National College of Ireland

BSHCDBT

Digital Business Transformation

2022/2023

Kayleigh Harris

x19471636

x19471636@student.ncirl.ie

Sign-ify: ISL Dictionary

Technical Report

Table of Contents

Executive Summary

This project is aimed to break the boundaries associated with communication in sign language communities, by having an app that people can use to look up the physical sign actions through a mobile phone.

The purpose of this report is to address the **technical details** and **requirements** for my project – an Irish sign language (ISL) dictionary mobile app, showcasing English words and phrases and what the equivalent is in Irish sign language (ISL). With the use of different diagrams, I have described the application's design and architecture, as well as highlight the use cases for each required functionality of the application.

This application can be usable by anyone but is targeted towards people who have family members or friends in the sign language communities, as well as businesses and industries to enable staff to communicate easier with people with these disabilities.

As well as the requirements needed, I have also addressed the **testing** I used throughout the project, including testing plans, tools and technologies. After testing, I will also **evaluate** the current state of the project, including the results received, system performance and its correctness.

1 Introduction

1.1 Background

Sign-ify: ISL Dictionary is an Irish sign language iOS mobile application. It is a mobile-friendly tool where users can look-up words or phrases in English and find the Irish Sign Language (ISL) equivalent.

In the Sign-ify mobile app, users can search for words or phrases and the application will process and display the user's input into the sign language physical action which is displayed on screen.
Users of the application can also save their favourite or most used words/phrases they have learnt via the application in the *favourites section* of the application.

I chose to undertake this project as sign language is such an important form of communication. In Ireland, its estimated that for every one Deaf person, there are nine hearing people who use ISL to communicate in their everyday lives. It is estimated that there are around 5000 Deaf sign language users for which ISL is their language of choice.

The use of sign language allows you to also communicate in areas including students in mainstream schools, deaf schools or college and universities. As well as deaf or hard-of-hearing residents and in areas of business. Sign language can also improve the communication amongst families who may have deaf, hard-of-hearing or mute family members.

Now with the help of technology and smartphones, a sign language mobile application can become a useful tool for users who wish to learn sign language, develop their skills or who wish to communicate with the above communities.

1.2 Aims

This project sets out to achieve the following goals:

⇒ To develop a mobile application which can be used to learn the vocabulary of Irish-Sign Language (ISL) through audio and text input.

⇒ To remove the communication barrier between hard-of-hearing, deaf, speech difficulty and mute communities.

1.3 Technology

The implementation language I have used in the development of my mobile application is **Swift** – which is an open-source programming language developed by Apple. The Swift programming language is based on other programming languages, specifically Objective C, C++ and C. As I have used XCode, Apple's iOS application development environment for my mobile application, the default language that is provided is Swift.

Within XCode, it offers many open-source libraries to use in its environment that I have used for my application development. I have used **SwiftUI** which helps the development of layout within the application.
Other technologies and tools I have used are:

⇒ **XCode** – the development environment I built the application in.

- ⇒ **Google Firebase** – the Database to hold datasets.
- ⇒ **XCTest** – a testing tool which is built into XCode.
- ⇒ **XCode Simulator** – the virtual device I used to visual my application and build on.

The important algorithms and data structures that I have used in the development of this mobile application are:

- ⇒ **Binding Algorithm** – I used this for the functionality of my side menu throughout the entire app. The Binding algorithm creates a connection between the views and the data.
- ⇒ **GetData Algorithm** – searches through datasets to find matches to an input and return a value.
- ⇒ **Import SwiftUI** – used to import the SwiftUI framework, which helps with development in layout and design.
- ⇒ **Strings** – used in search engines and spell checks.

I have used the **Native Approach** for the development of this mobile application. Apps developed using the Native approach operate at a high-level performance, they are fast and responsive due to being built for the specific platform and operating system. As well as performance, there is full control on creativity on the application content and visual aspects.

The Native Approach also means that apps made using this approach are developed in a specific platform programming language. As I have developed my project in XCode – an iOS development environment, the application is only developed using the Swift programming language.

1.4 Structure

This document is broken down into six sections.

1. **Introduction** – the current section we are in, this section consists of the project background, its aims and objectives, the technologies and tools used, and the overall structure of the document.
2. **System** – the next section consists of any requirements the project has. These requirements are broken down into the following: *functional, use cases, data, user, environmental, design and architecture, implementation, GUIs, testing* and *evaluations*.
3. **Conclusions** – in this section, I will discuss the advantages and disadvantages of the project, as well as any strengths and weaknesses found during this process.
4. **Further Development or Research** – any further developments I wish to make to this project, as well as discussing the right amount of resources needed will be discussed in this section.
5. **References** – any resources I have used will be noted in this section.
6. **Appendices** – this section will include my project proposal and my seven monthly journals.

2 System

2.1 Requirements

2.1.1 Functional Requirements

The list below contains the *functional requirements* that are needed for this application:

1. **User Registration –** This function is required because a user must have an account so they can save words and phrases that they've learnt on the application that they want to remember.

2. **User Sign-in –** This function is required because a user must have an account so they can save words and phrases that they've learnt on the application and to access their saved words and phrases.

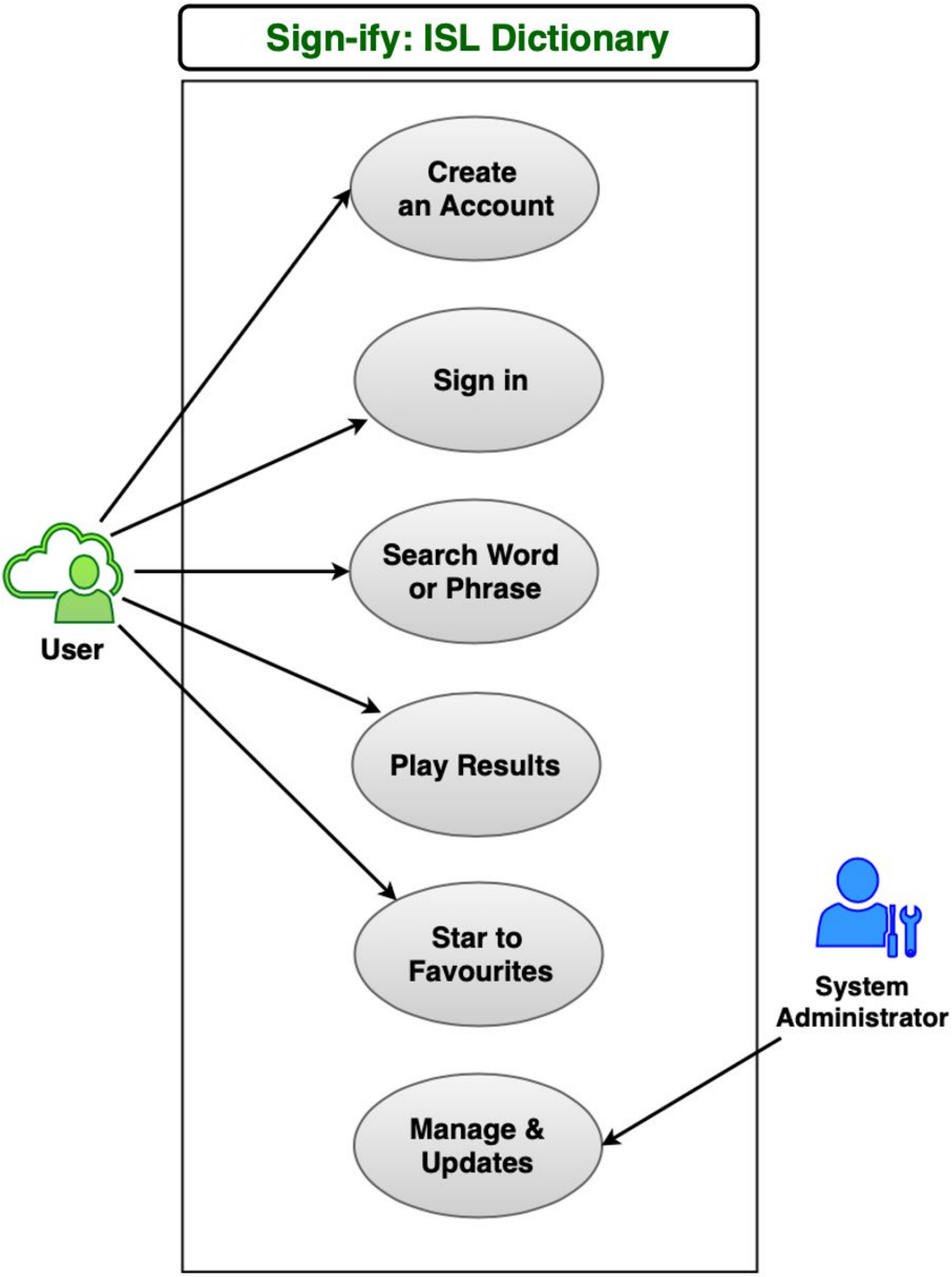
3. **Search bar –** This function is required because a user must search for a word or phrase before they can get access to the sign-action results.

4. **Sign Language Result –** This function is required because a user must access the results to their search.

5. **Starred to Favourites –** This function is required because a user can save words and phrases that they've learned to their account, to look back on later.

6. **Managing & Updating via System Administrator –** This function is required because the application will need updates made to the system, along with new functions that may be added in the future. As well as updates, the system administrator needs to manage the application on a regular basis so that the application runs smoothly.

2.1.2 Use Case Diagram

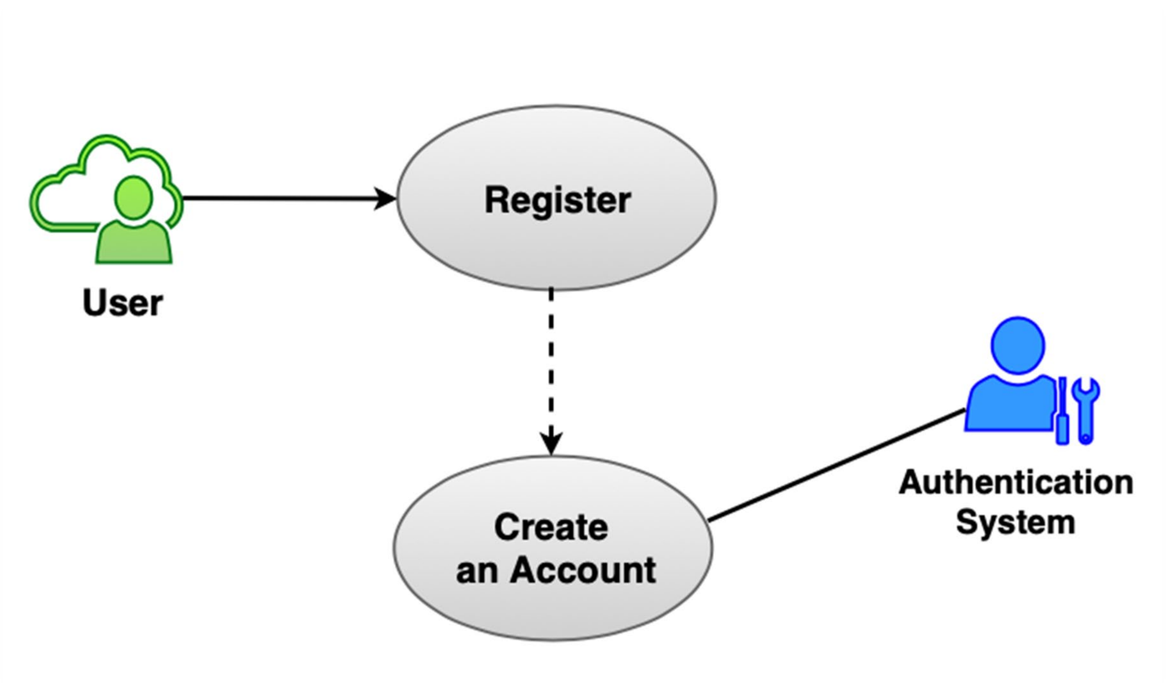


2.1.2.1 Requirement 1: User Registration

2.1.2.1.1 Description & Priority

Requirement 1 allows the application *user* to create an account so they can manage their saved favourite words and phrases in the application.

2.1.2.1.2 Use Case



Scope
The scope of this use case is to let the *user* create an account.

Description
This use case describes the how to register an account.

Flow Description

Precondition
The application has been opened by the *user*.

Activation
This use case starts when a *user* selects the login button from the sidebar menu.

Main flow

1. The application shows the option of logging in or to create an account.
2. The *user* selects the "Create an Account" button.
3. The *user* inputs their first-name, surname, a valid email address and a password.
4. The application continues to create the new account from the *authentication system*.
5. The *user's* account has been created and they login into their account on the mobile application

Alternate flow

A1: Incorrect Details

1. The *authentication system* notifies the *user* that they may have entered an incorrect email, or the password confirmation did not match.
2. The *user* can retry with their correct details.
3. The use case continues at position 4 of the main flow.

Exceptional flow

E1: Account Creation Error

1. An error occurs in the *authentication system* whilst trying to create the new account.
2. The *user* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The application presents the *user* with the login screen of the application.

Post condition

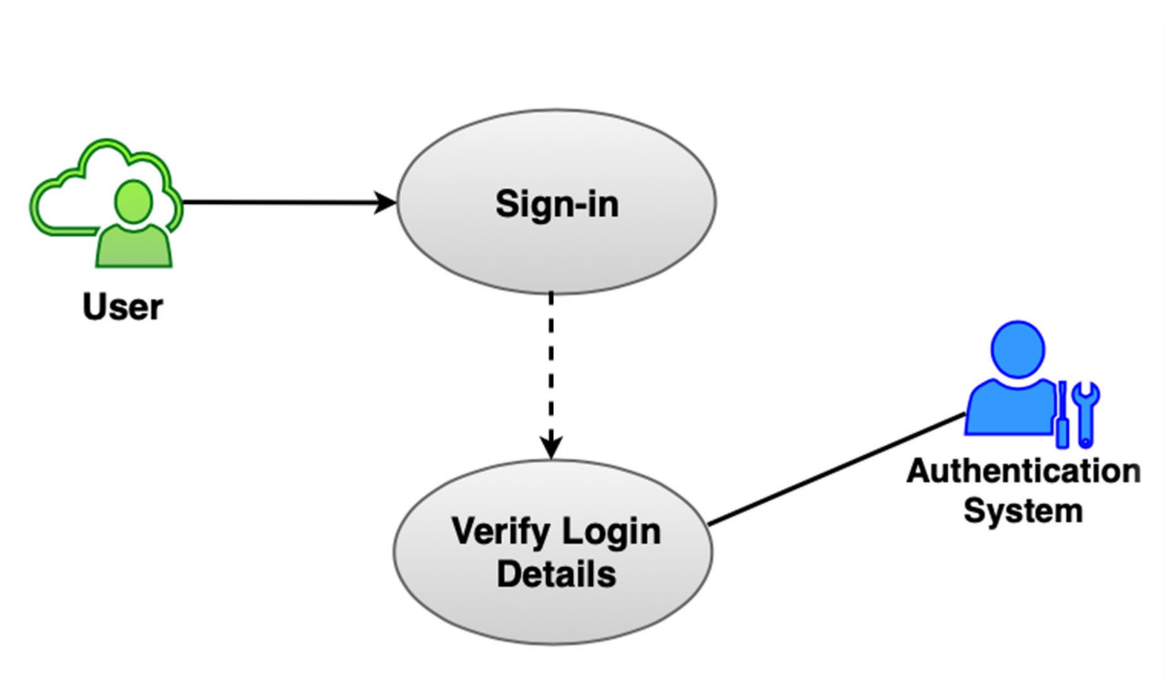
The *authentication system* creates a new account for the *user*.

