 5px; min-height: 100px;"><u>Comments:</u></div></div>											



Llanedeyrn Primary School

Mathematics Assessment



Name:

Y1 Y2 Y3 Y4 Y5 Y6

Autumn Spring Summer

☐ Formative Assessment
(Best Fit Level =)

☐ Focused Assessment Activity: Number (Abacus)
(Best Fit Level =) (app. pupil marks fielded)

Level 3

Pupils organise their work, check results and try different approaches. They talk about and explain their work. They use and interpret mathematical symbols and diagrams. They find particular examples that satisfy a general statement. They use place value in numbers up to a 1000 to make approximations. They use decimal notation in recording money, and recognise negative numbers in the context of temperature. They develop further mental strategies for adding and subtracting numbers with at least two digits. They use mental recall of the 2, 3, 4, 5 and 10 multiplication tables in solving whole-number problems involving multiplication and division, including those giving rise to remainders. They use standard units of length, capacity, mass and time. They classify shape in various ways. They extract and interpret information presented in simple tables and lists, and construct and interpret bar charts and pictograms.

Comments:

Level 4

Pupils develop their own strategies for solving problems, and present information and results systematically. They search for a solution by trying out ideas of their own. They use their understanding of place value to multiply and divide whole numbers by 10 and 100. They use a variety of mental and written methods for computation, including recall of multiplication facts up to 10×10 . They add and subtract decimals to two places. They check their results are reasonable by considering the context or the size of the numbers. They use simple fractions and percentages to describe approximate parts of a whole. They recognise and describe number patterns and relationships and use simple formulae expressed in words. They use their knowledge of shape to make 3-D mathematical models, draw common 2-D shapes in different orientations on grids, and reflect simple shapes in a mirror line. They choose and use suitable units and instruments, reading with appropriate accuracy, numbers on a range of measuring instruments. They find perimeters of shapes, areas by counting squares, and volumes by counting cubes. They use and interpret co-ordinates in the first quadrant. They collect discrete data, group data where appropriate, and use the mode and median as characteristics of a set of data. They draw and interpret frequency diagrams and construct and interpret simple line graphs. They understand and use simple vocabulary associated with probability.

Comments:



Llanedeyrn Primary School

Mathematics Assessment



Name:

Y1	Y2	Y3	Y4	Y5	Y6
----	----	----	----	----	----

Autumn	Spring	Summer
--------	--------	--------

☐ Formative Assessment
(Best Fit Level =)

☐ Focused Assessment Activity: _____
(Best Fit Level =)

Level 5

Pupils identify and obtain information to solve problems, and check whether their results are sensible in the context of the problem.

They describe situations mathematically using symbols, words and diagrams and draw their own conclusions, explaining their reasoning.

They make general statements of their own, based on available evidence.

They use their understanding of place value to multiply and divide whole numbers and decimals.

They order, add and subtract negative numbers.

They check their solutions by applying inverse operations or estimating using approximations.

They calculate fractional or percentage parts of quantities and measurements.

They construct and use simple formulae involving one or two operations.

They use co-ordinates in all four quadrants.

They measure and draw angles to the nearest degree.

They recognise, identify and describe all the symmetries of 2-D shapes.

They convert one metric unit to another and know the rough metric equivalents of imperial units in daily use. They make sensible estimates of a range of everyday measures.

They find areas of rectangles and triangles and volumes of cuboids.

They read scales on maps, plans and graphs.

They use the mean of discrete data and compare two simple distributions.

They interpret graphs, diagrams and pie charts.

They use the probability scale from 0 to 1, and appreciate that different outcomes may result from repeating an experiment.

Comments:

Appendix C – Use case flow of events

Register for teacher account

This use-case allows a user to set up a teacher account on the system enabling them to log-in to the website, add and remove students and view student data.

1. Flow of events

1.1 Basic flow of events

1. Select 'Register'

The website displays the options available to the teacher. The teacher selects 'Register'.

2. Enter teacher details.

The teacher chooses a username and password and enters this into the form.

3. Save teacher details.

The teacher saves the details to the database by selecting 'Save'. The teacher's username is compared with Teacher ID records in the database. If the teacher ID doesn't exist in the database details of teacher's username and password are added to the database.

1.2 Alternative flow of events

1.2.1 User already exists

If in step 2 of the use case the teacher enters a username which already exists in the database an error message will be displayed. The use case returns to step 2.

2. Pre-requisites

There are no pre-requisites for this use case.

