



Waterford Institute *of* Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

## Final Year Project Report **Final Report**

---

### **Ringforts of Ireland**

David Corrigan (20091410)

---

**Higher Diploma in Computer Science**

**Supervisor:** Peter Windle

Department of Computing & Mathematics  
Waterford Institute of Technology

## Declaration

I declare that the work which follows is my own, and that any quotations from any sources (e.g. Books, journals, the internet) are clearly identified as such by the use of 'single quotation marks', for shorter excerpt and identified italics for longer quotations. All quotations and paraphrases are accompanied by (date, author) in the text and a fuller citation is the bibliography. I have not submitted the work represented in this report in any other course of study leading to an academic award.

Student..... Date.....  
.....

Work Place Mento..... Date.....

## Table of Contents

<b>1. Introduction.....</b>	<b>4</b>
1.1 <i>Background</i> .....	4
1.2 <i>Objectives</i> .....	4
1.3 <i>Methodology</i> .....	5
<b>2. Requirements .....</b>	<b>6</b>
2.1 <i>Functional Requirements</i> .....	6
2.2 <i>Non-Functional Requirements</i> .....	7
<b>3. Analysis of technologies.....</b>	<b>8</b>
3.1 <i>Frontend Technology</i> .....	8
3.2 <i>Backend Technology</i> .....	12
3.3 <i>Mapping Technology</i> .....	14
<b>4. Design/Analysis.....</b>	<b>16</b>
4.1 <i>Screen Design</i> .....	16
4.2 <i>Data Modelling</i> .....	17
4.3 <i>Process Flow</i> .....	21
4.4 <i>Source Code</i> .....	24
<b>5. Development Phases .....</b>	<b>28</b>
5.1 <i>Sprint 1</i> .....	28
5.2 <i>Sprint 2</i> .....	28
5.3 <i>Sprint 3</i> .....	29
5.4 <i>Sprint 4</i> .....	30
5.5 <i>Sprint 5</i> .....	31
5.6 <i>Sprint 6/7/8/9</i> .....	33
<b>6. Critical Self Analysis.....</b>	<b>34</b>
6.1 <i>What I learned</i> .....	34
6.2 <i>What I achieved</i> .....	34
6.3 <i>Problems I encountered</i> .....	35
<b>Appendix.....</b>	<b>36</b>
Appendix A – Manually Drawn Screens .....	36
Appendix B – Screen Design using Mock-up Tool.....	38
Appendix C – User Stories .....	41
Appendix D – Original Sprint Plan.....	45
Appendix E – Updated Sprint Plan as at 01.03.2022 .....	47
Appendix F – Updated Sprint Plan as at 14.03.2022 .....	49
Appendix G – Firebase Pricing.....	51
Appendix H – Trello Sprint Tracking.....	52
<b>Bibliography.....</b>	<b>58</b>

## 1. Introduction

### 1.1 Background

According to (Hendicott, 2017), there is estimated to be 60,000 ringforts, sometimes referred to as fairy forts or Ráths, dotted around the Island of Ireland. These were mostly built during the early Christian period c. 500 – 1100AD and were erected as protected enclosures around farmsteads (Manning, 2004). Most of these are in rural Ireland, predominantly on farms where the farmer is the custodian. It is though that you are never more than a few kilometres from a Ringfort, yet most people are unaware of their presence.

My project is aiming to build a mobile application which would make the location of these sites more accessible to the general public, but also importantly allow for previously unmapped ringforts to be recorded. The app will allow these new ringforts to be recorded in three ways, firstly by people who discover them locally and can visit and record them on site. Secondly by people who want to search google satellite maps and discover them visually and record them this way. And finally, I have provided access to known locations from the National Monument Service database, which can be updated and saved to the apps live database of Ringforts.

### 1.2 Objectives

The objectives of this project are:

- To build a mobile application which is very easy to use which provides users with important information on the Ringforts around Ireland.
- To allow crowd sourcing of new Ringfort locations by users who know of them personally and by users who like to discover new ones by searching satellite map imagery.
- Investigate the feasibility of using existing records from the National Monument Service database archives (Archaeology.ie, 2019) to enable known locations of existing site to be used as a discovery tool.
- To investigate the current tools and frameworks which allow for cross platform mobile app development using a single code base.
- To become Perficient in a new technology and/or programming language not previously taught during the course.

## 1.3 Methodology

The plan is to use a hybrid of agile and prototyping during the project. The following practices will be employed:

- Writing User Stories based on conversations with the user.
- Prototyping using both manually drawn screen layouts and screen designs using the mock-up online tool.
- Breaking the project timeline down into the following 9 sprints over a period of 15 weeks.
  - Part 1 – This is three sprints of 2 weeks each. (**04.01.2022 – 13.02.2022**)
    - Sprint 1 – Preparatory Work
    - Sprint 2 – Analysis
    - Sprint 3 – User Stories, Development, and Interim Report
  - Part 2 – This again is three sprints of 2 weeks each. (**14.02.2022 – 27.03.2022**)
    - Sprint 4 – Development/Design
    - Sprint 5 – Development
    - Sprint 6 – Development
  - Part 3 – This is three sprints of 1 week each. (**28.03.2022 – 17.04.2022**)
    - Sprint 7 – Completing Build and Final Report
    - Sprint 8 – Prepare Demo Video
    - Sprint 9 – Prepare Presentation
- Incrementally building small testable features in each sprint and demo with the user with the aim of getting feedback which can be taken on board in the subsequent sprints. Each sprint would take the next highest priority stories into the next sprint along with feedback from the user from the previous sprint.
- See [Appendix D](#) for details of original Sprint Plan

## 2. Requirements

As part of requirements gathering for the project, I asked for the help of a local historian who has a keen interest in Ringforts, and who from this point forward will be referred to as the 'user'. After discussing my initial ideas about the mobile application with the user and getting their thoughts and feedback, I put together some manually drawn screen layouts for the App. These were then used during further discussions with the user where they were tweaked as we discussed them. See [Appendix A](#) for final version of the manually drawn screen prototypes.

I then used an online tool from [www.mockplus.com](http://www.mockplus.com) to build a prototype of the mobile app screens. See [Appendix B](#) for these. The output from my discussions with the user were used as a basis to write a number of user stories which split the apps functionality into discrete testable features. With the help of the user, I was able to prioritise those in order of importance. See [Appendix C](#) for User Stories

### 2.1 Functional Requirements

The following are the list of user stories which have been specified to date with the build estimated in hours They are in priority sequence, See [Appendix C](#) for User Stories details

Title	Estimate
(1) Add a New Ringfort	7
(2) Add New Ringfort Location	5
(3) Add New Ringfort Images	4
(4) List all Ringforts	5
(5) Edit Ringfort Details	5
(6) Delete Ringfort Details	3
(7) Make Data Available to All Users	8
(8) Allow User to Signup	7
(9) Allow User to Login	4
(10) Allow User to Logout	2
(11) Limit Access until User Logged-in	5
(12) Add Multiple Images per Ringfort	8
(13) Searching the Ringfort List	8
(14) Searching the Ringfort Map	5
(15) Add a Ringfort as a Favourite	7
(16) Make Photos Available to all	5
(17) Set up Approval System	10
(18) Load and Display National Monument Data	12
Total	110

## 2.2 Non-Functional Requirements

These are the non-functional requirement which I deemed as important to the project and the delivery of a successful mobile application experience:

- Performance - The application must be responsive such that the user is waiting for a minimal time to wait.
  - As part of this, an initial splash screen should be considered to distract the user while the application initially loads.
  - Responsiveness during application usage is key to its success, and caching of data should be considered to improve this performance, especially network images.
- Scalability – The ability of the application to grow both its user base and the number of ringforts stored is a key part of the success criteria. Some important factors are:
  - Choosing the correct backend which can scale while keeping costs to a minimum.
  - Choosing a mapping solution which can handle growth in the user base while keeping costs at an acceptable level.
- Availability – Making sure that the application is available when the users expect is important, but also it must be available on the device of their choice. Some of the areas to consider are:
  - Making the application available on the 2 main mobile platforms, android and IOS, is crucial to being able to see the application as a success.
  - Choosing a backend which has a high availability score was important.
- Security – Security is very important to users so keeping their login details secure is a must have and was one of the reasons that Firebase Authentication was chosen.

## 3. Analysis of technologies

This section will look at various technologies which could be used to build the cross platform mobile applications and I try to compare them objectively to determine which would be best suited to this project. I will breakdown the analysis into sections as follows:

1. Frontend technology
2. Backend/Database
3. Mapping Options

### 3.1 Frontend Technology

The frontend technology choice is a hugely important decision and will have a big impact on the direction of the project. Because one of the requirements is to make the mobile experience available on the main two mobile platforms, IOS and Android, it became clear from the beginning that building native apps was not a viable option, mainly because of the duplicity of coding required and the learning curve of IOS development. So, I decided to investigate solutions which could be used to deploy native standard applications to both platforms from a single code base.

I will try to compare the main contenders in this area, Flutter, React Native and Ionic.

#### 3.1.1 Flutter

Flutter, released by Google in 2018 is a comprehensive app SDK complete with widgets and tools which enables cross-platform development. It uses its own rendering engine and so controls every pixel on the screen. *To the underlying operating systems, Flutter applications are packaged in the same way as any other native app (docs.flutter.dev, 2021)*. It by default uses the material design screen style favoured by Google in Android apps. But it also can be coded to use Cupertino design which will deliver an iOS style app or both.

The advantages:

- Faster development and Delivery – for example tools like “hot reload”.
- Quality documentation
- Feature Rich user interfaces
- Backward compatibility
- Comprehensive library of plugins
- Very good performance, closest to Native performance

The disadvantages:

- Framework age
- Project size – project files occupy more space than other tools.

### 3.1.2 React Native

React Native is an open-source app framework launched in 2015 by Facebook and can also be used to build cross platform apps using a single codebase. It is based on React so has basic concepts like JSX, components, state and props. But it uses native component rather than web components. These native components compile directly to Android views and IOS UIViews. So, React Native app will look different depending on the platform. *React Native communicate with the native APIs via a JavaScript bridge. The JavaScript bridge doesn't perform well for all development requirements (Suranga, 2021).*

The advantages:

- Platform specific Native UI compiled components
- Web and ReactXP libraries for possible web version of the app
- Almost on Par performance with Native apps
- Large community of users to assist with issues encountered

The advantages:

- Poor documentation
- Requires a lot of external libraries

### 3.1.3 Ionic

Ionic was first introduced in 2013 and is an open-source SDK for building cross-platform mobile apps. Ionic uses traditional web languages like HTML, CSS and JavaScript to build mobile apps which is wrapped by the Cordova framework which will then render the app in a native WebView. Examples of Apps are MarketWatch, Pacifica and Sworkit.

