



CM3203 – COMPUTER SCIENCE (BSc) DISSERTATION

SPELLING PRACTICE

Tailored towards students with dyslexia.

AUTHOR : SAMANTHA HUGHES - 1936604

SUPERVISOR : HELEN PHILLIPS

MODERATOR : YULIA CHERDANTSEVA

CLIENT : MATTHEW MOLOUGHNEY

SCHOOL OF COMPUTER SCIENCE AND INFORMATICS, CARDIFF UNIVERSITY

13/05/2022

Abstract

It is believed that 1 in 6 adults in the UK have some form of dyslexia, and in a survey that has been conducted, it has been found that 16% of UK adults are deemed to be ‘functionally illiterate’. This project has a goal to create a spelling application which both educates people using the feedback and aiding people who have dyslexia to successfully use the application. Through this report, background research is conducted on how to aid people who have dyslexia, and improve the spellings for those people who have dyslexia. Also, a survey was conducted to research and gain knowledge on the best fonts, background, feedback and format of the spelling test, that real users would prefer. Discussed in this report is the XCode spelling test application that was coded for this proposal and within it’s test cases showing how the application prototype completes its tasks. The result and evaluation of the application shows a working prototype that fills the requirement and aims of the proposal. To conclude, this solution would aid and increase the literacy potential of the adult population in the UK, both with and without dyslexia.

Contents

1	Introduction	1
1.1	Aims and/or Goals	1
1.2	Beneficiaries	1
1.3	Scope of the project	1
1.4	Approach	1
1.5	Assumptions	1
1.6	Summary	2
2	Background	3
2.1	Dyslexia	3
2.1.1	Dyslexia 'Fonts'	3
2.1.2	Dyslexia 'Coloured Overlay'	3
2.1.3	Dyslexia 'Government Guidelines'	3
2.1.4	Dyslexia 'Accessibility Checker'	4
2.1.5	Dyslexia 'Dyslexic Struggles'	4
2.1.6	Dyslexia 'Accent and Language'	5
2.2	Spelling Applications	5
2.2.1	Spelling Application 'Dyslexia'	5
2.2.2	Spelling Application 'General'	5
2.3	My Survey	6
2.4	Application	6
2.4.1	Application 'Device'	6
2.4.2	Application 'Code'	7
2.4.3	Application 'Web Scraping'	7
3	Approach	8
3.0.1	Requirements Specification	8
3.0.2	Implementation	9
3.0.3	Design	9
3.0.4	Maintenance	9
3.0.5	Research	10
3.0.6	Testing	10
3.1	Module Participants	11
3.1.1	LoginViewController	11
3.1.2	StudentRegisterViewController	11
3.1.3	TeacherRegisterViewController	11
3.1.4	GeneralRegisterViewController	11
3.1.5	GeneralHomePageViewController	11
3.1.6	GeneralSpellingTestViewController	11
3.1.7	GeneralSettingViewController	12
3.1.8	StudentHomePageViewController	12
3.1.9	StudentSpellingTestViewController	12
3.1.10	StudentSettingViewController	12

3.1.11	TeacherHomePageViewController	12
3.1.12	TeacherClassViewController	12
3.1.13	TeacherTestViewController	13
3.1.14	Teacher setting page	13
3.2	Proposed Solution	13
4	Requirements and Design	14
4.1	Requirements	14
4.1.1	Personas '1'	14
4.1.2	Personas '2'	15
4.1.3	Personas '3'	15
4.2	Total Requirements	16
4.2.1	Must Requirements	16
4.2.2	Should Requirements	16
4.2.3	Could Requirement	16
4.3	Client Meetings	17
4.3.1	22-02-22	17
4.4	Results from Survey	17
4.5	UML Diagram	19
4.6	Interface Flowchart	20
4.7	User Interface Design	21
4.8	Aims Diagram	22
4.9	Data Flow Diagram	23
5	Implementation	24
5.1	Database	24
5.2	Login and Register	24
5.3	Help Button	25
5.4	Setting Page	25
5.4.1	Background Colour	25
5.4.2	Feedback	26
5.4.3	Font Size	26
5.4.4	Font	26
5.5	Student and General Home Page	27
5.5.1	Student Spelling Test – Teachers Set	27
5.6	Student and General Spelling Test Page	27
5.6.1	Web Scarping	27
5.6.2	Hint	30
5.6.3	Right Number and Level of Random Words	30
5.6.4	Displaying Spelling Test	30
5.6.5	Checking Users Spelling	31
5.6.6	Feedback - Hide and Show	32
5.7	Teacher Account	32
5.7.1	Home Page - Display/Adding/Deleting/Access Class	32
5.7.2	Class Page - Display Class Information - Display/Adding/Deleting Test	33

5.7.3	Student Feedback Change	34
5.7.4	Adding Test - Displaying on Student Account	34
5.8	Unforeseen Problems	34
6	Results and Evaluation	36
6.1	User Testing – Client	36
6.2	Test Case	37
6.3	Discussion	41
6.3.1	Must Requirements	42
6.3.2	Should Requirements	42
6.3.3	Could Requirement	42
7	Future Work	45
8	Conclusions	47
9	Reflection on Learning	48
A	Appendices	50
A.1	Survey	50
A.2	Test Case - General User (10-)	55
A.3	Test Case - Student User (20-)	59
A.4	Test Case - Teacher User (30-)	63
A.5	App Screenshot	67
B	Bibliography	78
C	References	79

List of Tables

1	The table is the test case for "Changing the feedback display for a class in that class view controller." for a Teacher User.	37
2	The table is the test case for "Starting a spelling test, of different levels and number of words." for a General User.	38
3	The table is the test case for "Starting a spelling test, that the teacher has set for the student." for a Student User.	39
4	The table is the test case for "Changing the background colour in the setting page." for a Teacher User.	40
5	The table is the test case for "Register and login as a general user." for a General User. . .	55
6	The table is the test case for "Changing the background colour in the setting page." for a General User.	56
7	The table is the test case for "Changing the font in the setting page." for a General User. .	56
8	The table is the test case for "Changing the font size in the setting page." for a General User.	57
9	The table is the test case for "Changing the feedback display in the setting page." for a General User.	58
10	The table is the test case for "Log off the account." for a General User.	58
11	The table is the test case for "Register and login as a student user." for a Student User. . .	59
12	The table is the test case for "Starting a spelling test, of different levels and number of words." for a Student User.	60
13	The table is the test case for "Changing the background colour in the setting page." for a Student User.	61
14	The table is the test case for "Changing the font in the setting page." for a Student User. .	61
15	The table is the test case for "Changing the font size in the setting page." for a Student User.	62
16	The table is the test case for "Log off the account." for a Student User.	62
17	The table is the test case for "Register and login as a teacher user." for a Teacher User. . .	63
18	The table is the test case for "Adding a class to the database, linked to that teacher." for a Teacher User.	64
19	The table is the test case for "Access a class to go to the class page where more information is displayed." for a Teacher User.	64
20	Table above is the test case for "Add a new spelling test to a class for the student to then complete." for a Teacher User.	65
21	The table is the test case for "Changing the font in the setting page." for a Teacher User. .	65
22	The table is the test case for "Changing the font size in the setting page." for a Teacher User.	66
23	The table is the test case for "Log off the account." for a Teacher User.	66

List of Figures

1	Graph for fonts results from 'Spelling Test' survey.	50
2	Graph for background colour results from 'Spelling Test' survey.	50
3	Graph for spelling test results from 'Spelling Test' survey.	51
4	Screenshot of the spelling test question from 'Spelling Test' survey.	51
5	Graph for feedback results from 'Spelling Test' survey.	52
6	Screenshot of the feedback question from 'Spelling Test' survey.	52
7	Screenshot of the feedback question from 'Spelling Test' survey.	52
8	Screenshot of the feedback question from 'Spelling Test' survey.	53
9	Screenshot of the font question from 'Spelling Test' survey.	53
10	Screenshot of the background question from 'Spelling Test' survey.	54
11	Help button.	67
12	General Account Setting Page.	67
13	Background Selector.	68
14	Cream.	68
15	Yellow.	68
16	Orange.	68
17	Pink.	68
18	Blue.	68
19	Green.	68
20	Font Selector.	68
21	Verdana.	68
22	Comic Sans.	68
23	Arial.	68
24	Helvetica.	68
25	San Forgetica.	68
26	General Account Spelling Test Page.	69
27	Teacher Account Add Test Page Filled.	69
28	Student Account Spelling Test Page.	69
29	Teacher Account, Class feedback, 1.	69
30	Student Account, Feedback View, 1.	69
31	Level and Word Stepper, 1.	69
32	General Account, Feedback, 1.	69
33	Level and Word Stepper, 2.	69
34	Level and Word Stepper, 3.	69
35	Pink Selector.	70
36	Pink Background.	70
37	Blue Selector.	70
38	Blue Background.	70
39	Yellow Selector.	70
40	Yellow Background.	70
41	Teacher Account, Feedback, 2.	70
42	Student Account, Feedback View, 2.	70
43	Teacher Account, Feedback, 3.	71

44	Student Account, Feedback View, 3.	71
45	Congratulation Alert.	71
46	Emails Don't Match Alert	71
47	Invalid Email Alert.	71
48	Email is Already in the System Alert.	71
49	Passwords Don't Match Alert.	71
50	All Fields need to be Filled Out Alert.	71
51	Email and/or Password Alert.	71
52	Arial Selected.	72
53	Comic Sans Selected.	72
54	Small Font Selected.	72
55	Large Font Selected.	72
56	General Account, Feedback View, 1.	72
57	General Account, Feedback, 2.	72
58	General Account, Feedback View, 2.	72
59	General Account, Feedback, 3.	73
60	General Account, Feedback View, 3.	73
61	Invalid Access Code Alert.	73
62	Delete a Class Alert.	73
63	Access or Delete Alert.	73
64	Access Code Already Used Alert.	73
65	Class been Deleted.	73
66	Teacher Account, Class Added, List.	73
67	Teacher Account, Class Page.	74
68	Deleted a Test Alert.	74
69	Yes or No, Delete Test, Alert.	74
70	Teacher Account, Test List, 2	75
71	Teacher Account, Add Test Page.	75
72	Full Feedback View.	76
73	Teacher Account, Add Class Filled.	76
74	General Account, Spelling Test Filled.	76
75	Teacher Account, Class Deleted.	77
76	Teacher Account, Test List, 1.	77
77	Teacher Account, Test List, 3.	77
78	Your Done Alert.	77

1 Introduction

1.1 Aims and/or Goals

This dissertation and application project is diving into creating a spelling test application which is tailored towards people with dyslexia. There will be features within the app meaning the user will be able to change the font, font size, background colour and feedback display. These features are all aimed to help people with dyslexia to find the app easier to use and aid all users of the app to improve their spelling and literacy. Also, within the application teachers can set tests for their classes to complete and set the feedback in which they believe best suits that class. The end goal is to create an application which will help improve the UK's literacy.

1.2 Beneficiaries

The application will be able to be used by a wide range of people and a range of ages, due to the simplicity of the design. The effect to appeal and be understood by everyone. There are 3 different types of beneficiaries to the application, the general public, teachers, and students. Each account has a change in feature to match what they can complete in their account, to aid the literacy improvement.

1.3 Scope of the project

The application will be compatible with IOS operating system subject to iPhones and the project will be designed and coded on XCode using swift. The project at this current point of time is unable to run on any other devices. The database which was chosen is called SQLite3. This coding language was chosen due to the fact of the accessibility of the programmer. The main reason for the chosen language being swift using XCode is because more people have access to smartphones over laptops or desktop computers, and the most popular phone, in the UK, is an apple product.

1.4 Approach

A survey will be conducted to enable the best design for the public, finding out what feedback the public would like best to appear when completing the spelling of the word. It is important information to find out through this project to satisfy the proposal. Then an XCode application is created that allows the user to load a random spelling test, of a level of their choice to help improve their literacy. Also, able to change the settings of the application to aid the user.

1.5 Assumptions

Some of the assumptions which have been used to create the application is that some of the main tools which help the dyslexic community, are coloured overlays, changing fonts and font size. Another assumption, however, one that has been explained and backed up is that the users of this application will have an IOS smartphone, as that is the only platform the application is able to be run on.

1.6 Summary

Throughout this report it will explore the background around dyslexia, what products are already in place to aid and help the dyslexic population with literacy. There will also be research on currently published spelling test applications, looking into device and coding platforms, and any other research that will be able to be used to aid the project. The approach describes how the solution is going to be worked and the report also discusses the implementation of the application. As well as looking at test cases of the final program. The results will be analysed, and a demo video is provided in the archives, demonstrating the application. Evaluation is included to check the applications effectiveness, and this will aid the users. Finally, the report will be reflecting on work that could be added to the system in the future and concluding the dissertation. The main outcome of the application is that it is able to create spelling tests which are able to help the general public to improve their literacy and also the dyslexic community improve their literacy and have the settings of the application best reflect their paper-based learning.

2 Background

2.1 Dyslexia

Looking at the NHS overview of their definition of dyslexia it is stated as *“Signs of dyslexia usually become apparent when a child starts school and begins to focus more on learning how to read and write”* [20].

2.1.1 Dyslexia 'Fonts'

When researching the relevant fonts which can make it easier for individuals which have dyslexia, it was stated to use “sans serif fonts” [5] some examples of these are “arial, comic sans and or verdana” the reason that these fonts are easier for people with dyslexia is due the fact that these fonts are less crowded as other font which are more pushed together. Font sizes can differ depending on the person as some dyslexic people may need larger fonts than others, however, it is said that the font size should be around “12-14 points” [5].

2.1.2 Dyslexia 'Coloured Overlay'

When reading and researching into the colour background and overlay's, and investigating into why they are helpful for people with dyslexia. It has been found that there is a link to overlays, specifically colours due to a syndrome called SSS which stands for “Scotopic Sensitivity Syndrome”. The syndrome is all to do with *“hypothesized as sensitivity to frequencies of the light spectrum that causes visual stress”* [7]. Coloured overlays are used to try to alleviate visual stress. The reason this is linked to dyslexia is because it is linked to symptoms that are commonly suffered in people with dyslexia.

The concept of using colour overlays or coloured paper is to have one single colour that is used for the background. However, it should be put onto paper that doesn't have any patterns and or pictures, this is so there is no distractions in the surrounding. When picking the colour of the overlay or paper there needs to be a sufficient contrast between the background and the text. One of the best contrasts is using dark text for example black and a light background which should not be white. Colours which should be avoided, because it can be difficult when people also have colour blindness, are both green and pink / red. Dyslexic people do have their own preference when it comes to colour of background. Common colours that work well are creams or a soft pastel colour, this is instead of white which can appear to be dazzling. [5]

2.1.3 Dyslexia 'Government Guidelines'

When trying to research what provisions are already in place for people with dyslexia through the government website, an outline was unable to be found. A research and analysis of a dyslexic user article was found. [12] Within this research paper it is mentioned the device and technology which the user already has in place to make day to day life easier, this includes a software which will read documents and web pages out loud for the user. The article also includes more reading for help with people with dyslexia.

There has been guidelines entered into the disability act of 1995 and the equality act of 2010. This is to stop any discrimination in either the workplace and/or in society for people who have dyslexia. Contained in these acts are protocols that employers have to adhere to, adjustments that are appropriate and

reasonable to help and reduce the impact of the user's disability on their work and the role they need to complete. [4]

2.1.4 Dyslexia 'Accessibility Checker'

Some applications for example word already have software in place which is designed to make the formatting of the documentation easier for the user who suffers with dyslexia to read and understand.

One of the main features of the words accessibility checker is, increasing the amount of white space between the sentence and the paragraph within a document. This is since some people who have dyslexia can say that the more compressed the lines and texts are it is harder to read as it tends to merge and or distort.

Font and colour are also talked about when discussing inclusivity of a word document, but there are many more built in features that word has created within their accessibility checker, to help make the document easier to read and understand. [26]

2.1.5 Dyslexia 'Dyslexic Struggles'

Researching more into the day to day difficulties that people with dyslexia struggle with, it is not only the more noted one in society of struggling with literacy but also there are many everyday activity's which could cause a difficulty to dyslexic's. Some of which include [27]:

- Social skills.
- Listening comprehension.
- Memory.
- Navigation.
- Time management.
- Finance Management.

For children they will be learning using the wrong techniques if it is not spotted that they might have dyslexia, some of the most common struggles that children aged 5 – 12 include [20]:

- Learning the spelling of their names.
- Unpredictable spelling.
- Letters that look the same.
- Ordering the letter correctly.
- Reading slower and making errors.
- Hard to write their answer down.
- Carrying out sequence of directions.

-
- Learning sequences.
 - Writing speed.
 - Handwriting.

Due to these struggles that children may have, means that children then face more struggles in their life within the school system. A study was conducted, title being: ‘Dyslexic learners experiences with their peers and teachers in special and mainstream private schools in North-West Province’. Within the findings of this research study, it was noted that students that are dyslexic will be subjected to more ill-treatment by their peers. This normally includes bullying of the individual. It was also found that teachers were not patient with the students. Teachers would also not give them extra attention and even worst some would embarrass the student by using negative comments. [16]

2.1.6 Dyslexia ‘Accent and Language’

People who have dyslexia find it easier to understand simple languages than more complex ones like English. However, when it comes to accent or change in voice, a study was conducted, and it was noted that people who have dyslexia have a reduced neural adaptation when it comes to hearing the same accent / tone of voice and even more reduced when they hear a changed accent / tone of voice. Therefore, accents don’t necessarily affect how a dyslexic person learns as their neural adaptation is reduced no matter what. [18]

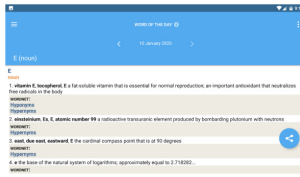
2.2 Spelling Applications

2.2.1 Spelling Application ‘Dyslexia’

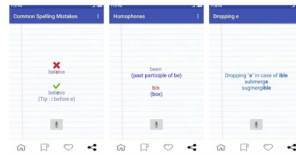
Through researching it was found there are many spelling practice books for people with dyslexia. When it comes to phone applications, these were few and far between. Many of them were tailored towards people of a younger age. An article which displayed some of the best spelling test applications for people with dyslexia included “Spell Better, Spell Check, Typ-O HD, Dragon Naturally Speaking, Dragon Dictate” [17]. There are other applications designed to help dyslexic people, however these are more for writing and aiding spellings for longer paragraphs not individual words. [9]

2.2.2 Spelling Application ‘General’

When looking into more general spelling practice applications there was the same issue that occurred with the application which were tailored towards people with dyslexia is that they were all directed to younger children. Not inclusive towards all ages, or many applications that weren’t for the very younger generation they were in fact paid for applications. Even applications that are for older grades like “SAT Vocab by MindSnack” which is for grades 8 – 12 still have a child like look to them, being very over crowded with imagery. They are not just simply to the point of spelling, with aesthetic design [6]. Applications which have been designed for adults that are to improve spelling are designed to be more of a scrabble or word search design, instead of spelling a word which is given to the user. A couple of applications were found that are tailored more towards adults that are writing / spelling a word given to them. As seen below the design then goes the other way, being plain and not enticing to the users. [25]



Advanced Dictionary
Thesaurus, [25]



English Spelling Rules, [25]



Spelling Notebook, [25]

2.3 My Survey

Along with the research which has been conducted through looking at articles, a user survey was instigated to gain relevant up to date data. Based on the research conducted the survey was designed. Using the data that is collected from the survey this will be used to design the application to aid people with dyslexia as well as the general population. [14]

The main couple of questions from the questionnaire which will have the most impact on the design of the application is as follows:

- 8 = “What font do you find the easiest to read?” [Figure 9]
- 9 = “What colour background do you find makes it the easiest to read the word?” [Figure 10]
- 11 = “What example below do you believe you could complete the spelling off?” [Figure 4]
- 12 = “When seeing the result of the spelling, how much information would you like to be displayed?” [Figure 8]

To generate the option for the user survey was based on research from articles, adding a control (option that are not tailored towards dyslexia). Within question 8, the options where the 3 best fonts suited to people with dyslexia, however added to this is the most common used font which is described as “Helvetica”, this was cross examined through multiple articles to make sure that the font was valid [19][23].

2.4 Application

2.4.1 Application 'Device'

The spelling test application could either be based on a desktop or on a smart phone. In research of the most commonly used device: *“More than half of the people worldwide access the internet using mobile devices, while only 49% do so via desktops”* [22]. Based on the quotation, the most popular smartphone device are

1. “iPhone 12 Pro”
2. ”iPhone 8” [28]

Based on a YouGov survey, the most popular devices are apple smartphone, which means for the application to be compatible with an apple smartphone device it would need to be coded with an IOS system.

2.4.2 Application 'Code'

When looking into coding languages and what would be the best for the application, also making sure that it was accessible to the coder, it was decided to use swift, XCode. Developer apple website describes swift as, quoted, *"Swift is a powerful and intuitive programming language for iOS, iPadOS, macOS, tvOS, and watchOS. Writing Swift code is interactive and fun, the syntax is concise yet expressive, and Swift included modern features developers love. Swift code is safe by design, yet also produces software that runs lightning-fast."* [3].