2.1.2.2 Requirement 2: Sign-In

2.1.2.2.1 Description & Priority

Requirement 2 allows the application *user* to login to their account so they can see their account information, manage their saved words and phrases, etc.

2.1.2.2.2 Use Case



Scope
The scope of this use case is to let the *user* sign-in to their account.

Description
This use case describes how the *user* can login to their account.

Flow Description
Precondition
The application has been opened by the *user.*

Activation
This use case starts when a *user* selects the login button from the sidebar menu.

Main flow

1. The application shows the option of logging in or to create an account.
2. The *user* inputs their correct login details and clicks the “login” button.
3. The *authentication system* confirms the login details are valid.
4. The *user* is logged-in successfully and they are shown the account screen of the application.

Alternate flow

A1: Incorrect Details

1. The application notifies the *user* that their login details are correct.
2. The *user* can retry with their correct details.
3. The use case continues at position 3 of the main flow.

Exceptional flow

E1: Login Error

1. An error in the *authentication system* whilst trying to login to the *user's* account.
2. The *user* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The application presents the *user* with the account screen of the application.

Post condition

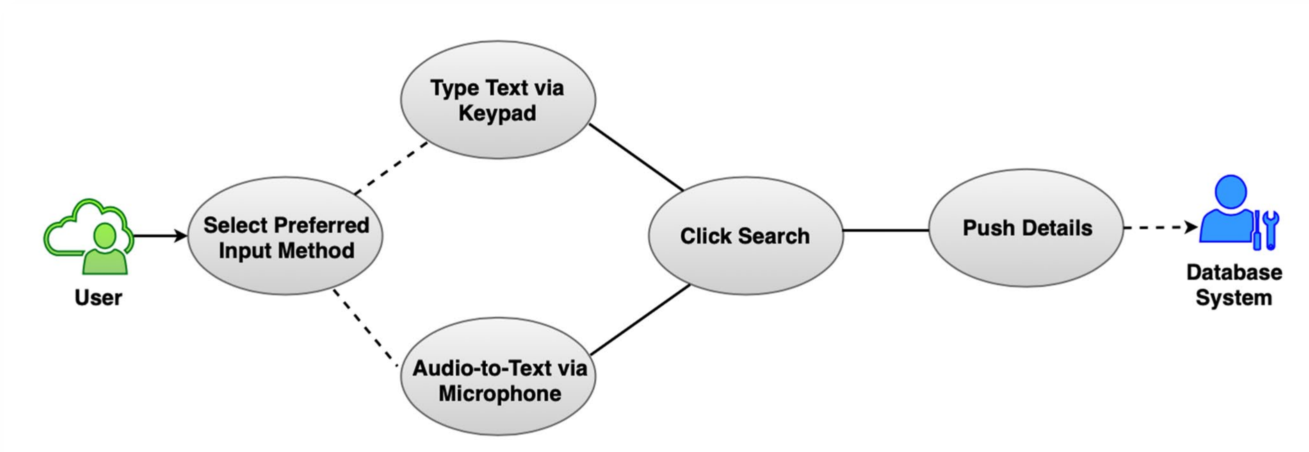
The application logs the *user* into their account.

2.1.2.3 Requirement 3: Search Word or Phrase

2.1.2.3.1 Description & Priority

Requirement 3 allows the application *user* to search for words or phrases via keypad or audio-to-text.

2.1.2.3.2 Use Case



Scope
The scope of this use case is to let the *user* search for a word or phrase of their choosing.

Description
This use case describes how a *user* can search for a word or phrase by either using the keypad or microphone.

Flow Description
Precondition
The application has been opened by the *user.*

Activation
This use case starts when a *user* opens the application and is presented the main screen.

Main flow

1. The application displays a search-bar.
2. The *user* selects their preferred method of input and clicks the “search” button.
3. The *database system* accepts the *user’s* search word and/or phrase from the search-bar and processes the request.
4. The application presents to the *user* the results page screen.

Alternate flow
A1: Incorrect Word or Phrase Inputted

1. The *user* clicks the “return-arrow” button on the results page screen.
2. The *user* returns to the main screen.
3. The *user* can retry with the correct word or phrase.
4. The use case continues at position 3 of the main flow.

Exceptional flow

E1: System Error

1. An error occurs in the *database system* whilst trying to process the *user's* search request.
2. The *user* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The application presents the *user* with the results screen of the application.

Post condition

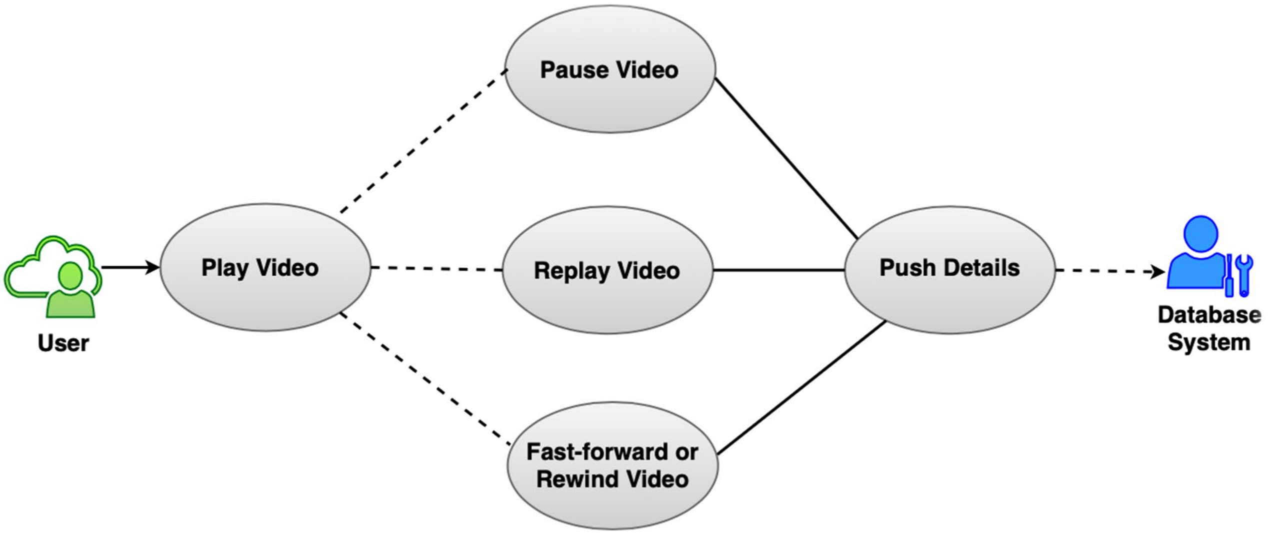
The *user* has selected their preferred video result on the application.

2.1.2.4 Requirement 4: Video Results

2.1.2.4.1 Description & Priority

Requirement 4 allows the application *user* to play the selected video result of their choosing.

2.1.2.4.2 Use Case



Scope
The scope of this use case is to let the *user* play the video result of their choosing.

Description
This use case describes how a *user* can play, pause, replay, fast-forward or rewind a video in the application.

Flow Description
Precondition
The application has been opened by the *user.*

Activation
This use case starts when a *user* has selected a specific video result from the results screen and clicks the "play" button.

Main flow

1. The *user* selects a video from the video results screen.
2. The application displays a video for the *user*.
3. The *user* clicks the "play" button to start the video.
4. The *database system* accepts the *user's* request and processes the request.
5. The application begins to play the video to the *user*.
6. The application displays the video options that are available; Pause, Replay, Fast-Forward or Rewind.

Alternate flow

A1: Incorrect Video Selected

1. The *user* clicks the “return-arrow” button on the video screen.
2. The *user* returns to the video results screen.
3. The use case continues at position 1 of the main flow.

Exceptional flow

E1: System Error

1. An error occurs in the *database system* whilst trying to process the *user’s* video request.
2. The *user* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The application presents the *user* with the video screen of the application.

Post condition

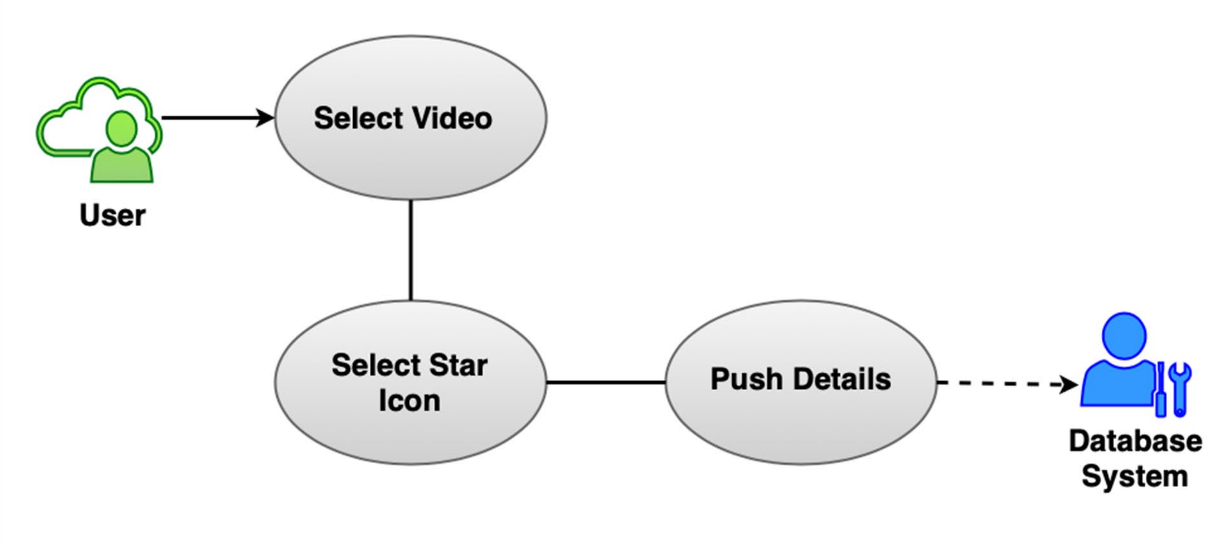
The *user* is watching their selected video on the application.

2.1.2.5 Requirement 5: Bookmark Video

2.1.2.5.1 Description & Priority

Requirement 5 allows the application *user* to save the selected video result of their choosing.

2.1.2.5.2 Use Case



Scope
The scope of this use case is to let the *user* save a video result of their choosing.

Description
This use case describes the how the *user* can save a video result in the application.

Flow Description
Precondition
The application has been opened by the *user*.

Activation
This use case starts when a *user* has selected a video result from the video screen and clicks the "star" button.

Main flow

1. The application displays the selected video to the *user*.
2. The *user* clicks the "star" button on the video screen.
3. The *database system* accepts the *user's* request and processes the request.
4. The application saves the video to the *user's* favourites section of their account.
5. The application presents to the *user* a "Saved to Favourites" message prompt on the video screen.

Alternate flow
A1: Incorrect Selection

1. The *user* unclicks the "star" button on the video screen.

2. The *database system* accepts the *user's* request and processes the request.
3. The application removes the video from the *user's* favourites section of their account.
4. The application presents to the *user* a "Removed from Favourites" message prompt on the video screen.

Exceptional flow

E1: System Error

1. An error occurs in the *database system* whilst trying to process the *user's* request.
2. The *user* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The application presents the *user* with the video screen of the application.

Post condition

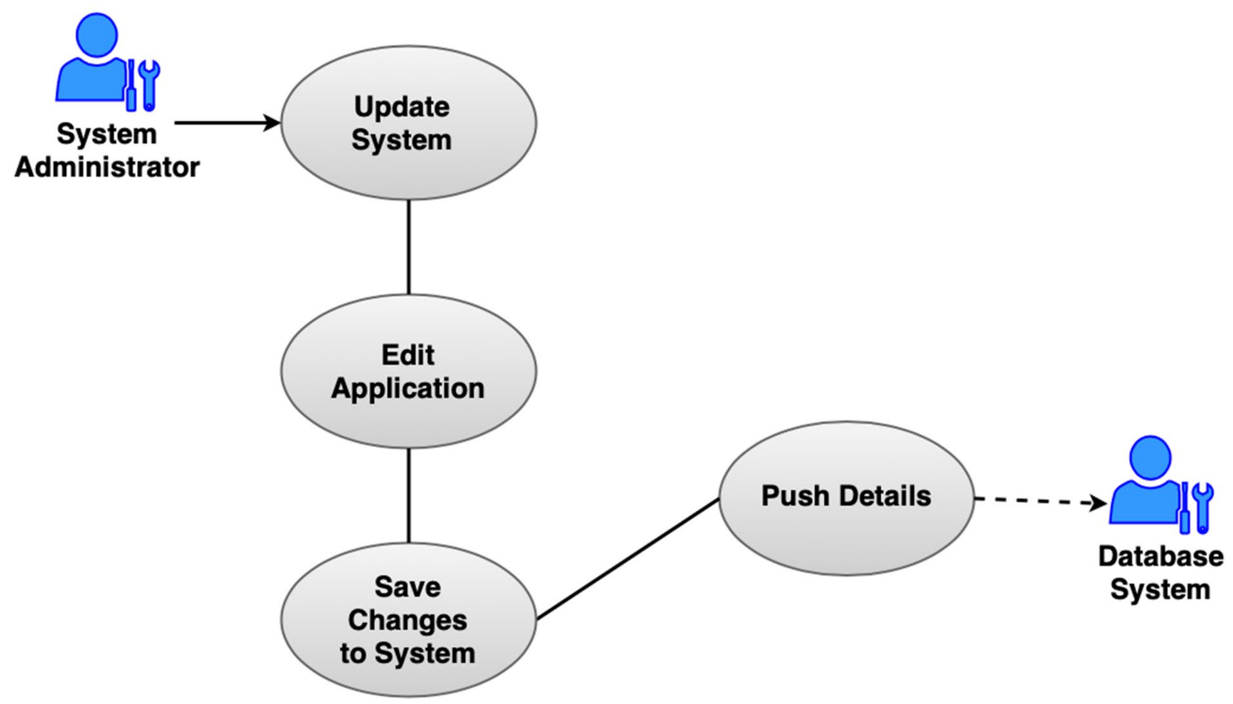
The *user* has saved their selected video on the application.

2.1.2.6 Requirement 6: Manage & Update – System Administrator

2.1.2.6.1 Description & Priority

Requirement 6 allows the application *administrator* to manage and update the application.

2.1.2.6.2 Use Case



Scope
The scope of this use case is to let the *system administrator* update and manage the application for *users.*

Description
This use case describes the how the *system administrator* can update and manage the application.