Source: (Strapi, 2021)

The advantages:

- Great for web developers who already know HTML, CSS and JavaScript.
- Any frontend framework can be used to create the app starting from Ionic 4.

The advantages:

- Using WebView for large applications can result in a significant reduction in performance.
- Can lack some native plugin support.

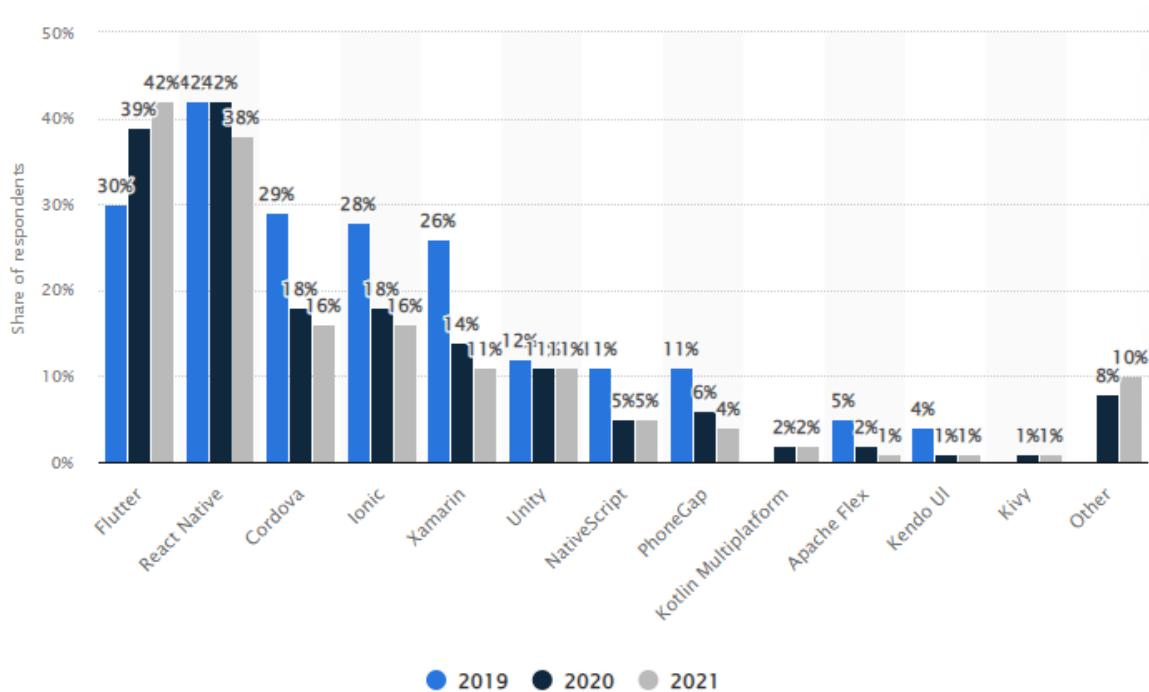
### 3.1.4 Head-to-Head Comparison

The diagram below compares the technologies across several categories. (Rykowski, 2021)

<b>Criteria</b>	<b>Flutter</b>	<b>React Native</b>	<b>Ionic</b>
<b>Created by</b>	Google	Facebook	Ionic
<b>Official Release</b>	December 2018	March 2015	November 2013
<b>Programming Language</b>	Dart	JavaScript/TypeScript	HTML, CSS, JavaScript
<b>Compilation</b>	Compiled Native App	Partly Compiled Native App. JavaScript not compiled but run using a bridge	Not compiled to Native, but Web App wrapped in Native app.
<b>UI Display</b>	Flutter renders the screen itself using the frameworks graphics engine. So by default the Android and iOS apps will be consistent across platforms. But can be customised where required.	UI Elements compile to Native Widgets so you get Native looking applications as default.	Styled as in any Web app, not compiled to Android or iOS.
<b>Performance</b>	Flutter has near Native performance. It is compiled Ahead-of-time in native machine language.	UI components get compiled to Native components, but the JavaScript isn't and runs in its own JavaScript thread. So, any interaction between JavaScript and native components is through a bridge which can impact performance.	Performs the worse of the 3 in almost all scenarios.
<b>Popularity</b>	135k stars GitHub Jan22	101k stars GitHub Jan22	46k stars GitHub Jan22
<b>Example App</b>	Google Ads, Alibaba, ULike, Hamilton	Instagram, Uber, Messenger	Accenture, NHS, UNTAPPD
<b>Documentation</b>	Comprehensive and easy to understand.	Documentation not easy to use.	A lot of documentation available

### 3.1.5 Cross-Platform Mobile Frameworks used by Software developers worldwide 2019-2021

Here we can see the steady increase in the usage of Flutter in the last 3 years and the stagnation and then decline in the market share of React Native and Ionic. With Flutter now the most popular Cross-platform mobile framework among Software Developers.  
 Source: (Statista, 2022)



### 3.1.6 Conclusion

I have chosen Flutter as my Front-End Development Framework for several reasons:

- Flutter gives a richer more consistent cross-platform UI experience, which is what I want for this app.
- The fast development environment is certainly a huge positive given the timescales of the development.
- Flutter Applications give the closest to Native performance of any of the options compared.
- There is great documentation and resources available. And community hubs like [Flutter Community](#) and [Flutter Awesome](#) are great for finding solutions.
- I wanted the opportunity to learn a new framework and programming language, one which from the statistics is becoming more and more popular. This allows me to demonstrate to myself that one of the most important skills learned on the Higher Diploma course is the ability and confidence to research and learn new technologies.

## 3.2 Backend Technology

The decision about which backend technology to use is another major project decision. For me the decision came down to two technologies which I covered during the course, the first being a custom REST API build on Node.js and the second option being Firebase backend services.

### Custom Node API

The first option, and the one I had initially planned to employ, was building a custom REST API on Node.js using the hapi framework. It would use a MongoDB NoSQL database with the help of the Mongoose library and use a service like Cloudinary to handle image storage.

The advantages of this approach (based on (kella, 2021)):

- The separation of the data service from the client allows for portability of the data service and re-use by other applications.
- The project has total control over reliability and scalability of the backend service.
- The data is cacheable which could be used to ease the burden on the server.

The disadvantages (based on (kella, 2021)):

- This requires a complete backend to be built from the outset including authentication of the endpoints. This is a large cost and time overhead for projects which wish to get off the ground quickly.
- Because reliability and scalability are project responsibilities, they will also become time and cost drains on the project.

### Firebase BAAS

The second option was to use the Googles Firebase suite of backend services, storing the data on Cloud Firestore and images on Cloud Storage. This is a paid service but gives a free layer which should be enough for most apps initially. See Firebase Pricing in [Appendix G](#) (Firebase, 2019)

The advantages (Clark, 2021):

- Reliable and Extensive databases, including the Realtime Database and Cloud Firestone
- Free layer which might suffice for a lot of projects. See [Appendix G](#).
- Google analytic works seamlessly with Firebase unsurprisingly.
- Free Multi-Platform Authentication
- Very Scalable solution
- Great plugin and documentation available for Flutter to use the Firebase APIs
- Very quick to integrate into a new app.
- High Availability > 99.999 per month. (Google, 2020)
- Very secure.

The disadvantages (Moqod, 2021):

- No fixed price, rather a Pay as You Go approach which can be unpredictable
- Limited querying ability in the Realtime database which are resolved by Firestore DB.
- Doesn't provide tools for migration off Firebase to other platforms

### ***Conclusion***

I have decided to proceed with the project use the Firebase BAAS both for my data needs and for the authentication for my application.

This was based on a number of factors including:

- The integration with the Flutter SDK using the various plugins available and the documentation available both for Firebase and the Flutter plugins are excellent.
- The pricing plans make it very likely the application could exist without any cost using the Blaze plan (See [Appendix G](#)).
- The speed of development using Firebase services was a major factor in the decision. After the initial sprint planning (see [Appendix D](#)) it became apparent quite early that building the back end as a custom API was not feasible in the timeframe.

### **3.3 Mapping Technology**

This application relies on maps as part of its core functionality, so picking the best map solution was important. Here I will evaluate some of the options available for mapping which could be used. For the comparison I have sourced the information from (Dziuba, 2021)

#### **MapBox**

MapBox offers a set of comprehensive features to integrate into Mobile apps and websites. These include data visualization, navigation, and search functionality. It is used by giants like Facebook, Snapchat and Foursquare.

The advantages:

- Very customizable and flexible
- Fast load times and performance
- Offline mode in the API
- Open Source

The disadvantages:

- Inferior map coverage in certain areas
- Learning curve with MapBox API.
- Can be unpredictable pricing

#### **OpenStreetMaps**

OpenStreetMaps is an open sourced, community-based project which supplied maps and data to many websites and apps. It's totally free but still can provide a high level of accuracy thanks to the efforts of volunteer map enthusiasts and developers who supply it with data.

The advantages:

- The API is free
- The is an open-source service.

The disadvantages:

- Limited number of queries. Excessive data exchange not welcome, and users may be blocked
- May require addition services. So may need to use MapBox which is based on OSM data.

### **Google Maps**

Google Maps is the original pioneer of online mapping technology and boasts over 99% coverage of the world. It can leverage a fleet of satellites, Street View cars and Android devices as data sources.

The advantages:

- Very Recognisable on mobile and web
- Excellent global and local data quality
- Multi Language support
- Street View

The disadvantages:

- Less Customisation
- Not open source
- Can be unpredictable pricing

### **Conclusion**

I have decided to choose Google Maps as the map solution for my App. This was based on the following:

- It provided all the functionality required in the App.
- I already had a google Map account with €200 euro credit a month which would cover my app in most scenarios. MapBox might have been slightly cheaper but both models have unpredictable pricing models.
- The Flutter plugin for Google maps is excellent and has very clear documentation.