3. Post-condition

Following successful completion of the use case the teacher will have a valid teacher ID and password which will enable them to log-in to the website.

Log in to website

This use case enables the user to access their account on the website.

1. Flow of events

1.1 Basic flow of events

1. Select 'Log in'.

The website displays the options available to the teacher. The teacher selects 'Log in'.

2. Enter user details

The teacher enters their username and password.

3. Check user credentials

The teacher input is validated against what is stored in the database. If both the username and password match a record in the database the teacher is logged in.

4. View user account

On successful log in the user will be redirected to their account.

1.2 Alternative flow of events

1.2.1 Unidentified teacher

In step 2 if the teacher username and/or password do not match with a record in the database an error message is displayed and the use case ends.

2. Pre-requisites

The teacher will need to complete the 'Register for teacher account' use case and have created a Teacher ID and password which has been stored in the database.

3. Post-condition

Following successful completion of the use case the teacher will have accessed their account on the website and will be re-directed to the main menu of their account.

Add new student

This use case allows a teacher to register a new student on to the system.

1. Flow of events

1.1 Basic flow of events

1. Select 'Add new student'.

The website displays the options available to the teacher. The teacher selects 'Add new student'.

2. Enter student details.

The teacher enters the students name, and creates a user ID and password for the student.

3. Save student details.

The teacher saves the details to the database by selecting 'Save'. The student ID entered is compared with Student ID records in the database. If the Student ID doesn't exist in the database details of student name, Student ID and password are added to the database.

1.2 Alternative flow of events

1.2.1 Log out

A teacher can log out of the website at any time. No input data is stored in the database and all input is gone on next log in. The use case ends.

1.2.2 User already exists

If in step 2 of the use case the teacher enters a student ID which already exists in the database an error message will be displayed. The use case returns to step 2.

2. Pre-requisites

The teacher will need to complete the 'Log in to website' use case.

3. Post-condition

Following successful completion of the use case a new student account will have been created.

Remove Student

This use case allows the teacher to remove students from the database and therefore from the system.

1. Flow of events

1.1 Basic flow of events

1. Select 'Remove student'.

The website displays the options available to the teacher. The teacher selects 'Remove student'.

2. Enter student ID.

The teacher enters the students ID of the student they wish to remove.

3. Remove a student from database.

The teacher selects the 'Remove student' button. The student ID entered is compared with Student ID records in the database. If the Student ID exists in the database details of student name, Student ID and password are removed from the database.

1.2 Alternative flow of events

1.2.1 User doesn't exists

If in step 2 of the use case the teacher enters a student ID which doesn't exists in the database an error message will be displayed. The use case ends.

2. Pre-requisites

The teacher will need to have completed the 'Log in to website' use case. The teacher will also need to have completed the 'Add new student' use case and as a result the database will contain one or more student account details.

3. Post-condition

Following successful completion of the use case the specified student account will have been deleted from the database.

View all Student records

This use case allows the teacher to view the level an individual student has achieved from the assessments they have completed. They will see the date an assessment was completed and the result for that assessment.

1. Flow of events

1.1 Basic flow of events

1. Select 'view all student records'.

The website displays the options available to the teacher. The teacher selects 'view individual student record'.

2. View student data.

Details of all students' assessments levels and assessment dates are retrieved from the database and displayed.

1.2 Alternative flow of events

1.2.1 Log out

A teacher can log out of the website at any time. The use case ends.

2. Pre-requisites

The teacher will need to complete the 'Log in to website' use case. The student the teacher wishes to view results of must have been added to the database, use case 'add new student', and the account of those students must have not been deleted.

3. Post-condition

Following successful completion of the use case the teacher will be presented with all students' assessment results.

Log in to application

This use case enables the student user to access their account on the application.

1. Flow of events

1.1 Basic flow of events

1. Select 'Log in'.

The application displays the options available to the user. The user selects 'Log in'.

2. Enter user details

The student enters their username and password.

3. Check user credentials

The user input is validated against what is stored in the database. If both the username and password match a record in the database the student is logged in.

4. View user account

On successful log in the user will be redirected to their account.

1.2 Alternative flow of events

1.2.1 Unidentified student

In step 2 if the student username and/or password do not match with a record in the database an error message is displayed and the use case ends.