There are also two main advantages of using swift, XCode, one being that it is safe, as the code encourages the coder to make sure their code is clean and consistent, this means there is less likely to be errors. [2]

The other advantage and positive point of using swift, XCode, is that it is fast in performance. The code is fast within two ways firstly in developing code as the syntax is simple. Secondly the code itself is faster, a quote includes, *"Swift is up to 2.6x faster than Objective-C and 8.4x faster than Python."* [2].

2.4.3 Application 'Web Scraping'

Web scraping will need to be used in the application to be able to gain information that will be displayed on the application. For example gaining the definition of the word to show to the user for their feedback. Web scraping is a process of using bots that are used to take content and data from another website that is passed in through the web-scraper. Web scraping gains the data using the HTML code and the process is able to duplicate the website content for the use of the application. [15]

3 Approach

For the approach for the spelling test application, a ‘hybrid waterfall’ will be used. This methodology is based off the ideation that the approach to the development is broken down into task which follow a linear sequence.

Within the hybrid waterfall methodology, which is more of a iterative approach. Steps have been taken that are already within sequences in the methodology and added multiple mini tasks within each selection.

A point which needs to be addressed before explaining each section is normally in a traditional waterfall methodology the development team does not have contact with the client. However due to the fact the programmer and reporter is a one-person team, and the project is client based, the development team will have contact with the client. When meeting the client there will be some iterative development for the coding side meaning, the feedback will be given by the client which will then affect and make changes within the development process.

As the application is a client-based project, the client being “Matthew Moloughney”. Throughout the pre-development and development process there will be meetings with the client to make sure that the application is still in compliance with his set requirements. Conducting two to three meetings over the space of the 12 weeks; however, more are able to be added if needed.

Meeting one will be to discuss the user requirement, and the user stories. The client will also be able to see the outline design of the application and decided if there is any detail that needs to be added or changed.

Meeting two will be in the middle of development making sure the client is happy with the process, the design. Making sure requirements match up with the application itself. The client at this point can give any feedback if they want something to be added, changed or taken away from the application.

Meeting three will be at the end of the development to receive feedback for the application and for the client to test the application.

Meeting two and three could be combined if the development process is quicker than expected and more efficient then planned.

3.0.1 Requirements Specification

There are three different types of requirements that can occur when creating an application. By placing the requirements under these headings means that there is organisation when it comes to creating the application. There is a hierarchy of processes that need to be completed. Below is the named sections and their definitions:

1. Must - These are requirements that must be a part of the application when it is completed. These are the standard requirements of the application.
2. Should - These are requirements that should be a part of the application when it is completed, however, if there is no time within the development stage then these don’t have to be accounted for.

-
3. Could - These are requirements that only need to be a part of the finished application, if there is extra development time. If they are not completed they could move to the future work stage.

3.0.2 Implementation

I have broken down the implementation into the parts that will need to be coded.

1. Login
2. Register
 - (a) Student
 - (b) Teacher
 - (c) Other
3. Student Account
 - (a) Home Page
 - (b) Spelling Test Page
 - (c) Setting Page
4. Teacher Account
 - (a) Home Page
 - (b) Class Page
 - (c) Spelling Test Additions
 - (d) Setting Page
5. Other/General Account
 - (a) Home Page
 - (b) Spelling Test Page
 - (c) Setting Page
6. General Design and System Code (to make the application fully run).

3.0.3 Design

1. Simple hand design model of the application.
2. Client meeting to discuss and have the design approved.
3. Computer design of the model, with updated changes that is wanted by the client.

3.0.4 Maintenance

This part of the waterfall model will not be conducted, however, if this application is to be placed on the app store, this would be needed for future work. This is discussed later in the report.

3.0.5 Research

To complete the application to a high standard, it was required to complete research to make sure that all the approaches to creating the application and the ideas for it are solid and secure. It was decided to conduct both research using a developed survey and also to read articles and websites which have a mixture of information already conducted and written by other people.

In the background section above, there is research into dyslexia and objects that have already been designed to help people with dyslexia. For example, colour overlays, these are used to change the background of a page to make it easier for the individual to read. Research was also conducted on the different and user friendly fonts for people with dyslexia. Deducing that more separate and spread-out fonts work the best to help the individual read.

Also, in the background section it was discussed that spelling test applications have already been coded and used in the market. In the research, background, section it was apparent that all the applications already on the market were made for the younger generation and not for adults or teenagers or those in-between.

Conducting a relevant survey will make sure that the application has the best results for both people with dyslexia and people without. The survey results will be able to support or disprove the background research. The results will include the greatest selected font, the favourite colour background, how the question is being displayed to the user and the way feedback presented. All these selection are the majority that survey highlights.

This application is needed to fill a gap in the market for a spelling test application which is not just tailored to the younger generation, but is tailored to all the population age ranges and has features which enable a person with dyslexia to find literacy easier.

3.0.6 Testing

Within the testing each part described in the implementation will be checked to make sure that they are running smoothly and correctly. Checking the application doesn't have any known bugs that will affect it's running. As this is a prototype the testing will be done by the developer and the client. Future testing would be needed to make the application app store ready. This will be discussed in the future work section of the report.

The testing process will involve 2 different types of testing:

- Going through each element of the front-end and back-end and using testing tables (test cases) to make sure they work.
- Asking the client and the supervisor to test the application / prototype so that a second opinion can be sought.

3.1 Module Participants

3.1.1 LoginViewController

Allow the user to input an email and password, then based on the user type of that login direct the user to the correct account. If the data is invalid an alert appears letting the user know an error has occurred. If the user doesn't have an account, they are able to transfer to the correct register page they need.

3.1.2 StudentRegisterViewController

The basic information is placed in the view controller, for example name, email, confirm email, age, password and confirm password. If any of this is invalid at the time of registering an error alert pops up letting the user know what they have placed invalidly. The student also needs to add an access code which is linked to a teacher so they can be assessed to that class. The error message appears at time of register if the access code is invalid. Once all values are valid the user gets added to the database and directed to the login page. From this view controller the user is also able to transfer back to the login page.

3.1.3 TeacherRegisterViewController

The basic information is placed in the view controller, for example name, email, confirm email, age, password and confirm password. If any of this is invalid at the time of registering an error alert pops up letting the user know what they have placed invalidly. Once all values are valid the user gets added to the database and directed to the login page. From this view controller the user is also able to transfer back to the login page.

3.1.4 GeneralRegisterViewController

The basic information is placed in the view controller, for example name, email, confirm email, age, password and confirm password. If any of this is invalid at the time of registering an error alert pops up letting the user know what they have placed invalidly. Once all values are valid the user gets added to the database and directed to the login page. From this view controller the user is also able to transfer back to the login page.

3.1.5 GeneralHomePageViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go to the setting page, also log off the app. Pick a random generate spelling test with the user chosen parameter of how many words and of what level.

3.1.6 GeneralSpellingTestViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off the account. The user can get a hint for the word they are being asked to spell and the word will be read out for them. The user fills in what they believe to be the correct spelling and then the background code checks the spelling and displays feedback based on what the user has chosen to see. Then the test either runs another word when the user presses next spelling based on the paramount 'number of words' chosen or lets the user know it is the end of the test.

3.1.7 GeneralSettingViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off the account. The user can update the font, font size, background colour and feedback. This gets updated in the database and the application default for that current session, and the new pre-sets are used.

3.1.8 StudentHomePageViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go to the setting page, also log off from the app. Pick a random generate spelling test with the user chosen parameter of how many words and of what level. If the teacher has set any test for the student class. The student is also able to complete that spelling test.

3.1.9 StudentSpellingTestViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off from the account. The user can get a hint for the word they are being asked to spell and the word will be read out for them. The user fills in what they believe to be the correct spelling and then the background code checks the spelling and displays feedback based on what the teacher has set for that class to see. Then the test either runs another word when the user presses next spelling. This is based on the paramount 'number of words' chosen or the teacher spelling test length. Or lets the user know it is the end of the test.

3.1.10 StudentSettingViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off from the account. The user can update the font, font size and background colour. This gets updated in the database and the application default for that current session, and the new pre-sets are used.

3.1.11 TeacherHomePageViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go to the setting page or log off from the account. The user is also able to access or delete classes from the home page, and add classes to their list, using a unique access code. When adding a new class it gets added to the database and a student can register to that class.

3.1.12 TeacherClassViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off from the account. The teacher is able to update the feedback the student sees. This gets updated in the database, and in turn changes the default setting for next time the student login. The user is also able to link to the add test view controller and delete test that have been previously set.

3.1.13 TeacherTestViewController

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the class page or log off the account. When all the fields are filled with valid input the test gets placed in the database, in turn showing on both the teacher account and that class student's account. If the values are invalid for example not all filled out then an error alert pops up letting the user know the problem.

3.1.14 Teacher setting page

A help alert can be displayed if the user needs help to understand the app page. The user can go back to the home page or log off from the account. The user can update the font, font size and background colour. This gets updated in the database and the application default for that current session, and the new pre-sets are used.

3.2 Proposed Solution

The proposed solution to the proposal given by the client, consists of using the XCode platform using the language swift combined with SQLite3 for the database to create a IOS compatible smartphone application. It will allow the user to create an account which matches their needs / uses for the application. A teacher, wanting to use the application to set homework for their student. Student, been introduced to the application by their teacher. General user, wanting to improve their spelling with no prompt from an outside party. The user can login to the app and be directed to the correct account. The teacher will be able to add and remove classes. Add, and remove tests for individual classes, set feedback setting for classes. For themselves they can change the font, font size and background. For both student and general user, they will also be able to change their font, font size and background and be able to start a random generated test based off chosen parameters. The Difference between these two accounts is a student can start a teacher set test whereas a general user cannot. A general user can change their feedback whereas a student is notable to. Help alerts are located all throughout the app to aid the user into understanding the features on each page. The app will have a limited number of steps to complete a task making it effected an easy to use and navigate. Overall, the proposed solution is aiming to help the user improve their literacy and aid people who have dyslexia.

4 Requirements and Design

4.1 Requirements

Within this requirement section, personas and user stories have been used to gain the most prominent requirements for the application. This is to make sure that it has good usability, functions and features that are attractive to the clientele.

The personas that have been created are all fictional descriptions of characters, that are, or could be the users of the application. They all have different lifestyles, reasons for using the app and goals for their literacy. From these personas, quotes have been taken. These are labeled as user stories. They link to one requirement, that would need to be a part of the app to make sure it is suitable for that character, persona. Another way of describing a user story, is a goal or a need for the persona to have in the application.

Below are three personas that have been created for this application linking to user stories which have created some of the base line requirements for the development.

4.1.1 Personas '1'

“As an 18-year-old dyslexic teenager / adult, with an iPhone, who is taking A Level English Language. I want to be able to improve my spelling and improve my spelling of important words my teacher has mentioned in class. I also want to be able to pick a font which are placed further apart, as this is what I find the easiest to read. This is so that I can feel more confident within class having not to ask for spelling and therefore it will benefit my education.”

User Story '1.1'

Quote - “I also want to be able to pick a font which are placed further apart, as this is what I find the easiest to read.”

Requirement '1.1'

Change the font of the application.

User Story '1.2'

Quote - “with an iPhone”

Requirement '1.2'

Application works on an iPhone.

User Story '1.3'

Quote - “improve my spelling of important word my teacher has mentioned in class”

Requirement '1.3'

Complete spelling test set by the teacher.

User Story '1.4'

Quote - “I want to be able to improve my spelling”

Requirement '1.4'

Complete random generated spelling test.

4.1.2 Personas '2'

"As a 56-year-old, with an iPhone, with SSS 'Scotopic Sensitivity Syndrome'. I want to improve my spellings and be able to change the background of the application due to my SSS and be able to control the feedback of my spelling tests, so the more I improve the more feedback I can allow myself. So that within my day-to-day life I don't need to ask my friends and family to help me write documents as I am unable to spell the key words."

User Story '2.1'

Quote - "be able to change the background of the application due to my SSS"

Requirement '2.1'

Change the background of the application to imitate colour overlays.

User Story '2.2'

Quote - "I want to improve my spellings"

Requirement '2.2'

Complete random generate spelling test.

User Story '2.3'

Quote - "be able to control the feedback of my spelling tests, so the more I improve the more feedback I can allow myself"

Requirement '2.3'

Edit the amount of feedback shown.

4.1.3 Personas '3'

"As a teacher in a college, who teaches English, with an iPhone. I want to be able to set classes homework tasks to be completed and set the amount of feedback for different classes and even change the feedback for different people in the class. This is so that the student will get the feedback which will best suit them and the spelling test that will best suit the class."

User Story '3.1'

Quote - "I want to be able to set classes homework tasks to be completed"

Requirement '3.1'

Set spelling test for different classes.

User Story '3.2'

Quote - "set the amount of feedback for different classes"

Requirement '3.2'

Set the feedback for different classes.

User Story '3.3'

Quote - "even change the feedback for different people in the class. This is so that the student will get the feedback which will best suit them and the spelling test that will best suit the class"

Requirement '3.3'

Set the feedback for different students.

4.2 Total Requirements

When creating an application, the requirements will be gained from the proposal, the client and the user stories. There are requirements for both the front-end and back-end of the application. There are also three different types of requirements, these include, must, should and could.

The must requirements must be a part of the application. The should requirement should be part of the application if there is time when creating the program and the could requirement are those that could benefit the application, but doesn't have to be a part of the development at this time if there are time restraints.

4.2.1 Must Requirements

- Allow the user to be able to login to their account.
- Allow the user to register for an account.
- Give the user random generated spelling practices to complete.
- Give the user feedback on the spelling based on the user choice or of the teacher.
- Let the user be able to change the colour of the background if needed.
- Let the user be able to change the font if needed.
- Give the user an overall result of how many questions they got right and wrong overall.
- The application must be able to be simple to navigate through. So that it is able to be used and be able to be accessed by all ages and all abilities.
- Allow the teacher to set the feedback for class.

4.2.2 Should Requirements

- When the user is signing up / registering to the application, they can pick if they are a student, teacher, or general user.
- Should be able to log off the system and be taken back to the login page.
- The teacher should be able to see a list of their classes and their students in that individual class.
- Allow the teacher to set spelling test for classes.

4.2.3 Could Requirement

- Could store the answer that were incorrect and allow the user to complete a spelling practise with those words.
- Allow the user to try again if the user has not put in the correct answer to the spelling question.
- Allow the teacher to set feedback to particular students.

4.3 Client Meetings

4.3.1 22-02-22

A meeting with the client was held at the beginning for the requirements and design part of the process, to make sure that the application is on the right lines with what the client expected.

The following were discussed with the client:

- Login system, it was agreed that there needed to be some secure login system, however, this is a 'could' to the client and just needed to be functional. A full login and register system would come after, if there is enough time in the development process. It was discussed about the differentiating between users and their interface as the user type are very important.
- The main functionality of the application is the spelling test so needs to be the main part of the program making sure that it is running smoothly and is effective.
- Another discussion point with the client was the feedback. It was discussed that the teacher will be able to apply the amount of feedback for the class, however, the student will still be able to hide or show different parts of the feedback that the teacher has set.

Also within the meeting with the client was the design of the application this included:

- Adding a hide button within the feedback section of the application.
- Introduction to another font which could be useful with this application called sans forgetica. [24]

4.4 Results from Survey

Throughout the research and background on the project a survey was created. So that evaluation and analysis for the most popular opinions of public preference could be gained. To accumulate the survey, the background research that is presented above was taken and questions compiled to create not only the default setting for the application but also how the spelling practice is implemented itself. Once ethic approval had been received the survey was conducted.

For the survey a high volume of data was collected from a range of different ages, starting at 18 to above 71. By gaining a large amount of data it was believed that this would reduce the bias in the program and the end goal was to create something that would appeal to many different generations and different people with or without learning difficulties.

It was decided to collect data from both people with and without dyslexia. This is because the project will be created to be tailored to people with dyslexia however, the application is be able to be used and accessible to anyone and everyone. These application is not to be restricted to one group of people. Due to the reasoning above the question was asked to everyone that participated in the survey, "Do you have dyslexia?". It was known that there would be more participants without dyslexia then with. This is because "6.3 million people (around 10% of the UK population) have dyslexia." [13]. Therefore, when conducting this survey, it would be expected to see about 10% of the participants to be dyslexic. Anything above this would be a blessing. Out of the 159 participants, it is noted that 35 are dyslexic, making out

of the sample 22% dyslexic. This is very helpful as it is above average.

Below, have been created 4 bar graphs. Each of the graphs have been separated into two group of results, dyslexia and non-dyslexia. This has been done to see if any patterns correlating people with dyslexia and people without could be found.

Examining the graph from appendix's [Figure 1], for the best font to set as the default setting for the application, (what font will be used when the program is loaded). The results show for people that haven't got dyslexia they prefer 'Arial' and for those with dyslexia prefer 'Verdana'. Since there is not a common font between people with and without dyslexia, the decision that was made was to use 'Verdana'. This is because the application is being made to help make it easy for people that have dyslexia to spell.

With regards to picking the background colour for the default, evaluating graph from appendix's [Figure 2], it was decided to go for 'Cream' this is because it was the most even vote for people that have dyslexia and people that don't. The other options can be picked in the setting page. The white was removed due to the fact of how close it is to cream.

Evaluating both graphs from appendix's [Figure 3], and [Figure 4], the spelling test is going to be based on option one. However, it was known there would be the problem that people might not be able to work out what the icon is meant to be. Therefore making a hint for the spelling test application which would be a speak button. So if the user is unable to work out the word from the icon they can hear the word too.

Examining graphs from appendix's [Figure 5], and [Figure 8], although option 2 was the overall preferred option, the conclusion was that because hide buttons are going to be implemented in the feedback pop up, option 1 is going to be the default option. The user can still add and change what feedback they want to see, but also show any feedback that they don't normally want displayed.

Also, when conducting the survey, to back up the reasoning for programming the application on an IOS device, e.g., an apple product, the question was asked: "Do you have an IOS device?" to everyone who filled in the survey. Out of 159 results, 41 said they did not have an IOS device and 118 said they did. The meant 70% do have an IOS device. This is over half, proving that it is the more popular device to have. To back-up the point further and to make sure the point still stand with people that have dyslexia as well. The result of the question "Do you have an IOS device?" from just the participants that answer yes to the question in the survey "Do you have dyslexia?". Out of 159 participants that answered yes to have dyslexia which was 35, 25 have an IOS device, 71% and 10 do not have an IOS device, 29%. Showing that the majority have an IOS, good reasoning to go ahead and code the application on XCode using swift, an IOS coding platform.

4.5 UML Diagram

Below is a UML diagram outlining a prototype on how the applications database will be structured, with the tables, using SQLite3 to code the database.

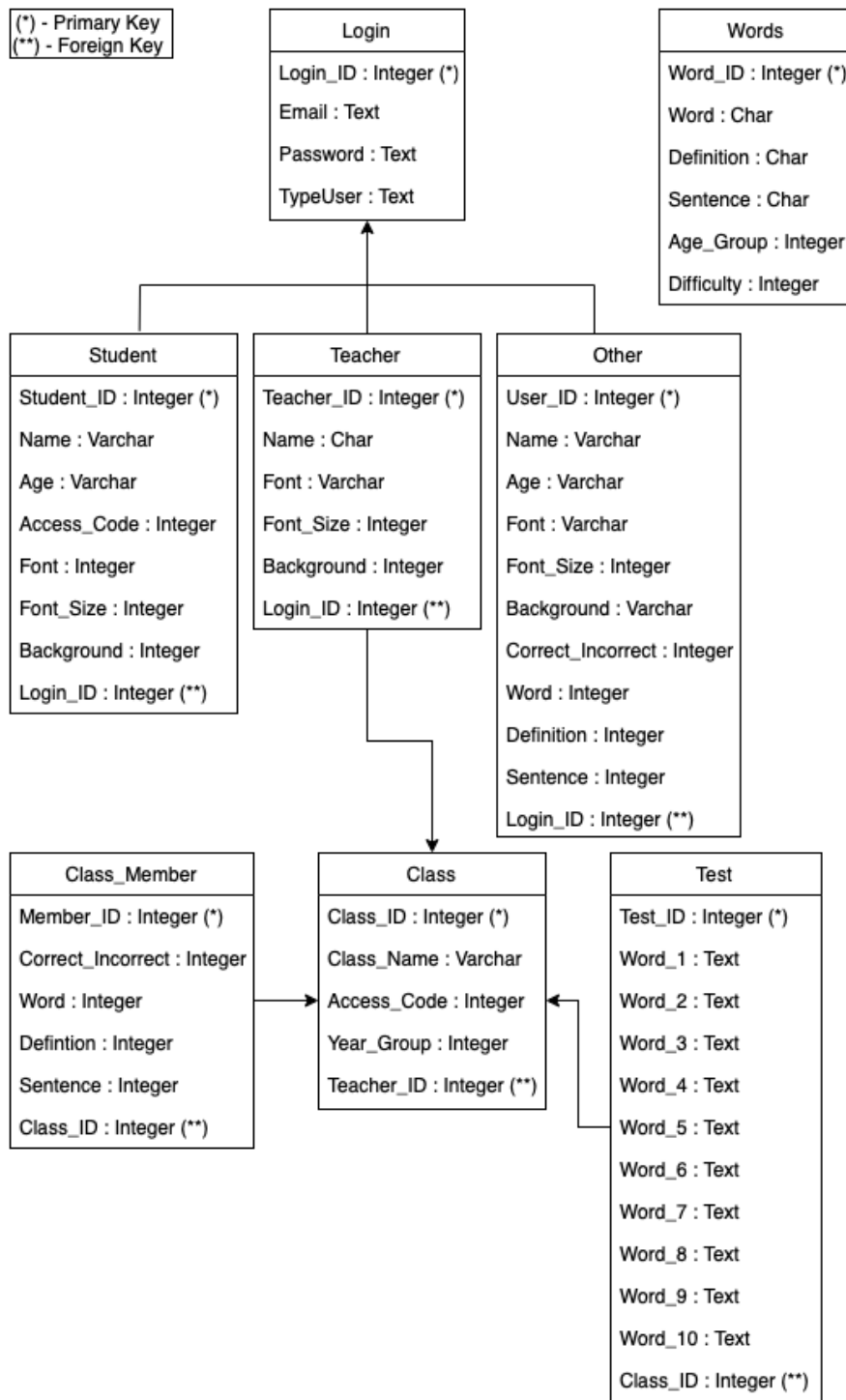


Figure 1: UML Diagram

4.6 Interface Flowchart

The image below is a user interface flowchart demonstrating how the prototype will move from view controller to another view controller in XCode using Swift.