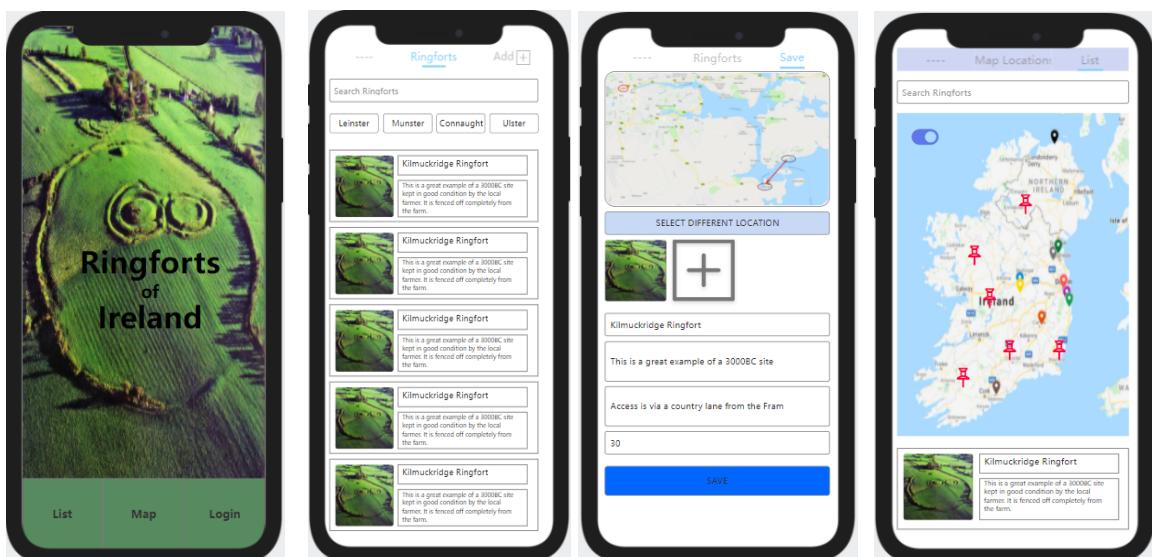
## 4. Design/Analysis

### 4.1 Screen Design

The design of the screens was the area where the users' feedback was of most benefit to me. I initially used hand drawn screens (see [Appendix A](#)) as an artifact to help draw out the users requirements and this evolved into these screen mock-ups below which were eventually agreed with the user. I found that having something concrete like the screen drawing to use during discussions helped tease out what was important for the user.

#### 4.1.1 Screen Mock-ups

These are the final mock-up agreed with the user before the 1<sup>st</sup> development sprint. These allowed the user to visualise what would be produced which lessened the probability of surprises after each sprint.



## 4.2 Data Modelling

This section will give details of the data required by the application and how it will be models in the Firebase Firestore DB. There are the 4 Firestore collections being used by the application:

1. **users** – This collection stores data about registered users of the system.
2. **historicSites** – This is a Firestore collection of the ‘live’ ringforts on the system. This is the collection where the app stores the ringforts which users add to the system.
3. **historicStagingSites** – This is a collection of staging updates which are either awaiting approval by an admin user, or a history of those which have either been approved or rejected previously.
4. **NMS-Ringforts** – This collection stores ringfort records downloaded from the National Monument Services database of national Ringfort monuments. (National Monument Services, 2022)

### 4.2.1 user Collection

This is the main **user** data collection for the application. A new user document is created when a new user registers on the App and this document defines the user on the system. The favourites array will be used store the **uid** of the users favourited ringfort sites. The adminUser Boolean field will indicate if the user has administrator rights on the data, with the ability to make instant updates to data and to approve/reject updates from other users.

<i>user</i>	
uid	String
email	String
adminUser	Boolean
favourites	[ ]

#### Sample Data from Firestore collection ‘users’

```

adminUser: false
email: "joshua@gmail.com"
▼ favourites
  0 "8kNtI8j7D42BVYEVQ4Uk"
  1 "pKX8IKddwQpfLifGqsza"
  2 "KtO2Js6NNCkMsMrUYVqq"
  3 "4ge8KCUFZhcvSSYBYxw9"
  4 "zI0AjuVISPCwocC2DYJq"
uid: "8rheHdpU7YWb7EkGivlRgeslY622"

```

#### 4.2.2 *historicSites Collection*

This is the main ‘live’ data collection for the application. It defines a ringfort in the application. When a user adds a new ringfort in the app it gets added this collection.

historicSite	
uid	String
siteName	String
siteDesc	String
siteAddress	String
siteCounty	String
siteProvince	String
siteAccess	String
siteSize	double
latitude	double
longitude	double
image	String
lastUpdatedBy	String
createdBy	String

#### *Sample Data from Firestore collection ‘historicSites’*

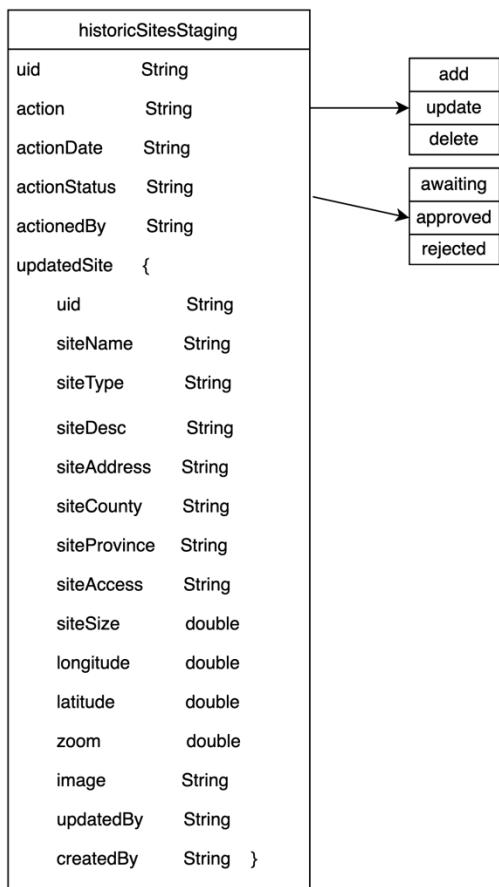
```

address: "V6Q8+Q9 Incharue, County Kerry, Ireland"
county: "County Kerry"
createdBy: "uNH5vYQsqidNUaX23iglYjSQYrF3"
image: "https://firebasestorage.googleapis.com/v0/b/ringforts-ireland.appspot.com/o/images%2Fimage-0ML3vJHiFTy7GcjZrKAv.jpg?alt=media&token=82175dd6-42d4-4147-a977-49df9bf3c2fd"
lastUpdatedBy: "chHXJDpEXrTd4eLKq5387vaTWwT2"
latitude: 51.88939322459631
longitude: -9.784055426716805
province: "Munster"
siteAccess: "Yes"
siteDesc: "This is a great extract of a Ringfort in Kerry"
siteName: "Kerry Ringfort"
siteSize: 32
uid: "0ML3vJHiFTy7GcjZrKAv"

```

#### 4.2.3 *historicSiteStaging* Collection

This is staging collection for changes made by users to ringfort records. When a non-admin user makes updates, i.e., adding a new ringfort, deleting a ringfort or updating a ringfort, the changes are stored in this collection with a status of ‘awaiting’. An admin user can then approve or reject this change at which point the changes get added to the ‘live’ collection *historicSites* if approved.



#### Sample Data from Firestore collection ‘historicSiteStaging’

```

action: "add"
actionDate: 26 February 2022 at 16:07:03 UTC
actionStatus: "approved"
actionedBy: "8rheHdpU7YWb7EkGivlRgeslY622"
uid: "9EuPaiSuAHNVW4TRqqvy"

updatedSite:
    address: "FPRH+CP Killincooley Beg, County Wexford, Ireland"
    county: "County Wexford"
    createdBy: "8rheHdpU7YWb7EkGivlRgeslY622"
    image: "https://firebasestorage.googleapis.com/v0/b/ringforts-ireland.appspot.com/o/images%2Fimage-9EuPaiSuAHNVW4TRqqvy.jpg?alt=media&token=c8ce7760-ad10-4e90-b27c-53a658dbf05"
    lastUpdatedBy: "8rheHdpU7YWb7EkGivlRgeslY622"
    latitude: 52.4910083
    longitude: -6.2707267
    province: "Leinster"
    siteAccess: "No"
    siteDesc: "Test Description is added here"

    siteName: "Test"
    siteSize: 32
    uid: "9EuPaiSuAHNVW4TRqqvy"

```

#### 4.2.4 NMS-Ringforts Collection

This is data collection uploaded from an extract from the National Monument Service database of historical sites. The data has been filtered before being uploaded to include only ringforts sites. Approximately 24000 records have been added using this extract from NMS using the tools at this site: <https://data.gov.ie/dataset/national-monuments-service-archaeological-survey-of-ireland> . (National Monument Services, 2022)

NMS-Ringforts	
siteName	String
siteDesc	String
latitude	double
longitude	double

#### Sample Data from Firestore collection 'NMS-Ringforts'

```
latitude: "52.392667"  
longitude: "-6.586131"  
siteDesc: "CORLICAN"  
siteName: "Ringfort - rath"
```

## 4.3 Process Flow

This section outlines the main process and data flows for the application. It outlines the different paths data will take depending on:

1. If the source is the National Monument Service Data (NMS) or not.
2. If the user is a normal user who required any updates to be approved before becoming live.
3. If the user is an Admin user whose updates are immediately reflected in the live data.

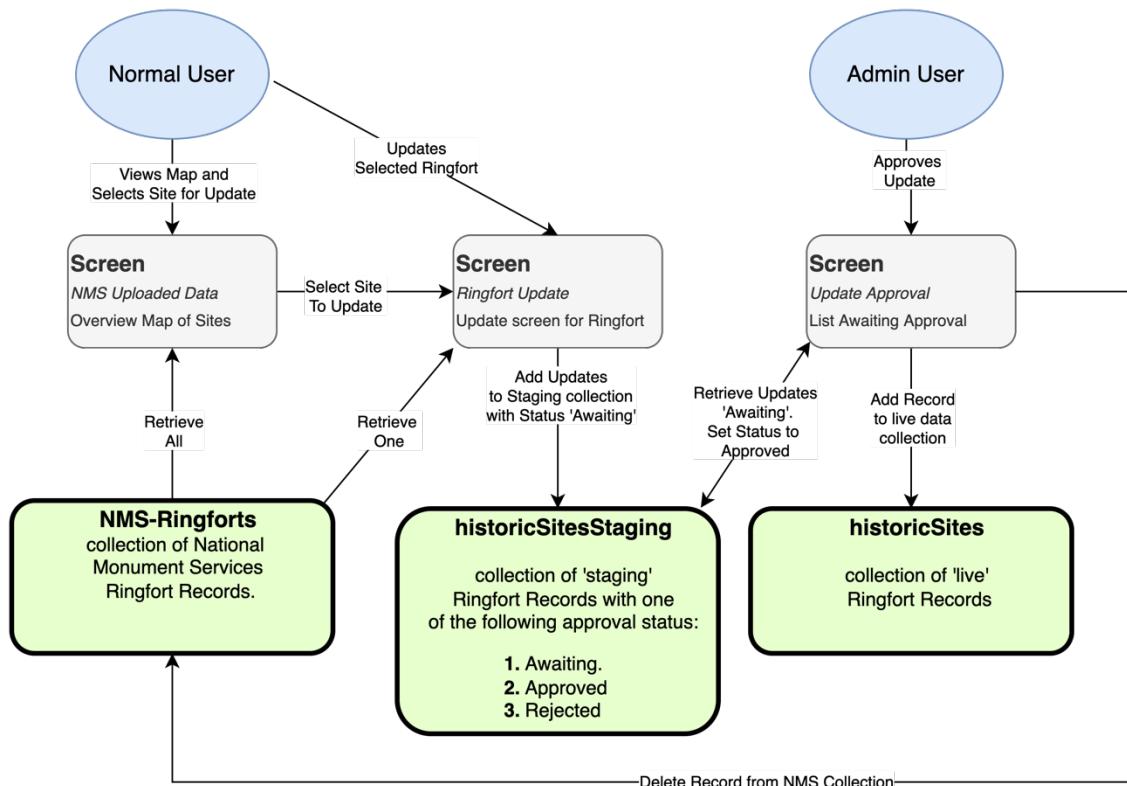
### 4.3.1 Scenario: Normal User Adds Ringfort via NMS Map Screen

This scenario shows how data flows when a *normal* user adds a new Ringfort via the NMS data map overview screen. This screen shows ringfort locations on a map using data from the 'NMS-Ringforts' collection, this collection was populated from a download of data from the National Monument Services website.