Flow Description
Precondition
The *system administrator* has accessed the application's database system.

Activation
This use case starts when a *system administrator* opens the database system and chooses to make changes to the application.

Main flow

1. The database system presents the *system administrator* with a login for administrators only.
2. The *system administrator* inputs their login details and clicks "login".
3. The database system grants the *system administrator* access.

4. The *system administrator* manages, updates and/or applies any changes that they wish to be made to the database system.
5. The *system administrator* saves the changes that are made, and the details are pushed to the database system.

Alternate flow

A1: Incorrect Details

1. The application notifies the *system administrator* that their login details are incorrect.
2. The *system administrator* can retry with their correct details.
3. The use case continues at position 3 of the main flow.

Exceptional flow

E1: System Error

1. An error occurs in the *database system* whilst the *system administrator* is trying to apply changes.
2. The *system administrator* understands that an error has occurred.
3. The use case continues at position 1 of the main flow.

Termination

The database system has been changed and updated by the *system administrator*.

Post condition

The *system administrator* has managed and updated the application.

2.1.3 Data Requirements

The list below contains the *data requirements* that are needed for this application:

1. **Logical Data Models –** these models help define how the application needs to be designed and implemented but not specific to the database. These models can be in the shape of a UML system architecture diagram, seen in *section 2.2*. These models also help to organise and describe the project's concepts and aims.

2. **Entities –** the entities are data types or tasks required in an application. The data types in this project consist of:
   - ⇒ ***Name –*** either a singular or plural value, that's a unique value.
   - ⇒ ***Parent Page –*** this determines the site hierarchy, where a specific page – such as the main page of the application – is specified as the *parent page* and the screen is the root of the application.
   - ⇒ ***Email –*** text box with a maximum number of characters and an email address validation.
   - ⇒ ***Text –*** text box a maximum number of characters.
   - ⇒ ***Boolean –*** a true or false data type. This will be used in the *terms of services* and *privacy policy* agreement checkbox seen in the create an account section of the application.

3. **Data Reports –** reports help give an insight of the overview of the application. The reports can consist of:
   - ⇒ ***Extracted Data –*** using tools, such as HKHealthStore which is built into XCode or Google Firebase, it can fetch real-time data from the database.
   - ⇒ ***Data Processing –*** using HKHealthStore or Firebase, they can run real-time database tests on the database's functionality so that you can optimize security, deploy live services into the app and interact with the back-end.
   - ⇒ ***Data Visualization - using*** the extracted and processed data, and applying it to a data studio, you can build customized reports to filter and categorize data.

2.1.4 User Requirements

The list below contains the *user requirements* that are needed for this application:

1. **End-User –** the end-user, or the main user of the application, is the target for this project. The end-user will use the application to create an account, learn Irish sign language and save videos to their favourite section of their account, if they wish to go back and re-watch them in the future.

2. **System Administrator User –** the system administrator is responsible to app maintenance and updating the app for any new changes being introduced to the system. The system administrator will use the application to login to the back end of the application, review the application's maintenance and update the system, if there is any.

3. **Developer –** the developer is responsible in the early stages of the application. The developer will build, code and implement the application from the beginning of development, as well as develop any future changes made to the application, which the system administrator will apply to the application.

2.1.5 Environmental Requirements

The list below contains the *environmental requirements* that are needed for this application:

1. **Development Environment –** this environment is where the application's designing, programming, debugging and implementing takes place. As stated in section *1.3 Technology*, I will be using **XCode** as my development environment.

2. **Test Environment –** for my test environment, I will also be using **XCode**. Within XCode, I will use **XCTest** which is a testing framework that breaks down into three testing phases: *Arrange, Act* and *Assert*.

3. **Staging Environment –** Once again, I will be using **XCode Simulator**. Within XCode, there is an **emulator** software. This emulator allows for an iOS device simulator to be ran to see the mobile application function off a virtual device.

As well as the above environments, a **Production Environment** can be used if the system administrator wishes to make any future changes or updates to the application. In the environment, any new builds or updates are applied into the production environment where users can see these new changes.

2.1.6 Usability Requirements

The list below contains the *usability requirements* that are needed for this application:

1. **Platform Compatibility –** the application must function correctly on the desired platform. The platform I have chosen to develop my application on is *iOS*, hence why I have chosen to use *XCode* as my development environment.

2. **Simple Navigation –** navigation is one of the main concerns with mobile application development. Clear navigation needs to limit the number of barriers a user might encounter whilst using the application. The navigation should be easy to understand for the user so that they won't get lost within the application.

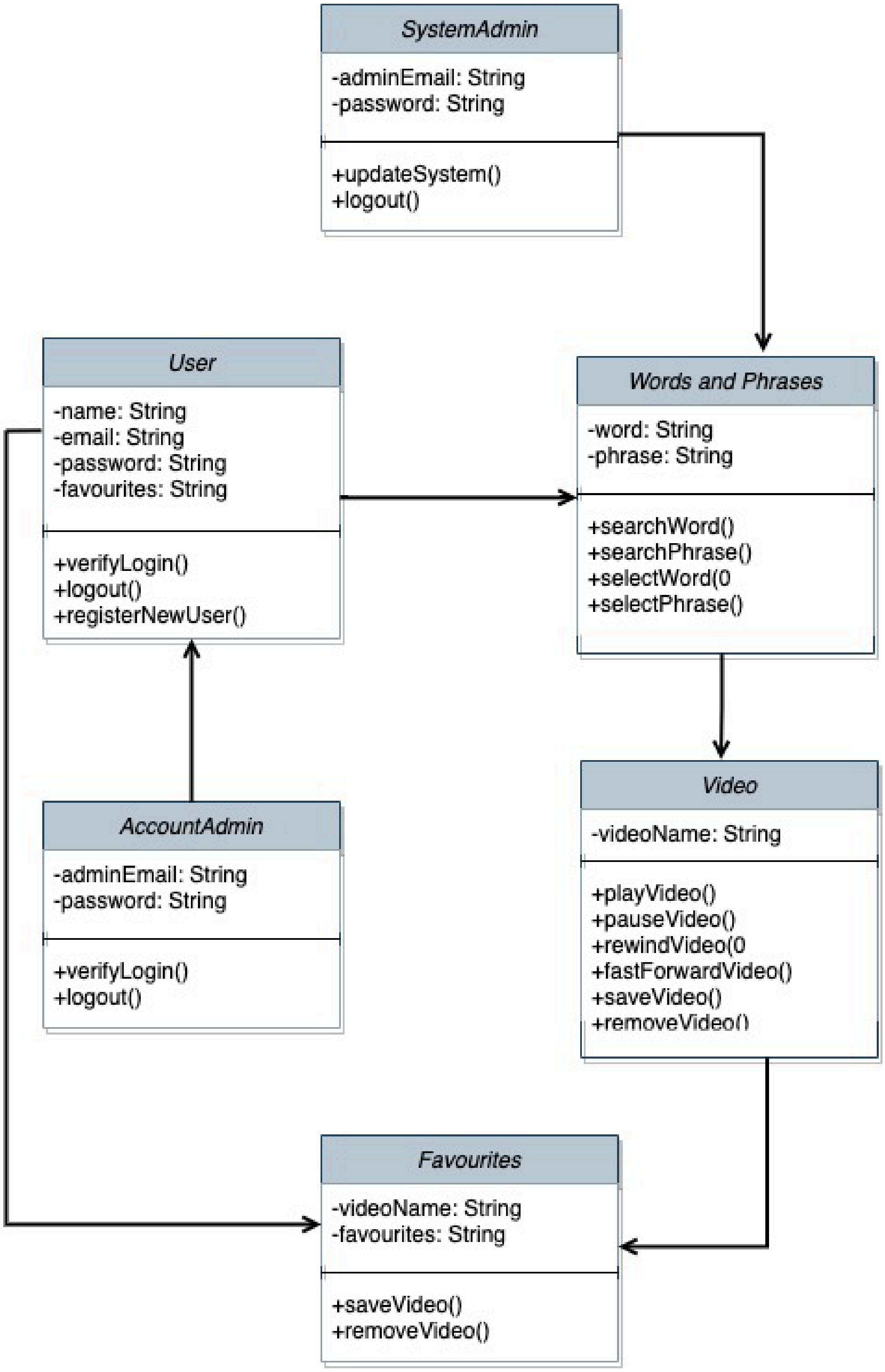
3. **Clear and Concise Content –** the simpler the content, the more practical the application is for the user. Including too much content on an application can be overwhelming for the user and can cause poor performance on the application for users, which can lead to frustration amongst users.

4. **Minimal Movements –** similar to *simple navigation*, clear navigation and minimal movements made by the user to receive the desired results. The fewer movements a user has to make, the closer the user is to completing their goal on the application, which makes a happy customer.

5. **Providing Service Immediately –** the application needs to deliver its main objective as soon as possible once the application has been opened by a user. The application needs to coax users into staying on the app which leads on to users keeping the application. It's critical to make sure you offer the service you are providing as soon as possible.

2.2 Design & Architecture

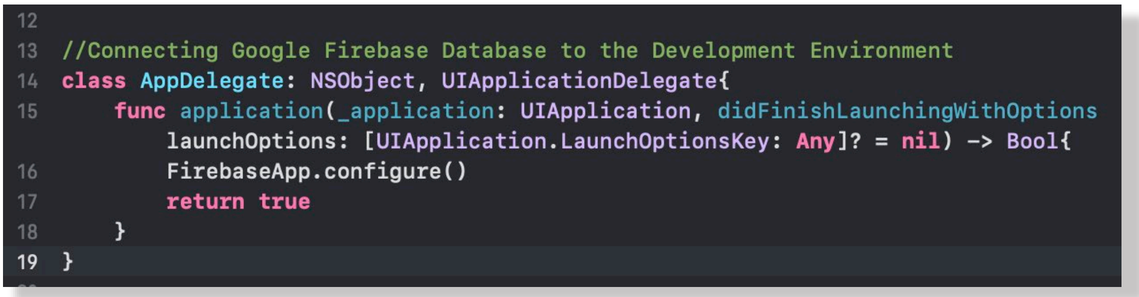
The below diagram represents the system's architecture. It consists of the following components: ***SystemAdmin, WordsAndPhrases, User, AccountAdmin, Video*** and ***Favourites.***



2.3 Implementation

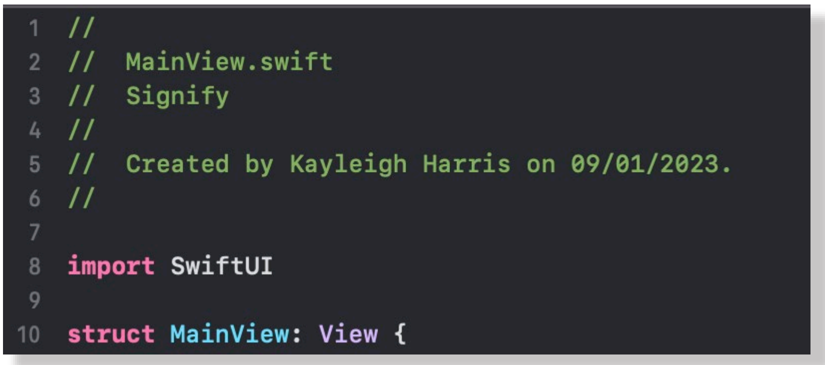
In my build for the application, I have a number of classes, functions and algorithms. The below consists of some of the classes, functions and algorithms within the application:

⇒ **SignifyApp** – this is the main class, or as XCode labels them ‘views’. This is the parent class that all other classes revolve around. The configuration for the Firebase database is coded into this class.



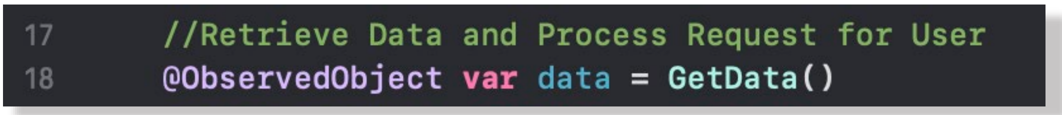
```
//Connecting Google Firebase Database to the Development Environment
class AppDelegate: NSObject, UIApplicationDelegate{
    func application(_application: UIApplication, didFinishLaunchingWithOptions
        launchOptions: [UIApplication.LaunchOptionsKey: Any]? = nil) -> Bool{
        FirebaseApp.configure()
        return true
    }
}
```

⇒ **Import SwiftUI –** this imports the SwiftUI framework into my classes on XCode. With this framework, I was able to have more control over the views, layouts and structures of the application. This framework also provided me with the functions for on-tap gestures, inputs into the application and controls on visibility. This framework was imported on majority of the classes/views in my development and it is the first line of code in these classes:

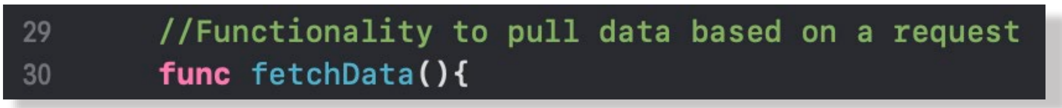


```
//
//  MainView.swift
//  Signify
//
//  Created by Kayleigh Harris on 09/01/2023.
//

import SwiftUI

struct MainView: View {
```

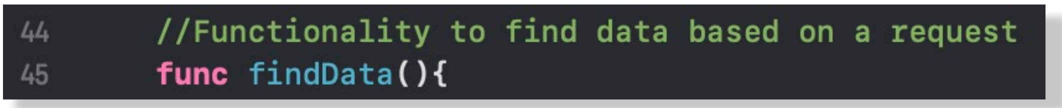
⇒ **GetData() –** this algorithm connects the classes/views with the data model. With this algorithm, it allows for the application to search through the data and find anything the matches the user’s search request. This algorithm works with two other lines of code which are **fetchData()** and **findData()** which finds any data matching the user’s input and the fetches the found data, to display out to the user:



```
    //Retrieve Data and Process Request for User
    @ObservedObject var data = GetData()
```