2. Pre-requisites

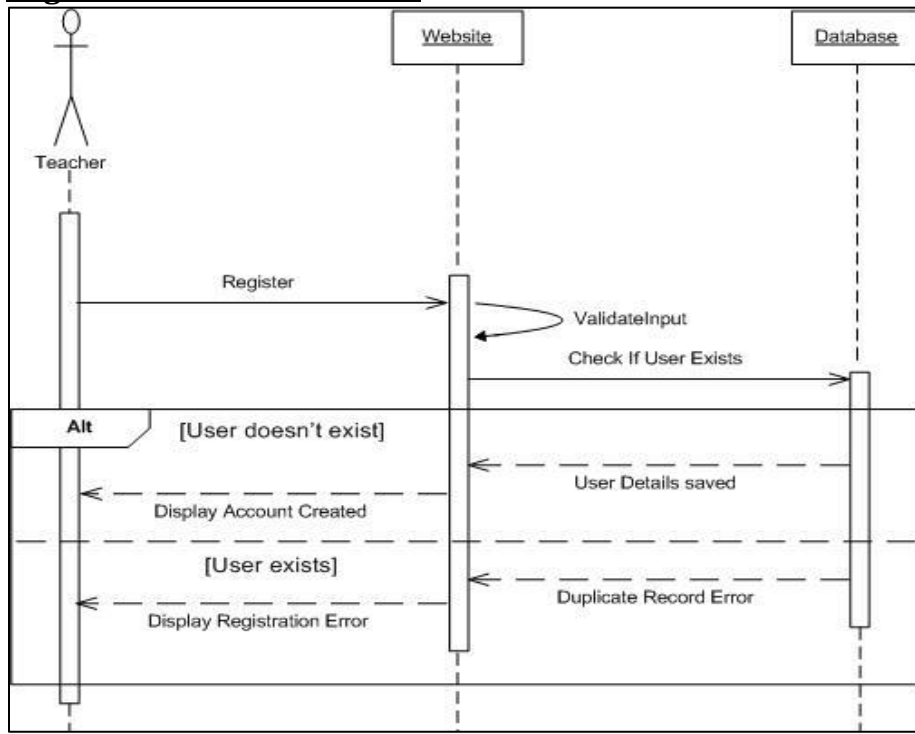
The teacher will need to complete the 'Add new student' use case and have created a Student ID and password for the student which has been stored in the database.

3. Post-condition

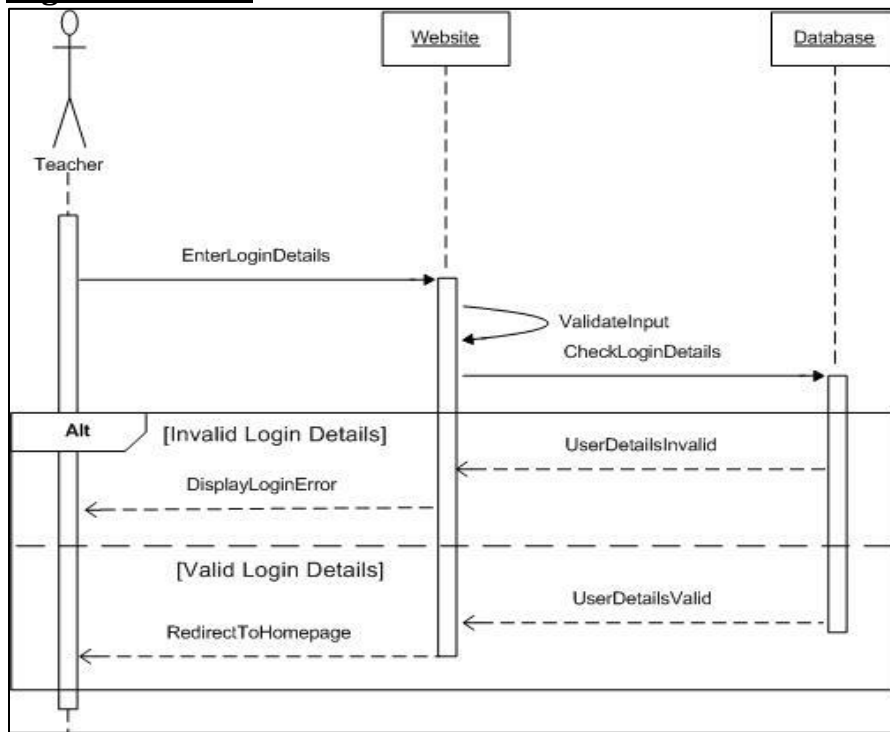
Following successful completion of the use case the student will have accessed their account on the application and will be re-directed to the main menu of their account.