On the map a user can then select the ringfort they want to add, which triggers the update screen to launch with the selected ringfort's location and some details pre-populated. The user then completes the updates and saves. This triggers the data to be sent to a staging '*historicSitesStaging*' collection for approval by an admin user.

The *admin* user can use the approval screen to view all changes awaiting approval and to either approve or reject. Once approved the ringfort is added to the live '*historicSites*' data collection and deleted from the NMS data collection. The staging table gets its status updated from awaiting to either approved or rejected. If rejected, then the data is not added to the 'live' data collection.

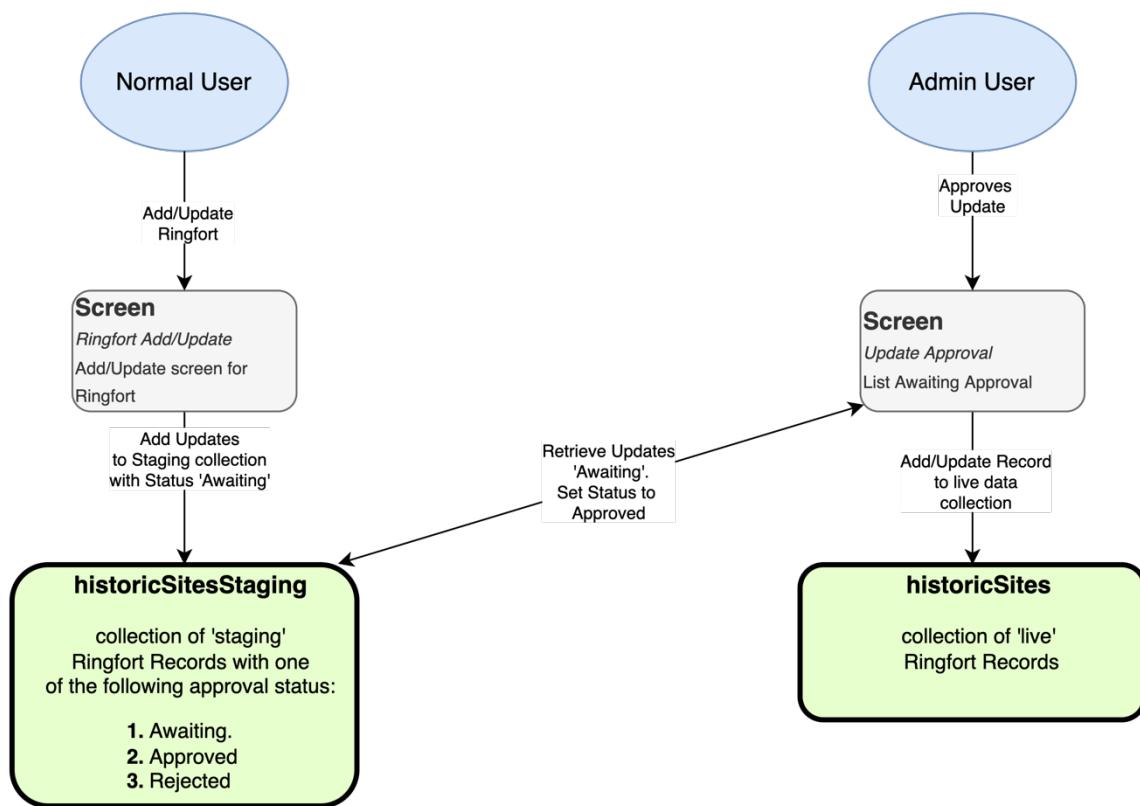
Using this process users can systematically move data from the NMS collection to the apps 'live' data collection while adding useful and consistent data.



#### 4.3.2 Normal User Adds/Updates Ringfort via Add/Update Screen

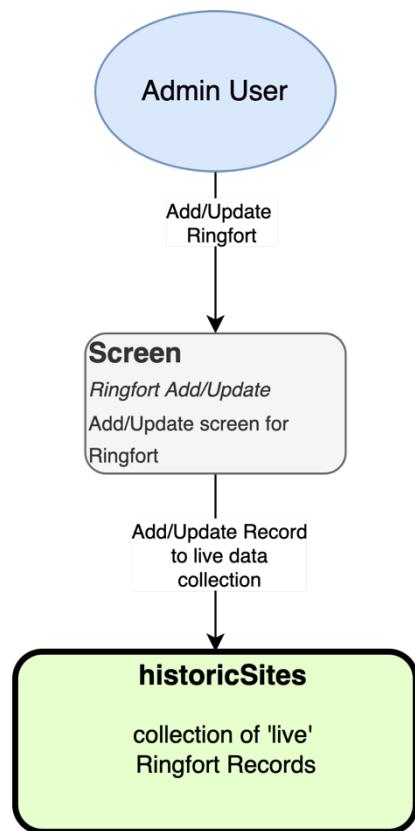
This scenario shows how data flows when a *normal* user adds or updates a Ringfort via Add or Update screen. This updated or new Ringfort data will get added to the staging 'historicSitesStaging' collection with a status of 'awaiting'.

The *admin* user can use the approval screen to view all changes awaiting approval and to either approve or reject. Once approved the new or updated data is add or updated on the live 'historicSites' data collection. The staging table gets its status updated from awaiting to either approved or rejected. If rejected, then the data is not added to the 'live' data collection.



#### 4.3.3 Admin User Adds/Updates Ringfort via Add/Update Screen

This scenario shows how data flows when a *admin* user adds or updates a Ringfort via the Add or Update screen. This updated or new Ringfort data will get added directly to the live '*historicSites*' data collection. There is no approval or staging process for admin users.



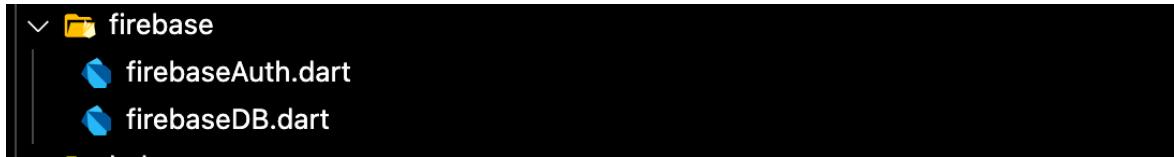
## 4.4 Source Code

### 4.4.1 Code Structure

I have broken the code into a number of libraries to help maintainability and readability. I have kept as much logic out of the screen classes as possible and added to the state management provider classes.

#### Firebase Access Classes

The firebaseDB class provide the state management classes with access to the Firestore functions to add, update, delete and query the collections. The firebaseAuth class handles registration and authentication of users.



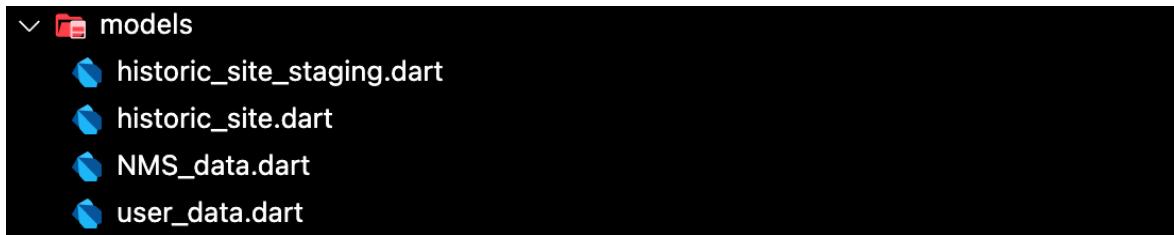
#### General Helper Classes

The **location\_helper** class handles location tasks like getting the static satellite image for a location. It also handles getting the current address for a chosen position including the county and province. While the **map\_helper** class calculate the bounds needed by google maps to focus the maps to include all the markers.



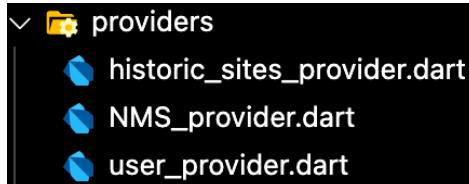
#### Data Model Classes

These classes define the data models for the Firestore collections and define methods to convert data from dart classes to JSON maps and vice versa to help manage data to and from Firestore.



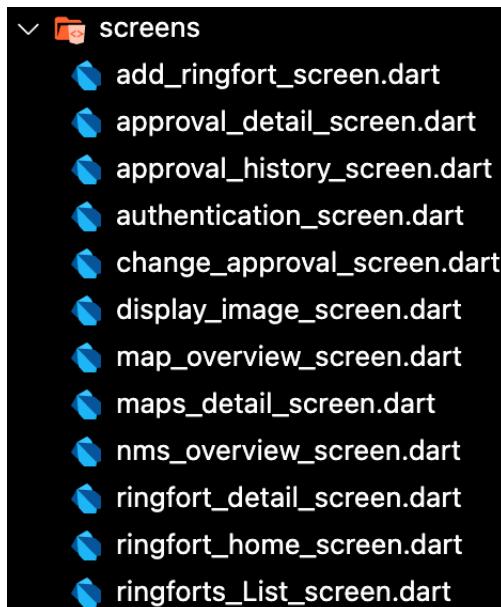
### **State Management Provider Classes**

These classes use the flutter **provider** package to implement state management. These classes are implemented using ‘**with ChangeNotifier**’ syntax which will ‘mixin’ capabilities from the ChangeNotifier class without inheriting from it. Other classes can then listen to updates to fields in these provider classes. E.g., a list of ringfort sites.