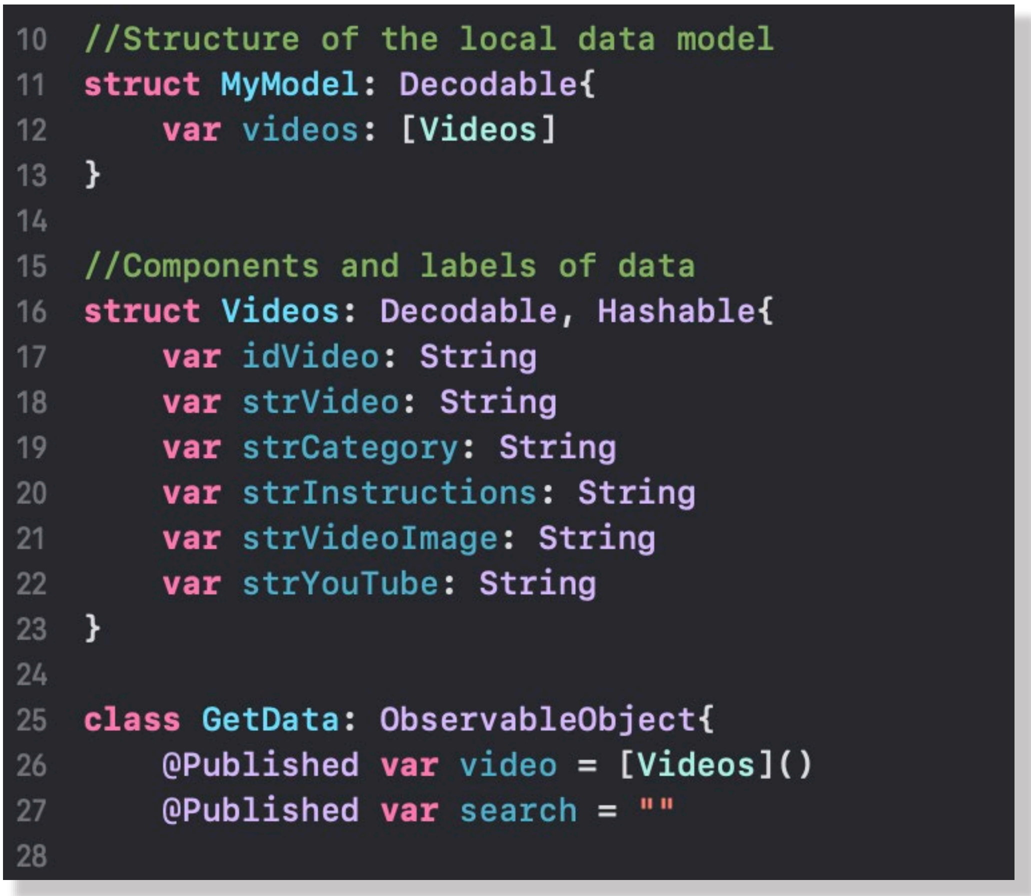


```
    //Functionality to pull data based on a request
    func fetchData(){
```



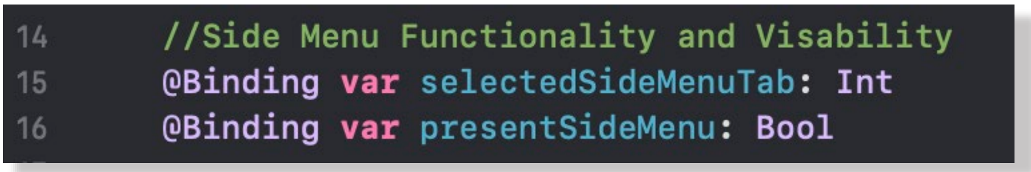
```
    //Functionality to find data based on a request
    func findData(){
```

⇒ **GetData Model –** this class is a very important part of the application. This class is responsible for retrieving, searching and filtering through data based on a user's request. This class works with the **GetData()** algorithm which I discussed above to find and fetch data requests. Unlike the other classes, this class doesn't act as a view and is used more for functionality. It holds the types of data, the data structure and its components:



```
//Structure of the local data model
struct MyModel: Decodable{
    var videos: [Videos]
}

//Components and labels of data
struct Videos: Decodable, Hashable{
    var idVideo: String
    var strVideo: String
    var strCategory: String
    var strInstructions: String
    var strVideoImage: String
    var strYouTube: String
}

class GetData: ObservableObject{
    @Published var video = [Videos]()
    @Published var search = ""

```

⇒ **@Binding –** this algorithm connects the side menu view with all the other views in the application. This algorithm also creates a connection to the data in the application. This algorithm is used in majority of the views/classes in the application. This algorithm also works with the SideMenu functionality so that the side menu can be visible on any class/view that this code in embedded into and delivers a user's request.



```
    //Side Menu Functionality and Visability
    @Binding var selectedSideMenuTab: Int
    @Binding var presentSideMenu: Bool
```

2.4 Graphical User Interface (GUI)

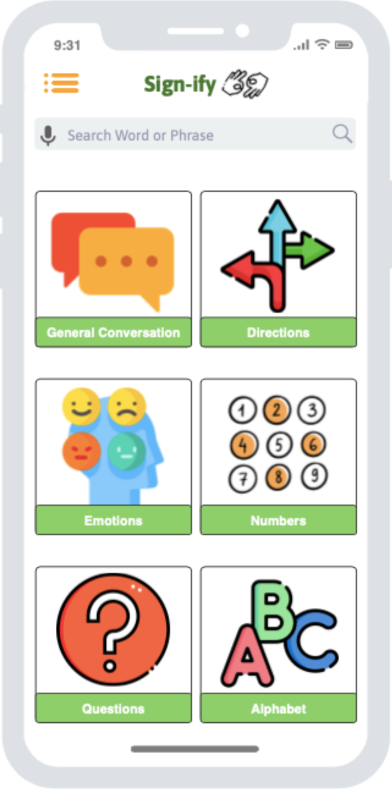


Figure 1 - *Main Screen*

In **fig. 1** is the *main screen* of the application. The name of the application and logo is displayed in the top-centre of the screen, as well as in the top left, we see an orange button to access the *side-bar menu* which you will see in **fig. 2.**

Underneath the name of the application, is the search-bar used to search throughout the application, including the *microphone symbol* to signify the use of audio-to-text is available.
After using the search-bar, the user will be brought to the *video results* screen, seen in **fig. 4**.

Underneath the search-bar, there are a range of categories that the user can search through with different video results, seen in **fig. 3.**

In **fig. 2** is the *side-bar menu* of the application. The *user's account* and information is displayed at the top of the side-bar menu - once the user has logged into their account, which you will see how in **fig. 5**.

Underneath the user's account, is the button to access the *main screen*, seen in **fig. 1**.
Underneath the main screen button, is the button users will use to access their *account details*, which is seen in **fig. 11**.

Users can also use the *favourites* button to access their favourite videos they've saved on the application, seen in **fig. 10.**
And at the very bottom of the side-bar menu, is the button to access the *settings* of the application.

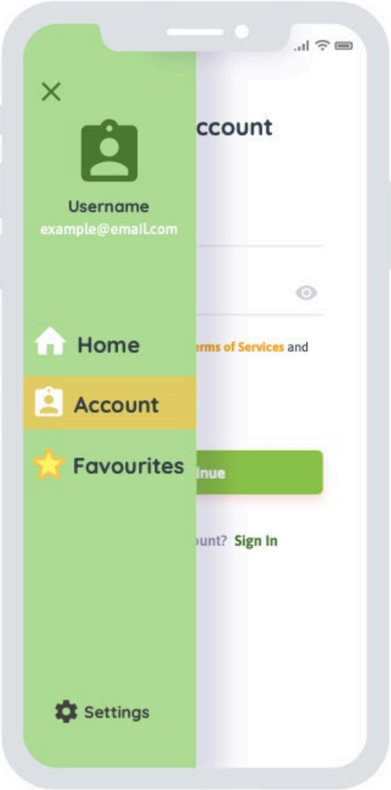


Figure 2 - *Sidebar Menu*

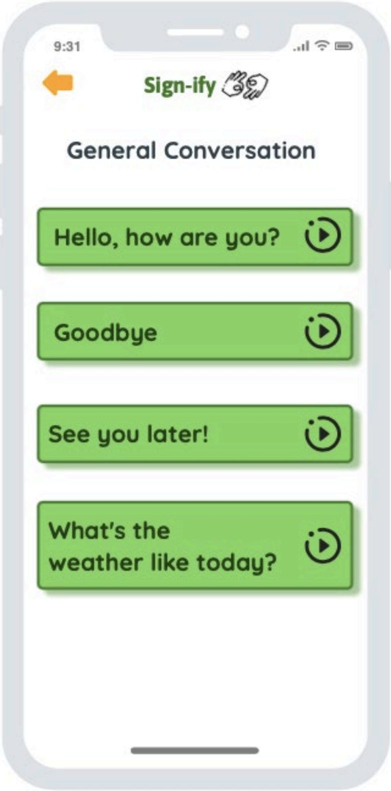


Figure 3 - *Category Results Screen*

In **fig. 3** is the *category results* screen of the application. After a user has selected a category from the *main screen* of the application, they can scroll through the different video results each category holds.

The user can click on a result which will bring them to the video screen, seen in **fig. 7**.

In the top left corner, there is a *return arrow* which will bring the user back to the *main screen* of the application.

In **fig. 4** is the *video results* screen of the application. After a user have used the search-bar, seen in **fig. 1**, any video results will be presented to the user.

The user can scroll through the results and select the desired video.

In the top left corner, there is a *return arrow* which will bring the user back to the *main screen* of the application.

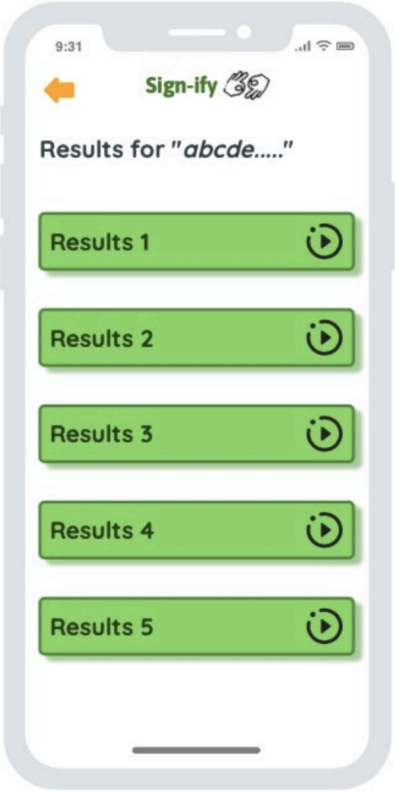
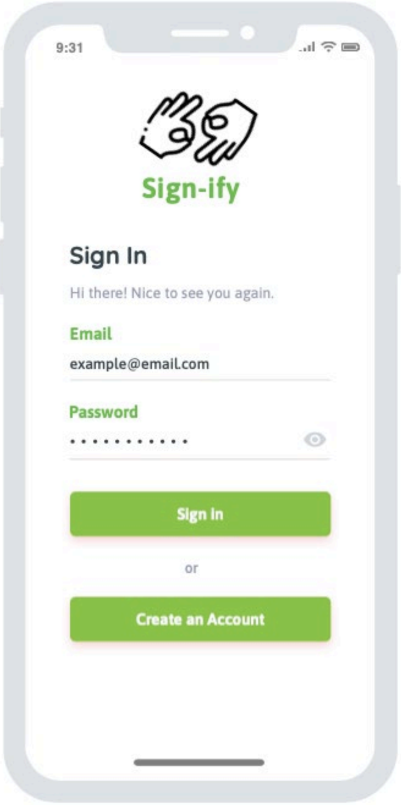


Figure 4 - *Video Results Screen*



Sign In

Hi there! Nice to see you again.

Email: example@email.com

Password: •••••••••••

Figure 5 - *Login Screen*

In **fig. 5** is the *login* screen of the application. When the user initially opens the application for the first time, they will be presented this screen. On the top of the screen is the application name and logo.

There is then a greeting found underneath the *Sign In* title, followed by the input boxes for entering an email and password.

Once the user creates an account, seen in **fig. 6**, the user can enter the account details, consisting of an email and password, and login into their account on the application.

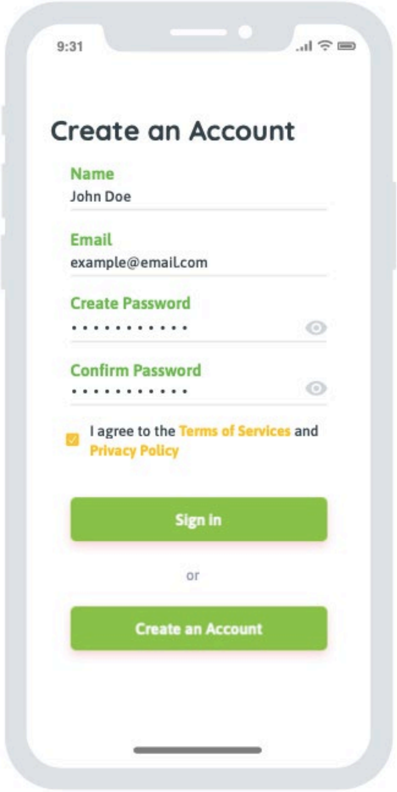
At the bottom of the *login screen*, there are two buttons. A *Sign In* button that the user can click once they've entered their details. And a *Create an Account* button which the user can click to sign-up to use the application.

In **fig. 6** is the *create an account* screen of the application.

There are four input boxes for entering a name, an email, a password and confirming the password. Once the user enters these details, they will need to agree to the *terms of services and privacy policy*, by ticking the box below the password input box.

Once the user has entered valid details, the application will communicate with the database to create a new account using the entered credentials.

Once the system has successfully created the account and the user has clicked the *continue button*, the user can login to the application.



Create an Account

Name: John Doe

Email: example@email.com

Create Password: •••••••••••

Confirm Password: •••••••••••

☑ I agree to the Terms of Services and Privacy Policy

Figure 6 - *Create an Account Screen*

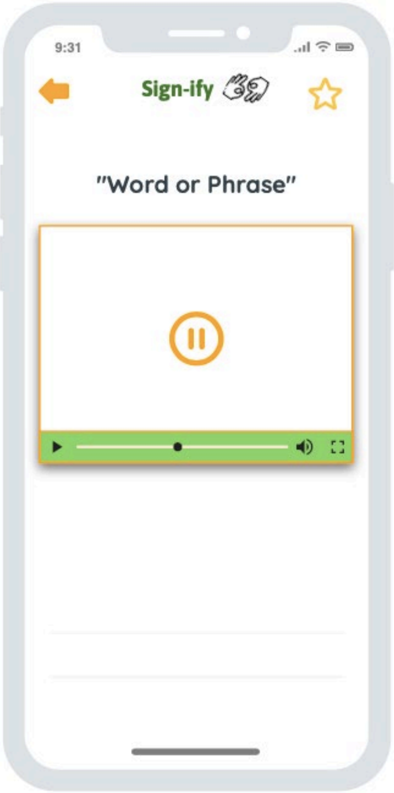


Figure 7 - *Video Screen*

In **fig. 7** is the *video* screen of the application. Once the user has selected a video result, seen in **fig. 4**, the user will be displayed a video to watch.

There is a video box, which has a play/pause feature for the user to use. As well as a video progress bar that they can use to rewind or fast-forward into the video.

There is also a volume button to increase/decrease the sound of the video.

In the top right corner, there is a *star button*. The users can click this button which will save this video to their favourites section of their account.
In the top left corner, there is a *return arrow* which will bring the user back to the *video results screen* of the application.

In **fig. 8,** there is a dialog box that appears once a user has clicked the *star button* in the top right corner of the screen.

If a user selects the button, the message, "*Saved to Favourites*" will appear on the screen and the video will be added to their *favourites section*, seen in the side-bar menu and in **fig. 10**.