Appendix D – Sequence Diagrams

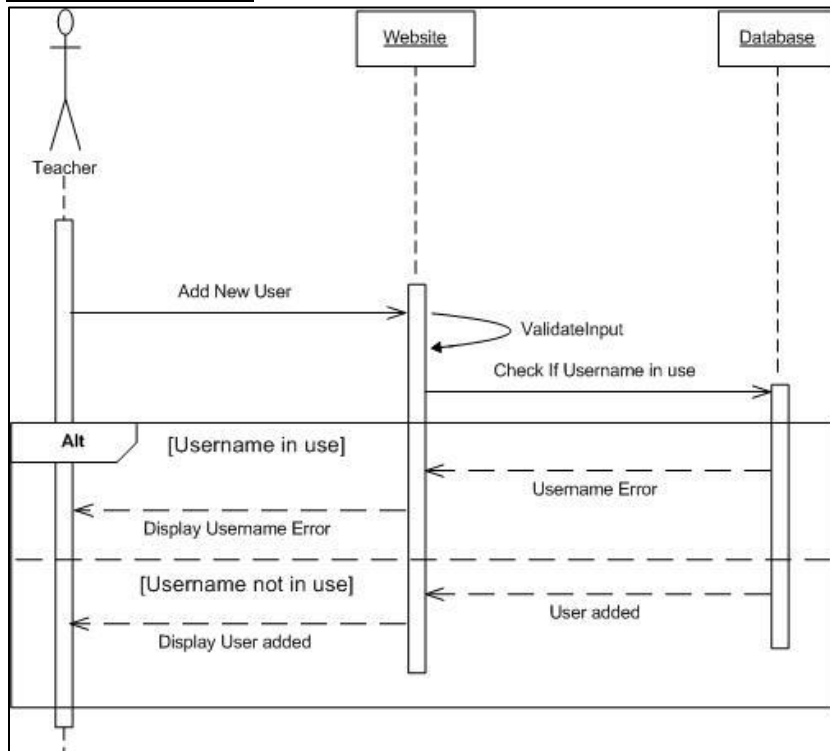
Register for teacher account



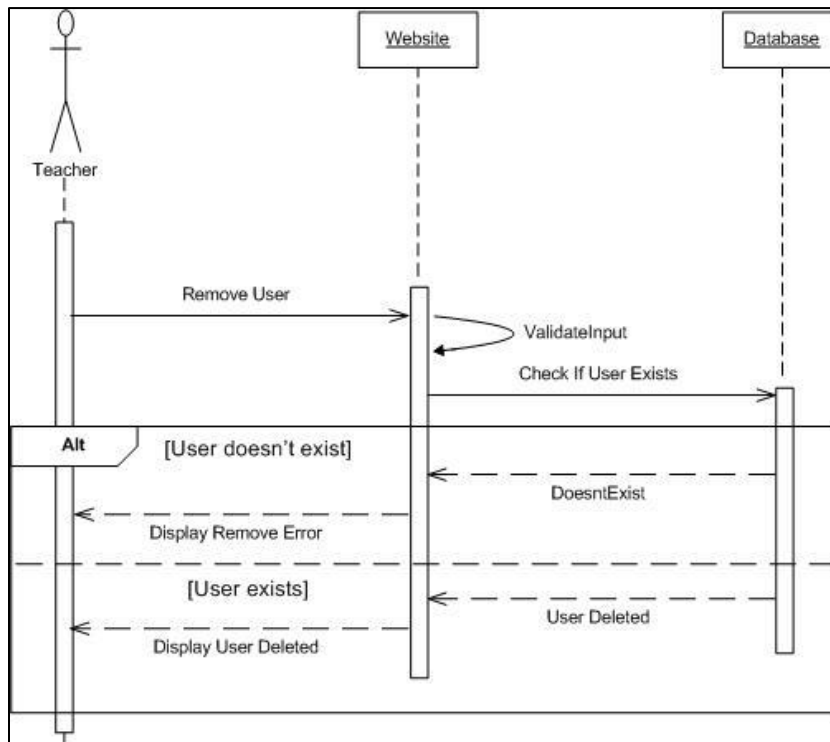
Login to Website



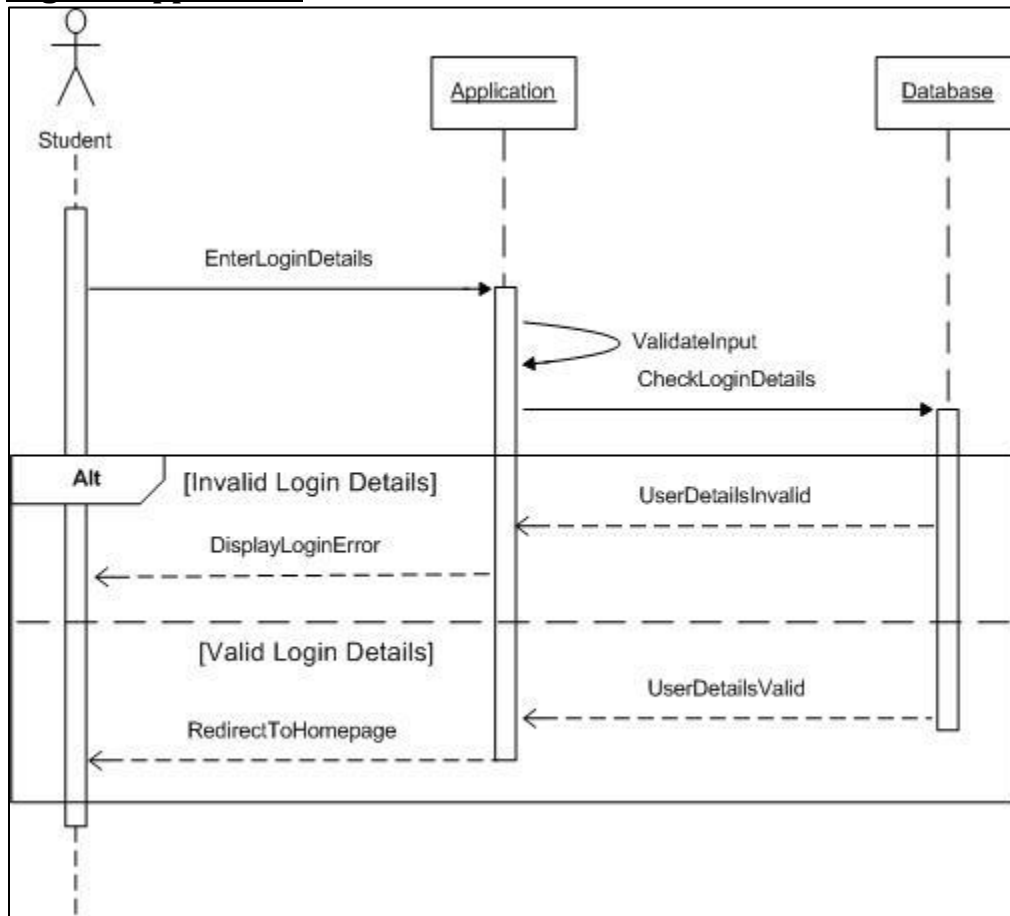
Add new student



Remove User



Login to Application



Appendix E - Algorithms to generate questions

```
Level2() {  
  Question number = +1;  
  int X = random number between 0 and 100;  
  
  for QuestionNumber 1 to 3 do  
    y = random number between 0 and 100;  
    Question = What is larger X or Y?;  
    If X > Y  
      Answer = X;  
    End if  
    If Y > X  
      Answer = Y;  
    End if  
  end for  
  
  for QuestionNumber 4 to 7 do  
    Question = What number is next after X?;  
    Answer = X + 1;  
  end for  
  
  for QuestionNumber 8 to 11 do  
    int y = 11 - x;  
    int z = random number between 0 and y;  
    Question = What is X + Y?;  
    Answer = X + Y;  
  end for  
  
  for QuestionNumber 12 to 15 do  
    y = X + 1  
    z = random number between 0 and y;  
    Question = What is X - Y?;  
    Answer = X - Y;  
  end for  
}
```

```
Level3() {  
  Question number = +1;  
  
  for QuestionNumber 1 to 3 do  
    int y = 2;  
  end for
```

```

for QuestionNumber 4 to 6 do
    int y = 3;
end for

for QuestionNumber 7 to 9 do
    int y = 4;
end for

for QuestionNumber 10 to 12 do
    int y = 5;
end for

for QuestionNumber 13 to 15 do
    int y = 10;
end for

int x = random number between 0 and 10;
Question = What is  $X \times Y$ ?
Answer =  $X * Y$ ;

}

```

```

Level4( ){
    Question number =+1;