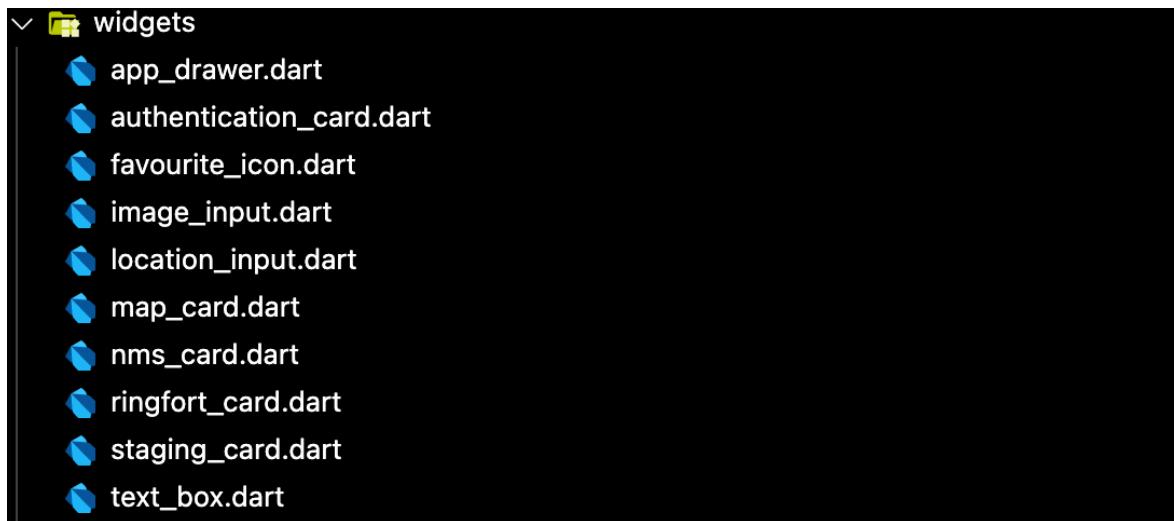
### **App Screen Classes**

These classes define the screens for the application. Each of these classes has a build method which returns a Widget. The Widget returned is usually a hierarchical tree of Widgets. A simple example might be that a **Text** widget is used to display text on the screen, and this might be wrapped in a **Padding** Widget for to add padding, which might be wrapped in a **Container** Widget to define width and height and then wrapped in a **Scaffold** Widget which defines a screen.



## App Widget Classes

To make the code more readable and maintainable, and to enable code re-use, it sometimes makes sense to define a part of the screen in its own class. For instance, below I have defined what I want a favourite icon to look like in the **favourite\_icon** class below. I have defined input parameters and defined it as stateful so that it can react to user interaction on the screen. I can then use this like any flutter defined widget in my screen classes or in other user defined widgets classes. I use this **favourite\_icon** widget in my **ringfort\_card** widget class, so that the icon appears in each ringfort card in the list and responds to being pressed by turning red or clear, and calling a function to update the user's favourites.



## Main Class

The main dart class is where the app is controlled from, and where the set-up is defined. I use using a **MultiProvider** package from Flutter to allow me to use multiple state management classes. I have a state management classes for the following

1. StreamProvider<User> for the firebase authentication state of a user.
2. ChangeNotifierProvider<HistoricSitesProvider> for list of Ringforts
3. ChangeNotifierProvider<NMSProvider> for list of NMS Ringforts
4. ChangeNotifierProvider<UserProvider> for details on current logged on user.

The **home** is defined as the **RingfortHomeScreen** which is the entry point to the app.

And the app **routes** has all the possible screen routes defined.

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MultiProvider(
      providers: [
        // The StreamProvider will keep the user authentication status
        // up to date across all widgets that consume it.
        StreamProvider<User>(
          create: (context) => FirebaseAuth.getUserStatus(),
          initialData: null,
        ),
        ChangeNotifierProvider<HistoricSitesProvider>.value(
          value: HistoricSitesProvider(),
        )
      ],
    );
  }
}
```