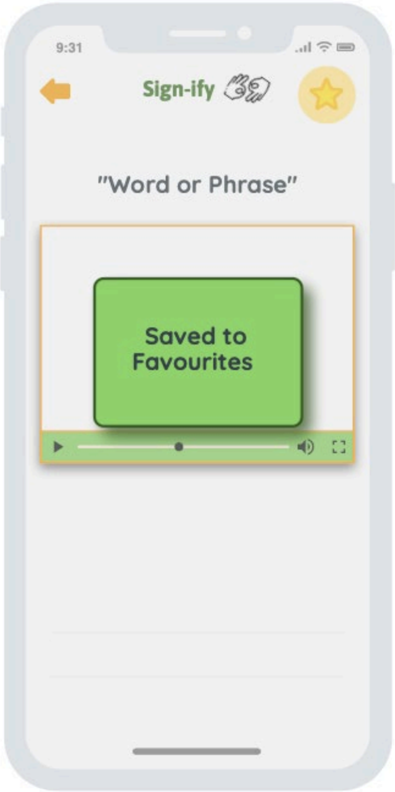


Figure 8 - *Dialog Box*

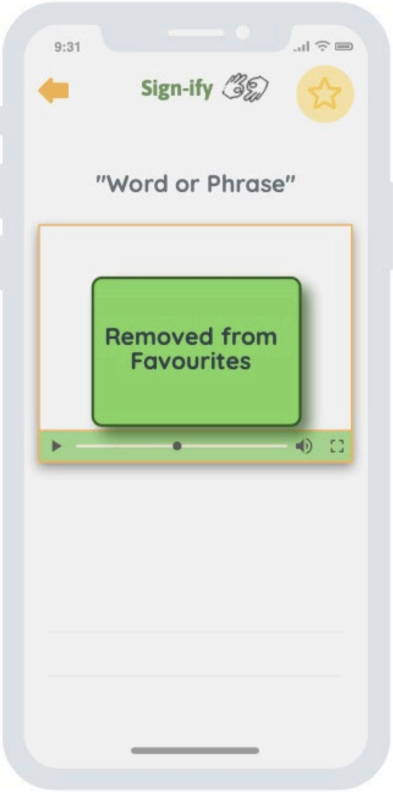


Figure 9 - *Dialog Box (2)*

In **fig. 9,** similar to **fig.** 8, there is a dialog box that appears once a user has unclicked the *star button* in the top right corner of the screen.

If a user deselects the button, the message, "*Removed from Favourites*" will appear on the screen and the video will be removed from their *favourites section*, seen in the side-bar menu and in **fig. 10**.

In **fig. 10,** we see the *favourites* screen of the application. The user can access this section of the app through the side-bar menu seen in **fig. 2**.

The user can see any videos they may have saved whilst using the account and access the videos through here.

In the top left corner of the screen, there is a *menu button* which the user can use to access the side-bar menu and navigate through the application.

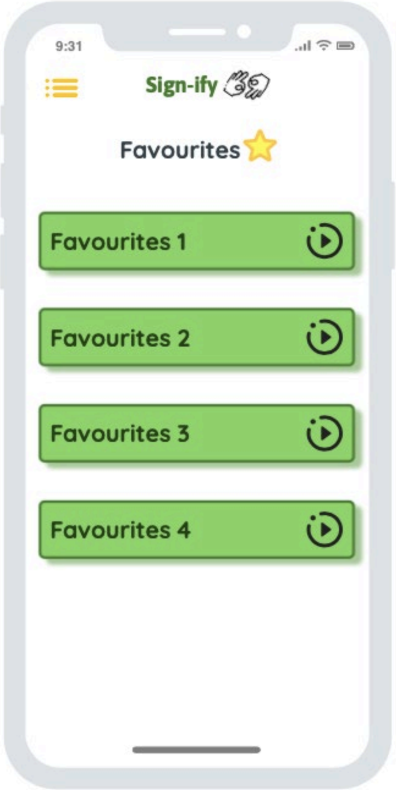
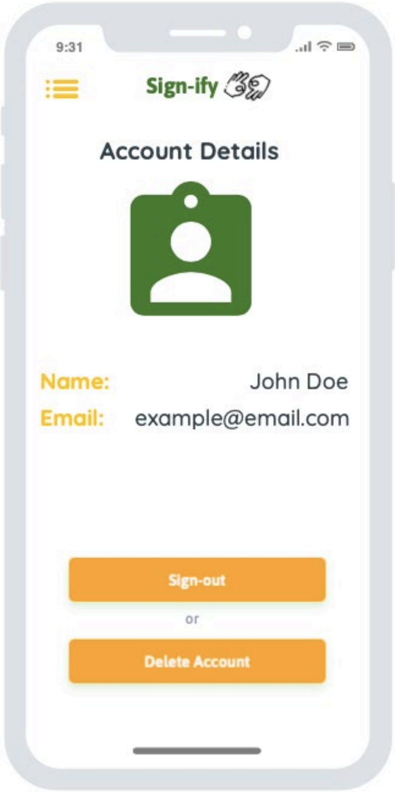


Figure 10 - *Favourites Screen*



Name: John Doe
Email: example@email.com

Figure 11 - *Account Details Screen*

In **fig. 11,** we can see the *account details* screen. There is a profile picture section, which the user can upload a picture of themselves to their account.

Underneath the profile picture, the user can find their account details, including their name and email address linked to their account.

At the bottom of the screen are two buttons. The first button is a *sign out,* if the user wishes to sign-out of their account. The second button is a *delete account* button, if the user wishes to delete their account entirely.

In the top left corner of the screen, there is a *menu button* which the user can use to access the side-bar menu and navigate through the application.

In **fig. 12,** we see the *settings* screen of the application. The user can access this section of the app through the side-bar menu seen in **fig. 2**.

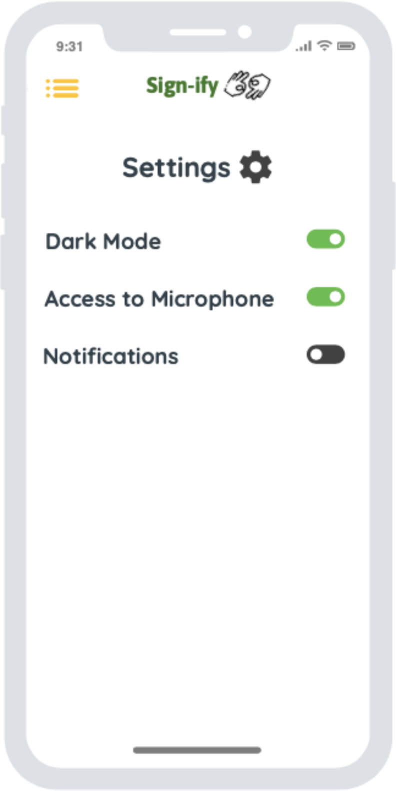
The application displays the different options available to the user that they can enable/disable.

These options include:

⇒ Dark Mode.

⇒ Access to the device’s microphone settings.

⇒ Push notifications.



Dark Mode ●
Access to Microphone ●
Notifications ○

Figure 12 - *Settings Screen*

2.5 Testing

2.5.1 Testing Approaches

Testing in projects is key! By testing the code I have written, it ensures that the code behaves the way you want it to and that any changes to code won't cause any affects to the rest of the application.

In my project, I used **Unit Testing** for testing all the components in my application. By using Unit Testing, I was able to test each element of the application, including its layout, design and functionality.

I also used the **Arrange-Act-Assert model** during my testing phases. This model consists of the following test phases:

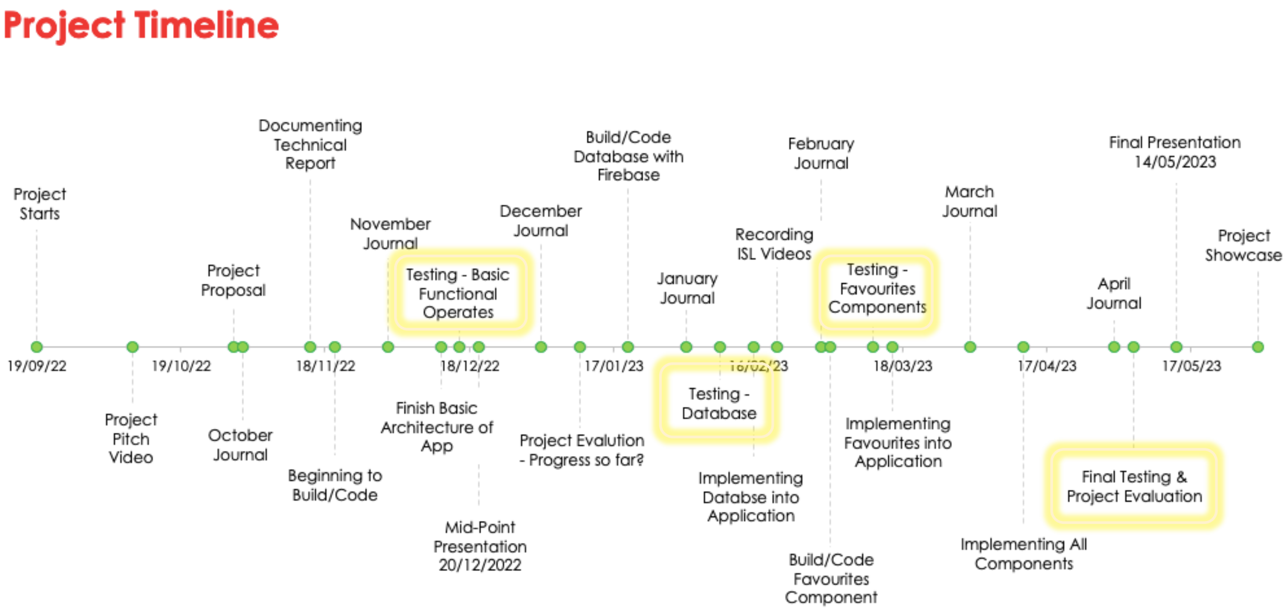
1. **Arrange –** setting up the components and conditions for the test.
2. **Act –** declaring the component, function or method for the test.
3. **Assert –** state that one form of end-condition is true.

I used the following testing tools during the testing phases of my project:

⇒ **XCTest –** as I am using *Swift* as my main programming language and XCode as my development environment, I used XCTest for testing within the environment. As I mentioned previously, XCTest is a framework within XCode that allows for testing to be ran within the environment. It includes testing elements such as unit tests, performance tests and UI tests.

⇒ **Firebase Test Lab –** this test lab can be used to test the application on a range of different devices and configurations, so you can see how well the application will perform in the hands of the end-users.

As stated in the *timeline* in my project proposal, the diagram below shows the full project timeline for this application. In the diagram below, I have highlighted when I had planned for testing to occur during my project timeline:


Project Timeline
Project Starts
19/09/22
Project Pitch Video
19/10/22
Project Proposal
October Journal
Documenting Technical Report
18/11/22
Beginning to Build/Code
November Journal
Testing - Basic Functional Operates
Finish Basic Architecture of App
18/12/22
Mid-Point Presentation 20/12/2022
December Journal
Project Evalution - Progress so far?
17/01/23
Build/Code Database with Firebase
January Journal
Testing - Database
16/02/23
Implementing Databse into Application
Recording ISL Videos
February Journal
Build/Code Favourites Component
Testing - Favourites Components
18/03/23
Implementing Favourites into Application
March Journal
17/04/23
Implementing All Components
April Journal
Final Testing &amp; Project Evaluation
Final Presentation 14/05/2023
17/05/23
Project Showcase

Starting from left to right, the first highlighted testing phase from the diagram above is testing for the ***basic architecture*** of the application. This includes the navigating from each screen on the application and accessing the side-bar menu.

The second highlighted testing phase is testing for the ***database*** created using Google Firebase. This component will test the databases and the application's capability to create account on the application.

The third highlighted testing phase is testing for the component of the application to ***save videos*** to a user's account. This includes testing whether the database communicates and processes the user' request.

The final highlighted testing phase on the far right of the diagram is for testing the ***overall application***. This includes testing whether all the individual components of the application work together – including the search-bar, creating an account, playing videos and saving videos.

2.5.2 Testing Evidence

Testing Phase:		**Basic Functionality/Search**		
No:	**Test Description:**	**Expected Result:**	**Actual Result:**	**Pass/Fail:**
1.1	Side menu button	Side menu button appears once button is tapped.	When clicked, button did not display side menu.	**Fail**
1.2	Side menu button	Side menu button appears once button is tapped.	When clicked, side menu displayed on screen.	**Pass**
2	Home screen button	Home screen displays once button is tapped in the side menu.	When clicked, home screen displayed on screen.	**Pass**
3	Search screen button	Search screen displays once button is tapped in the side menu.	When clicked, search screen displayed on screen.	**Pass**
4	Account details screen button	Account screen displays once button is tapped in the side menu.	When clicked, account details screen displayed on screen.	**Pass**
5	Favourites screen button	Favourites screen displays once button is tapped in the side menu.	When clicked, favourites screen displayed on screen.	**Pass**
6	Settings screen button	Settings screen displays once button is tapped in the side menu.	When clicked, settings screen displayed on screen.	**Pass**

7.1	Typing in Search bar	Device keyboard should appear and user should be able to type.	Device keyboard did not appear	**Fail**
7.2	Typing in Search bar	Device keyboard should appear and user should be able to type.	Device keyboard appears once search bar is clicked.	**Pass**
8.1	Video Player	Video should play once tap gesture made by user.	No video or audio played – Only first image of the video.	**Fail**
8.2	Video Player	Video should play once tap gesture made by user.	Video did not display, only Audio sound was heard.	**Fail**
8.3	Video Player	Video should play once tap gesture made by user.	Video and audio plays on-tap gesture on display.	**Pass**
9.1	Dark Mode toggle - Yes	Dark mode appearance displays once toggle slides to 'Yes'.	Display appearance changed to Dark Display.	**Pass**
9.2	Dark Mode toggle – No	Dark mode appearance displays once toggle slides to 'No'.	Display appearance changed to default Display.	**Pass**

Testing Phase:		**Firebase Database**		
No:	**Test Description:**	**Expected Result:**	**Actual Result:**	**Pass/Fail:**
1.1	Firebase to XCode	Connection made between XCode environment and Google Firebase.	Connection Failed.	**Fail**
1.2	Firebase to XCode	Connection made between XCode environment and Google Firebase.	Connection Failed.	**Fail**
1.3	Firebase to XCode	Connection made between XCode environment and Google Firebase.	Connection established between XCode and Firebase.	**Pass**

Testing Phase:		**Favourites/Star Videos**		
No:	**Test Description:**	**Expected Result:**	**Actual Result:**	**Pass/Fail:**
1.1	Star button - Clicked	Star button display changes and highlights	Button display does not change	**Fail**

1.2	Star button – Clicked	Star button display changes and highlights	Button display does change	**Pass**
1.3	Star button – Unclicked	Star button display changes and un-highlights	Button display does change	**Pass**
2.1	Dialog Box – Saved	When clicked, "Saved to Favourites" dialog box appears to users.	Dialog box appears	**Pass**
2.2	Dialog Box – Removed	When clicked, "Removed from Favourites" dialog box appears to users.	Dialog box appears	**Pass**
3.1	Video – Saved	When clicked, video is saved to Favourite screen and database.	Display changes and dialog appears but functionality to save does not work.	**Fail**
3.2	Video – Saved	When clicked, video is saved to Favourite screen and database.	Display changes and dialog appears, and Video saves to Favourites screen and DB.	**Pass**
3.3	Video – Removed	When clicked, video is removed from Favourite screen and database.	Display changes and dialog appears but functionality to remove does not work.	**Fail**
3.4	Video - Removed	When clicked, video is removed from Favourite screen and database.	Display changes and dialog appears, and Video is removed from Favourites screen and DB.	**Pass**