```

```

for QuestionNumber 1 to 3 do
    int x = random number between 0 and 10;
    int y = random number between 0 and 10;
    Question = What is  $X \times Y$ ?;
    Answer =  $X * Y$ ;
end for

for QuestionNumber 4 to 6 do
    float x = random number between 0 and 1000;
    if x < 10 do
        x = x + 10;
    end if
    float y = 10;
    Question = What is  $x/y$ ?;
    Answer =  $x / y$ ;
end for

for QuestionNumber 7 to 9 do
    float x = random number between 0 and 1000;
    if x < 100 do
        x = x + 100;

```

```

        end if
        float y = 100;
        Question = What is x/y?;
        Answer = x / y;
    end for

    for QuestionNumber 10 to 12 do
        float x = random number between 0.00 and 9.99;
        float y = random number between 0.00 and 9.99;
        Question = What is x + y?;
        Answer = x + y;

    end for

    for QuestionNumber 13 to 15 do
        float x = random number between 0.00 and 9.99;
        float y = random number between 0.00 and 9.99;
        Question = What is x - y?;
        Answer = x - y;
    end for

}

```

Level5(){

Question number +=1;

```

    for QuestionNumber 1 to 4 do
        float x = random number between 0.00 and 9.99;
        float y = random number between 0.00 and 9.99;
        Question = What is X × Y?;
        Answer = X * Y;
    end for

```

```

    for QuestionNumber 5 to 8 do
        float X = random number between 0.00 and 49.99;
        Answer = random number between 0.00 and 49.99;
        float y = X * Answer;
        Question = What is Y/X?;
    end for

```

```

    for QuestionNumber 9 to 11 do
        float x = random negative number between 0.00 and 10.99;
        float y = random number between 0.00 and 10.99;
        Question = What is X + Y?;
        Answer = X + Y;
    end for

```

```
for QuestionNumber 12 to 15 do
    float x = random negative number between 0.00 and 10.99;
    float y = random number between 0.00 and 10.99;
    Question = What is x - y?;
    Answer = x - y;

end for

}
```

Appendix F – Test cases

Test 1	Test whether the teacher can register for an account
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions No preconditions for this test.	
Process: <ol style="list-style-type: none">1. Select 'register' button from the navigation bar on the website.2. Input username "TestUser" in the username field.3. Input email "TestUser@cardiff.ac.uk" in the field 'email address'4. Input email "TestUser@cardiff.ac.uk" in the field 'confirm email address'5. Input "password1" in 'Password' field.6. Input "password1" in 'Confirm Password' field.7. Select 'Register' button.8. Check that the above details are stored in the teacher accounts table of the database	
Expected Outcome The following record should be stored in the accounts table of the database: TeachUsername: 'TestUser', TeachEmail: 'TestUser@Cardiff.ac.uk', TeachPass: 'password1'	
Actual Outcome: Pass / Fail	

Test 2	Test whether the teacher can login to the website
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher must be registered, (Database must contain the test data outlined in section 7.1).	
Process: <ol style="list-style-type: none">1. Select 'Login' button from the navigation bar on the website.2. Input username "Greenvr" in the username field.3. Input "Rach1410" in 'Password' field.4. Select 'Login' button.5. Check that the page has redirected to the 'mainmenu.php' page.	
Expected Outcome The home page of the user is displayed.	
Actual Outcome: Pass / Fail	

Test 3	Test whether the teacher can add a new student
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher logged in as Bob123.	
Process: <ol style="list-style-type: none"> 6. Select 'Add Student' button from the navigation bar on the website. 7. Input username "Student1" in the username field. 8. Input name "Fred Jones" in the field 'Student Name' 9. Input "abc1" in 'Password' field. 10. Input "abc1" in 'Confirm Password' field. 11. Select 'Add Student' button. 12. Check that the above details are stored in the student accounts table of the database 	
Expected Outcome The following record should be stored in the accounts table of the database: StudentUsername: 'Student1', StudName: 'Fred Jones', StudPass: 'abc1', TeachID: 1, CurrentLvl: 0, Stars:0	
Actual Outcome: Pass / Fail	

Test 4	Test whether the teacher can remove a student
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher logged in as Bob123. Database must contain the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none"> 1. Select 'Remove Student' button from the navigation bar on the website. 2. Input username "LJones" in the username field. 3. Select 'Remove Student' button. 4. A message is displayed asking whether you wish to remove the student. 5. Check the student name is 'Lisa Jones'. 6. Select 'Yes' button. 7. Check that students account table in the database and ensure there is no record containing student Username 'LJones'. 	
Expected Outcome There should be no record in the student accounts table of the database where studentUserName is 'LJones'.	