```

),
ChangeNotifierProvider<NMSProvider>.value(
  value: NMSProvider(),
),
ChangeNotifierProvider<UserProvider>(
  create: (_) => UserProvider(),
),
],
child: MaterialApp(
  title: 'Ringforts of Ireland',
  theme: ThemeData(
    primarySwatch: Colors.green,
    primaryColor: Colors.grey,
  ),
  home: RingfortHomeScreen(),
  // Routing table for the app screens
  routes: {
    //Route - Ringfort Home screen
    RingfortHomeScreen.routeName: (context) => RingfortHomeScreen(),
    //Route - add a ringfort screen
    AddRingfortScreen.routeName: (context) => AddRingfortScreen(),
    //Route - Ringfort List screen
    RingfortsListScreen.routeName: (context) => RingfortsListScreen(),
    //Route - Ringfort Detail screen
    RingfortDetailScreen.routeName: (context) => RingfortDetailScreen(),
    //Route - AuthScreen
    AuthenticationScreen.routeName: (context) => AuthenticationScreen(),
    //Route - Map Overview screen
    MapOverviewScreen.routeName: (context) => MapOverviewScreen(),
    //Route - Update Approval screen
    ChangeApprovalScreen.routeName: (context) => ChangeApprovalScreen(),
    //Route - Approval Detail screen
    ApprovalDetailScreen.routeName: (context) => ApprovalDetailScreen(),
    //Route - User Approval History screen
    ApprovalHistoryScreen.routeName: (context) => ApprovalHistoryScreen(),
    //Route - Display Image screen
    DisplayImageScreen.routeName: (context) => DisplayImageScreen(),
    //Route - NMS Map Overview screen
    NmsOverviewScreen.routeName: (context) => NmsOverviewScreen(),
  },
),
);
}
}
}

```

## 5. Development Phases

The next section describes what will be included in each sprint of the project. For the full original sprint plan see [appendix D](#). I have been using the Trello tool to manage my sprint progress and [Appendix H](#) shows the sprint progress for the first 6 sprints.

### 5.1 Sprint 1

This start-up sprint centred about determining the best Front end technology framework to use in building the app. Once Flutter was picked, the next task was to become proficient using it. This involved reading the Flutter documentation and taking online tutorials and courses.

Sprint No:	<b>1</b>
Start Date	04.01.2022
End Date	16.01.2022
Tasks	1) Compare and Select Front End Technologies (4 hours) 2) Study selected Front End Technology to become competent (25 hours) TOTAL: 29 hrs
Status	Complete

### 5.2 Sprint 2

This sprint saw the start of the requirements gathering process, where I spoke about the project with the local historian and based on that drew sample screens for discussion and update. I also managed to do some more study on the Flutter front technology.

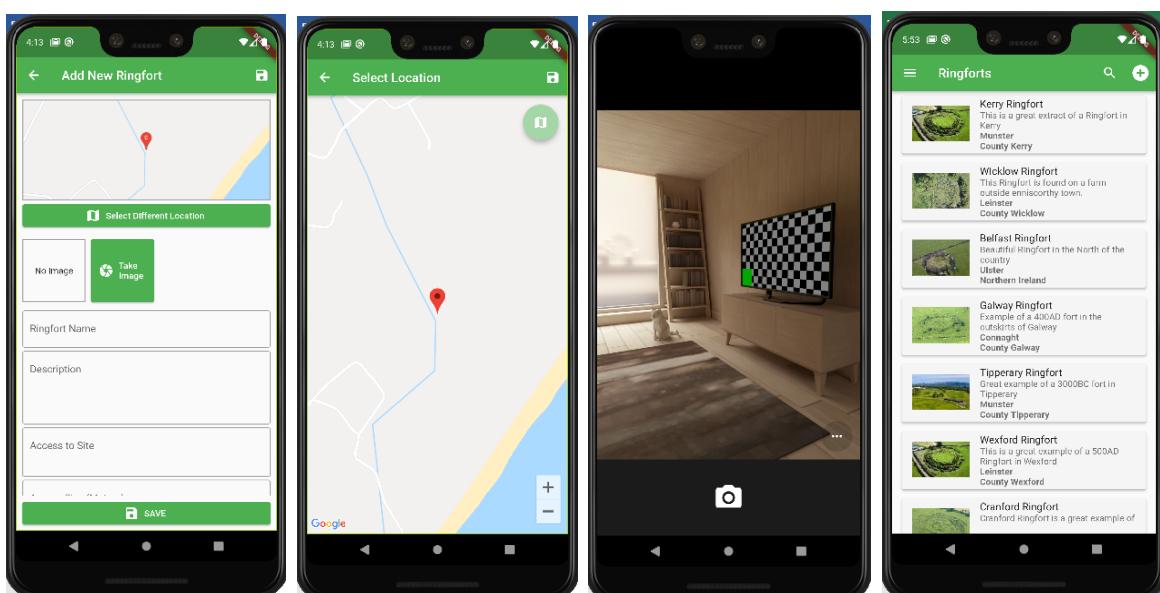
Sprint No:	<b>2</b>
Start Date	17.01.2022
End Date	30.01.2022
Tasks	1) Contact User (Local Historian) and discuss requirements (1 hour) 2) Mock up sample screens by manually drawing and confirm with user (4 hours) 3) Study selected Front End Technology to become competent (12 hours) 4) Start project report layout and structure (3 hours) 5) Document design of Data model (3 hours) TOTAL: 23
Status	Complete

## 5.3 Sprint 3

This 3<sup>rd</sup> sprint involved getting the user requirement written up into user stories and taking the manual screen drawings and mocking them up using an online tool. The first of the ‘build’ user stories were taken into this sprint which resulted in a working “Add Ringfort” screen with map location and images being captured. This interim report is also due at the end of this sprint.

Sprint No:	3
Start Date	31.01.2022
End Date	13.02.2022
Tasks	<ul style="list-style-type: none"> <li>1) Write user Stories for highest priority tasks (3 hour)</li> <li>2) Turn screen drawings into Wireframes (4 hours)</li> <li>3) Write project report to interim report level (10 hours)</li> <li>4) User Story 1 Dev (7 hours) – (Add a New Ringfort)</li> <li>5) User Story 2 Dev (5 hours) – (Add New Ringfort Location)</li> <li>6) User Story 3 Dev (4 hours) – (Add New Ringfort Images)</li> <li>7) User Story 4 Dev (5 hours) – (List all Ringforts)</li> </ul> <p>TOTAL: 38</p>
Status	Complete

Here are the results of the first development sprint which is the first iteration of the ‘Add Ringfort’ screen, which triggers a Map location selection screen and the camera app.

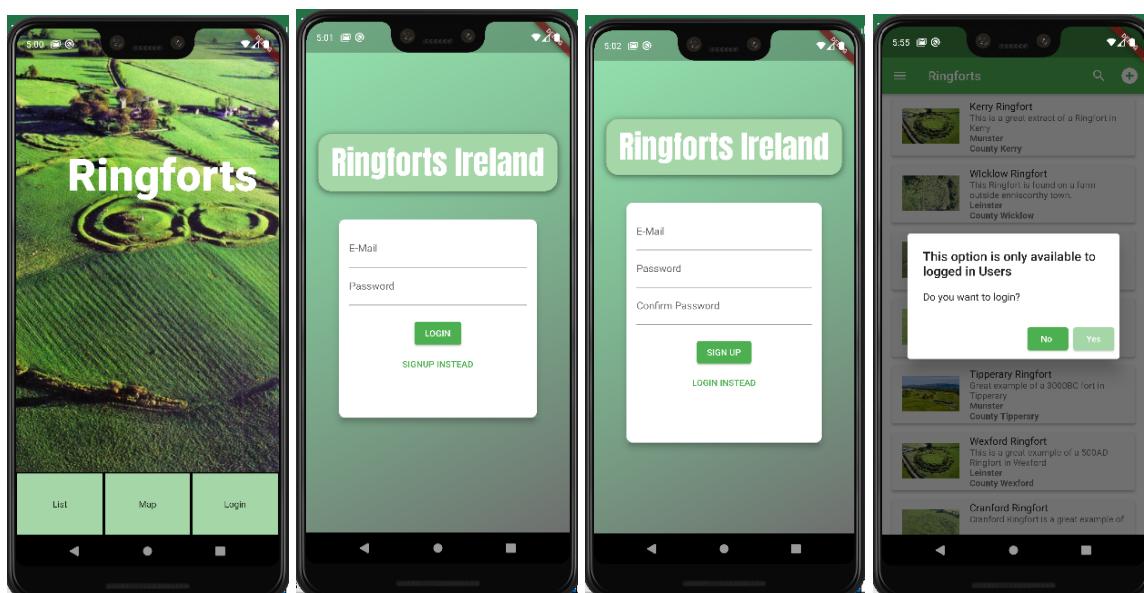


## 5.4 Sprint 4

Based on the progress in sprint 3, the next sprint was re-planned. It is predominantly a development sprint but includes the design of the login and Signup screens and the continuation of the Project Report.

Sprint No:	4
Start Date	14.02.2022
End Date	27.02.2022
Tasks	1) Continue Project Report (2 hour) 2) User Story 5 Dev (5 hours) – (Edit Ringfort Details) 3) User Story 6 Dev (4 hours) - (Delete Ringfort Details) 4) User Story 7 Dev (5 hours) - (Make Data Available to All Users) 5) User Story 8 Dev (5 hours) - (Allow User to Signup) 6) User Story 9 Dev (4 hours) - (Allow User to Login) 7) User Story 10 Dev (2 hours) - (Allow User to Logout) 8) User Story 11 Dev (5 hours) -(Limit Access until User Logged-in)  <b>TOTAL: 32</b>
Status	Complete

Here are the results from the this development sprint. This sprint focused on authenticating users with the Firebase Auth SDK for flutter.

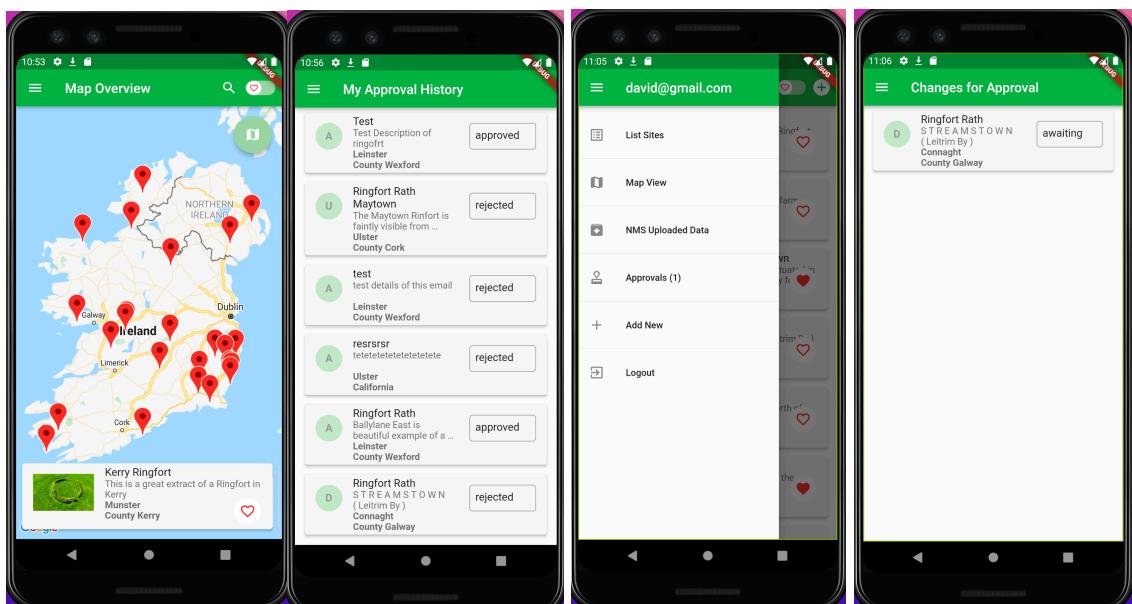
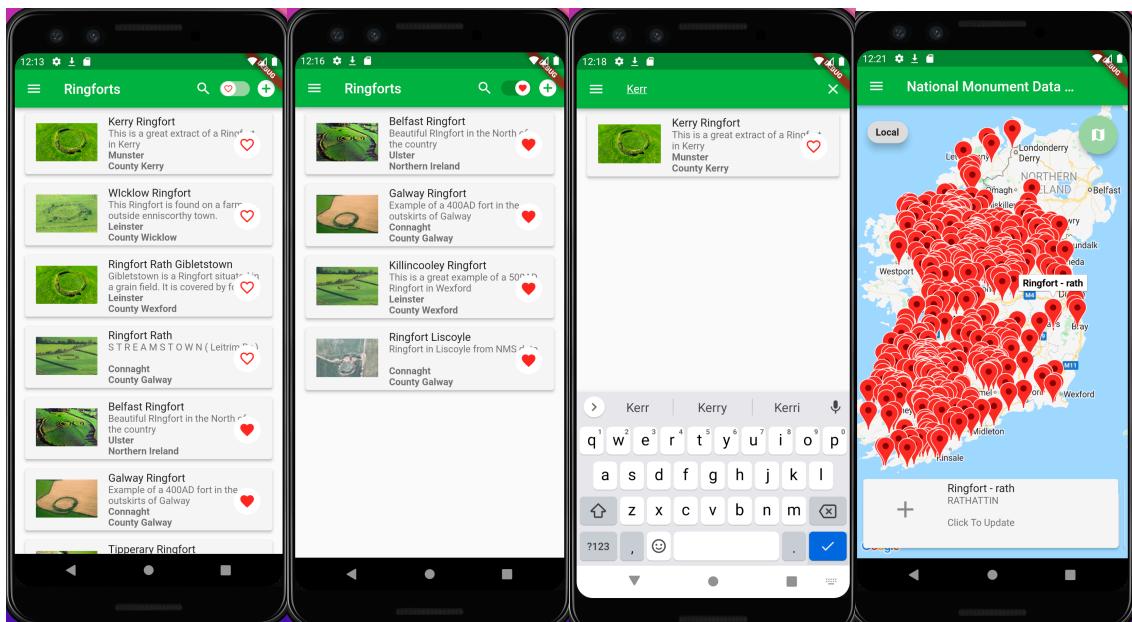