Testing Phase:		**Final Testing – Overall Application**		
No:	**Test Description:**	**Expected Result:**	**Actual Result:**	**Pass/Fail:**
1.1	Search bar	When input made to search bar, app displays related results.	App displayed all videos, no filter to match results.	**Fail**
1.2	Search bar	When input made to search bar, app displays related results.	App displayed related videos first but still displayed all videos	**Fail**
1.3	Search bar	When input made to search bar, app displays related results.	App displayed only related videos.	**Pass**

2.6 Evaluation

After finishing with Unit Testing, I began to evaluate the application as a whole. As shown previously in the unit tests, I have unit tested in four sections:

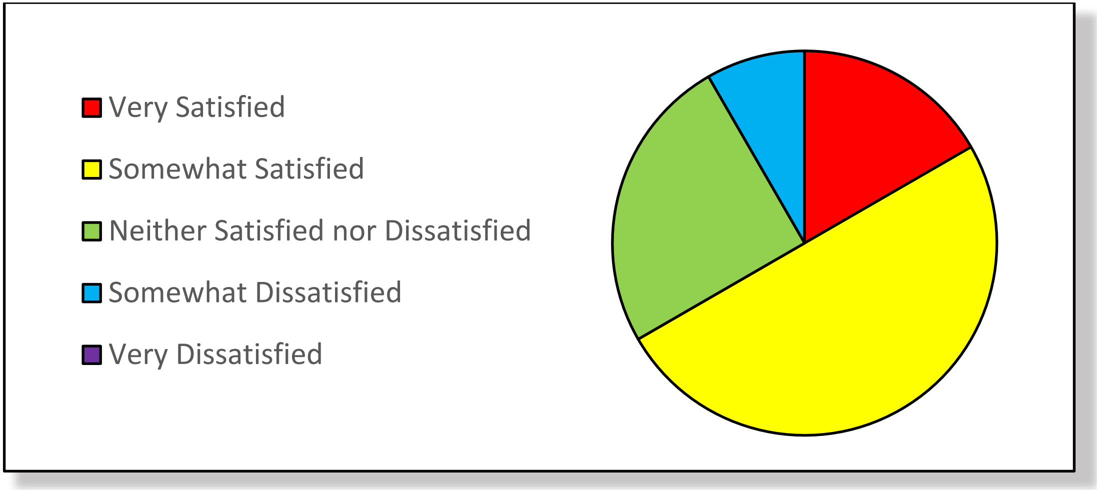
1. **Basic Functionalities and Search**.
2. **Firebase Database.**
3. **Favourites Functionality.**
4. **Overall Functionality.**

I have found that the basic functionality of the app works and with its easy design, the app is usable and easy to navigate through with the use of the side menu.

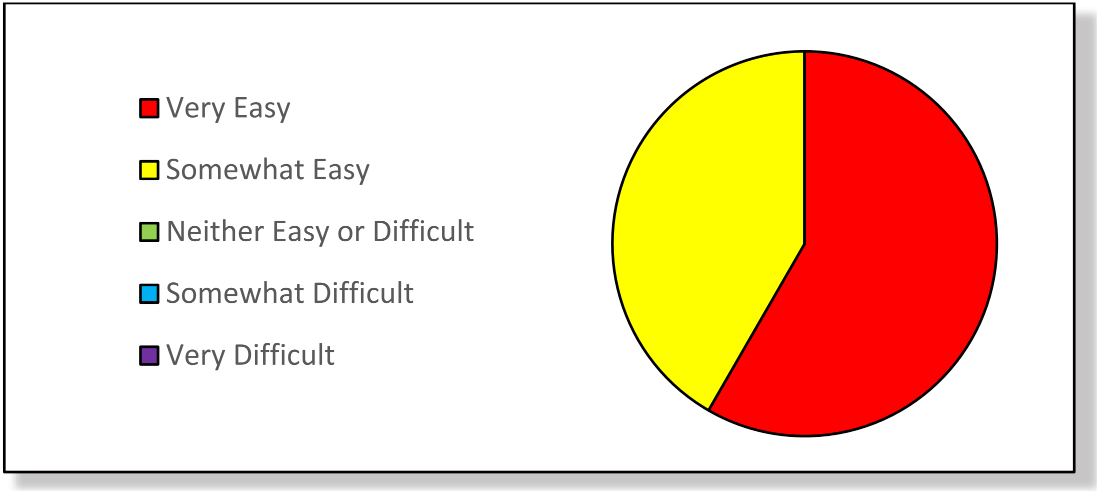
I decided to create a **survey** with seven questions and surveyed people on the application. I wanted to evaluate the application's functionality and usability. By using **Microsoft Forms,** I was able to create a survey and share a link with the people to submit their answers. With their responses, Microsoft Forms was able to convert their responses into data, which then creates different graphs and percentages – making it easier to sort through the data and responses.

The below were the questions I asked and the responses, with the statistics from the data.

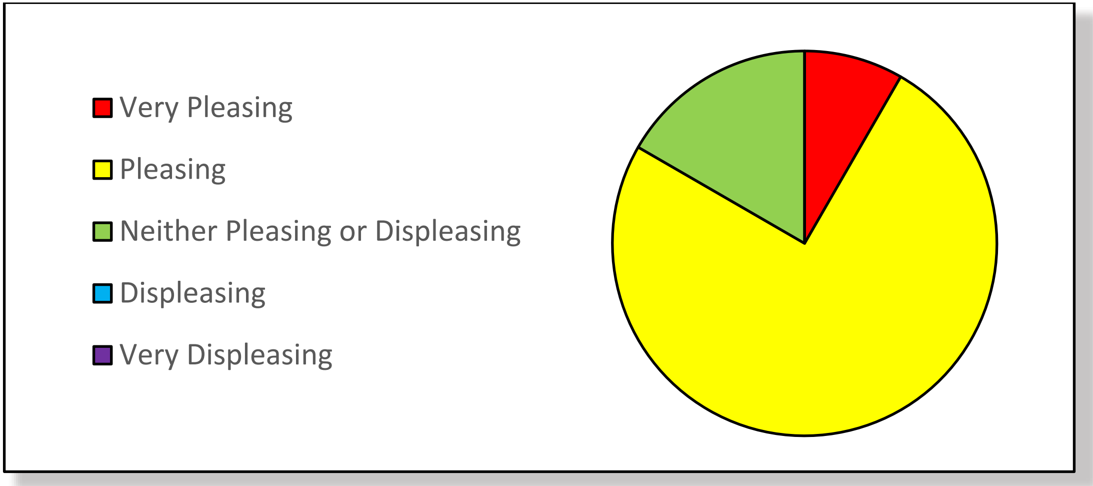
Question 1: *How satisfied were you with how the application performed/functioned?*



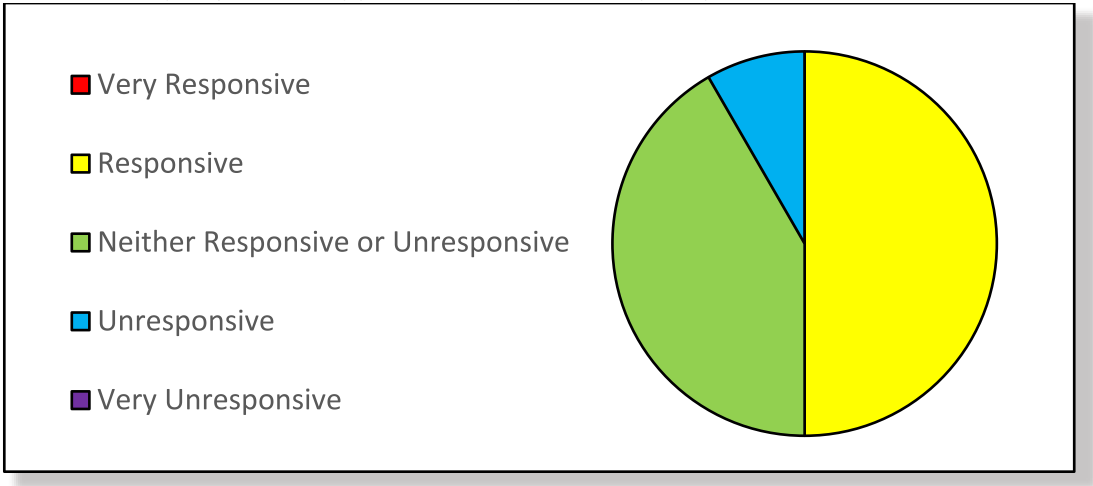
Question 2: *Did you find the application user friendly? (ie. Easy to Use/Understand)*



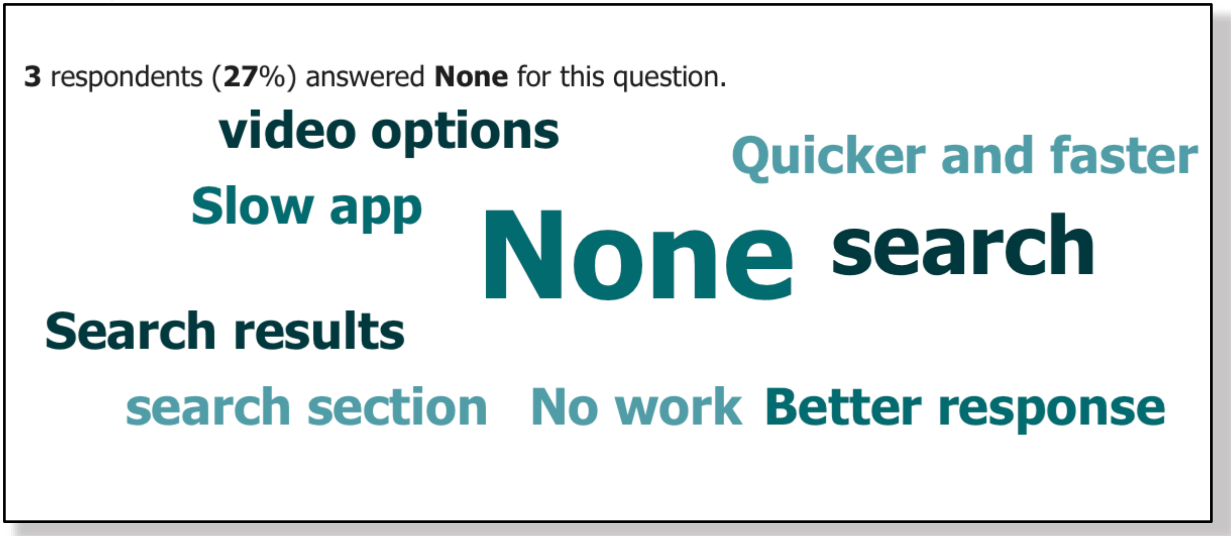
Question 3: *Did you find the application visually pleasing?*



Question 4: *Did you find the application responsive/speedy?*

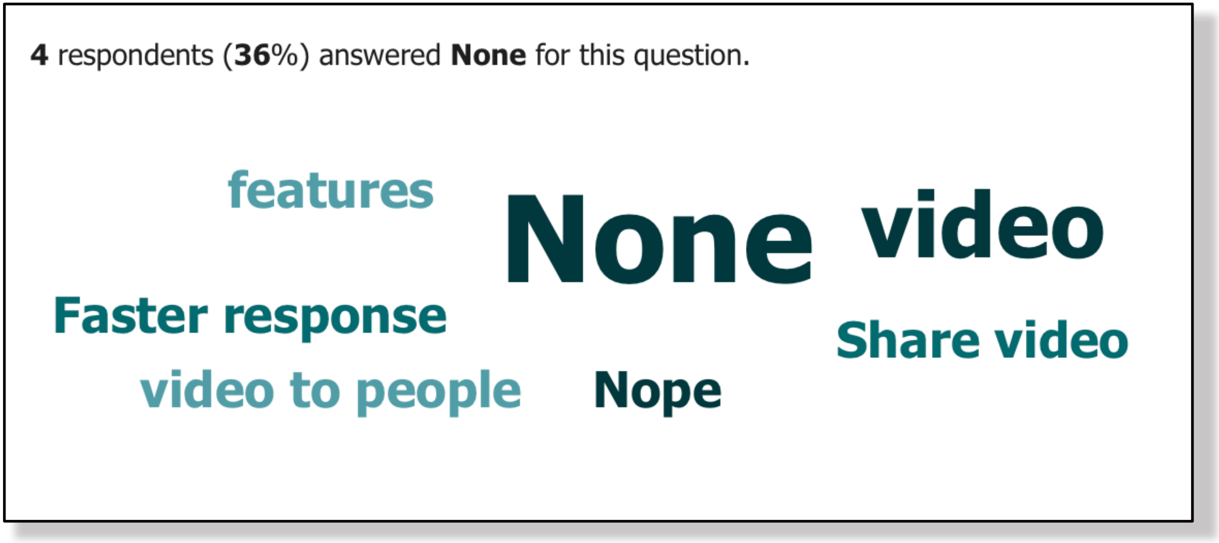


Question 5: *Were there any features of the application you found needed more work/adjustments?*



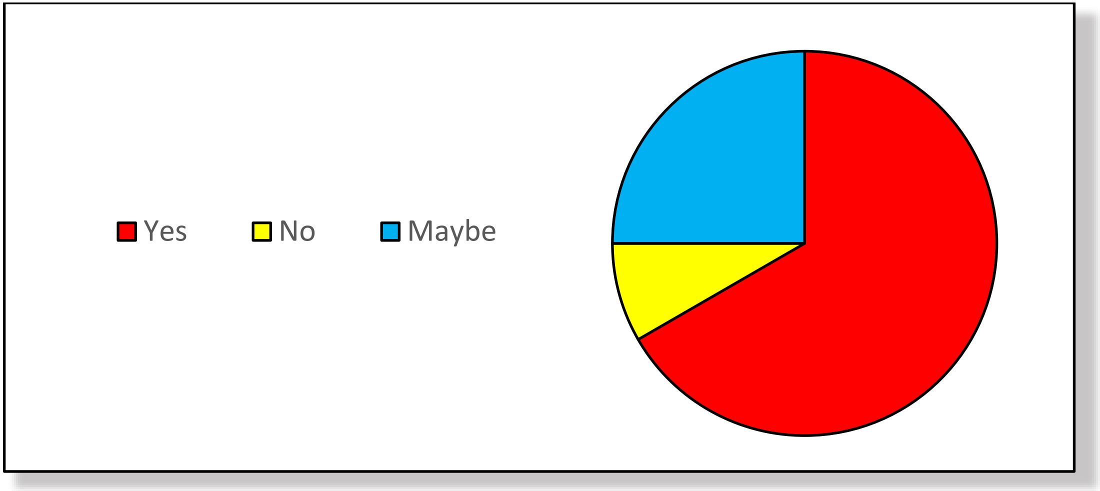
3 respondents (**27%**) answered **None** for this question.