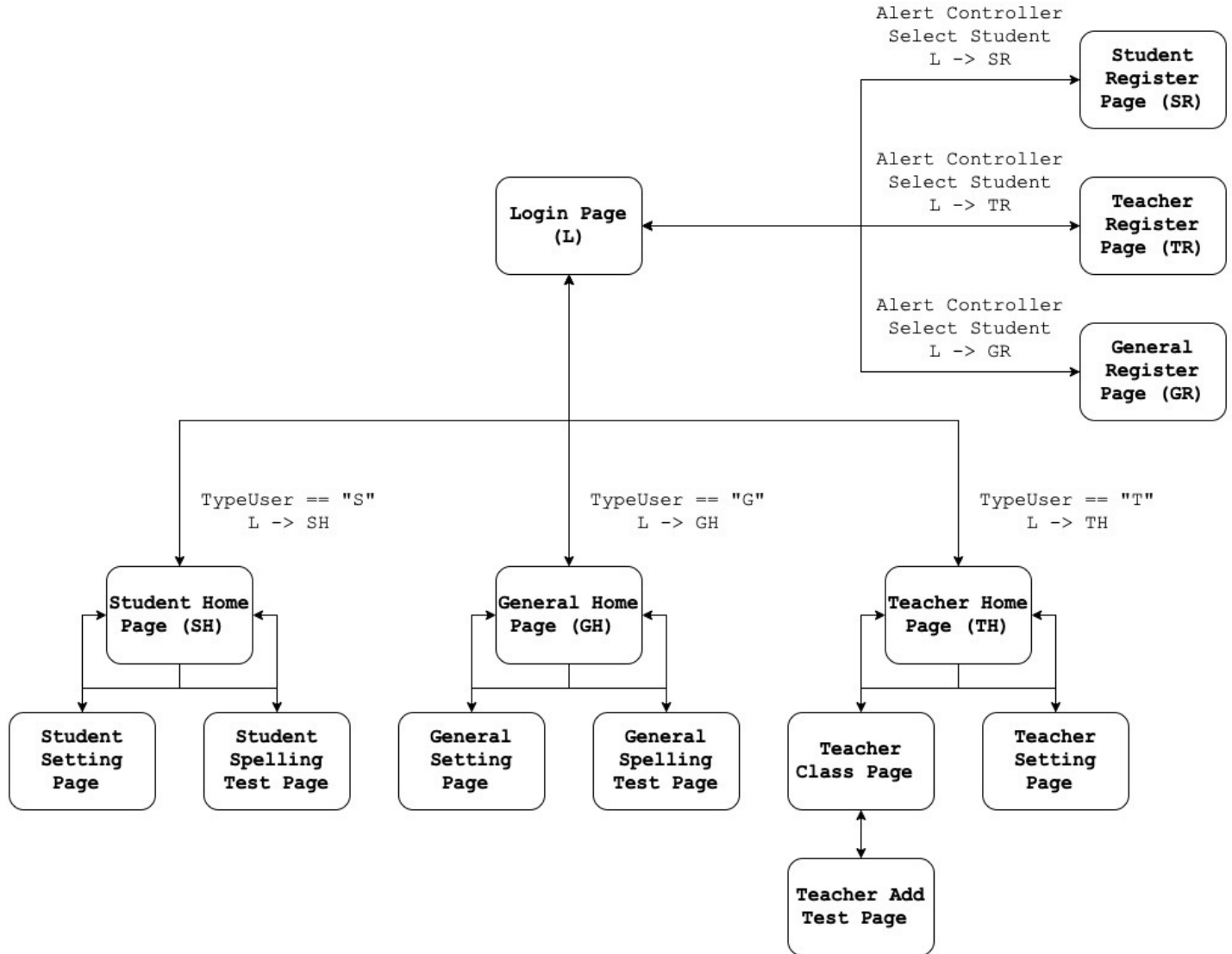


Figure 2: Interface Flowchart

L - "Login"
SR - "Student Register"
TR - "Teacher Register"
GR - "General Register"
S - "Student"
T - "Teacher"
G - "General"
SH - "Student Home"
GH - "General Home"
TH - "Teacher Home"

4.7 User Interface Design

The image below is the initial interface design, however it is and will be subject to change over the development process.

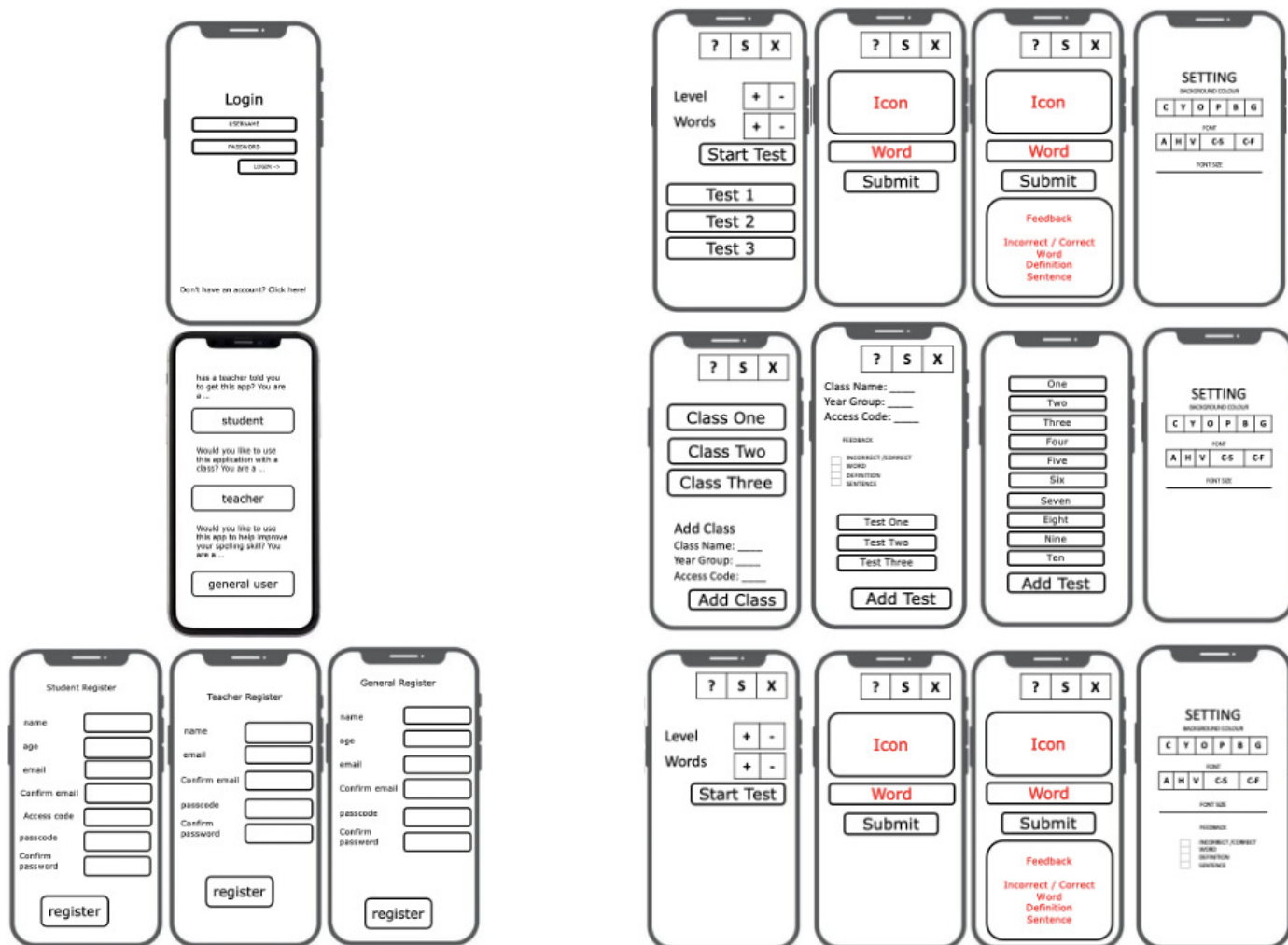


Figure 3: User Interface Design

4.8 Aims Diagram

This aim diagram is based on the aims extracted from the induction. This is a very plain simple goal that needs to be implemented in the program, this is not including all the requirements. Green is the project, yellow is the goals and blue is how they will be achieved.

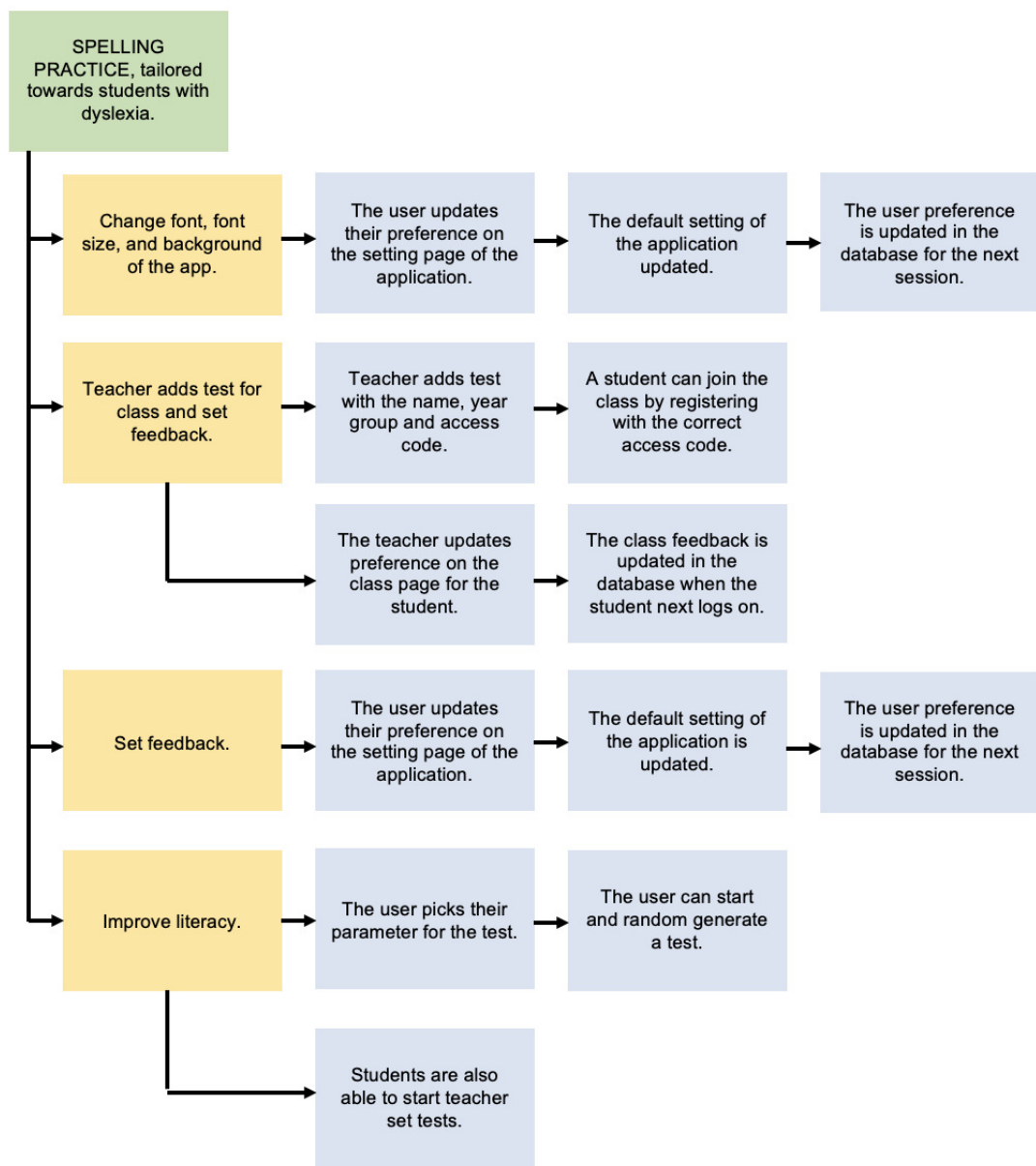


Figure 4: Aims Diagram

4.9 Data Flow Diagram

This data flow diagram shows a simplified version on how and which data get transferred from the database throughout the system. Some parts of the system don't use the database these are limited. The default input is passed to this view controller.

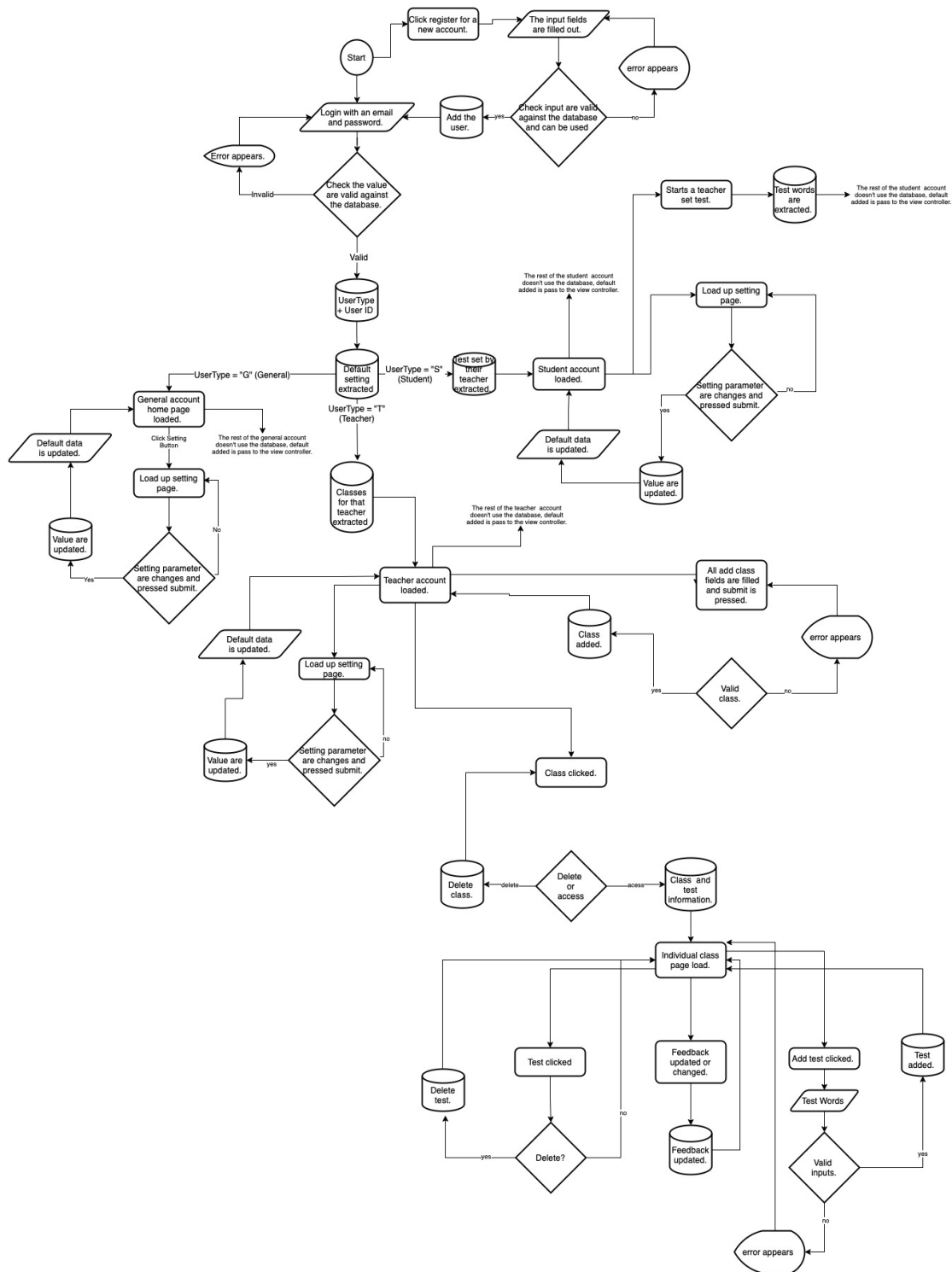


Figure 5: Data Flow Diagram

5 Implementation

The ‘SPELLBEE’ application has been coded on XCode, using a mixture of Swift and SQLite3. Other packages imported into the code include SwiftSoup and AVFoundation.

SwiftSoup was imported into the view controllers coding pages so that the process of web-scraping was able to be used to not only gain the random words the user is to spell, but also to gain the correct information that the user receives in their feedback. Also the icon of the word the user needs to spell. SwiftSoup allows for the code to extract the specific areas, class, div, html of a web page.

AVFoundation package (import), is a foundation import, this is included in the view controller code pages of the application due to the fact that it creates and defines a layer of functionality for the development. This means that classes are able to be used in the coding, however without this being imported this function would not be able to be used.

Through the implementation section many parts of the application are disused about, in the archive there are videos demonstration the working application.

Storyboards are a built in part of IOS 5 XCode. These are an interactive feature which allows the user to be able to build a user interface without using code. Storyboards mainly means that prototypes of the interface are able to be made and to create links and transitions between storyboards. [10]

5.1 Database

The database which was used to store the data throughout the application was **SQLite3**. SQLite3 is a C language library which has and is able to provide, to the application, *"lightweight disk-based database"* [8]. There is a database which is able to be used on IOS, this was a key important factor due to the application being coded on XCode. Another factor of SQLite3 is that, it can be easily transfer the code to another database which is larger for example Oracle, whereas SQLite3 is a smaller prototype database.

5.2 Login and Register

To start the application the user selects either the login or register page. All these pages have many validation steps to make sure that the login email and password is valid and all the information in the register form is valid. Within the login system the validation includes that the email is in a valid form, the text fields are all filled out and that the email and password are within the database together. The login page redirects the user to their account once validated. The program directs the user to the correct account based on the TypeUser, e.g., “S” = Student Account, “T” = Teacher account and “G” = General Account. For the register account for a general account the validation steps include, email is in a valid form, not in the system already and that it matches the confirm email. The passwords match and finally all the text fields are filled out. These validations are the same for the teachers and students. The only other validation is for students account, making sure the access code is valid and is in the database, connected to a teacher and a class.

5.3 Help Button

Throughout the application, help buttons [Figure 11] have been added to aid the user on how the pages work and enable them to get the best results out of the application. The help button is coded using swift UIAlertController that pops up on the screen giving advise to the user. An example of this code, is from the ‘TeacherHomePageViewController’.

```
@IBAction func HelpButtonTeacherAccount(_ sender: Any) {
    // Display an alert with a help message to help the user
    understand the page.
    let HelpAlert = UIAlertController (title: "Help?", message: "Here
you are able to see all the classes you have and you are able to add new
classes, you also able to click on the classes to gain more information!",
preferredStyle: .alert)
    HelpAlert.addAction(UIAlertAction(title: "Thank You!", style:
UIAlertAction.Style.default, handler: nil))
    self.present(HelpAlert, animated: true, completion: nil)
}
```

5.4 Setting Page

All the accounts have a setting page, however, depending on what account the user is logged into depends on how much the user can edit. A student account is only able to edit their background colour, font, and font-size, this is the same for the teacher account. The general account is the same again however the user is also able to edit their feedback as well. [Figure 12]

All the setting choices are saved to the database connected to the user. At the beginning of the user session, when they log into their account this information is extracted from the database. Each of the choices that the user can pick from have been selected due to the background research conducted and the spelling test survey.