Actual Outcome: Pass / Fail

Test 5	Test whether the teacher can schedule an assessment for a single student.
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher must be logged in as GreenVR. Database must contain the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none">1. Select 'Set Assessment' button from the navigation bar on the website.2. Select the date 04/02/2014 for the 'Start date' field.3. Select the date 07/02/2014 for the 'End date' field.4. Select assessment Level 4 from drop down list.5. Select 'Individual Student' radio button.6. Input username "JFields" in the username field.7. Select 'Set Assessment' button.	
Expected Outcome The following record should be stored in the assessments table of the database: Stud username: JFields, startDate:2014-02-04, endDate:2014-02-07, assessLvl: 4	
Actual Outcome: Pass / Fail	

Test 6	Test whether the teacher can schedule an assessment for all students.
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher must be logged in as GreenVR. Database must contain the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none">1. Select 'Set Assessment' button from the navigation bar on the website.2. Select the date 09/02/2014 for the 'Start date' field.3. Select the date 12/02/2014 for the 'End date' field.4. Select assessment Level 2 from drop down list.5. Select 'All Students' radio button.6. Select 'Set Assessment' button.	
Expected Outcome The following records should be stored in the assessments table of the database: Stud username: JFields, startDate:2014-02-09, endDate:2014-02-12, assessLvl: 2 Stud username: RJops, startDate:2014-02-09, endDate:2014-02-12, assessLvl: 2 Stud username: LBishop, startDate:2014-02-09, endDate:2014-02-12, assessLvl: 2	

Actual Outcome: Pass / Fail	
------------------------------------	--

Test 7	Test whether the teacher can view details of all students log in credentials.	
Environment	Windows 7	
Browser	Google Chrome	
Pre-Conditions Teacher must be logged in as GreenVR. Database must contain the test data outlined in section 7.1.		
Process: <div><div>1.</div>Select 'Student Records' from the navigation menu.</div> <div><div>2.</div>Select 'View students login information' radio button.</div> <div><div>3.</div>Select 'View Records' button.</div>		
Expected Outcome The following displayed in a table:		
Student Name	Username	Password
Jim Fields	JFields	Pet3
Rhys Jopson	RJops	Alffie123??
Lily Sue	LBishop	lily
Actual Outcome: Pass / Fail		

Test 8	Test whether the teacher can view details of all students current levels.		
Environment	Windows 7		
Browser	Google Chrome		
Pre-Conditions Teacher must be logged in as GreenVR. Database must contain the test data outlined in section 7.1.			
Process: <div>1. Select 'Student Records' from the navigation menu.</div> <div>2. Select 'View all students current levels' radio button.</div> <div>3. Select 'View Records' button.</div>			
Expected Outcome The following displayed in a table:			
Username	Student Name	Current Level	Stars
JFields	Jim Fields	3	15
RJops	Rhys Jopson	3	4
LBishop	Lily Sue	2	0

Actual Outcome: Pass / Fail

Test 9	Test whether the teacher can view results of a single student's assessment data.		
Environment	Windows 7		
Browser	Google Chrome		
Pre-Conditions Teacher must be logged in as GreenVR. Database must contain the test data outlined in section 7.1.			
Process: <div>1. Select 'Student Records' from the navigation menu.</div> <div>2. Select 'View results of individual's assessment' radio button.</div> <div>3. Input username 'LBishop' in text box.</div> <div>4. Select 'View Records' button.</div>			
Expected Outcome The following displayed in a table:			
Assessment Date	Assessment Level	Mark	Achieved Level
2014-04-05	1	15	1
Actual Outcome: Pass / Fail			

Test 10	Test whether a teacher can log out of the website.
Environment	Windows 7
Browser	Google Chrome
Pre-Conditions Teacher must be logged in	
Process: 1. Select 'Logout' button from navigation bar.	
Expected Outcome User should be redirected to the 'homepage2FYP.php' page.	
Actual Outcome: Pass / Fail	

Test 11	Test whether a student can log in to the application.
Environment	Apple iOS 7
Browser	N/A
Pre-Conditions	
Database must be populated with the test data outlined in section 7.1.	