## 5.5 Sprint 5

Based on the progress in sprint 3 and 4, this sprint was also replanned. I was able to move most of the outstanding development tasks into this sprint as progress was ahead of schedule and I was able to allocate more of my time to the sprint. This sprint dealt with search and map functionality but also included the loading of the national monument service data to a new collection on Firestore. Another major part if the sprint was the approval system, whereby normal users updates need to be approved by admin users. The app was installed on my personal IOS iPhone in this sprint.

Sprint No:	5
Start Date	28.02.2022
End Date	13.03.2022
Tasks	<ul style="list-style-type: none"> <li>1. Continue Project Report (5 hour)</li> <li>2. User Story 16 Dev (5 hours) (Make Photos Available to all)</li> <li>3. User Story 13 Dev (6 hours) (Searching the Ringfort List)</li> <li>4. User Story 14 Dev (4 hours) (Searching the Ringfort Map)</li> <li>5. User Story 15 Dev (7 hours) (Setting Ringfort as User Favourite)</li> <li>6. User Story 18 Dev (12 hours) (Adding NMS data and displaying on Map and allow to update)</li> <li>7. User Story 17 Dev (10 hours) (Approval System for Normal users)</li> <li>8. Install Xcode on apple IMAC and build the project for IOS. (4hrs)</li> </ul> <p>TOTAL:52</p>
Status	Complete

A sample of the screens developed in this sprint are shown below.



## 5.6 Sprint 6/7/89

Sprint 6 is the last one with development tasks, while sprint 7, 8 and 9 focus on the deliverables of the project, like this report, the demo video, and the presentation.

Sprint No:	<b>6</b>
Start Date	14.03.2022
End Date	27.03.2022
Tasks	<ul style="list-style-type: none"> <li>1. Get Handbook Entries ready (10 hours)</li> <li>2. Complete self-reflection in report (6 hours)</li> <li>3. User Story 12 Dev (8 hours) (Display image on new screen - zoomable)</li> <li>4. Bug fixes and commenting (4 hours)</li> </ul> <p>TOTAL: 28</p>
Status	In-progress

Sprint No:	<b>7</b>
Start Date	28.03.2022
End Date	03.04.2022
Tasks	<ul style="list-style-type: none"> <li>1. NOTE: 1 Week Sprint Complete Project Report</li> <li>2. Fix any code bugs and see if any small additions possible...</li> <li>3. Complete README</li> </ul>
Status	Awaiting

Sprint No:	<b>8</b>
Start Date	04.04.2022
End Date	10.04.2022
Tasks	<p>NOTE: 1 Week Sprint</p> <ul style="list-style-type: none"> <li>1. Prepare Demo Video</li> </ul>
Status	Awaiting

Sprint No:	<b>9</b>
Start Date	11.04.2022
End Date	17.04.2022
Tasks	<p>NOTE: 1 Week Sprint</p> <ul style="list-style-type: none"> <li>1. Prepare Presentation</li> </ul>
Status	Awaiting

## 6. Critical Self Analysis

### 6.1 What I learned

#### 6.1.1 Flutter SDK and Dart programming Language

One of the challenges which I set myself for this project was to learn a new skill. This was to demonstrate to myself that what I had learned during the course gave me the ability and confidence to take on a new programming language and development environment and produced an application to a high standard. There was definitely a 4 or 5 week lag at the start of the project where I was studying the Flutter SDK and the Dart programming language and as a consequence worrying that I was making no actual progress. But I stuck to the plan and once the development sprints started I was able to hit the ground running and make progress quickly because of the time spent on that learning.

### 6.2 What I achieved

#### 6.2.1 Building Cross Platform Mobile Application

The main achievement of the project was the production of a high quality cross platform mobile application. Being able to build and install a native standard application on both an Android device and an IOS device ticks a big box on my objectives.

Because I built this using a relatively new technology which wasn't covered in the course and which hadn't the mature online community to search for answers to problems, which is available for some of the technologies which have been around longer. This also meant starting the project from the basic 'hello world' starter project, rather than from basics built in the labs which was possible in previous course projects.

#### 6.2.2 Making NMS Data available to my application.

One of the objectives of the project was to use the existing data archives from the national monument service in my App. But how to use this data in the best way was not obvious. I wanted the live ringfort data to be consistent and accurate as possible, and the existing NMS extract did not provide this consistency.

The approach I took was to use the NMS data as a tool to find Ringforts around the country which then could be converted to 'live' data when the user performed the updates and saved. Because of the volume of NMS ringfort data, approx. 24000 records, I limit the number of ringforts which appear on the NMS map screen to 500. This is to protect the cost structure of the project which relies on 50000 reads from Firebase per day in the free price tier. See [Appendix G](#) for the Firebase pricing structure.

Future development could look at different options around how to retrieve the full collection of NMS ringforts in a cost-efficient way. One suggested approach is to use Firestore data bundles. The data bundles are a *serialized group of documents, retrieved using a specific query*. (Firebase, 2021). These data bundles can then be stored on a Content Distribution Network (CDN) which can improve the speed and cost of how data is retrieved with the help of caching, (Cloudflare, 2022).

## 6.3 Problems I encountered

### 6.3.1 Proving the Application on an IOS device

During development I used an Android emulator to test the new features being added, but because a major project objective was to create a cross platform mobile app, I really needed to test the app on an IOS device.