5.4.1 Background Colour

From the background and survey research, coloured overlays is very subjective to the individual. Meaning that the colour the user can pick from is a soft version of the primary colours and some of the other most popular coloured overlays. Below is the code of the colour which is then transferred to rgb colour and how the background is set once gained from the database using UIColor, this is how it is coded in swift. [Figure 13], [Figure 14], [Figure 15], [Figure 16], [Figure 17], [Figure 18] and [Figure 19].

```
let Colours = ["Cream": [1.00, 1.00, 0.89, 1.00], "Yellow": [1.00,
1.00, 0.72, 1.00], "Orange": [1.00, 0.76, 0.49, 1.00], "Pink": [1.00,
0.82, 0.97, 1.00], "Blue": [0.62, 0.80, 1.00, 1.00], "Green": [0.68, 1.00,
0.68, 1.00]]
```

```
BackgroundColourIndex = Int(BackgroundDatabaseQuery)
let ColourKey : String =
TheColours[BackgroundColourIndex]
BackgroundColour = Array(Colours[ColourKey]!)
self.view.backgroundColor = UIColor(red:
CGFloat(BackgroundColour[0]), green: CGFloat(BackgroundColour[1]), blue:
CGFloat(BackgroundColour[2]), alpha: CGFloat(BackgroundColour[3]))
```

5.4.2 Feedback

The general and teacher accounts are able to change the feedback which is shown when the submit spelling button is pressed. The user is still able to access the other feedback by pressing the show button, however, it will be hidden when the feedback is switch off. The feedback is split into four and each set has a switch. These include, correct/incorrect, word, definition, and sentence. The choices are saved into the database and become the users new preset. The user is therefore able to see the feedback that is desirable to them. If feedback has not been chosen then only correct/incorrect is the preset choice. The feedback preset if the user has set them previously will be changed on the home page and extracted from the database.

5.4.3 Font Size

Throughout the settings page the user can change the font size of the spelling test and adding test. The preset font size is a larger font to make it easier for the user however they can make it up to 20 and lowest 7. This will then be updated in the database so the next time the user logs in to their account it uses their new preset.

5.4.4 Font

The user is also able to change the font for the spelling test and adding test. The first four fonts are the ones gained from background research and verdana is the preset font based off the survey. 'San Forgetica' has been added due to a conversation with the client who mentioned that the font was interesting, and challenges the user concentration when reading, therefore it was added to the font choice. As with the background and font size the user choice is saved to the database so that their new preset is used for the new font when a new login session occurs. [Figure 20], [Figure 21], [Figure 22], [Figure 23], [Figure 24] and [Figure 25]. The fonts which were picked for the application, these are the font that the user could change between to find the one that best suited them and there links include.

- Verdana - <https://www.dafontfree.io/verdana-font/>
- Comic Sans - <https://www.dafontfree.net/comic-sans-ms-regular/f100934.htm>
- Arial - <https://freefontsfamily.com/arial-font-family-free-download/>
- Helvetica - <https://freefontsfamily.com/helvetica-font-family/>
- Sans Forgetica - <https://sansforgetica.rmit.edu.au>

```
var FontName = ""
if (Font == "Verdana") {
    FontName = "Verdana"
} else if (Font == "Comic Sans") {
    FontName = "Comic Sans MS"
} else if (Font == "Arial") {
    FontName = "ArialMT"
} else if (Font == "Helvetica") {
    FontName = "Helvetica"
} else if (Font == "SansForgetica") {
    FontName = "SansForgetica-Regular"
}
```

```
OneText.font = UIFont(name: FontName, size: CGFloat(FontSize))
```

5.5 Student and General Home Page

On both the student and general user home page they can generate their own spelling test. They are able to pick what level they want the spelling to be, higher the level, harder the test and the user is able to pick the number of words they want to complete. The maximum level is 3. Each level is connected to the amount of letter in the words.

- Level 1 - No more than 4 letter in the word.
- Level 2 - No more than 6 letter in the word.
- Level 3 - No more than 8 letter in the word.

The maximum number of words is 30, this is only to set a limit for the user to stop them picking an over zealous number of words which mean the application could potently stop running, and, or slowing down an extreme amount. Once the user has pressed the 'Start Spelling' button, the level and number of words integers are transferred to the spelling test view controller so that the spelling test is conducted with the value the user wanted. [Figure 26]

5.5.1 Student Spelling Test – Teachers Set

On the student account, not the general account, there is also the option of the student taking the spelling test that the teacher has set. The teacher will add a test to the class. This test will be saved into the database with a unique Test_ID and the foreign key of the matching Class_ID for that test. The test consisting of ten words of the choice of the teacher. The spelling test view controller is loaded and run in the same way, however, the web scraping for the random words is not conducted instead the words are taken from the database test table that have the correct Test_ID. [Figure 27] and [Figure 28].

5.6 Student and General Spelling Test Page

The spelling test page uses functions to make this page easy to navigate due to the fact that this page needs to conduct many different tasks to make the spelling test successful. The spelling test conducts different types of tasks including web scraping for information and images and speech synthesizer, getting the randomization of the letter being removed from the users word to spell, checking the users answer and also the feedback of the spelling test.

5.6.1 Web Scarping

Using SwiftSoup lets the application scrape the website and then select the correct part of the website using the html, div and class or attributes. SwiftSoup is an imported, package which is added to the application on XCode as an external platform, meaning that the functions are able to be used within the development code. The application is able to scrape the website and extract information to be able to gain what is needed for the application to run smoothly. Within the spelling test view controller web scraping, SwiftSoup, is used a total of 4 times to gain the random words that the user will be asked to spell, the feedback part including the definition and the sentence, and finally the icon, image, the user will

be spelling the word of.

```
func ScrapRandomWords() -> Array<String> { // Receiving the random words
from a random word generator website.
    let RandomWordURL =
URL(string:"https://creativitygames.net/random-word-
generator/randomwords/8")!
    let HTML = try! String(contentsOf: RandomWordURL)
    let Document = try! SwiftSoup.parse(HTML)
    let Results = try! Document.select("#randomwordslist").first()! //
Receive the correct part of the html from the website.
    let PlainTextResult: String = try! Results.text() // Removing the
html.
    let ArrayWords = PlainTextResult.components(separatedBy: " ") //
Setting each word to a new index in the array of words.
    return ArrayWords
}
```

Above is the code used to scrape the website <https://creativitygames.net/random-word-generator/randomwords/8> and gain the random word list. This list is then separated into individual words and placed into an array. The function is called as many times as the application needs to gain the number of words the user would like to complete.

```
func DictionaryWebScraping() -> String { // Receiving the definition from
a dictionary website.
    let DefinitionURL =
URL(string:"https://www.collinsdictionary.com/dictionary/english/(Spellin
gTestWords[0])")!
    let DefinitionHTML = try! String(contentsOf: DefinitionURL)
    let DefinitionDocument = try! SwiftSoup.parse(DefinitionHTML)
    let DefinitionSelect = try!
DefinitionDocument.select(".sense").first()! // Selecting the correct part
of html.
    let DefinitionResult = try! DefinitionSelect.text() // Removing
the html.
    let Definition = DefinitionResult.components(separatedBy: ".") //
Only taking the first sentence.
    return Definition[0]
}
```

Above is the code that is used to scrape the Collins dictionary to gain the definition of the words. The code takes the word that the user is currently spelling and gains the definition to then later be placed in the feedback array. Since the Collins dictionary normally gives a paragraph for the definition, the result has been separated by sentence and only the first sentence is being returned as can be seen in the last two lines of the function.

```

func SentenceWebScraping() -> String { // Receiving the sentence from a
dictionary website.
    let SentenceURL =
URL(string:"https://www.collinsdictionary.com/dictionary/english/\(Spellin
gTestWords[0])")!
    let SentenceHTML = try! String(contentsOf: SentenceURL)
    let SentenceDocument = try! SwiftSoup.parse(SentenceHTML)
    let SentenceSelect = try!
SentenceDocument.select(".quote").first()! // Selecting the correct part
of html.
    let SentenceResult = try! SentenceSelect.text() // Removing the
html.
    return SentenceResult
}

```

Above is the code that is used to scrape the Collins dictionary for the sentence, it passes in the URL with the spelling test word that the user is completing. Then it selects the class called quote, however, it just chooses the first one as there can be many on the one page. It then turns the HTML that was selected into plain text. This means it is just a sentence that is returned. Later in the code this is added to the feedback array.

```

func DisplayImage() { // Receiving the icons from an icon generator
website.
    let IconURL = URL(string:
"https://www.flaticon.com/search?word=\(SpellingTestWords[0])")!
    let IconHTML = try! String(contentsOf: IconURL)
    let IconDocument = try! SwiftSoup.parse(IconHTML)
    let IconSelect = try! IconDocument.select(".icon--
holder").first()!
    let IconSelectNext = try! IconSelect.select("a").first()!
    let IconResult = try! IconSelectNext.attr("href")
    // The code above receives the link to the icon page so that the
url for just to icon can be received.
    let IconPrintURL = URL(string: IconResult)!
    let IconPrintHTML = try! String(contentsOf: IconPrintURL)
    let IconPrintDocument = try! SwiftSoup.parse(IconPrintHTML)
    let IconPrintSelect = try! IconPrintDocument.select(".main-icon-
without-slide").first()!
    let IconPrintSelectNext = try!
IconPrintSelect.select("img").first()!
    let IconPrintResult = try! IconPrintSelectNext.attr("src") //
Receive the url for the image
    UIImageView.load(urlString: IconPrintResult) // Allows the image url
to be able to be displayed.
    view.addSubview(UIImageView) // Displays the icon.
}

```

The code above is how the application scrapes for the icon. The icon is completely random and there is no way of controlling the outcome of what the user will see in the icon. First the function scrapes the flat icon website with the spelling word to gain the link to the individual icon being displayed on the flat icon website. Then the SwiftSoup will re-scrape the website with the URL that has been gained by the website itself to get the JPEG. This is then put through an extension on the app delegate that means it can be loaded onto the UIImageView.

5.6.2 Hint

An extra hint of the word is it being read to the user. This is done by importing AVFoundation and then creating a synthesizer that has the word input into the utterance. The voice is then selected. This would be something to update later as the voice is not the most life like and easy to listen to and then the application speaks the word.

```
@IBAction func HintPressed(_ sender: Any) {
    // When the hint button is click a voice reads the word to give
    the user a hint.
    let Synthesizer = AVSpeechSynthesizer()
    let Utterance = AVSpeechUtterance(string: SpellingTestWords[0])
    Utterance.voice = AVSpeechSynthesisVoice(language: "en-GB")
    Utterance.rate = 0.4
    Synthesizer.speak(Utterance)
}
```

5.6.3 Right Number and Level of Random Words

To make sure that only the correct number and level of words are presented in the spelling test that the user has previously chosen, there needs to be two checks completed. The function check looks at the length of the word based of the level. If this is correct then it is added to the array, storing the word that will later be used for the spelling test. Then in the main body of the code when the spelling test view controller is loaded up, once it has got a list of words that are in the correct level, it should check the length of the array as too many numbers of word can be added. So, an 'If' statement is used to remove any extra words until the number of words is equal to the length of the array, fitting the two values the user has preset.

```
func SpellingTestWord(Arraywords: Array<String>) { // Making sure the word
length fits the level the user has picked.
    let ArrayWords = Arraywords
    for Words in ArrayWords {
        if (Level == 1) && (Words.count <= 4) {
            SpellingTestWords.append(Words)
        } else if (Level == 2) && (Words.count <= 6){
            SpellingTestWords.append(Words)
        } else if (Level == 3) && (Words.count <= 8){
            SpellingTestWords.append(Words)
        }
    }
}
```

```
if (SpellingTestWords.count > NumberWords) {
    var RemoveWords = SpellingTestWords.count - NumberWords
    repeat {
        SpellingTestWords.removeLast()
        RemoveWords = RemoveWords - 1
    } while RemoveWords != 0
}
```

5.6.4 Displaying Spelling Test

When displaying the word for the spelling test, depending on the length of the word, a certain amount of letters will be removed for the user to input. The number of letters to be removed is based on the

length divided by two. Remainders are ignored in the formula. Less character can still be removed due to the randomisation of the removal. A random number is generated based of the length of the word and then that character is removed and replaced with a char holder, '&'. If the random number reappears the process will just carry on as normal. Once the repeat loop is complete the updated word will be placed in the spelling test word update array and then the 'for loop' will carry on. This process means that the spelling test is completely random. Randomisation makes the application process more interesting and means if there are repeated words the spelling test would most likely be different.

```
func UpdateWordList(SpellingTestWords: Array<String>) -> Array<String> {
// Removing some letters for the user to input for the test.
    for Words in SpellingTestWords { // Looping all the words in the
        spelling test.
            var ArrayHolder = Array(Words) // Turning the word into a char
            array.
            var RemoveLetterAmount = ArrayHolder.count / 2 // Working out
            the amount of letters to remove.
            repeat {
                let Letter2Remove =
Int(arc4random_uniform(UInt32(ArrayHolder.count))) // Generating a random
number within the length of the word.
                ArrayHolder[Letter2Remove] = "&" // Removing the letter
from the word and replacing it.
                RemoveLetterAmount = RemoveLetterAmount - 1
            } while RemoveLetterAmount != 0
            let StringHolder = String(ArrayHolder) // Putting the array
back to a string.
            SpellingTestWordsUpdated.append(StringHolder)
        }
    }
    return SpellingTestWordsUpdated
}
```

5.6.5 Checking Users Spelling

Below is the code for the checking the user spelling and determining if it is correct or not. This is done by counting how many texts fields there are. If the text field input is equal to the index of the spelling word, then that is counted as correct. Finally, once it runs through all the text fields if the amount of text field and amount of correct counted are equal, then the word is spelt correctly. This was the simplest solution that could be thought of since the user is not inputting the whole word only certain letter of the word.

```

func UserAnswer() -> String { // Checking if the users answer is
correct or incorrect.
    var CorrectNeed = 0
    var CountCorrect = 0
    let SpellingWord = SpellingTestWords[0]
    var Index: Int = Level - 3
    for TextFields in TextCollection! { // Running through the 8 text
fields.
        if (TextFields.isHidden) == false { // If the text field is
shown, e.g. the user has input a char into it.
            CorrectNeed = CorrectNeed + 1 // Seeing how many answer
the user need to input.
            let UserValue = TextFields.text // Receive the user input.
            let RightLetter = String(Array(SpellingWord)[Index]) //
Getting the correct value that should be displayed.
            if (RightLetter == UserValue?.lowercased()) { // Checking
the correct answer and the user answer are the same.
                CountCorrect = CountCorrect + 1 // If they are add to
user is correct count.
            }
            Index = Index + 1
        }
    }
    if CorrectNeed == CountCorrect { return "Correct" } // Checking
that user correct and count correct is the same e.g. the word is spelt
correctly.
    else { return "Incorrect" }
}

```

5.6.6 Feedback - Hide and Show

The feedback for the user is split into 4 parts, these include if the word the user spells is correct or incorrect, the word spelt correctly, the definition of the word and the word used in a sentence. The definition and sentence are gained through web scraping as describe above. The word spelt correctly is gained from the spelling test word array and the correct or incorrect is gained by a mathematical process again described above. All the feedback can be accessed in the feedback view, however, the feedback is either shown or hidden based on either what the user has picked, or the teacher has picked for the student. If, however the user wants to see a piece of feedback that is hidden they can press the show button and it will be displayed on the feedback view.

5.7 Teacher Account

5.7.1 Home Page - Display/Adding/Deleting/Access Class

In the teachers account, home page, it displays all the classes that are in the database which match their Teacher.ID and places them onto buttons so that the teacher can either delete or access the class to gain more information. This is the code displayed below. The teacher is able to add a class to a maximum of 6 and can only add if there is no more than 6 or a class is deleted. If the teacher deletes a class it gets deleted from the database. The student is still able to access their account but there will only be the basic feedback shown and it cannot be edited and there will not be any teacher test set. If there are less than 6 classes the teacher can add a class. They input the class name and year group which is able to be anything the teacher decide to be desirable but the access code needs to be unique. So if it is already in

the database the class will not be able to be added. If the teacher wants to access the class to gain more information, they click the button of the desired class and from the alert presses 'access' this moves to the class page (discussed later).

```
var Select: OpaquePointer?
    let ClassSelectQuery = "SELECT Class_Name FROM Class WHERE
Teacher_ID = '\(Teacher_ID)';"
    // Receiving all the teacher classes, the class name from the
    database.
    if sqlite3_prepare(db, ClassSelectQuery, -1, &Select, nil) ==
    SQLITE_OK {
        while(sqlite3_step(Select) == SQLITE_ROW) {
            guard let Class_Name = sqlite3_column_text(Select, 0) else
{
                print(sqlite3_errmsg(db)!) // Print the error.
                return
            }
            // Placing the class names into an array.
            let Result_Class_Name = String(cString: Class_Name)
            Class_Name_Array.append(Result_Class_Name)
        }
        var Position = 0
        // Loop through all the button and setting the title to the
        class name if there is no class names left otherwise hiding the button.
        for button in ButtonCollection {
            let Statement = (Class_Name_Array.count > Position)
            if (Statement == true) {
                button.setTitle(String(Class_Name_Array[Position]),
for: .normal)
            } else {
                button.isHidden = true
            }
            Position = Position + 1
        }
    } else {
        print(sqlite3_errmsg(db)!) // Print the error.
    }
    sqlite3_finalize(Select)
```

5.7.2 Class Page - Display Class Information - Display/Adding/Deleting Test

When the teacher accesses a class, it takes them to a class page that displays the class name, year group, access code, feedback (setting/choice), any tests in the database for that class and also allows an add test feature.

The teacher will only be able to go to the add test view controller as long as there is less than 3 spelling tests already present in the database. If there are 3 tests already connecting that Class_ID, then a test will need to be deleted. To do this the teacher clicks on the test that they want to delete and then presses delete. This in turn removes the test from the database meaning that it no longer will appear on both the teacher and student account. This allows the teacher to add a new test to that Class_ID.

5.7.3 Student Feedback Change

Above it was described that the class page has the feedback feature, this displays switches and the different types of feedback including correct/incorrect, word, definition, and sentence. This is based on what the teacher wants shown to the student. Teachers are able to switch the feedback either on or off and that will be portrayed in the student account when they have conducted a spelling test and the feedback view appears. This is coded through the database as it saves the teachers preference to the database and the student account accesses the class member table to extract the feedback that will be displayed. [Figure 29] and [Figure 30]

5.7.4 Adding Test - Displaying on Student Account

The test works the same way as the feedback setting. Once the teacher has added the test to the class that the student is a member of. The test will get saved to the database with the foreign key Class_ID, this is linked to both the student and teacher. Then when accessing the student account, the application gains any test in the database that matches the Class_ID to the student and displays them on the home page allowing the student to access and complete these tests. [Figure 27] and [Figure 28]

5.8 Unforeseen Problems

When coding the application, some unforeseen problems were encountered. There were three main ones that will be discussed in the paragraphs below. These were as follows: difficulties with the database, Problems with the web scraping and gaining the correct amount and finally, the lack of documentation for the language chosen.

Connecting the application to the database seemed to incur some errors that were not expected. It was believed, due to the fact that there was no updated SQLite3 and swift help documentation for the current software updates that the way in which it was known to implement the database with the application no longer worked. To overcome this problem inspiration from multiple websites were compiled to get a working database. This was applied away from the application to conduct testing to make sure that all parts were working. Then the database was placed back into the working application. The app was then able to extract, and input information into the database, and update and change the database were needed.

Another problem that occurred when coding 'SPELLBEE' was gaining the correct information from websites for the web scraping and making sure that the website allowed the web scraping tools to work. To overcome this problem a similar solution was used as for the database as described above. Separate too the application, test code was created to make sure that the website could be scraped and have the correct data extracted from the website in plain English. Not including all the HTML. Once the code was fully working and all the external websites had been chosen, the code was placed back into the 'SPELLBEE' application. The variables were all changed to match the application and connective paths were made to the class. This in turn made the code run smoothly overcoming this unforeseen problem.

The code which was chosen for this project was Swift. This is a small coding platform and does not have as widespread use as other platforms, for example, python, java or c++. There is considerably less documentation to help the programmer learn all the professional coding techniques of the language. When problems were encountered help documentation would have been of great aid. Other websites created by

swift programmers or conversation platforms, like Stack Overflow enabled the problems to be fixed but the coding process would have been able to be sped up if there was a designated help documentation.

6 Results and Evaluation

The decision was made due to the fact that the development was just a prototype that user testing on random people of the public would not take place. This was also due to the fact that it would require more ethical approval and because of the time frame it did not seem possible. The time frame given for this project was 12 weeks, from start to finish. The testing that was undertaken was conducted after development, and each part of the app was tested with different values and variables. For example the application would work to its design specification or an error message would appear as there was a human error. Below are a couple of the test cases from this testing, the test case below show the main functionality of the program, and the main requirements set by the client and the proposal. Further examples are able to be located in the appendix, this included all the other functionality and features of the code. Some other tasks which would be considered to be part of the testing process was showing the app to the client and receiving feedback. Improvements that could be made to meet the standard of application that the client themselves imagine when writing the proposal. This is able to be found later on in the '*User Testing - Client*' section of the report, 6.1.