Process: <ol style="list-style-type: none"> 1. Insert 'JFields' in the username textbox. 2. Insert 'Pet3' in the password textbox. 3. Select 'Login' button. 	
Expected Outcome User should be redirected to main menu. The username 'JFields' should be displayed at the top of the screen.	
Actual Outcome: Pass / Fail	

Test 12	Test whether questions are generated for the practise function and that the answers generated for these questions are correct.
Environment	Apple iOS 7
Browser	N/A
Pre-Conditions User must be logged in as JFields. Database must be populated with the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none"> 1. Select 'Practise Mathematics' button. 2. A level 3 question should be generated. 3. Insert the answer to the question in the textbox labelled "Enter answer below". 4. Select 'Check Answer'. 5. Check feedback is correct. 6. Repeat steps 3 to 6 for all 15 questions. 	
Expected Outcome The score displayed should correspond to the number of correctly answered questions. This score should be added to the user's stars in the studaccount table of the database.	
Actual Outcome: Pass / Fail	

Test 13	Test whether questions are generated for the assessment and that the answers generated for these questions are correct.
Environment	Apple iOS 7
Browser	N/A
Pre-Conditions User must be logged in as JFields. Database must be populated with the test data outlined in section 7.1. Test must be conducted between the dates 2014-04-13 and 2014-04-20.	
Process: <ol style="list-style-type: none"> 1. Select 'Practise Mathematics' button. 2. A level 4 question should be generated. 	

<ol style="list-style-type: none"> 3. Insert the answer to the question in the textbox labelled “Enter answer below”. 4. Select ‘Check Answer’. 5. Repeat steps 3 to 5 for all 15 questions.
Expected Outcome The score displayed should correspond to the number of correctly answered questions. Assessment details should be store in the assessresults table of the database. If the score is more than 10 the student’s current level should be updated to level 4 in the studaccount table of the database.
Actual Outcome: Pass / Fail

Test 14	Test whether the student can log out of the application.
Environment	Apple iOS 7
Browser	N/A
Pre-Conditions Student must be logged in	
Process: <ol style="list-style-type: none"> 1. Select ‘Logout’ button on top right of the screen. 	
Expected Outcome Application should redirect to the login page.	
Actual Outcome: Pass / Fail	

Test 15	Test application feedback for wrong username or password
Environment	Apple iOS 7
Browser	N/A
Pre-Conditions Database must be populated with the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none"> 1. Enter ‘Jfox’ in the username field. 2. Enter ‘abc’ in the password field. 3. Select ‘Login’ button. 	
Expected Outcome An error message should be displayed informing the user of incorrect password and/or username.	
Actual Outcome: Pass / Fail	

Test 16	Test to ensure a student account can’t be used to log in to the website
Environment	Windows 7

Browser	Google Chrome
Pre-Conditions Database must be populated with the test data outlined in section 7.1.	
Process: <ol style="list-style-type: none"> 1. Enter 'JFields' in the username field. 2. Enter 'pet3' in the password field. 3. Select 'Login' button. 	
Expected Outcome An error message should be displayed informing the user of incorrect password and/or username.	
Actual Outcome: Pass / Fail	

Appendix G – Implementation code

The implementation code for the system has been uploaded with this report.

For full source code for connection between application and database implementation see the following file:

C1120237Project/connection.php

For full source code for application implementation see the following files:

C1120237Project/xcode/LevelSelector.h

C1120237Project/xcode/LevelSelector.m

C1120237Project/xcode/GeneratedQuestion.h

C1120237Project/xcode/GeneratedQuestion.m

C1120237Project/xcode/Navigation/LoginPageController.h

C1120237Project/xcode/Navigation/LoginPageController.m

For full source code for website implementation see the following files:

C1120237Project/WebendFYP/AddError.php

C1120237Project/WebendFYP/AddNewStudent.php

C1120237Project/WebendFYP/checkLogin.php

C1120237Project/WebendFYP/delete.php

C1120237Project/WebendFYP/homepage2FYP.php

C1120237Project/WebendFYP/Login.php

C1120237Project/WebendFYP/mainmenu.php

C1120237Project/WebendFYP/processAddStudent.php

C1120237Project/WebendFYP/processAddTeacher.php

C1120237Project/WebendFYP/processRemoveStudent.php

C1120237Project/WebendFYP/RegError.php

C1120237Project/WebendFYP/Register.php

C1120237Project/WebendFYP/RemoveStudent.php

C1120237Project/WebendFYP/set.php

C1120237Project/WebendFYP/setAssessment.php

C1120237Project/WebendFYP/studentRecords.php

C1120237Project/WebendFYP/studRegError.php

C1120237Project/WebendFYP/stylessheetFYP.css

C1120237Project/WebendFYP/successfullAdd.php

C1120237Project/WebendFYP/successfullReg.php

C1120237Project/WebendFYP/view.php