Question 6: *Are there any new features you'd like to see on the application?*



4 respondents (**36%**) answered **None** for this question.

Question 7: *If this application was on the App Store, would you support/use this application if the opportunity arose?*



3 Conclusions

As I specialised in **Digital Business Transformation**, how the evolution of technology and its continuous growth in today's world has always fascinated me. By undertaking this project, I got to see how technology has helped so many businesses to advance in the world.

By undertaking this project, I've been given the opportunity to learn new programming languages, new development environments and various other technologies I would of never have heard of in my previous experience, and it's now given me the ability to develop my skills that I can use for future projects.

Although I was familiar with iOS devices, I am an Android mobile user. I had initially planned to develop an Android-based platform app, see in my original project proposal, however due to complications I decided to take a risk and undertake a whole new environment and platform. I had never developed using XCode before, and had little experience with iPhones in the past. I had little experience with Google's Firebase from my second year of college (Team Project Module) but had to learn even further as I was integrating it with my new development environment, XCode.

Even though there were many set-backs throughout this entire process, it has helped me to learn even more from the mistakes that were made and I can now apply what I learnt in any situation in the future.

I feel like this project was a success, even though I didn't achieve everything I set out at the start of this project, I learnt so much from this experience and I feel like I have developed as a programmer and a person.

4 Further Development or Research

I believe if I had the opportunity and additional time to carry out further development and research, and the available resources that I could accomplish the below:

- ⇒ **Multiple Languages –** With enough resources and additional time, I would like to incorporate multiple other sign languages into the application, such as ASL (American Sign Language) and BSL (British Sign Language). With these additional languages, this expands the audience for the application and further breaks the barriers of communication. I'd like to even show the differences (if any) between two sign languages and similarities.

- ⇒ **Further Categories -** As seen from my graphical user interfaces (GUIs) in section 2.4, there are six categories, such as *General Conversation* and *Directions*. I would like to expand these categories even further to include categories such as; *Animals, Locations* and *Objects*.

- ⇒ **Audio Input –** In my original proposal, I wanted to not only have keyboard input, I also wanted audio-to-text input as an option. Unfortunately, due to set-backs in the project and learning an entire new environment and language, I had to sacrifice the audio input functionality in the project. With additional resources and time, I'd like to incorporate this functionality into the application.

5 References

⇒ Apple Inc., 2023. *Testing your apps in XCode | Apple Developer Documentation.* [Online]
Available at: https://developer.apple.com/documentation/xcode/testing-your-apps-in-xcode
[Accessed January 2023].

⇒ ClearBridge, 2020. *7 Best Practices to Overcome Mobile Apps Usability Issues.* [Online]
Available at: https://clearbridgemobile.com/7-best-practices-to-overcome-mobile-app-usability-issues/
[Accessed November 2022].

⇒ Fuerte, F. & Ching, C., 2019. *How to Use a Swift Enum.* [Online]
Available at: https://codewithchris.com/swift-enum/
[Accessed March 2023].

⇒ Google Developers, 2023. *Add Firebase to your Apple project | Firebase for Apple platforms.* [Online]
Available at: https://firebase.google.com/docs/ios/setup
[Accessed March 2023].

⇒ Hamilton, T., 2023. *Unit Testing Tutorial - What is, Types & Test Example.* [Online]
Available at: https://www.guru99.com/unit-testing-guide.html
[Accessed March 2023].

⇒ Lee, C., 2021. *What Is a Logical Data Model? | GoodData.* [Online]
Available at: https://www.gooddata.com/blog/how-build-logical-data-models-scale-analytical-applications/
[Accessed December 2022].

⇒ M.Abbas, 2023. *Side Menu in IOS SwiftUI..* [Online]
Available at: https://medium.com/geekculture/side-menu-in-ios-swiftui-9fe1b69fc487
[Accessed March 2023].

⇒ OS System, 2023. *Mobile App Architecture (Key Components & Factors to Consider).* [Online]
Available at: https://os-system.com/blog/mobile-app-architecture-how-to-design-it/
[Accessed May 2023].

⇒ Pitre, B. G., 2014. *Swift for Beginners.* Illustrated ed. San Francisco: Peachpit Press.

⇒ Pysmenna, K. & Baryshevskiy, A., 2022. *How to Write a Mobile App Requirements Document (+ Free Template Download) - Mind Studios.* [Online]
Available at: https://themindstudios.com/blog/mobile-app-requirements-document/#userrequirements
[Accessed October 2022].

6 Appendices

6.1 Project Proposal

National College of Ireland

Project Proposal
Sign-ify
31st October 2022

BSHCDBT
Digital Business Transformation
2022/2023
Kayleigh Harris
x19471636
x19471636@student.ncirl.ie

Contents

1.0 Objectives

Sign-ify is a Text-to-Sign Language and Audio-to-Sign Language mobile application. It is a mobile-friendly tool where users can input text or audio words and phrases and learn or grow their knowledge of sign language.

In the Sign-ify mobile app, users can type words or phrases, or using their microphone on their mobile device, can speak words or phrases and the application will process and translate the user's input into the sign language physical action.
Users of the application can also save their favourite or most used words/phrases they have learnt via the application in the *starred section* of the application.
This project sets out to achieve the following goals:

- ⇒ To develop a mobile application which can be used to learn the vocabulary of Irish-Sign Language (ISL) through audio and text input.
- ⇒ To remove the communication barrier between hard-of-hearing, deaf, speech difficulty and mute communities.
- ⇒ To ease the learning of sign language by making a more convenient and affordable solution rather than paying-out and attending classes.

2.0 Background

I chose to undertake this project as sign language is such an important form of communication – it allows you to communicate with a wide range of individuals, including hard-of-hearing, deaf individuals, speech difficulty and mute individuals.

The use of sign language allows you to also communicate in areas including students in mainstream schools, deaf schools or college and universities. As well as deaf or hard-of-hearing residents and in areas of business. Sign language can also improve the communication amongst families who may have deaf, hard-of-hearing or mute family members.
Now with the help of technology and smartphones, a sign language mobile application can become a useful tool for users who wish to learn sign language, develop their skills or who wish to communicate with the above communities.

I plan to meet the objectives that I have set out in Section 1 by:

- ⇒ Developing a user-friendly mobile application with the use of many technical tools, such as Android Studio, Firebase by Google and different programming libraries.
- ⇒ Discussing and consulting with disability support services and members of the deaf, hard-of-hearing, speech difficulty and mute community to understand the benefits an application like this could help them.
- ⇒ Providing a more favourable alternative to learning sign language in classes or courses.

3.0 State of the Art

From my prior research into sign language applications that already exist, there are a variety on the Android Google Play-Store as well as the Apple App Store. Even though there are a mixed variety of sign language applications, including apps focusing on ISL (Irish Sign Language), ISL* (Indian Sign Language), BSL (British Sign Language) and ASL (American Sign Language), compared to the actual of number of mobile apps that exist, the percentage of sign language applications is incredibly small.

Google Play Store (2021):
2.783 million apps and games, of which **30** apps were based on Sign Language.

Apple App Store (2021):
3.593 million apps and games, of which **60+** apps were based on Sign Language.

My focus for my application is highlighting ISL (Irish Sign Language). From my search, I have so far only discovered one mobile application that focuses on Irish Sign Language (Concise Irish Sign Language by K Doherty – Google Play Store). Because of my focus, I think my project, i.e., mobile application, will stand out compared to other existing applications because of ISL.
My application will differ from similar works of others as I plan to include an audio input for a search function for words and/or phrases. My application will also include a media/news section to inform users of events, media and news for the hard-of-hearing, deaf, speech difficulty and mute communities. And as stated in the previous paragraph, due to my focus being in ISL (Irish Sign Language), my application will differ from others due to the extremely low existence of ISL mobile applications on the mobile app market.

4.0 Technical Approach

For my mobile application, I plan to take the approach of either **Native Development** or **Hybrid Development**.

With **Native** Development, this approach uses platform-specific programming languages, development environments and other tools, such as Android Studio, Android SDK and Java programming language. When developing using a Native Approach, the application usually performs better and allows complete customization, the use of interface elements and more complex animations. The only disadvantage of using the Native Approach for development is that the application needs to be separately developed in each platform (iOS/Android).

With **Hybrid** Development, this approach uses WebView which are platform-specific components that allow web content to be displayed within the application instead of opening it on a web browser (Safari/Chrome). The Hybrid approach uses web technologies and tools2, such as HTML, CSS and JavaScript. The Hybrid approach is more ideal for my project when it comes to the news/media article components of my application as well as incorporating the Native approach for complete customization and development environments.

I have broken down the tasks and activities for this project in key stages/phases. I have a **7-phase** process for this project, which I have discussed in further detail in Section 7. In these phases, I have set out the tasks I wish to accomplish in these phases whilst working with a timeline (also attached in Section 7) that I have targeted these phases to be completed in.
I have several rounds of testing planned, including a final round of testing in early May for any last-minute adjustments and end-user tests. I have broken down the project planning and pre-development preparation tasks within phases 1-4, application development in phase 5, application testing in phase 6 (With calendared testing rounds in the Timeline) and the application launch in the final phase, phase 7.

5.0 Technical Details

The implementation languages I plan to use in the development of my mobile application are **Python** as the main implementation language, as well as **HTML, CSS** and **JavaScript** for customizing WebView components of the application.

The principal libraries I plan to use in the development of my mobile application are **Python programming libraries**. This includes, Pyaudio to transform the speech input to text and PyCharm code compiler.
Other technologies and tools I plan to use are:

- ⇒ **Android Studio** – the development environment I will build the application in.
- ⇒ **Google Speech API** – transforms audio to text.
- ⇒ **Dependency Parser** – to analyse sentence structure and word relationships.
- ⇒ **Google Firebase** – the Database to hold datasets.

The important algorithms and data structures that I plan to use in the development of this mobile application are:

- ⇒ **Sort Algorithm** – arranges the data, depending on the requirement. The sort algorithm can consist of *Quick Sort, Heap Sort* and *Counting Sort*.
- ⇒ **Search Algorithm** – searches through sorted datasets to find matches to an input and return a value.
- ⇒ **Language Detection Algorithm** – takes the text input and determines which language the text inputted is written in.
- ⇒ **Arrays** – used to store data. In this case, speech processing arrays to which each speech signal is an array.
- ⇒ **Strings** – used in search engines and spell checks.
- ⇒ **Linked Lists** – used to connect elements through links, such as webpages and images.

As mentioned in the previous section, I am considering using either the **Native** or **Hybrid approach** for the development of this mobile application. Apps developed using the Native approach operate at a high level of performance, they are fast and responsive due to being built for the specific platform and operating system. As well as performance, there is full control on creativity on the application content and visual aspects.

Apps developed using the Hybrid approach can be targeted to all platforms (iOS/Android). This is easier for developers to create the application as their code can be used on all platforms, whilst using the Native Approach, the code must be built per platform.

6.0 Special Resources Required

For this mobile application, I may need to use the following special resources for development purposes and visual purposes for the application:

⇒ **Android Studio** – development environment for Android applications.

⇒ **XCode** – development environment for Apple applications.

⇒ **Firebase** – App development platform to build application databases.

⇒ **Menu Resources** – A sidebar menu for users to access saved phrases/words they have learned on the app.

⇒ **Style Resources** – styles and designs to apply to the application's visual components.

⇒ **Layout Resources** – layout styles to apply to the application for the platform's resolutions and screen sizes.

⇒ **Animation Resources** – set animations with graphical descriptions for the physical action of the sign.

⇒ **Icon Resources** – material icons generally used in most mobile applications.

7.0 Project Plan

I have broken down the project into **7 phases**; including initial planning, development, testing and launch.

Phase 1 – Project Planning:
In this first phase (current phase), I will have defined the *objectives* and *goals* of the project (mobile application). I will determine who my *target audience* is and if there is a need/demand for this project. I will also do research into *existing projects* that are already on the market and how I can make my project differ from others, and how I can make my project stand-out. During this phase, I will have submitted my Project Pitch Video, as well submitting my Project Proposal documentation. I will not be submitting an ethics form yet, due to lack of datasets currently available for the project.

Phase 2 – Storyboard/Use Cases:
In this second phase, I will brainstorm the visual concepts of the application with the use of *storyboards*. I will also brainstorm the types of technologies and tools I can use to develop this project, as well as create Use Cases diagrams to imagine the project's use. These components will be included in my Project Analysis and Planning documentation for the Mid-Point presentation.

Phase 3 – Technical Resources:
In this phase, I will have a final understanding of what I visually want my application to look like, as well as know what technologies, systems and tools I will use to develop my application. I will have chosen which platform I want to build my application for (Android/iOS) and the format the application can function on. These components will be included in my Requirements Engineering documentation for the Mid-Point presentation. I will also have ethics forms submitted during this phase to gain access to datasets I may need during my project.

Phase 4 – Prototypes, Design and Prep for Development:
This will be a short phase, where I will design a brief prototype for the application I plan to develop, so I can map out the individual components that will need design elements, coding and data structures. This phase will help me have a blueprint of the app and how it should function visually. I will include visuals and basic coding of the prototype(s) in my Mid-Point Presentation.

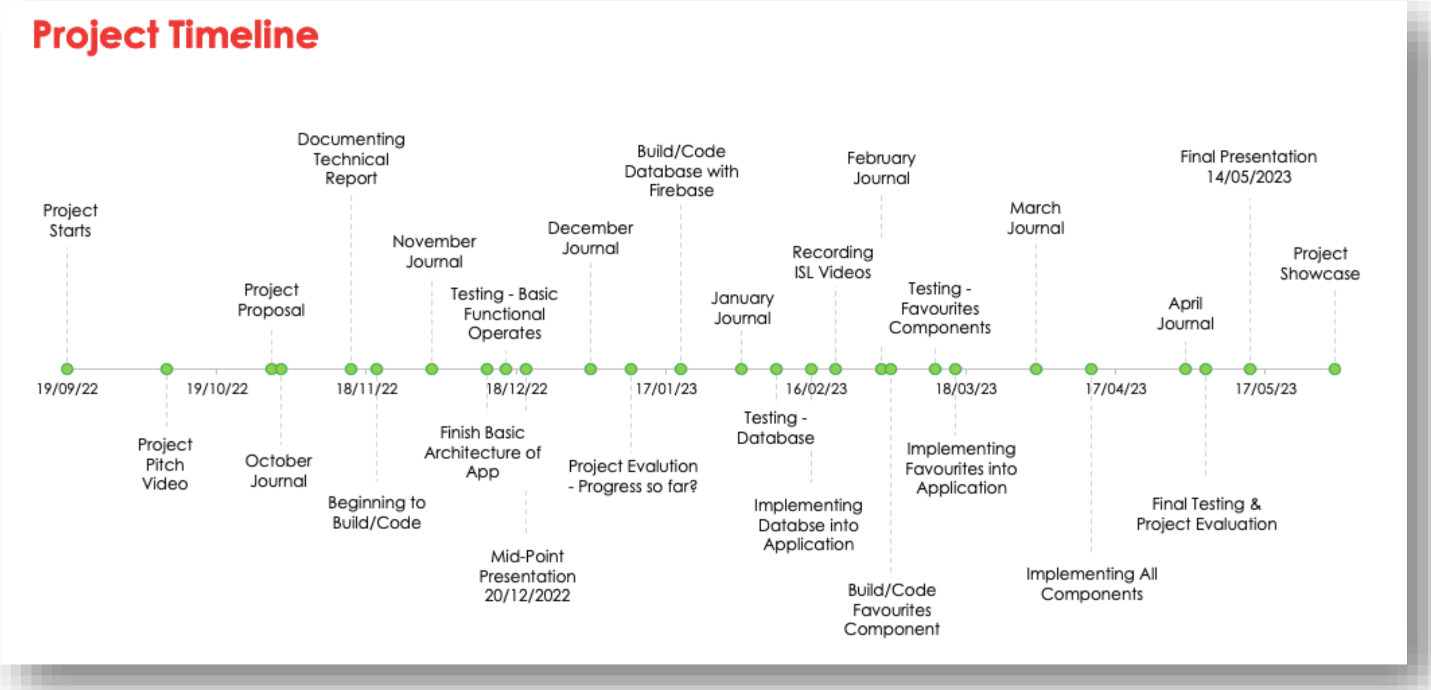
Phase 5 – Build of the Application:
This will be the most technical phase and longest phase of the project. Now that I will have all the individual components designed and mapped out, with the help of the blueprints and prototypes from phase 4, I need to build these components and the application and then I can move onto testing phases.