6.1 User Testing – Client

During the course of the development of the project due to the fact, that the application is only a prototype, the decision was made not to conduct user testing. Mainly as the app was not at its point of full development. Test cases are conducted as can be seen below and all these passed successfully. Demo videos can be located in the archives to show the successful application. The application was shown to the client to get client testing and feedback, during this meeting the feedback consisted of:

- Making sure there is more consistency throughout, meaning capital letters throughout on the feedback word and changing of the font and font-size. The feedback word was updated to be capital letters. A future work from the client's feedback would be to make sure the setting page will update all the fonts and font sizes.
- Another area of opportunity was on the test that the teacher set. This was for there to be a name describing the test to aid both the teacher and student. This would also be future work to be conducted in development later.
- The speed of the application was noted to be slow, not to the point of making the application unusable, yet it was noticed. Running parts of the code simultaneously instead of one part at a time. This would mean in turn the application would process and run faster. Linking to this it was mentioned by the client that their preference was for the user to have the option and feature they currently have than sacrificing that for speed of the application.
- The icon could allow the teacher to pick the image out of 3 – 5 random generated icons for that word. This would mean the teacher could have control removing any issue that could arise due to an error of sensitive icon being displayed.
- Finally allowing the teacher to gain the results of the student's test once it has been completed and stopping the student from taking the test again. This would give the teacher even more feedback from the student account and help the teacher improve the student's spelling further.

One part of the feedback that was discussed was since the letter being taken out was completely random, a process called gamification has occurred, this was pointed out by the client himself expressing that it draws him in. This means that the application is more engaging.

6.2 Test Case

Test Case ID: 308		Test Purpose: Changing the feedback display for a class in that class view controller.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Access the class.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	The user changes the switches based on if they want the feedback on or off. correct/incorrect - on word - on definition - off sentence - off	The correct/incorrect switch is turn on. The word switch is turn on. The definition switch is turn off. The sentence switch is turn off.	Pass [Figure 29]
1.2	Log off teacher account.	The feedback selection updates on both the application and the database.	Pass
1.3	Log on to student account.	The student account is accessed.	Pass
1.4	Start a spelling test.	The spelling test view controller is loaded on the application.	Pass
1.5	Complete first spelling and press submit spelling.	The feedback is the same as what the teacher has set.	Pass [Figure 30]
2.1	The user changes the switches based on if they want the feedback on or off. correct/incorrect - off word - off definition - on sentence - on	The correct/incorrect switch is turn off. The word switch is turn off. The definition switch is turn on. The sentence switch is turn on.	Pass [Figure 41]
2.2	Log off teacher account.	The feedback selection updates on both the application and the database.	Pass
2.3	Log on to student account.	The student account is accessed.	Pass
2.4	Start a spelling test.	The spelling test view controller is loaded on the application.	Pass
2.5	Complete first spelling and press submit spelling.	The feedback is the same as what the teacher has set.	Pass [Figure 42]
3.1	The user changes the switches based on if they want the feedback on or off. correct/incorrect - off word - on definition - on sentence - on	The correct/incorrect switch is turn off. The word switch is turn on. The definition switch is turn on. The sentence switch is turn on.	Pass [Figure 43]
3.2	Log off teacher account.	The feedback selection updates on both the application and the database.	Pass
3.3	Log on to student account.	The student account is accessed.	Pass
3.4	Start a spelling test.	The spelling test view controller is loaded on the application.	Pass
3.5	Complete first spelling and press submit spelling.	The feedback is the same as what the teacher has set.	Pass [Figure 44]
Comments: All the test passed successfully.			
Related Test: 303			
Author: Samantha Hughes			

Table 1: The table is the test case for "Changing the feedback display for a class in that class view controller." for a Teacher User.

Test Case ID: 102		Test Purpose: Starting a spelling test, of different levels and number of words.		
Layout: XCode, Version 13.2.1, MacBook Pro				
Preconditions: Logged into their account.				
Test Case Steps:				
Step No	Procedure	Response	Pass/Fail	
1.1	Selecting the level to number 1, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 31]	
1.2	Selecting the number of words to 1, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 31]	
1.3	Press Start Spelling button.	The spelling test view controller is loaded with the first spelling to complete.	Pass	
1.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass [Figure 32]	
1.5	Press submit spelling.	Feedback view appears.	Pass [Figure 72]	
1.6	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass [Figure 45]	
2.1	Selecting the level to number 2, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 31]	
2.2	Selecting the number of words to 2, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 31]	
2.3	Press Start Spelling button.	The spelling test view controller is loaded with the first spelling to complete.	Pass	
2.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
2.5	Press submit spelling.	Feedback view appears.	Pass	
2.6	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
2.7	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
2.8	Press submit spelling.	Feedback view appears.	Pass	
2.9	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass	
3.1	Selecting the level to number 3, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 31]	
3.2	Selecting the number of words to 3, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 31]	
3.3	Press Start Spelling button	The spelling test view controller is loaded with the first spelling to complete.	Pass	
3.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.5	Press submit spelling.	Feedback view will appear.	Pass	
3.6	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
3.7	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.8	Press submit spelling.	Feedback view appears.	Pass	
3.9	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
3.10	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.11	Press submit spelling.	Feedback view appears.	Pass	
3.12	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass	
Comments: All the test passed successfully.				
Related Test: 101				
Author: Samantha Hughes				

Table 2: The table is the test case for "Starting a spelling test, of different levels and number of words." for a General User.

Test Case ID: 203		Test Purpose:Starting a spelling test, that the teacher has set for the student.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account.			
Test Case Steps:			
Step	Procedure	Response	Pass/Fail
1	Press the Test.ID – Click here to start button.	The spelling is loaded, with the first teacher set word and run through all the words that have been set.	Pass [Figure 77]
2	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
3	Press submit spelling.	Feedback view will appear.	Pass
4	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
5	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
6	Press submit spelling.	Feedback view will appear.	Pass
7	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
8	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
9	Press submit spelling.	Feedback view will appear.	Pass
10	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
11	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
12	Press submit spelling.	Feedback view will appear.	Pass
13	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
14	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
15	Press submit spelling.	Feedback view will appear.	Pass
16	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
17	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
18	Press submit spelling.	Feedback view will appear.	Pass
19	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
20	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
21	Press submit spelling.	Feedback view will appear.	Pass
22	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
23	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
24	Press submit spelling.	Feedback view will appear.	Pass
25	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
26	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
27	Press submit spelling.	Feedback view will appear.	Pass
28	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass
29	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass
30	Press submit spelling.	Feedback view will appear.	Pass
31	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass [Figure 45]
Comments: All the test passed successfully.			
Related Test: 201			
Author: Samantha Hughes			

Table 3: The table is the test case for "Starting a spelling test, that the teacher has set for the student." for a Student User.

Test Case ID: 305		Test Purpose: Changing the background colour in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the background colour to pink.	On the segment controller pink is now selected.	Pass [Figure 35]
1.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 36]
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the background colour to blue.	On the segment controller blue is now selected.	Pass [Figure 37]
2.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 38]
3.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
3.2	Selecting the background colour to yellow.	On the segment controller yellow is now selected.	Pass [Figure 39]
3.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 40]
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 4: The table is the test case for "Changing the background colour in the setting page." for a Teacher User.

6.3 Discussion

Below are all the requirements that were met and the test cases that links to the specific requirement.

- Fully Complete - Is where the code is fully functional and no improve would need to made apart from normal wear and tear before the application goes online.
- Part Complete - Is where the code is functional however improvements would need to be made before the application goes online.

Requirement	Levels of Completeness	Test Case	Reasoning
Allow the user to be able to login to their account	Fully Completed	101, 201, 301	N/A
Allow the user to register for an account.	Fully Completed	101, 201, 301	N/A
Give the user random generate spelling practices to complete.	Part Completed	102, 202	A checker for rude or incentive word would need to be implemented to make the application ready to go online. However the requirement is fully complete for the specifications of the proposal for the prototype.
Give the user feedback on the spelling based on the user choice or of the teacher.	Fully Completed	308	N/A
Let the user be able to change the colour of the background if needed.	Fully Completed	103, 204, 305	N/A
Let the user be able to change the font if needed.	Fully Completed	104, 105, 205, 206, 306, 307	N/A
The application must be able to be simple to navigate through, so that it is able to be used and accessed by all ages and all abilities.	Part Completed	ALL	User testing would need to be conducted to be able to label this requirement complete.
Allow the teacher to set the feedback to class.	Fully Completed	308	N/A
When the user is signing up / registering to the application, they can pick if they are a student, teacher, or general user.	Fully Completed	101, 201, 301	N/A
Should be able to log off the system and be taken back to the login page.	Fully Completed	107, 207, 309	N/A
Allow the teacher to set spelling test for class.	Fully Completed	203, 304	N/A

Not all requirements where able to be completed. Below is a list of all the requirements from earlier in the report and the **bold** requirements are ones that didn't get completed and the reason why.

6.3.1 Must Requirements

- Allow the user to be able to login to their account.
- Allow the user to register for an account.
- Give the user random generated spelling practices to complete.
- Give the user feedback on the spelling based on the user choice or of the teacher.
- Let the user be able to change the colour of the background if needed.
- Let the user be able to change the font if needed.
- **Give the user an overall result of how many questions they got right and wrong overall.**
 - This was not completed due to time constraints and due to the fact that when discussed with the client it was better to have first hand quick feedback then having to wait until the test is completed. Instead, the user gets to see the result of their input for the spelling after each word and the feedback all appears.
- The application must be able to be simple to navigate through. So that it is able to be used and be able to be accessed by all ages and all abilities.
- Allow the teacher to set the feedback for class.

6.3.2 Should Requirements

- When the user is signing up / registering to the application, they can pick if they are a student, teacher, or general user.
- Should be able to log off the system and be taken back to the login page.
- **The teacher should be able to see a list of their classes and their students in that individual class.** - This requirement was completed mainly for time constraints, and ended up being more of a 'could' requirement for the client as it was more important that the teacher was able to change the feedback for the class, and be able to add tests for the class instead of seeing a list of the students names.
- Allow the teacher to set spelling test for classes.

6.3.3 Could Requirement

- **Could store the answer that were incorrect and allow the user to complete a spelling practise with those words.** - Due to the size of the database, as only a prototype small database was used it was better to hierarchy the more important uses for the database than creating individual tests for the user based off their incorrect answers.
- Allow the user to try again if the user has not put in the correct answer to the spelling question.

-
- **Allow the teacher to set feedback to particular students.** - This requirement was not completed mainly due to time constraints and for the client as it was more important that the teacher was able to change the feedback for the class.

The application also meets the aim of being a simple and effective design making it easy to navigate through, meaning the usability of the application is to a high standard and meeting this goal of the proposal. It also meets the proposal as it does not only help the user improve their spelling and literacy's but also helps people that have dyslexia understand further.

To gain a feeling on how usable the application created was using the 10 principles of Nielsen's heuristics have been discussed below: [1]

1. **Visibility of system status** - This principle is about making sure that the user stays informed at all times when using the application. In the 'SPELLBEE' app it doesn't give the user much feedback in the way of when the application is loading, however the run time of the application even though longer it is still a reasonable response time.
2. **Match between system and real world** - Principle number two is making sure that the application is speaking the user language, throughout the 'SPELLBEE' app, simple but effective language was used and icons for buttons were common and well used in public. Also the average time it takes for the 'SPELLBEE' application to load the spelling test view controller from the home page is 4 seconds 14 milliseconds on average. This transition was picked at it is the longest gain of view controller. For the real world this is acceptable as 10 sec is normally the limit for keeping user attention, anything passed this, the user is most likely to exit to application. [21]
3. **User control and freedom** - This is making sure that the user has a clear exit strategy, on all of the view controllers is a log off button to take the user back to the login page and/or a back button to take the user back to the previous page they were on.
4. **Consistency and standards** - Throughout the application there is consistency, meaning not only the same colour scheme and features, but also has consistency with the language used for buttons and questions.
5. **Help users recognize, diagnose, recover from errors** - In the application any errors that occur get changed into plain English and then tells the user what error has occurred by an alert popping up. This directs user on how to fix the issue.
6. **Error prevention** - Not much error prevention was needed in the application, but when the teacher deletes a class test then error prevention is used to make sure that the teacher doesn't delete the test by accident.
7. **Recognition rather than recall** - As described above the buttons used either consist of icons or plain easy to understand language, and the application is made too be able to get from one view controller to another in a short and precise time, reducing the need for recall.
8. **Flexibility and efficiency of use** - This principle would be one to look into improving in the future work of the application.

-
9. **Aesthetic and minimalist design** - The 'SPELLBEE' app has a design which flows through each view controller with the same colour scheme and features. The same language is used which is plain and only contains the key information and the wording is short and precise.
 10. **Help and documentation** - Extra documentation was not needed in the case of the 'SPELLBEE' app. There are help buttons through the application letting the user know a brief summary of how the page works.

Through the principals above it has proven the theory that the application have good usability and functionality, which was a main aim of the proposal.

Within the application as seen above when it came to the test case all passed successfully. This shows that the application is working well, even with some improvement still needed to be made. Also, in the application there are key aspect which have gone well and aided the creation of the application these include web scraping, which was discussed in more detail in the implementation section of the report and the randomisation of the letter removal, this is discussed in more detail in the 'User Test – Client' section.

With all applications there is always room for improvement and area that could have been coded better in the initial stages. There are two main parts of the 'SPELLBEE' application that fits into that description. One is the randomisation of the web scraping for the word in the spelling test. There is no filter of the words and so it is possible that a rude or inappropriate or sensitive word would be able to appear on the spelling test view controller. There is no security or child safety with the random words. The second parts that falls into this description is that similarly the random icons that are selected are unknown the similar issues can occur. This means unfortunately that the icon that could appear, may not be friendly to the user themselves. This problem would need to be fixed before the app was to go online. The improvement which have just been discussed are considered and talked over in the future work 7 section of this report.

7 Future Work

Through this process of creating the spelling application there have been many learning's enabling the production of a spelling application have has been completed and the aims and targets reach. However, there was points in the application where extra functionality would have been added to improve the end result.

- Spelling Rule

There are lots of rules when it comes to spelling for example

“‘i’ before ‘e’ expect after ‘c’”

“‘u’ after ‘q’”

And many more

The government have a pdf document with many rules and guidance on spelling for up to year 5 and 6. [11]. Adding rules to the application as a help to the user to improve their spelling future. This could either be implement mimicking the hint already implemented and the correct rule could be displayed for that word. If the user only wants to see the rules of spellings there could be another help feature that gives an alert that displays the website with all the spelling rules. This would be a good addition to the application and help the user to improve further.

- No limit of amount of Class and Test

At this current point in time the application only allows the teacher to set 3 tests per class and only allows the teacher to have 3 classes. Therefore to improve this application in the future the limit could be taken off to allow the teacher to set as many tests as they desire and further let the teacher set a due date for the student to complete the test. Then once the student has completed that test marked as done, no longer showing the student the test and letting the teacher see who has completed the test. Making the test more like homework style. The teacher also has a limit on how many classes they are allowed to use in the application which could reduce the use of the application at the present time. In future work the limit would be removed.

- Android

The application currently can only be run on IOS applications, and more particular can only be able to be run on a smartphone that is compatible with IOS applications. In the future the application would need to be able to be run on not only a smartphone but tablets, laptops and any other types of technology. Also create the application so that it could be used on other operating systems, for example android, this would make the application more accessible to a wider range of people.

- User testing

There is also the topic of making the application public on an app store. To complete this task and to make it successful there would need to be user testing completed. To conduct this ethical approval would be required. For the user testing a split of participants into equal groups of people that have dyslexia and do not have dyslexia, then split the two groups into equal groups of teachers, general, student and all. Then ask all groups to test the application and give feedback based on their experience. From the feedback the application can be updated and then sent out onto the app store.

- Icon Database

As discussed through the results and evaluation sections, there could be issues when it comes to the icon being randomly generated, therefore a future work plan would be to create a database that will be constantly updated with safe icons. This will mean that there will be no inappropriate icons appearing.

- Spelling Words Database - Rude or Insensitive

There is the same issues with the random words which are generated for the spelling test as there is for the icons. As there is no safety net the words that could appear which are rude or insensitive, in this case over time there could spelling words database of rude and incentive, there is the random word for the user is to spell falls into that database then it would be removed from the spelling test. There is also the opinion of putting children safety parameter on the application so the words that appear are child friendly.

8 Conclusions

Through the research and approach the program was planning to help assist any user of any age improve their literacy by completing one word spelling test and gaining the appropriate amount of feedback. The application was also planning to aid people with dyslexia by allowing them to change the background, and font and font size, to try and mimic document tools that help people read such as coloured overlays. All these aims were achieved in the programming and development stage. Backed up with the research conducted at the beginning of the report and the results of the test cases confirm that the application is working successfully.

Overall, it is believed that the approach taken to complete the proposal has been successfully achieved. All the aims and targets completed to make an effective project. There are still issues within the code including, performance time and the success rate of completion. The code takes a slower than average speed time to complete tasks. On talking to the client the speed it takes to complete is not an issue for the client. The success rate of completion is an issue due to the fact that occasionally the code will crash while running, however, there is no visible problem with the code. It is believed to crash when it is the first time of running the application when XCode has firstly been loaded up on the computer. These issues would be fixed later during the user testing, which was encouraged by the supervisor not to complete at the current time, and final development staging, as the current code was only designed to be a prototype. The research conducted at the beginning of the project aided in giving justification for the application design, features, and functionality. These application created is not only an effective spelling test app to improve the UK adult average literates' skill and or age, but the features combined aid in the usability of people that suffer with learning disability, manly looking toward people that suffer with dyslexia. There are demo videos located in the archive demonstrating the application working.

9 Reflection on Learning

Throughout this process of completing this report and developing the application, skills have not only been important but also gained, to make this project successful.

In the individual parts of the project for example development, technical skill were learnt. The main skill that were gained was learning the software XCode and the coding language swift. Alongside this learning the database SQLite3 that was connected. Some coding techniques learnt include:

- Function
- Extension
- Web Scraping
- Randomization

Understanding and being able to implement these coding functions became a keen learning of the coding and development.

Within the reporting area there were skills learnt around obtaining background research. Skills were created on extracting the key information from articles and website. In turn making and defining referencing the correct sites and with the correct information. This research skill that was developed throughout the background research was also relevant for the approach section.

The ethical approval and survey at the beginning of the dissertation went well as it did not only allow the ability to gain information which is not able to be found online, but also allow for the application to be tailored to the requirements of the population and people which dyslexia both having a need for the application. Having the survey reduced the need for user testing at the end of the development, due to the fact that this was only a prototype but also because the interface was already designed in line with the survey results.

Reflection on not completing user testing at the time, there would have most likely been time to get ethic approval looking back. As the project management was of a high standard and the management of task was consistently being updated, when this were completed on time, early or needing more time. However the ethical approval process would have needed to be to started as soon as the first approval was given. Even if ethical approval was receive in time there is also the issues of trying to getting an unbiased group of participants, and finding a way of conducting the user testing with the program only being a prototype at this current point. The decision to not go through with user testing was made at time in conjunction with the supervisor at the time was the most feasible.

The development process was something that was challenging but also very rewarding. This is due to XCode, swift, being a brand-new language that needed to be learnt. As it was the case there are parts of the code that looking back on them now with increased knowledge of XCode could have been coded in a more streamlined and effective way. An example of this is discussed below.

If this project was to be started again, some decisions would have been made differently. This includes the

interface flow. As standing, the application is created using different settings page for all 3-account types, and 2 similar home pages. To improve this next time the pages which were almost duplicate would have not been repeated and deepening on the account type would determine what is hidden or shown to the user.

It is still apparent that XCode at the time was the most sensible and viable language and platform to choose. There is an obvious downfall that the application is not able to be used for android devices, but the application that was produced fulfilled the proposal and fitted the requirements of the user stories, personas.

Another part that would have been completed if the project was to be started again, would be that user testing would be done. Due to the condition of needing to get ethical approval to complete user testing on the public, and because of the time constraints of the project the decision was made in conjunction with my supervisor; not to undertake testing on the public for the purpose of this development. The evaluation would have been improved and there would have been more information produced through user testing being completed.

Working with a client was very beneficial. It provided formal information which guided the direction of the development to fit the functionality and align with the client's needs. Having a client separate from the supervisor meant more insight could be gained in all areas such as development and reporting, to gain the full potential from this dissertation.

Throughout the project meetings were arranged and placed evenly in the report and development process. One meeting at the beginning to discuss the proposal and what the client wanted to from the application. One in the beginning of the approach and development step. This was to make sure that the development of the application was the correct idea for what the client had in mind. Finally, there was a last meeting at the end of the development. This was to receive the client feedback to gain an idea if the proposal had been fulfilled completely and correctly. This meeting really aided in the completion of the dissertation and getting some user feedback due to the fact that there was not any user testing once the development was finish. All meetings were well placed in the timeline and were well prepared for, with both questions and documentation to show the client, and the knowledge of what feedback needs to be retained.

Within the approach and looking at the methodology of the hybrid waterfall method was the correct one chosen. It turned out to be a great deal more agile in it's working, mainly since XCode, swift was being learnt as the process was developing, learning on the job. All the requirements were laid out before, helping the whole process flow very smoothly.

Overall, this dissertation has been an amazing learning curve to gain more skills in reporting work which has been completed and learning a new language for the implementation, that's meant that the proposal was met and achieved.

A Appendices

A.1 Survey

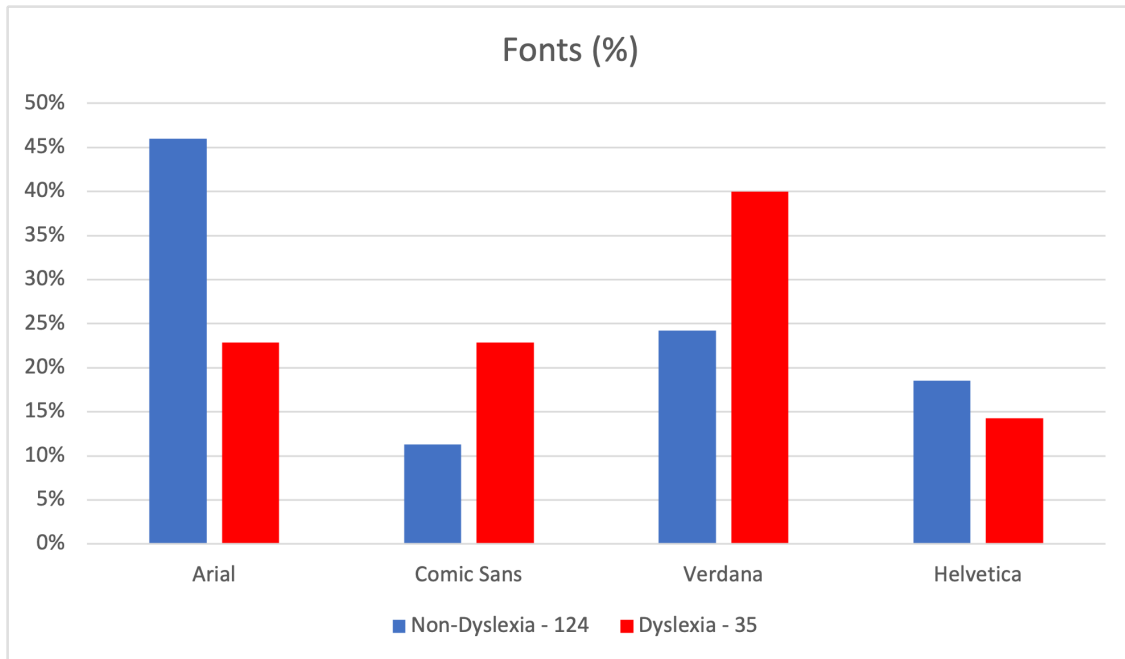


Figure 1: Graph for fonts results from 'Spelling Test' survey.

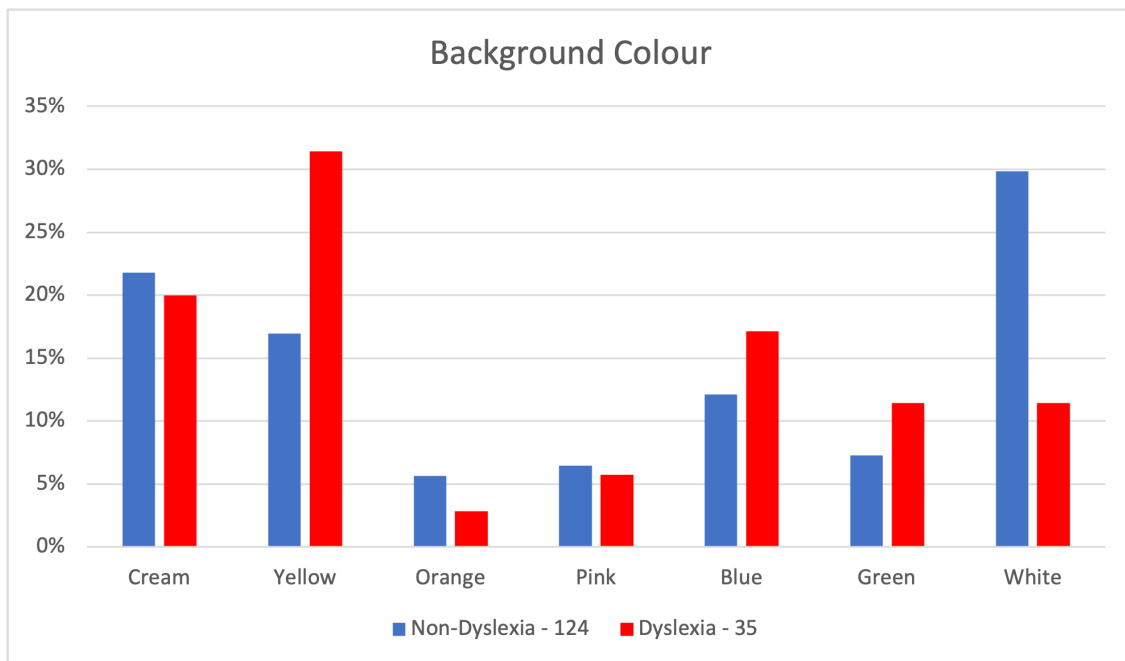


Figure 2: Graph for background colour results from 'Spelling Test' survey.

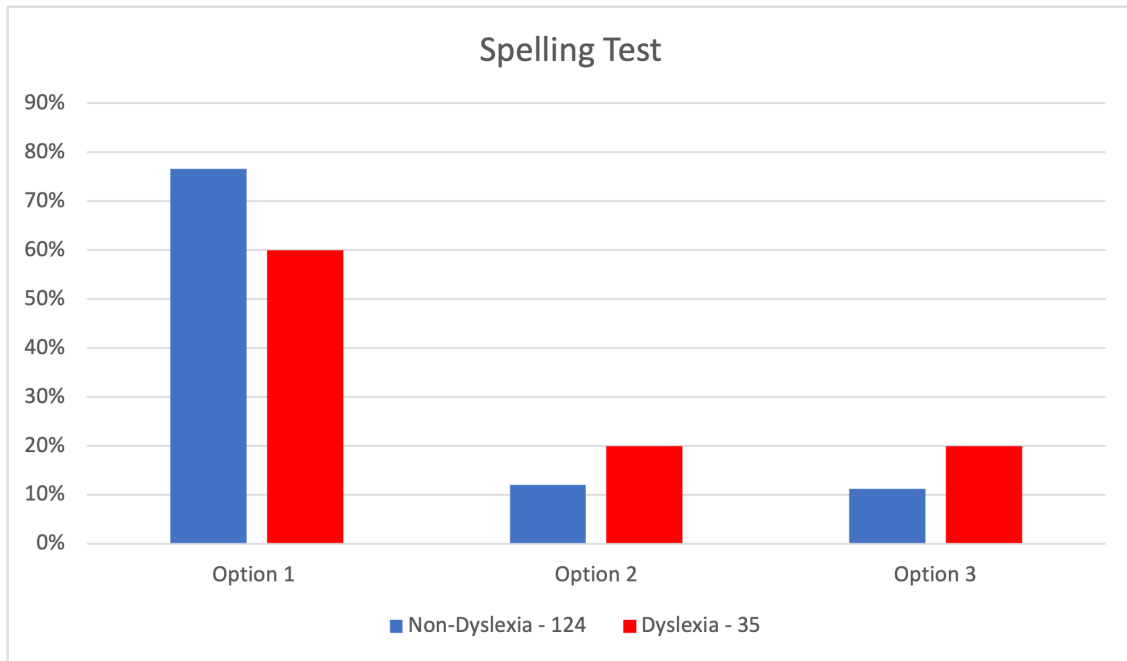


Figure 3: Graph for spelling test results from 'Spelling Test' survey.

☐ Option 1

H _ _ S E 

☐ Option 2


The _ _ _ _ _ was yellow!

☐ Option 3

 (ONCLICK "house, spell house?")

Figure 4: Screenshot of the spelling test question from 'Spelling Test' survey.

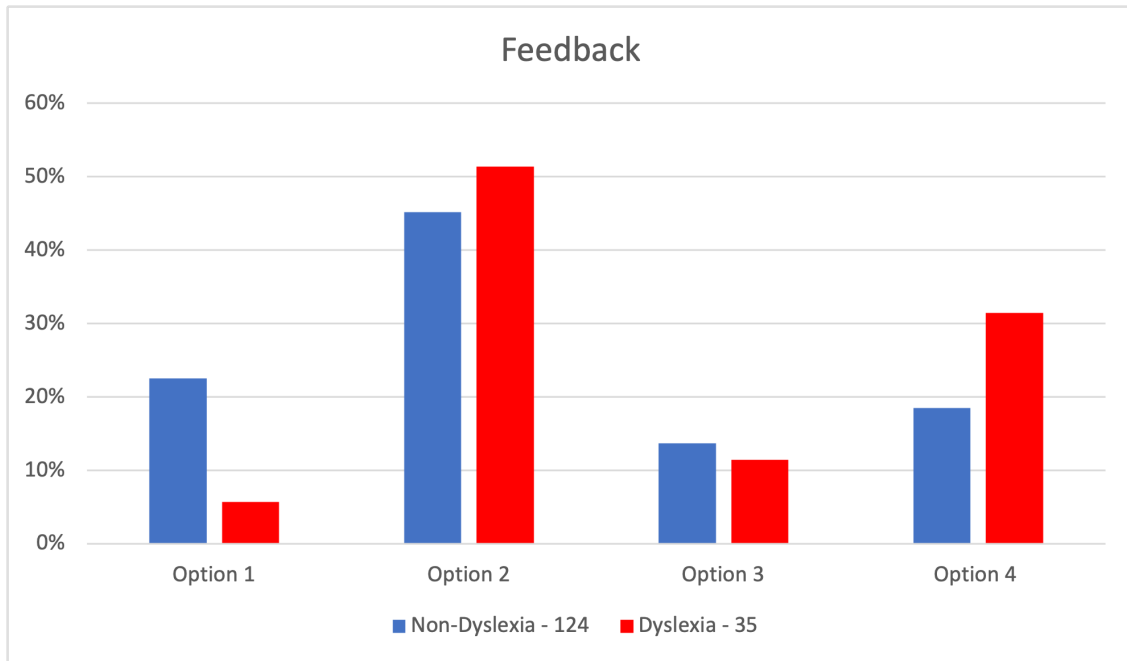


Figure 5: Graph for feedback results from 'Spelling Test' survey.

- ☐ Correct / Incorrect
- ☐ Correct / Incorrect + word spelt correctly
- ☐ Correct / Incorrect + word spelt correctly + definition of word
- ☐ Correct / Incorrect + word spelt correctly + definition of word + word in a sentence

Figure 6: Screenshot of the feedback question from 'Spelling Test' survey.

- ☐ Correct / Incorrect
- ☐ Correct / Incorrect + word spelt correctly
- ☐ Correct / Incorrect + word spelt correctly + definition of word
- ☐ Correct / Incorrect + word spelt correctly + definition of word + word in a sentence

Figure 7: Screenshot of the feedback question from 'Spelling Test' survey.

-
- ☐ Correct / Incorrect
 - ☐ Correct / Incorrect + word spelt correctly
 - ☐ Correct / Incorrect + word spelt correctly + definition of word
 - ☐ Correct / Incorrect + word spelt correctly + definition of word + word in a sentence

Figure 8: Screenshot of the feedback question from 'Spelling Test' survey.

What font do you find the easiest to read? *

Example	Example
<input type="radio"/> Option 1	<input type="radio"/> Option 2
Example	Example
<input type="radio"/> Option 3	<input type="radio"/> Option 4

Figure 9: Screenshot of the font question from 'Spelling Test' survey.

What colour background do you find makes it the easiest to read the word? *

Example

☐ Option 1

Example

☐ Option 2

Example

☐ Option 3

Example

☐ Option 4

Example

☐ Option 5

Example

☐ Option 6

Example

☐ Option 7

Figure 10: Screenshot of the background question from 'Spelling Test' survey.

A.2 Test Case - General User (10-)

Test Case ID: 101		Test Purpose:Register and login as a general user.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: The user has loaded the application and pressed they do not have an account and picked to register as a student.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	The register fields are filled with the correct values. Name: Test General Age: 04/06/1994 Email: TestGeneral@email.com Confirm Email: TestGeneral@email.com Password: general123 Confirm Password: general123	An alert will appear letting the user know, that they have registered successfully.	Pass [Figure 78]
1.1	The email fields don't match. Email: TestGeneral@email.com Confirm Email: TestLogin@email.com	An error message will appear saying that the email doesn't match, and in turn doesn't let the user register.	Pass [Figure 46]
1.2	The email fields are not of a valid form. Email: TestGeneral Confirm Email: TestGeneral	An error message will appear saying that the email is invalid, and in turn doesn't let the user register.	Pass [Figure 47]
1.3	The email is already in the system. Email: TestLogin@email.com Confirm Email: TestLogin@email.com	An error message will appear saying that the email is already in the system, and in turn doesn't let the user register.	Pass [Figure 48]
1.4	The password fields don't match. Password: general123 Confirm Password: 123general	An error message will appear saying that the password doesn't match, and in turn doesn't let the user register.	Pass [Figure 49]
1.5	All the register fields are empty.	An error message will appear saying that the user needs to make sure all the fields are filled, and in turn doesn't let the user register.	Pass [Figure 50]
2	The login fields are filled with the correct values. Email: TestGeneral@email.com Password: general123	Will transfer the user to their correct account.	Pass
2.1	The login is invalid. Email: TestLogin@email.com Password: general123	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 51]
2.2	All the login fields are empty.	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 50]
Comments: All the test passed successfully.			
Related Test: N/A			
Author: Samantha Hughes			

Table 5: The table is the test case for "Register and login as a general user." for a General User.

Test Case ID: 103		Test Purpose: Changing the background colour in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the background colour to pink.	On the segment controller pink is now selected.	Pass [Figure 35]
1.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 36]
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the background colour to blue.	On the segment controller blue is now selected.	Pass [Figure 37]
2.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 38]
3.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
3.2	Selecting the background colour to yellow.	On the segment controller yellow is now selected.	Pass [Figure 39]
3.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 40]
Comments: All the test passed successfully.			
Related Test: 101			
Author: Samantha Hughes			

Table 6: The table is the test case for "Changing the background colour in the setting page." for a General User.

Test Case ID: 104		Test Purpose: Changing the font in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font to Arial.	On the segment controller Arial is now selected.	Pass [Figure 52]
1.3	Pressing submit.	The font updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font to Comic Sans.	On the segment controller Comic Sans is now selected.	Pass [Figure 53]
2.3	Pressing submit.	The font updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 101			
Author: Samantha Hughes			

Table 7: The table is the test case for "Changing the font in the setting page." for a General User.

Test Case ID: 105		Test Purpose: Changing the font size in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font size to small.	The slider is on the lower end with the smaller font size.	Pass [Figure 54]
1.3	Pressing submit.	The font updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font size to large.	The slider is on the higher end with the larger font size.	Pass [Figure 55]
2.3	Pressing submit.	The font updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 101			
Author: Samantha Hughes			

Table 8: The table is the test case for "Changing the font size in the setting page." for a General User.

Test Case ID: 106		Test Purpose: Changing the feedback display in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	The user changes the switch based on if they want the feedback on or off. correct/incorrect - on word - on definition - off sentence - off	The correct/incorrect switch is turn on. The word switch is turn on. The definition switch is turn off. The sentence switch is turn off.	Pass [Figure 32]
1.3	Pressing submit.	The feedback selection updates on both the application and the database.	Pass [Figure 56]
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	The user changes the switches based on if they want the feedback on or off. correct/incorrect - off word - off definition - on sentence - on	The correct/incorrect switch is turn off. The word switch is turn off. The definition switch is turn on. The sentence switch is turn on.	Pass [Figure 57]
2.3	Pressing submit.	The feedback selection updates on both the application and the database.	Pass [Figure 58]
3.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
3.2	The user changes the switches based on if they want the feedback on or off. correct/incorrect - off word - on definition - on sentence - on	The correct/incorrect switch is turn off. The word switch is turn on. The definition switch is turn on. The sentence switch is turn on.	Pass [Figure 59]
3.3	Pressing submit.	The feedback selection updates on both the application and the database.	Pass [Figure 60]
Comments: All the test passed successfully.			
Related Test: 101			
Author: Samantha Hughes			

Table 9: The table is the test case for "Changing the feedback display in the setting page." for a General User.

Test Case ID: 107		Test Purpose: Log off the account.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	From the home page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.2	From the test page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.3	From the setting page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
Comments: All the test passed successfully.			
Related Test: 101			
Author: Samantha Hughes			

Table 10: The table is the test case for "Log off the account." for a General User.

A.3 Test Case - Student User (20-)

Test Case ID: 201		Test Purpose: Register and login as a student user.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: The user has loaded the application and press they do not have an account and picked to register as a student.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	The register fields are filled with the correct values. Name: Test Student Age: 03/07/2000 Email: TestStudent@email.com Confirm_Email: TestStudent@email.com Access Code: 12345 Password: student123 Confirm Password: student123	An alert will appear letting the user know, that they have register successful.	Pass [Figure 78]
1.1	The email fields don't match. Email: TestStudent@email.com Confirm_Email: TestLogin@email.com	An error message will appear saying that the email doesn't match, and in turn doesn't let the user register.	Pass [Figure 46]
1.2	The email fields are not of a valid form. Email: TestStudent Confirm_Email: TestStudent	An error message will appear saying that the email is invalid, and in turn doesn't let the user register.	Pass [Figure 47]
1.3	The email is already in the system Email: TestLogin@email.com Confirm_Email: TestLogin@email.com	An error message will appear saying that the email is already in the system, and in turn doesn't let the user register.	Pass [Figure 48]
1.4	The password fields don't match. Password: student123 Confirm Password: 123student	An error message will appear saying that the password doesn't match, and in turn doesn't let the user register.	Pass [Figure 49]
1.5	The access code is invalid, not connected to a teacher. Access Code: 00000	An error message will appear saying that the access code is invalid, and in turn doesn't let the user register.	Pass [Figure 61]
1.6	All the register fields are empty.	An error message will appear saying that the user needs to make sure all the fields are filled, and in turn doesn't let the user register.	Pass [Figure 50]
2	The login in fields is filled with the correct value.	Will transfer the user to their correct account.	Pass
2.1	The login is invalid. Email: TestLogin@email.com Password: student123	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 51]
2.2	All the login fields are empty.	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 50]
Comments: All the test passed successfully.			
Related Test: N/A			
Author: Samantha Hughes			

Table 11: The table is the test case for "Register and login as a student user." for a Student User.

Test Case ID: 202		Test Purpose: Starting a spelling test, of different levels and number of words.		
Layout: XCode, Version 13.2.1, MacBook Pro				
Preconditions: Logged into their account.				
Test Case Steps:				
Step No	Procedure	Response	Pass/Fail	
1.1	Selecting the level to number 1, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 31]	
1.2	Selecting the number of words to 1, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 31]	
1.3	Press Start Spelling button.	The spelling test view controller is loaded with the first spelling to complete.	Pass	
1.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass [Figure 28]	
1.5	Press submit spelling.	Feedback view appears.	Pass [Figure 30]	
1.6	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass [Figure 45]	
2.1	Selecting the level to number 2, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 33]	
2.2	Selecting the number of words to 2, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 33]	
2.3	Press Start Spelling button.	The spelling test view controller is loaded with the first spelling to complete.	Pass	
2.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
2.5	Press submit spelling.	Feedback view appears.	Pass	
2.6	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
2.7	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
2.8	Press submit spelling.	Feedback view appears.	Pass	
2.9	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass	
3.1	Selecting the level to number 3, using the stepper.	One will be displayed in the text field next to the level label.	Pass [Figure 34]	
3.2	Selecting the number of words to 3, using the stepper.	One will be displayed in the text field next to the number of words label.	Pass [Figure 34]	
3.3	Press Start Spelling button.	The spelling test view controller is loaded with the first spelling to complete.	Pass	
3.4	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.5	Press submit spelling.	Feedback view will appear.	Pass	
3.6	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
3.7	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.8	Press submit spelling.	Feedback view appears.	Pass	
3.9	Press next spelling.	Loads the spelling test with the next word for the user to complete.	Pass	
3.10	Placing the letter in the text field to complete the word.	The user can type the letter they believe to be correct in the text fields.	Pass	
3.11	Press submit spelling.	Feedback view appears.	Pass	
3.12	Press next spelling.	An alert pops up letting the user know they have completed the spelling test, then when the user pressed ‘Yay!’ It goes back to the home page.	Pass	
Comments: All the test passed successfully.				
Related Test: 201				
Author: Samantha Hughes				

Table 12: The table is the test case for "Starting a spelling test, of different levels and number of words." for a Student User.