Phase 6 – Testing Phase:
This is probably the most important phase throughout the project, as testing is crucial in making sure your application is functional, responsive, usable and consistent. I plan to do at least two rounds of testing before the submission of my Mid-Point presentation on any

components/elements that I have built on. Following on into the second semester, I plan to do more in-depth testing on every element of the application and then move into integrated testing to see if the application works together as a whole. The forms of testing I plan to use are discussed in further detail in section 8.

Phase 7 – Launch:
The final phase of the project, the launch of the project. This phase can only be implemented once the testing phase of the project has surpassed and has met all the objectives listed in the initial proposal for this project.

Below I have attached a timeline for my project, with key dates for submissions and testing. I have also included in this timeline, an **evaluation stage** – this will occur after the submission of my Mid-Point Presentation (Early to Mid-January) where I will evaluate the progress of the project, any difficulties I have faced so far and whether it is meeting the objectives set-out at the beginning of this project.



Project Timeline

Project Starts

19/09/22

Project Pitch Video

19/10/22

Project Proposal

October Journal

Documenting Technical Report

18/11/22

Beginning to Build/Code

November Journal

Finish Basic Architecture of App

18/12/22

Testing - Basic Functional Operates

Mid-Point Presentation 20/12/2022

December Journal

Project Evalution - Progress so far?

17/01/23

Build/Code Database with Firebase

January Journal

Testing - Database

16/02/23

Implementing Databse into Application

Recording ISL Videos

February Journal

Build/Code Favourites Component

Testing - Favourites Components

18/03/23

Implementing Favourites into Application

March Journal

Implementing All Components

17/04/23

April Journal

Final Testing & Project Evaluation

Final Presentation 14/05/2023

17/05/23

Project Showcase

8.0 Testing

Without thorough testing during the development phase, testing phase and final phase of projects, there are high chances of encountering bugs, unexpected app crash, bad user experience and performance issues. Without thorough testing, the release of the application is risky and could lead to potential loss of customers, time and revenue.

During the development of my mobile application, I plan to do thorough system testing, testing for the application's **functionality**, **usability** and **consistency**. I plan to use either **PyTest** or **Robot** testing tools, which are a part of the python libraries I plan to use.

Functional Testing is required to make sure the application **functions** based on the objectives defined in section 1. An important factor for functionality testing is human interaction – the use of human interaction for a mobile application is key to testing if the application is user-friendly, visually pleasing and functional.

Usability Testing is required to make sure the application is **efficient, effective** and **satisfactory**. Efficiency in the application to see if it is accurate and complete where users can accomplish the objectives set out for the application. Effectiveness in the application to see if the resources used in the development of the application are also accurate and complete. And satisfaction in the application to see if the application is user-friendly and is acceptable as a working system for users.

Consistency Testing is required to make sure the application is up to quality and robustness. This is also an important factor has it is necessary to assess the aspect of robustness in the application.

Integration Testing is an important phase of testing to see all individual elements/components of the application come together and see if they function well as a group. During the development of my application, I plan to do integration in phases. I will integrate individual components together in small numbers and test. Once testing on this initial integrated test group succeeds, I will add to this integrated group further and slowly test and integrate until all individual components are grouped together for one larger test.

As mentioned previously in the *functionality* testing, **End-User Testing** is incredibly important to see if the application is usable, functional and visually pleasing. End-user testing can help gain insight to application changes that may be needed, user suggestions that can help performance and visual design of the application, and it can help detect issues that could occur in user interface and experience.

6.2 Reflective Journals

October Journal:

Supervision & Reflection	
Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: October

What?

In the month of October:

- ⇒ **Brainstormed –** I brainstormed different ideas and concepts for my project.
- ⇒ **Project Pitch Video –** I submitted my Project Pitch Video, where I pitched the idea of a mobile application that translate text and/or audio input into the physical action of sign language.
- ⇒ **Pitch Results –** I received my results to my Project Pitch Video. My pitch was approved with amendments to be made.
- ⇒ **Supervisor Meeting –** After submitting my pitch, I had a meeting with my project supervisor to discuss the amendments and questions suggested for my project. We also organised another meeting to follow up on the project idea and to discuss my Project Proposal documentation.
- ⇒ **Project Proposal –** I submitted my Project Proposal Documentation, where I discussed the project objectives, technical approaches, plans and testing.

So What?

Project Progress:

⇒ I have submitted my first two assignments for the project.

⇒ After discussing the project with my supervisor in further detail, I have been delayed in submitting my Ethics form as there was a lack of resources for sign language datasets.

Successes:

⇒ Creating an idea for my project.

⇒ Submitting project pitch video.

⇒ Having project idea approved with amendments.

⇒ Submitting project proposal documentation.

⇒ Having a meeting with my project supervisor and organising a further follow up meeting.

Challenges:

⇒ Lack of resources and datasets available for my project.

⇒ Delay in submitting Ethics forms.

⇒ With the lack of resources and datasets available for sign language, I may have to manually record signs myself which could take out a large amount of project development time.

Student Signature	

November Journal:

Supervision & Reflection

Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: November

What?

In the month of November:

- ⇒ **Supervisor Meeting –** After submitting my proposal, I had another meeting with my project supervisor to discuss the amendments that needed to be made on my proposal and any questions I had. We also organised another meeting to follow up on the project idea and to discuss my datasets/resources.
- ⇒ **Project Proposal –** I have started to make some adjustments to my second version of my project proposal, including adjusting the timeline and adding more detail into the testing parts of my timeline.
- ⇒ **Android Studio –** I installed Android Studio as this will be the platform and environment, I plan to use to build my application off.
- ⇒ **Basic Design –** I have started to create the basic architecture for my mobile application on Android Studio.
- ⇒ **Resources -** I reached out to the Irish Deaf Society in search for Irish sign language (ISL) datasets. They were kind enough to supply me with contact information for other developers who have created applications for ISL.

So What?

Project Progress:

⇒ I have submitted my first two assignments for the project and my first two journals.

⇒ I have begun to build the foundations of my mobile application using Android Studio.

⇒ I have still yet to submit my Ethics form as there I am still searching for datasets and resources in Irish Sign language (ISL).

Successes:

⇒ Submitting my first two assignments and first two journals.

⇒ Having my project proposal approved with amendments.

⇒ Having further contacts to reach out to for datasets.

⇒ Having a meeting with my project supervisor and organising further follow up meetings.

Challenges:

⇒ Still the lack of resources and datasets available for my project.

⇒ Still the delay in submitting Ethics forms.

Student Signature	

December Journal:

Supervision & Reflection

Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: December

What?

In the month of December:

- ⇒ **Supervisor Meetings –** I still proceed to have meetings with my supervisor, discussing progress updates, resources and any questions.
- ⇒ **Documentation –** I completed the needed amount of documentation for the Mid-Point presentation, including Use Cases, System Requirements and Testing. I submitted this via Moodle.
- ⇒ **Coding –** I continued to build my application on my development environment, using Android Studio.
- ⇒ **Errors –** Whilst trying to get the built-in simulator Android Studio provides, it caused the software to crash and wipe all of the code to date.
- ⇒ **Presentation/Demonstration –** I created a PowerPoint presentation for my Mid-Point submission, including key-points to my project, its aims and a video demonstration of my project so far. The video was also uploaded to YouTube.
- ⇒ **GitHub –** I created a GitHub account to upload my code/resources for my project.

So What?

Project Progress:

⇒ I have submitted my Mid-Point presentation, including my project proposal, documentation, code and demo for the project and my three journals for the first semester.
⇒ I have made a clearer plan to follow for my project, which is included in my project demonstration.

Successes:

⇒ Submitting my Mid-Point presentation and three journals for semester one.
⇒ Having my project designed and coded halfway.
⇒ Having a more thorough plan to follow for the rest of the project.
⇒ Having a meeting with my project supervisor and organising further follow up meetings.

Challenges:

⇒ Having to manually record video data myself as a form of resource for the project.
⇒ Rebuilding all elements of the application after software issues caused all code to be wiped.
⇒ Fixing the device emulator in Android Studio.

Student Signature	

January Journal:

Supervision & Reflection

Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: January

What?

In the month of January:

⇒ **Supervisor Meetings –** I had my first supervisor meeting since starting back semester 2.

⇒ **Documentation –** I re-evaluated my project from all aspects, including the project plan, technology and documentation.

⇒ **Project Code and Technology –** Due to issues with the original development environment I had planned my project around, I began testing another development environment to possibly build my project - **xCode**.

⇒ **Firebase –** I created a Google Firebase account and began creating a database and authenticator. I have also successfully connected the Firebase to the new development environment, xCode.

So What?

Project Progress:

⇒ Due to significant loss of code, which I mentioned in my last journal, I have made not as much progress as planned.

⇒ I have begun coding from scratch using the new development environment and have a fully functioning simulator to visualise the application on a virtual mobile.

⇒ I have begun making adjustments to my documentation as the technology and coding languages I had planned to use are now changed due to the development environment changing.

Successes:

⇒ Submitting my first journal of the second semester.

⇒ Finding a better and more reliable solution for my coding environment.

⇒ Having a fully functioning virtual mobile simulator so I can now visualise my application in production.

⇒ Creating a Google Firebase account and connecting it to the new development environment, xCode.

Challenges:

⇒ Adjusting my project plan due to set backs.

⇒ Learning a new development environment with new coding language.

⇒ Coding as much of the application to catch up to what my plan progress should be at.

⇒ Having to manually record video data myself as a form of resource for the project.

Student Signature	

February Journal:

Supervision & Reflection	
Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: February

What?

In the month of February:

- ⇒ **Project Code and Technology –** I continued learning the new development environment I have decided to build my application in: **xCode**.
- ⇒ **Progress –** I am nearly caught up with my original project plan, after starting from scratch with development. I have to finalise and clean-up the last few components that I am supposed to be scheduled on.
- ⇒ **Videos/Date –** I have pushed back my recordings of my videos for the application, that were originally supposed to take place Mid-February so that I could focus on project progress and development.

So What?

Project Progress:

- I am roughly a week behind on my original project plan, due to starting from scratch on my application and learning a whole new language/development environment.
- I am nearly finished editing my Mid-Point documentation to adjust to the changes, such as my environment and coding languages.
- I have done more research on sign language videos so I can familiarise myself with the language when it comes to recording my data, making the process easier.

Successes:

- Submitting my first two journal of the second semester.
- Familiarising myself with my new environment development and learning a brand-new coding language.
- Catching up with my project plan schedule after setbacks at Christmas.

Challenges:

- Pushing back my video recordings until Reading Week due to setbacks.
- Factoring in other projects/CA that are due for this semester, that they don’t clash with project plans.

Student Signature	

March Journal:

Supervision & Reflection	
Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: March

What?

In the month of February:

- ⇒ **Project Code and Technology –** I am still coding and developing my mobile application, nearing towards the final stages of development of the project.
- ⇒ **Firebase –** I have finished implementing and integrating the Firebase cloud database into the application architecture and I have tested with small videos to see if the bridge between the two works.
- ⇒ **Documentation –** I have progressed in my project's documentation, while still correcting previous mistakes and adjustments from my previous submission.
- ⇒ **Journals –** I submitted my February journal.

So What?

Project Progress:

⇒ I am currently developing the last ‘major’ part of the application, the favourites section of the application.

⇒ I have completed integrating the Firebase database into my application’s architecture.

⇒ I am nearly finished editing my Mid-Point documentation to adjust to the changes, such as my environment and coding languages and progressed into the next sections of the documentation.

⇒ I have done more research on sign language videos so I can familiarise myself with the language when it comes to recording my data, making the process easier.

Successes:

⇒ Submitting three journals of the second semester.

⇒ Nearing the end of the development stage of the project.

⇒ Having two out of the three major components of the application completed and tested.

⇒ Continuing to familiarise myself with my new environment development and learning a brand-new coding language.

⇒ Progressing into the next sections of my project’s documentation.

Challenges:

⇒ Developing the favourites section of my application due to the lack of resources and videos to learn from.

⇒ Introducing more videos into my Firebase application and seeing how it impacts the application performance.

Student Signature	

April Journal:

Supervision & Reflection	
Student Name	Kayleigh Harris
Student Number	X19471636
Course	BSHCDBT
Supervisor	Frances Sheridan

Month: April

What?

In the month of February:

- ⇒ **Project Code and Technology –** I am finished all the large elements of development and coding any small, additional elements/designs into the application.
- ⇒ **Testing –** I am in the final stages of testing. I have tested all the individual elements on their own, and I am now testing the application as a whole.
- ⇒ **Documentation –** I continue to write my documentation for my technical report and I have begun creating the PowerPoint for my video presentation.
- ⇒ **Project Poster –** I have begun creating my project showcase poster.

So What?

Project Progress:

- I am testing the project as a whole, and have finished testing all the individual elements of the application.
- I am finished editing my technical report, from sections 1-2 which had been submitted previously in my mid-point presentation.
- I am still writing up my technical report, from sections 3-5 which were not needed in the mid-point presentation.
- I am creating my PowerPoint slides for my presentation.
- I am creating my Project Showcase poster for submission.

Successes:

- Submitting all my journals up-to-date.
- Nearly finishing the testing phase of my project.
- Continuing to familiarise myself with my new environment development and learning a brand-new coding language.
- Perfecting my documentation from my previous mid-point submission.
- Preparing for the Project Showcase.

Challenges:

- Testing the project as a whole, with all the elements working together without causing any issues.
- Testing in a new environment, using new testing tools that are unfamiliar to me.

Student Signature