Test Case ID: 204		Test Purpose: Changing the background colour in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the background colour to pink.	On the segment controller pink is now selected.	Pass [Figure 35]
1.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 36]
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the background colour to blue.	On the segment controller blue is now selected.	Pass [Figure 37]
2.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 38]
3.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
3.2	Selecting the background colour to yellow.	On the segment controller yellow is now selected.	Pass [Figure 39]
3.3	Pressing submit.	The background colour updates on both the application and the database.	Pass [Figure 40]
Comments: All the test passed successfully.			
Related Test: 201			
Author: Samantha Hughes			

Table 13: The table is the test case for "Changing the background colour in the setting page." for a Student User.

Test Case ID: 205		Test Purpose: Changing the font in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions:			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font to Arial.	On the segment controller Arial is now selected.	Pass [Figure 52]
1.3	Pressing submit.	The font updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font to Comic Sans.	On the segment controller Comic Sans is now selected.	Pass [Figure 53]
2.3	Pressing submit.	The font updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 201			
Author: Samantha Hughes			

Table 14: The table is the test case for "Changing the font in the setting page." for a Student User.

Test Case ID: 206		Test Purpose: Changing the font size in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font size to small.	The slider is on the lower end with the smaller font size.	Pass [Figure 54]
1.3	Pressing submit.	The font updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font size to large.	The slider is on the higher end with the larger font size.	Pass [Figure 55]
2.3	Pressing submit.	The font updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 201			
Author: Samantha Hughes			

Table 15: The table is the test case for "Changing the font size in the setting page." for a Student User.

Test Case ID: 207		Test Purpose: Log off the account.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	From the home page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.2	From the test page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.3	From the setting page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
Comments: All the test passed successfully.			
Related Test: 201			
Author: Samantha Hughes			

Table 16: The table is the test case for "Log off the account." for a Student User.

A.4 Test Case - Teacher User (30-)

Test Case ID: 301		Test Purpose: Register and login as a teacher user.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: The user has loaded the application and press they do not have an account and picked to register as a teacher.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	The register fields are filled with the correct values. Name: Test Teacher Email: TestTeacher@email.com Confirm_Email: TestTeacher@email.com Password: teacher123 Confirm Password: teacher123	An alert will appear letting the user know, that they have register successful.	Pass [Figure 78]
1.1	The email fields don't match. Email: TestTeacher@email.com Confirm_Email: TestLogin@email.com	An error message will appear saying that the email doesn't match, and in turn doesn't let the user register.	Pass [Figure 46]
1.2	The email fields are not of a valid form. Email: TestTeacher Confirm_Email: TestTeacher	An error message will appear saying that the email is invalid, and in turn doesn't let the user register.	Pass [Figure 47]
1.3	The email is already in the system. Email: TestLogin@email.com Confirm_Email: TestLogin@email.com	An error message will appear saying that the email is already in the system, and in turn doesn't let the user register.	Pass [Figure 48]
1.4	The password fields don't match. Password: teacher123 Confirm Password: 123teacher	An error message will appear saying that the password doesn't match, and in turn doesn't let the user register.	Pass [Figure 49]
1.5	All the register fields are empty.	An error message will appear saying that the user needs to make sure all the fields are filled, and in turn doesn't let the user register.	Pass [Figure 50]
2	The login in fields is filled with the correct value.	Will transfer the user to their correct account.	Pass
2.1	The login is invalid. Email: TestLogin@email.com Password: teacher123	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 51]
2.2	All the login fields are empty.	An error message will appear saying the email and or password is invalid, and in turn doesn't let the user login.	Pass [Figure 50]
Comments: All the test passed successfully.			
Related Test: N/A			
Author: Samantha Hughes			

Table 17: The table is the test case for "Register and login as a teacher user." for a Teacher User.

Test Case ID: 302		Test Purpose: Adding a class to the database, linked to that teacher.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	Valid class name, year group and access code, press submit. However, there is 6 classes already.	An alert pops up letting the user know that they need to delete a class before they can add another one.	Pass [Figure 62]
2	Press class button that is wanted to be deleted.	An alert pops up allowing the user to either delete the class or access the class for more information.	Pass [Figure 63]
3	Press delete, on the alert.	The class is deleted from the database and no longer displayed on the teacher class list.	Pass [Figure 65]
4	Press 'Add Class' when none of the fields are filled out.	An error appears letting the user know to make sure all the fields are filled out, and in turn not allowing the class to be added to the database.	Pass [Figure 50]
5	All 'add class' fields are filled out however input an invalid access code, teacher already used that access code. Class Name: K12 Year Group: 12 Access Code: 12345	An error appears letting the user know it is an invalid access code they have input, and in turn not allowing the class to be added to the database.	Pass [Figure 64]
6	All field filled out with value inputs. Class Name: K12 Year Group: 12 Access Code: 10101	The class is added to the database and displayed on the teacher class list. Students are also now able to register with this access code.	Pass [Figure 66]
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 18: The table is the test case for "Adding a class to the database, linked to that teacher." for a Teacher User.

Test Case ID: 303		Test Purpose: Access a class to go to the class page where more information is displayed.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	Pressing the class that the teacher would like to receive more information about.	An alert appears letting the user pick between accessing the class or deleting the class.	Pass [Figure 63]
2	Clicking access on the alert.	The class view controller loads with the correct information relating to that class.	Pass [Figure 67]
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 19: The table is the test case for "Access a class to go to the class page where more information is displayed." for a Teacher User.

Test Case ID: 304		Test Purpose: Add a new spelling test to a class for the student to then complete.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Access the class.			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1	Press ‘Test Class’ when there are 3 classes already attached to the class.	An alert appears telling the teacher to delete one of the tests first before being able to add another one.	Pass [Figure 68]
2	Press test button that is wanted to be deleted.	An alert appears letting the user either delete the test or cancel and the test will not be deleted.	Pass [Figure 69]
3	Press delete, on the alert.	The test will be removed from the database, and in turn not display the test anymore on the teacher and student page.	Pass [Figure 70]
4	Press ‘Test Class’ when there are less than 3 classes already attached to the class.	Moves to the add test view controller where the teacher can input 10 words.	Pass [Figure 71]
5	Press the add test button when there is nothing in the text field.	An error message appears telling the user to make sure the fields are filled out.	Pass [Figure 50]
6	All fields filled out with value inputs and press ‘Add Test’. — Word One: cow Word Two: sheep Word Three: dog Word Four: cat Word Five: fish Word Six: horse Word Seven: mouse Word Eight: bird Word Nine: goat Word Ten: lamb	The page will redirect to the class page and the test is displayed, added to the database, and displayed on the students account too.	Pass [Figure 27]
Comments: All the test passed successfully.			
Related Test: 303			
Author: Samantha Hughes			

Table 20: Table above is the test case for "Add a new spelling test to a class for the student to then complete." for a Teacher User.

Test Case ID: 306		Test Purpose: Changing the font in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font to Arial.	On the segment controller Arial is now selected.	Pass [Figure 52]
1.3	Pressing submit.	The font updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font to Comic Sans.	On the segment controller Comic Sans is now selected.	Pass [Figure 53]
2.3	Pressing submit.	The font updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 21: The table is the test case for "Changing the font in the setting page." for a Teacher User.

Test Case ID: 307		Test Purpose: Changing the font size in the setting page.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
1.2	Selecting the font size to small.	The slider is on the lower end with the smaller font size.	Pass [Figure 54]
1.3	Pressing submit.	The font size updates on both the application and the database.	Pass
2.1	Setting page is entered from the home page.	The setting page view controller loads on the application.	Pass
2.2	Selecting the font size to large.	The slider is on the higher end with the larger font size.	Pass [Figure 55]
2.3	Pressing submit.	The font size updates on both the application and the database.	Pass
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 22: The table is the test case for "Changing the font size in the setting page." for a Teacher User.

Test Case ID: 309		Test Purpose: Log off the account.	
Layout: XCode, Version 13.2.1, MacBook Pro			
Preconditions: Logged into their account			
Test Case Steps:			
Step No	Procedure	Response	Pass/Fail
1.1	From the home page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.2	From the class page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.3	From the test page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
1.4	From the setting page the logoff button is pressed.	Takes the user back to the login view controller.	Pass
Comments: All the test passed successfully.			
Related Test: 301			
Author: Samantha Hughes			

Table 23: The table is the test case for "Log off the account." for a Teacher User.

A.5 App Screenshot

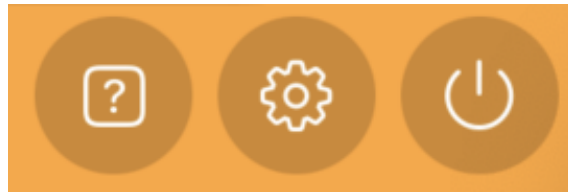


Figure 11: Help button.

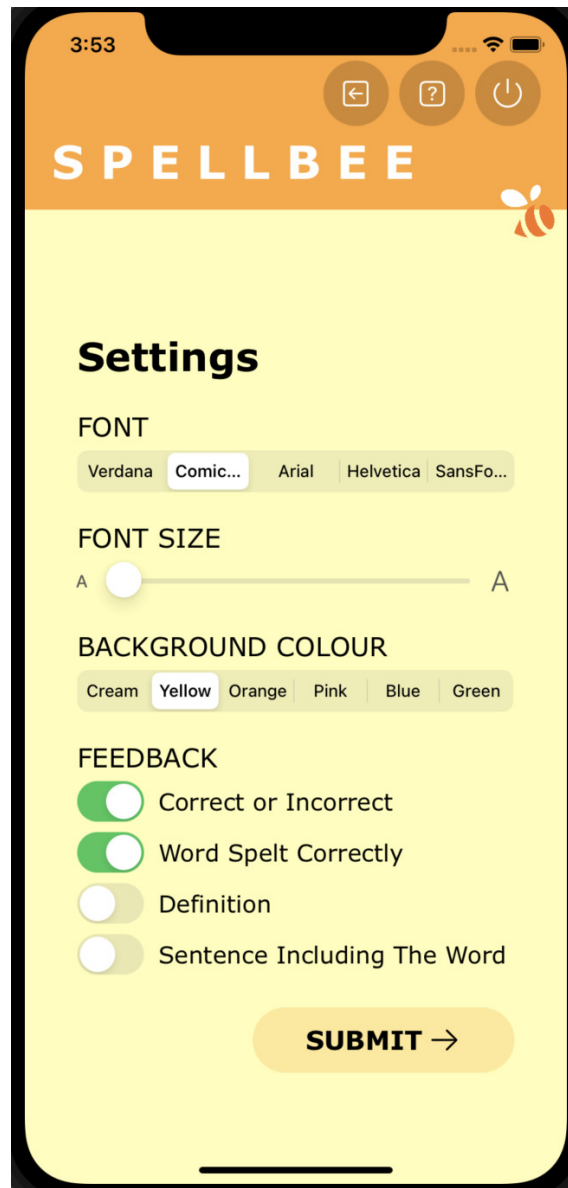


Figure 12: General Account Setting Page.

BACKGROUND COLOUR

Cream Yellow Orange Pink Blue Green

Figure 13: Background Selector.

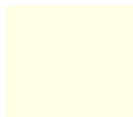


Figure 14: Cream.

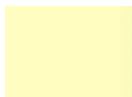


Figure 15: Yellow.



Figure 16: Orange.



Figure 17: Pink.

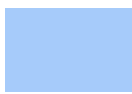


Figure 18: Blue.

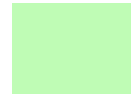


Figure 19: Green.

FONT

Verdana Comic... Arial Helvetica SansFo...

Figure 20: Font Selector.

Correct

Figure 21: Verdana.

Correct

Figure 22: Comic Sans.

Correct

Figure 23: Arial.

Correct

Figure 24: Helvetica.

Correct

Figure 25: San Forgetica.

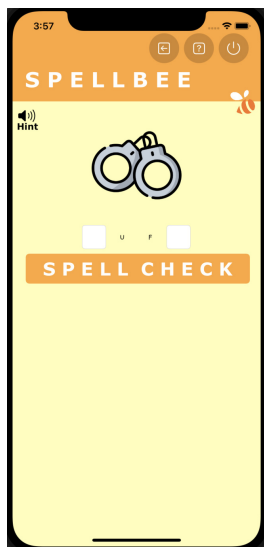


Figure 26: General Account Spelling Test Page.

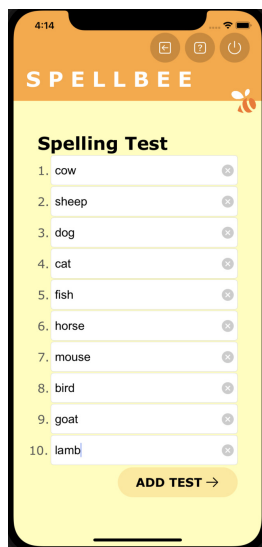


Figure 27: Teacher Account Add Test Page Filled.



Figure 28: Student Account Spelling Test Page.

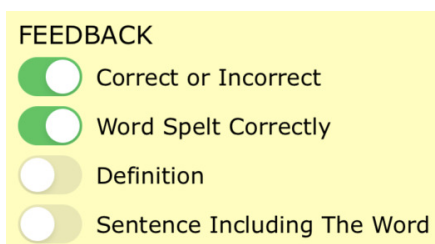


Figure 29: Teacher Account, Class feedback, 1.

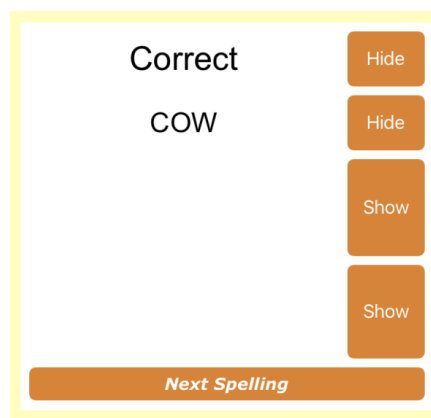


Figure 30: Student Account, Feedback View, 1.

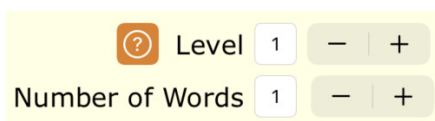


Figure 31: Level and Word Stepper, 1.

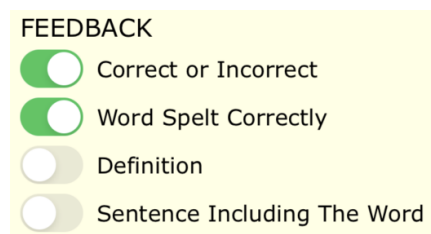


Figure 32: General Account, Feedback, 1.

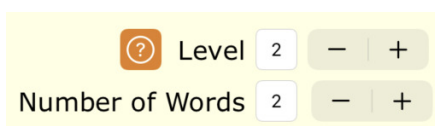


Figure 33: Level and Word Stepper, 2.

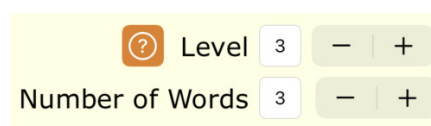


Figure 34: Level and Word Stepper, 3.

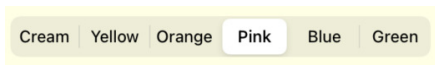


Figure 35: Pink Selector.

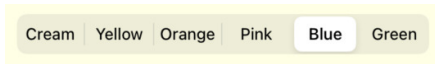


Figure 37: Blue Selector.



Figure 39: Yellow Selector.

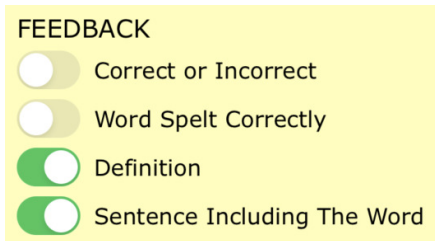


Figure 41: Teacher Account, Feedback, 2.

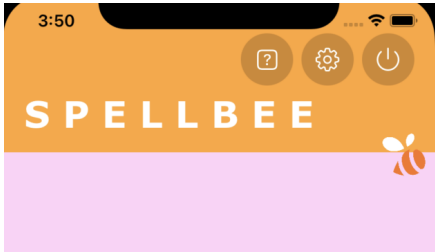


Figure 36: Pink Background.

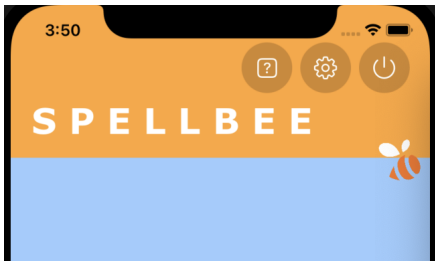


Figure 38: Blue Background.



Figure 40: Yellow Background.

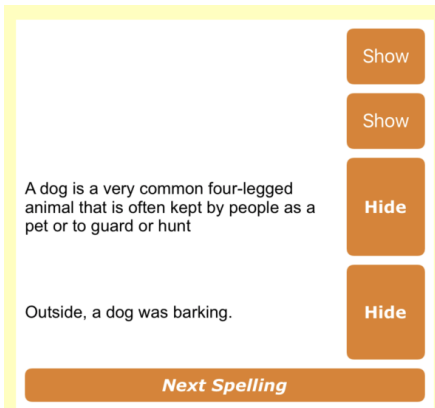


Figure 42: Student Account, Feedback View, 2.

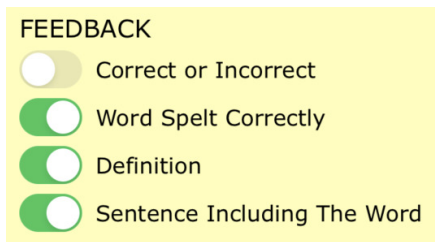


Figure 43: Teacher Account, Feedback, 3.



Figure 44: Student Account, Feedback View, 3.

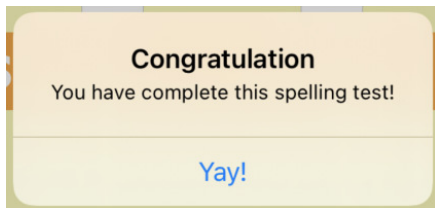


Figure 45: Congratulation Alert.

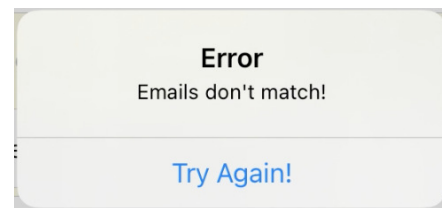


Figure 46: Emails Don't Match Alert

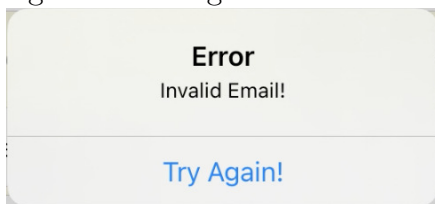


Figure 47: Invalid Email Alert.

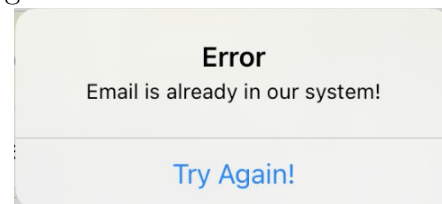


Figure 48: Email is Already in the System Alert.

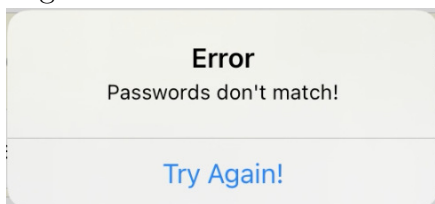


Figure 49: Passwords Don't Match Alert.

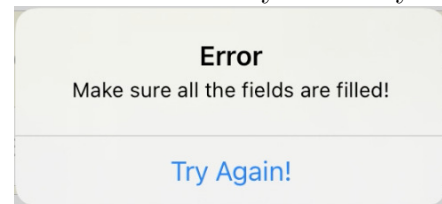


Figure 50: All Fields need to be Filled Out Alert.

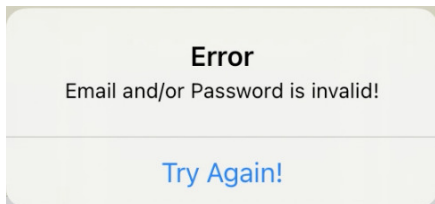


Figure 51: Email and/or Password Alert.



Figure 52: Arial Selected.



Figure 53: Comic Sans Selected.



Figure 54: Small Font Selected.



Figure 55: Large Font Selected.



Figure 56: General Account, Feedback View, 1.

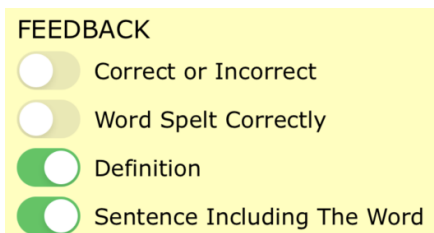


Figure 57: General Account, Feedback, 2.

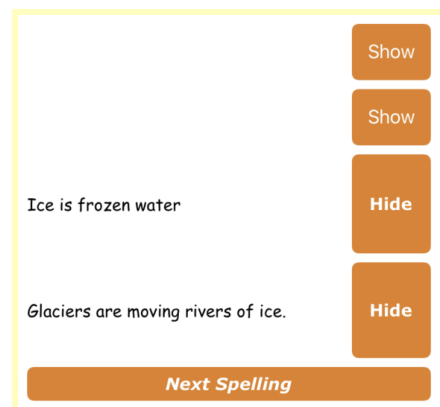


Figure 58: General Account, Feedback View, 2.

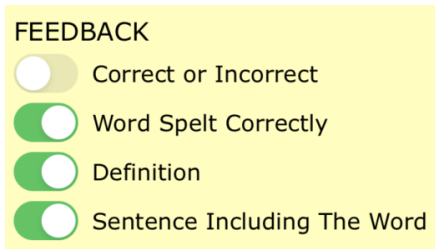


Figure 59: General Account, Feedback, 3.

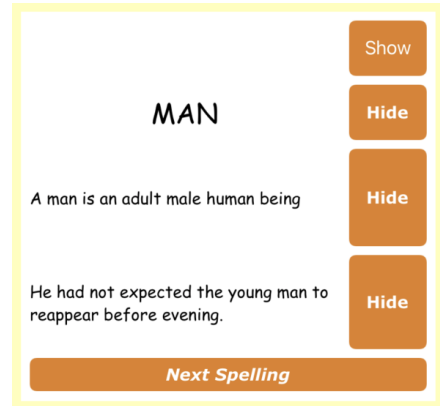


Figure 60: General Account, Feedback View, 3.

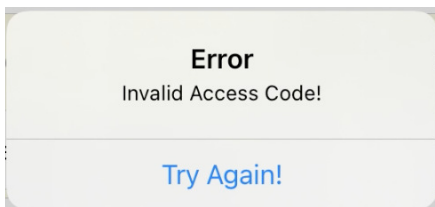


Figure 61: Invalid Access Code Alert.

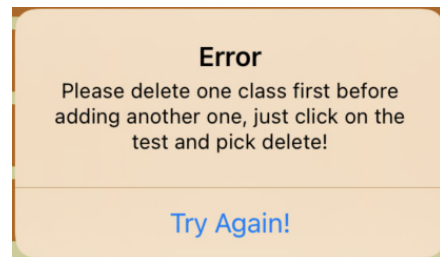


Figure 62: Delete a Class Alert.

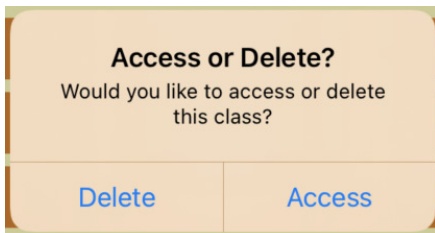


Figure 63: Access or Delete Alert.

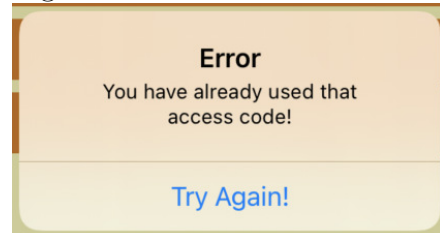


Figure 64: Access Code Already Used Alert.

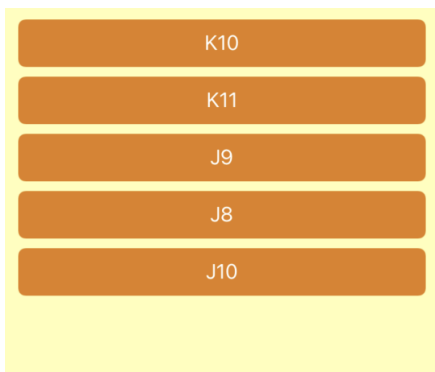


Figure 65: Class been Deleted.



Figure 66: Teacher Account, Class Added, List.

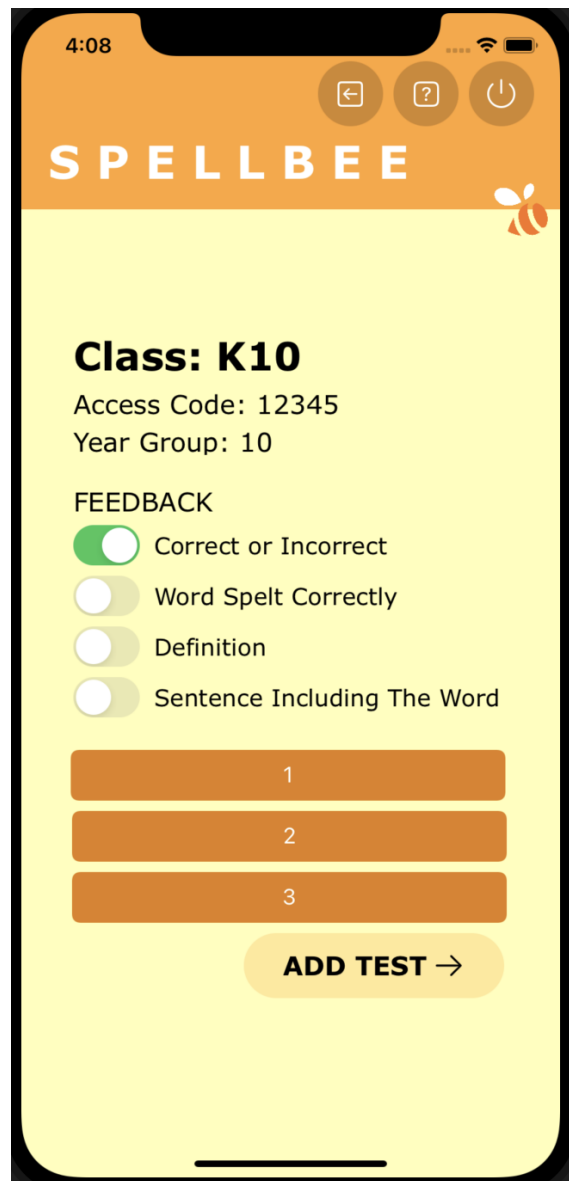


Figure 67: Teacher Account, Class Page.

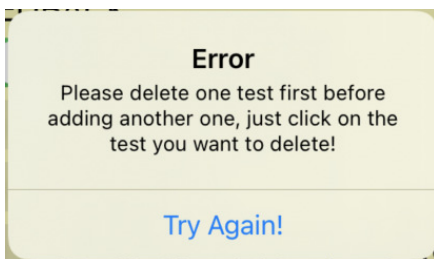


Figure 68: Deleted a Test Alert.

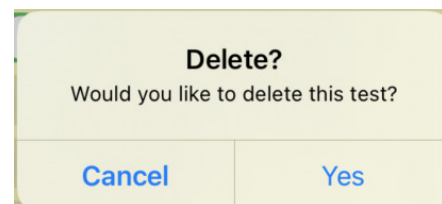


Figure 69: Yes or No, Delete Test, Alert.

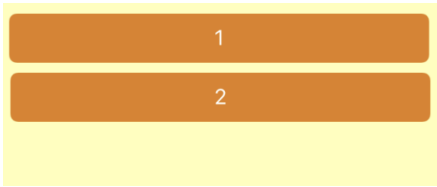


Figure 70: Teacher Account, Test List, 2

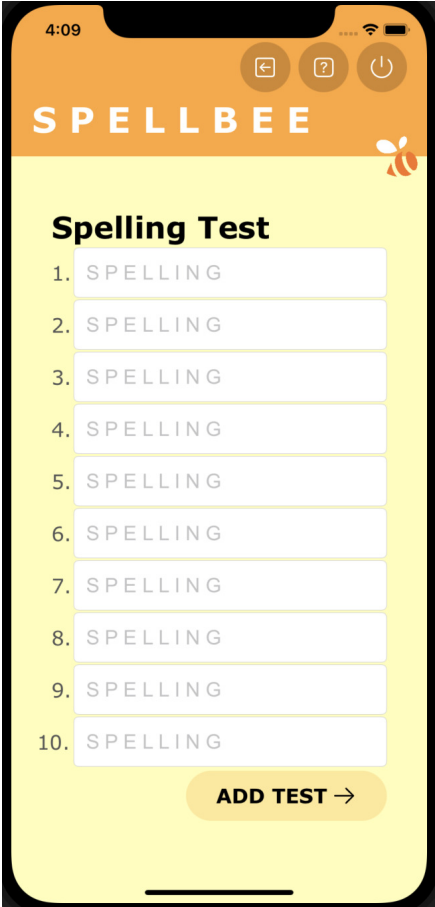


Figure 71: Teacher Account, Add Test Page.

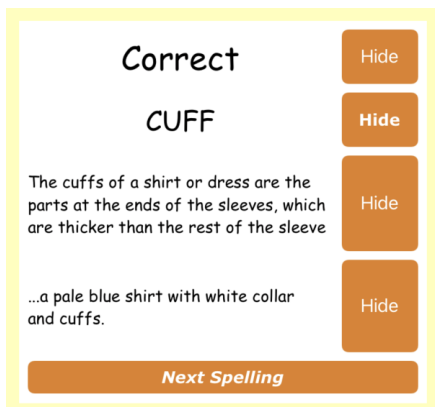


Figure 72: Full Feedback View.

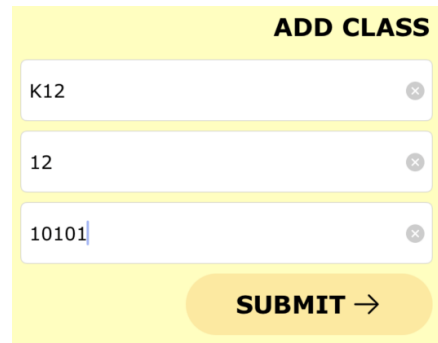


Figure 73: Teacher Account, Add Class Filled.

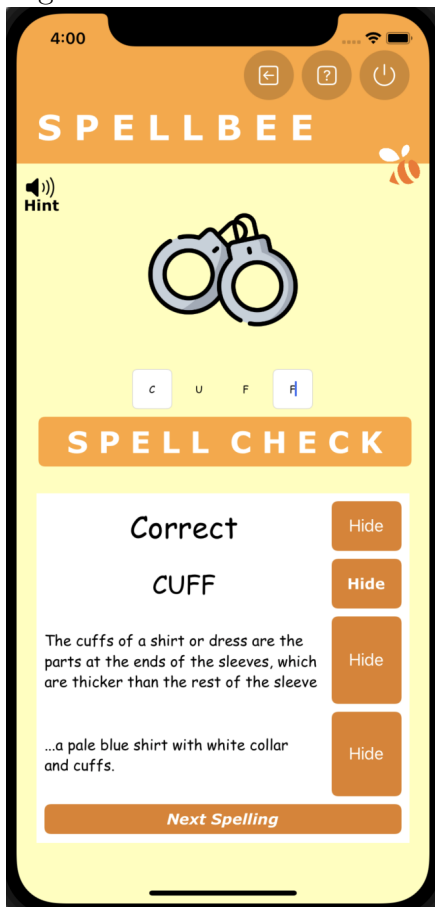


Figure 74: General Account, Spelling Test Filled.

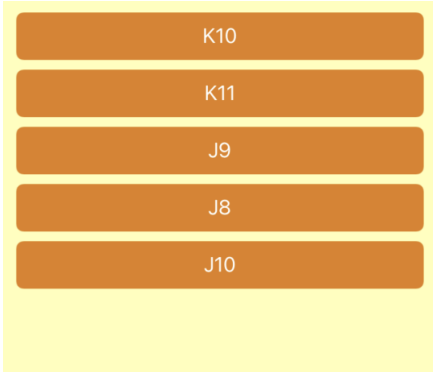


Figure 75: Teacher Account, Class Deleted.

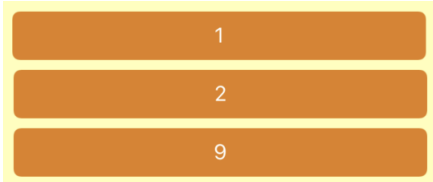


Figure 77: Teacher Account, Test List, 3.

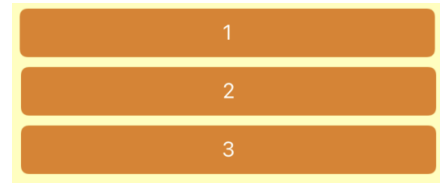


Figure 76: Teacher Account, Test List, 1.

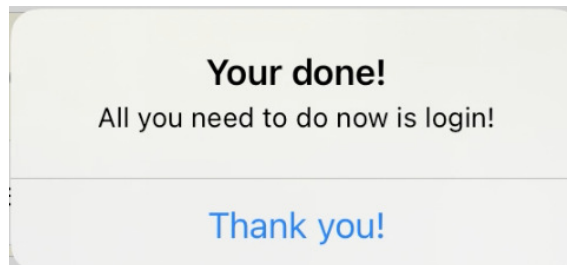


Figure 78: Your Done Alert.

B Bibliography

1. <https://www.codecademy.com/learn/learn-swift>
2. <https://www.raywenderlich.com/6620276-sqlite-with-swift-tutorial-getting-started#toc-anchor-022>
3. <https://www.w3schools.com/sql/default.asp>
4. <https://developer.apple.com/swift/>
5. <https://developer.apple.com/forums/thread/653723>
6. https://developer.apple.com/documentation/uikit/windows_and_screens/getting_the_user_s_attention_with_alerts_and_action_sheets#2940105
7. <https://developer.apple.com/documentation/swift/collection>
8. <https://developer.apple.com/documentation/uikit/uicolor>
9. https://developer.apple.com/documentation/uikit/uicolor/ui_element_colors
10. <https://stackoverflow.com/questions/42475384/uialertcontroller-to-go-to-new-viewcontroller>
11. <https://medium.com/@bfozdar/outlet-collections-a70b09ca61a3>
12. <https://blog.kulman.sk/web-scraping-with-swift/>
13. <https://stackoverflow.com/questions/29887869/how-to-present-ios-uiactionsheet-in-swift>
14. <https://stackoverflow.com/questions/39998412/how-to-make-a-select-query-using-a-local-variable-in-sqlite-on-swift>
15. https://www.youtube.com/watch?v=_9BT01z1DKw
16. <https://www.youtube.com/watch?v=zzhG2MlZ8WU>
17. <https://www.youtube.com/watch?v=73XRidPrLwY>
18. https://www.wfonts.com/font/comic-sans-ms#google_vignette
19. https://www.cufonfonts.com/font/sans-forgetica#google_vignette
20. <https://www.pinterest.co.uk/pin/48061921062765966/>
21. <https://creativitygames.net/random-word-generator/randomwords/8>

C References

References

- [1] Abdelmoty, A. (2021). Usability Principles (Guidelines and Heuristics). Cardiff University.
- [2] AltexSoft. (2019). The Good and the Bad of Swift Programming Language. [online] Available at: <https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-swift-programming-language/> [Accessed Feb. 9AD].
- [3] Apple.com. (2019). Swift - Apple Developer. [online] Available at: <https://developer.apple.com/swift/> [Accessed Feb. 9AD].
- [4] British Dyslexia Association (2010). Legislation - British Dyslexia Association. [online] British Dyslexia Association. Available at: <https://www.bdadyslexia.org.uk/advice/employers/how-can-i-support-my-dyslexic-employees/legislation> [Accessed 2 Apr. 2022].
- [5] British Dyslexia Association (2018). Dyslexia friendly style guide. [online] British Dyslexia Association. Available at: <https://www.bdadyslexia.org.uk/advice/employers/creating-a-dyslexia-friendly-workplace/dyslexia-friendly-style-guide> [Accessed 31 Jan. 2022].
- [6] Common Sense Education. (2014). Spelling Apps for Students. [online] Available at: <https://www.commonsense.org/education/top-picks/spelling-apps-for-students> [Accessed 31 Jan. 2022].
- [7] Denton, T.F. and Meindl, J.N. (2015). The Effect of Colored Overlays on Reading Fluency in Individuals with Dyslexia. *Behavior Analysis in Practice*, 9(3), pp.191–198.
- [8] docs.python.org. (n.d.). sqlite3 — DB-API 2.0 interface for SQLite databases — Python 3.8.2 documentation. [online] Available at: <https://docs.python.org/3/library/sqlite3.html> [Accessed 27 Apr. 2022].
- [9] dyslexiahelp.umich.edu. (n.d.). Apps for Dyslexia and Learning Disabilities. [online] Available at: <http://dyslexiahelp.umich.edu/tools/apps> [Accessed 1 May 2022].
- [10] Ehab Yosry Amer (2019). iOS Storyboards: Getting Started. [online] raywenderlich.com. Available at: <https://www.raywenderlich.com/5055364-ios-storyboards-getting-started> [Accessed 1 May 2022].
- [11] English -Appendix 1: Spelling 1. (n.d.). [online] Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/239784/English_Appendix_1_-_Spelling.pdf [Accessed 17 Mar. 2022].
- [12] GOV.UK. (n.d.). Simone: dyslexic user. [online] Available at: <https://www.gov.uk/government/publications/understanding-disabilities-and-impairments-user-profiles/simone-dyslexic-user#devices-and-technology> [Accessed 2 Apr. 2022].
- [13] Government Digital Service (2017). Simone: dyslexic user. [online] GOV.UK. Available at: <https://www.gov.uk/government/publications/understanding-disabilities-and-impairments-user-profiles/simone-dyslexic-user> [Accessed 28 Feb. 2022].

-
- [14] Hughes, S. (2022). Google Forms - create and analyze surveys, for free. [online] accounts.google.com. Available at: <https://docs.google.com/forms/d/e/1FAIpQLSf52ssHJ9V6GM-h.5v822FramYrbisq5YFpkPYpMMvfP6l0FQ/viewform> [Accessed 1 Feb. 2022].
- [15] Learning Center. (n.d.). What Is Scraping — About Price Web Scraping Tools — Imperva. [online] Available at: <https://www.imperva.com/learn/application-security/web-scraping-attack/> [Accessed 1 May 2022].
- [16] Leseyane, M., Mandende, P., Makgato, M. and Cekiso, M. (2018). Dyslexic learners' experiences with their peers and teachers in special and mainstream primary schools in North-West Province. African Journal of Disability, [online] 7(0). Available at: <https://ajod.org/index.php/ajod/article/view/363https://ajod.org/index.php/ajod/article/view/363> [Accessed 2 Apr. 2022].
- [17] Marianne (2015). The Best Spelling Apps for Dyslexia. [online] Homeschooling with Dyslexia. Available at: <https://homeschoolingwithdyslexia.com/the-best-spelling-apps-for-dyslexia/> [Accessed 31 Jan. 2022].
- [18] Murnane, K. (n.d.). This Is Your Brain On Dyslexia. [online] Forbes. Available at: <https://www.forbes.com/sites/kevinmurnane/2017/01/05/this-is-your-brain-on-dyslexia/?sh=750c77dd64ed> [Accessed 2 Apr. 2022].
- [19] Nabin Paudyal (2016). Top 20 Most Popular Fonts Of All Time. [online] Lifehack. Available at: <https://www.lifehack.org/428846/top-20-most-popular-fonts-of-all-time> [Accessed 1 Feb. 2022].
- [20] NHS (2018). Dyslexia. [online] NHS. Available at: <https://www.nhs.uk/conditions/Dyslexia/> [Accessed 31 Jan. 2022].
- [21] Nielson, J. (1993). Response Time Limits: Article by Jakob Nielsen. [online] Nielsen Norman Group. Available at: <https://www.nngroup.com/articles/response-times-3-important-limits/> [Accessed 10 May 2022].
- [22] Petrov, C. (2019). 55+ Mobile Vs Desktop Usage Stats You Should Know in 2020. [online] TechJury. Available at: <https://techjury.net/blog/mobile-vs-desktop-usage/#gref> [Accessed 1 Feb. 2022].
- [23] printsupport-copi. (n.d.). 2020 Top Ten Fonts. [online] Available at: <https://www.copiprintsupport.com/2020-top-ten-fonts> [Accessed 1 Feb. 2022].
- [24] sansforgetica.rmit.edu.au. (n.d.). Sans Forgetica. [online] Available at: <https://sansforgetica.rmit.edu.au> [Accessed 25 Feb. 2022].
- [25] Singh, M. (2020). Top 10 Spelling Games Apps for Adults. [online] Number Dyslexia. Available at: <https://numberdyslexia.com/top-10-spelling-games-apps-for-adults/> [Accessed 1 May 2022].
- [26] support.microsoft.com. (n.d.). Make your Word documents accessible to people with disabilities. [online] Available at: <https://support.microsoft.com/en-us/office/make-your-word-documents-accessible-to-people-with-disabilities-d9bf3683-87ac-47ea-b91a-78dcacb3c66d> [Accessed 2 Apr. 2022].

-
- [27] UNESCO MGIEP. (n.d.). Living with Dyslexia. [online] Available at: <https://mgiep.unesco.org/article/living-with-dyslexia> [Accessed 2 Apr. 2022].
- [28] yougov.co.uk. (n.d.). The most popular phone models in the UK — Technology — YouGov Ratings. [online] Available at: <https://yougov.co.uk/ratings/technology/popularity/phone-models/all> [Accessed 1 Feb. 2022].