The only way this was possible was to build and install the app using XCode which is only available on MAC, but I was using a windows laptop!. So after exploring all options, I took the plunge and bought a 2015 iMac and installed XCode. In my sprint planning I had the task estimated at 4 hours to build and install the app for IOS, and **20+ hours** later I had managed to get the build and install to my iPhone to work.

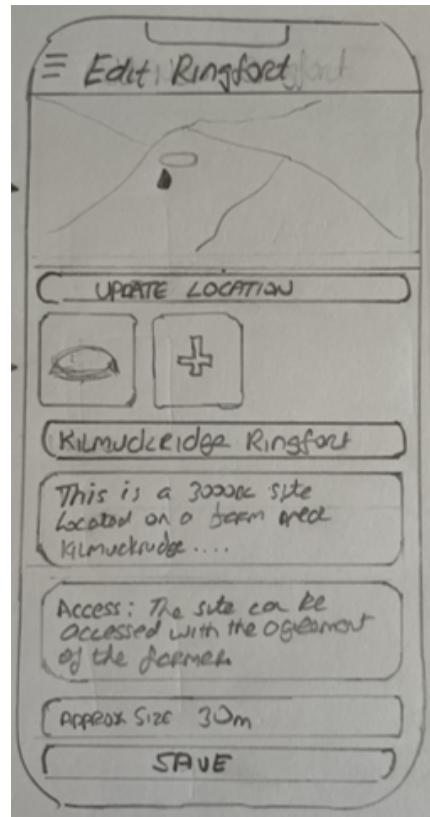
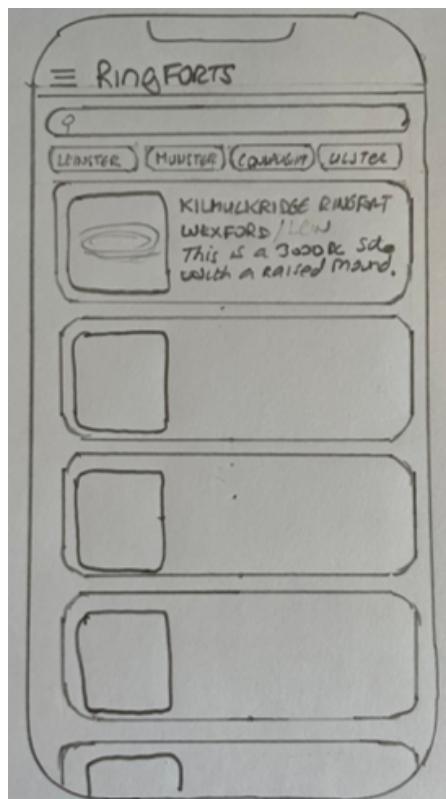
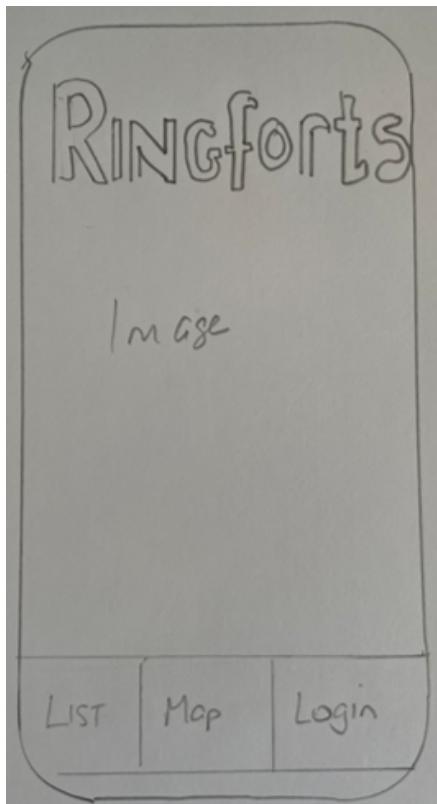
A number of issues were encountered, including initially not being part of the developer program which meant the release version did install but was only valid for 7 days before needing to be re-validated with a new build and install. This was resolved by discovering that WIT had access to this program for students. Another issue seemed to be caused by my apple watch being synced with my iPhone which seems to be an issue which caused development update for the watch to try install endlessly. In the end I need to make sure my apple watch was turned off when building and installing the app on IOS.

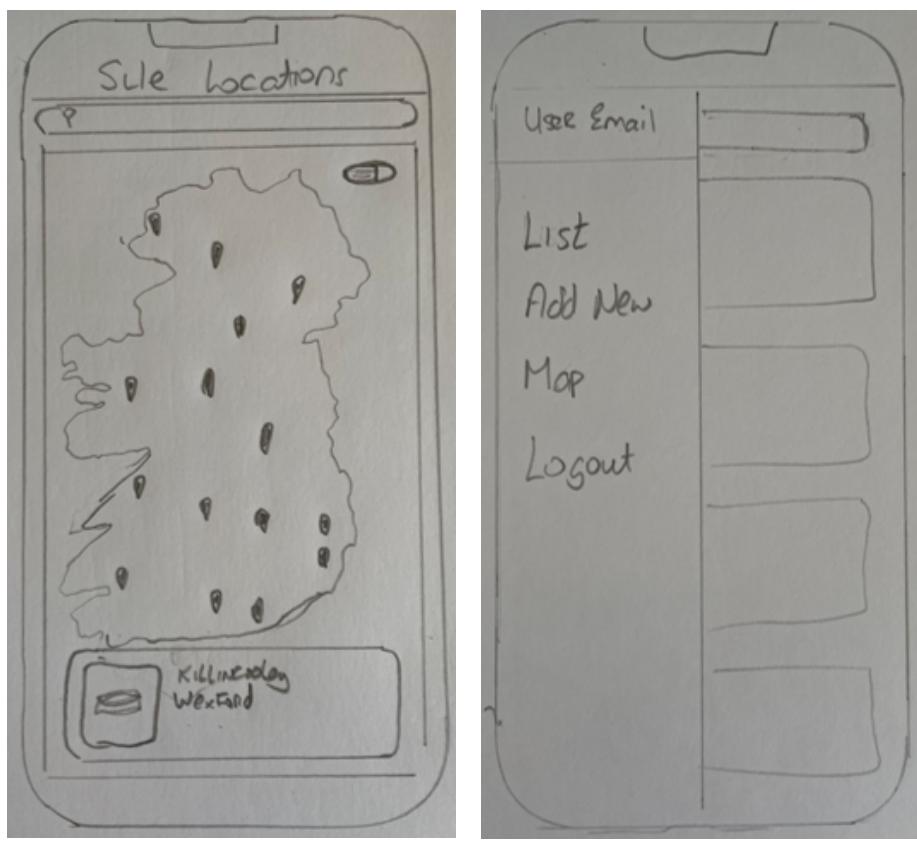
### 6.3.2 State management in the App

An area I struggled with for a while was implementing successful state management within the app. I settled on using a flutter package call '**provider**' to assist with this, which allowed me to define multiple provider classes in my top-level main class. This allowed multiple screens to have access to the state data and methods on that data. For example, this allowed me to implement an app wide **setFilteredSites** method which updated a ringfort state variable List called **filteredSites** when it was called from the search bar, or if triggered from the favourites chip, or triggered from the 'local' chip. This method and variable could be used in the List Ringforts screen and the Map overview screen to give consistent results..

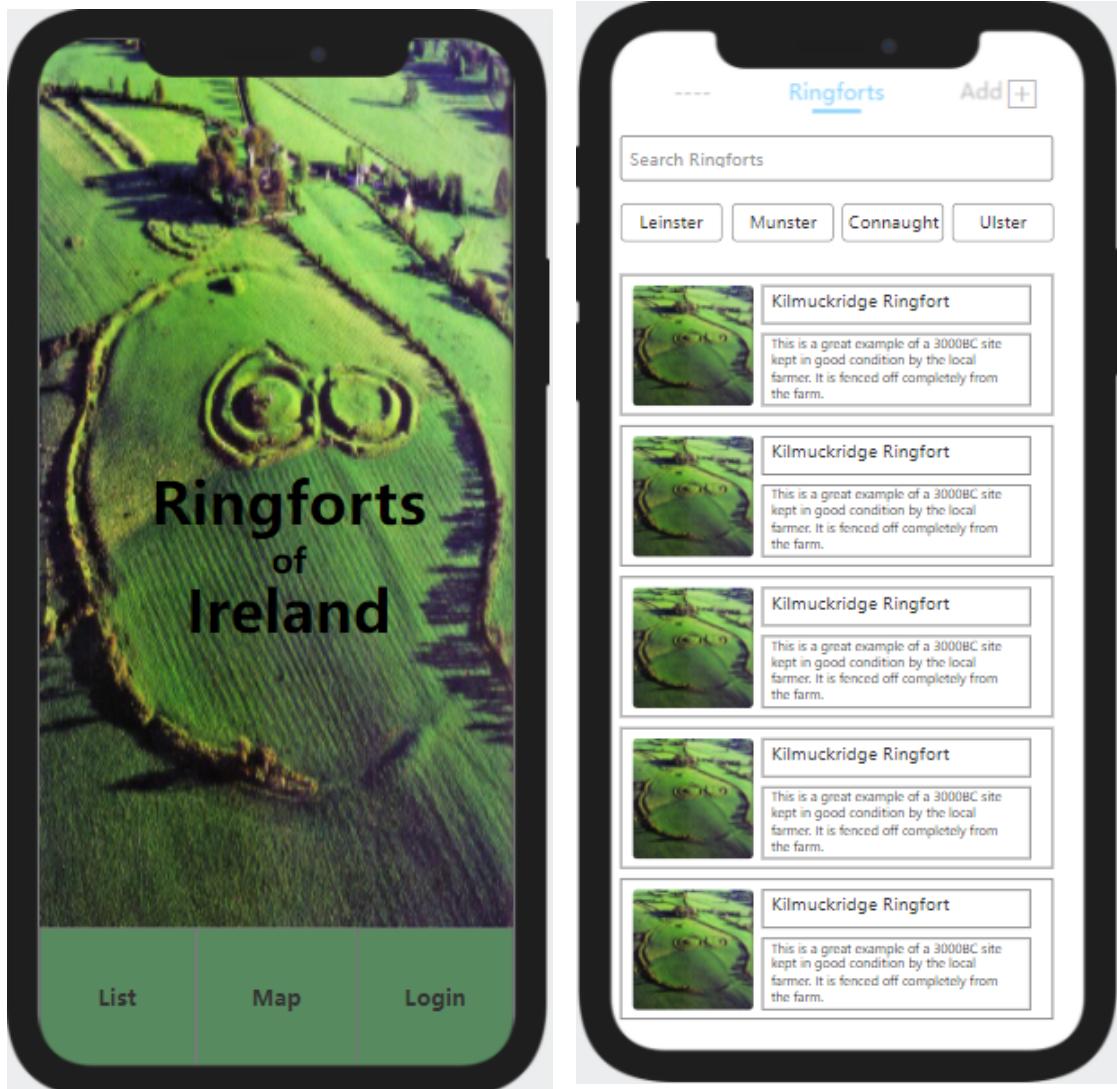
# Appendix

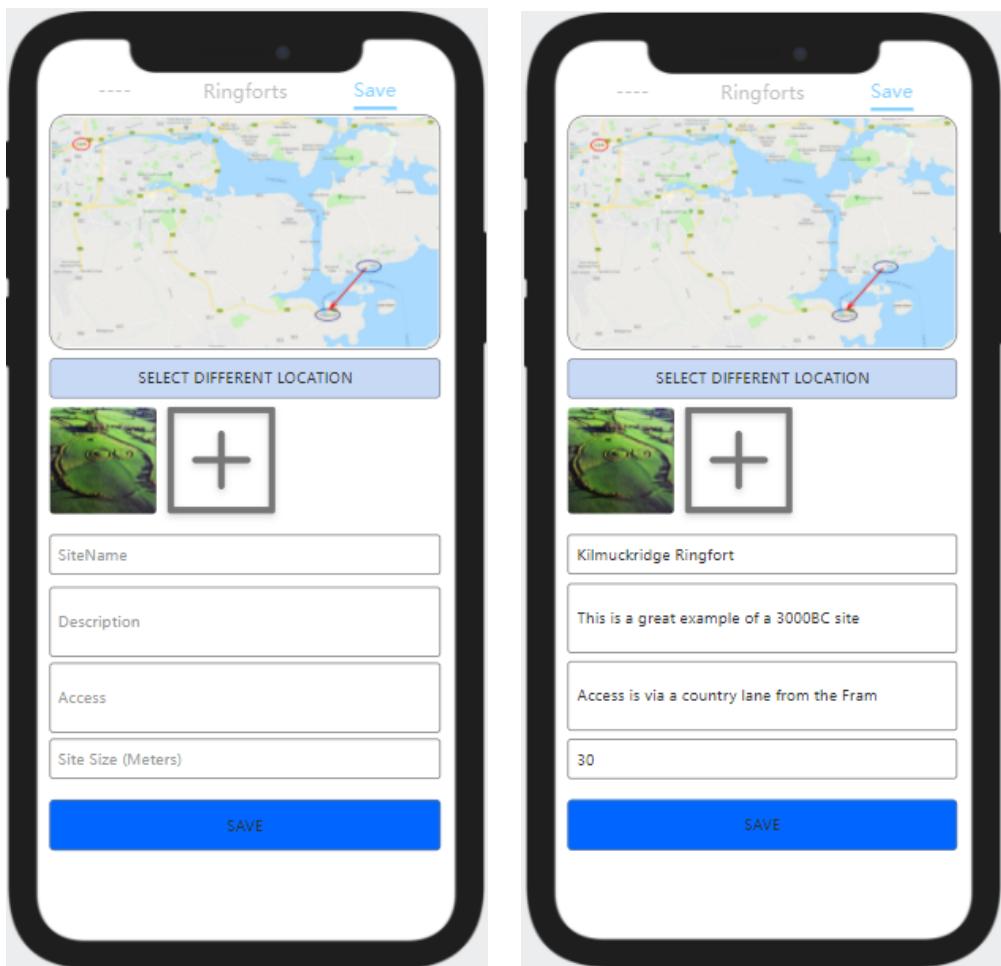
## Appendix A – Manually Drawn Screens

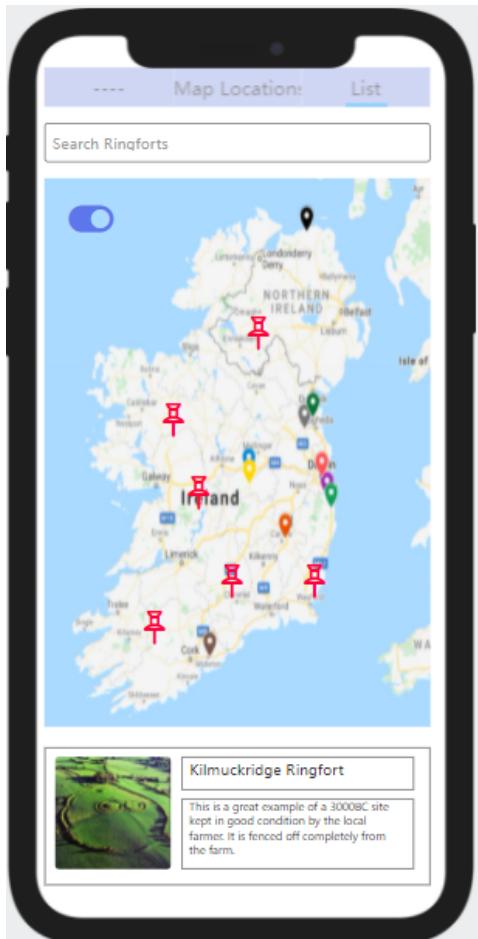




## Appendix B – Screen Design using Mock-up Tool







## Appendix C – User Stories

Title	(1) Add a New Ringfort	Title	(2) Add New Ringfort Location
As a	Ordinary User	As a	Ordinary User
I Want	To have the ability to add a new Ringfort to the system in an easy way with basic information.	I Want	To have the ability to easily add the location of a ringfort when adding a new one.
So That	The following important basic information is stored about the Ringfort 1) Ringfort Name 2) Short Description or known information 3) Is the site accessible 4) The approximate size of the site	So That	The exact location of the Ringfort is known with information such as 1) GPS Coordinates 2) Address Details 3) Country 4) Province
Priority	1	Priority	2
Estimate	7 hours	Estimate	5 hours

Title	(3) Add New Ringfort Images	Title	(4) List all Ringforts
As a	Ordinary User	As a	Ordinary User
I Want	To have the ability to easily take a camera image of a ringfort when adding a new one.	I Want	To have the ability to see a list of all Ringforts which have been added to the system.
So That	1) The image is available to be viewed 2) The image is stored locally 3) The location of the image is available and linked to the ringfort	So That	1) A list of all ringforts is available and scrollable 2) Each Ringfort entry displays the Image taken 3) Each Ringfort entry displays the Ringfort Name 4) Each Ringfort entry displays the Ringfort Description
Priority	3	Priority	4
Estimate	4 hours	Estimate	5 hours

Title	(5) Edit Ringfort Details	Title	(6) Delete Ringfort Details
As a	Ordinary User	As a	Ordinary User
I Want	To have the ability to edit an existing Ringfort on the system in an easy way.	I Want	To have the ability to delete an existing Ringfort on the system in an easy way..
So That	The following important basic information is updateable: 1) Ringfort Name 2) Short Description or known information 3) Is the site accessible 4) The approximate size of the site 5) An new image can be added 6) The Ringfort location can be updated	So That	1) When the delete is complete the data is completely removed from storage.  2) I will be given the chance to confirm that the delete should happen before it is executed
Priority	5	Priority	6
Estimate	5 hours	Estimate	3 hours

Title	(7) Make Data Available to All Users	Title	(8) Allow User to Signup
As a	Ordinary User	As a	Ordinary User (New)
I Want	To have the ability to see Ringforts added by other users and that they can see those add by me.	I Want	To be able to sign up to the App using an email and password.
So That	1) When a new Ringfort is added other users can immediately see it in their list when they refresh 2) When a Ringfort is deleted it is removed from other users when they refresh 3) When a Ringfort gets updates the changes are available to other users when they refresh <b>Note:</b> Data will be stored centrally on Firebase Firestone	So That	1) When I click Sign-up, I am presented with a screen to allow be to enter an email and password (and confirm password) 2) If the email exist already then I will be notified and allowed to select another email 3) If successful then I will be authenticated to the App. <b>Note:</b> Authentication will be managed using Firebase Auth services.
Priority	7	Priority	8
Estimate	8 hours	Estimate	7 hours

Title	(9) Allow User to Login	Title	(10) Allow User to Logout
As a	Ordinary User (Existing)	As a	Ordinary User (Existing)
I Want	To be able to login to the App using an email and password which I Signed-up with previously.	I Want	To be able to logout of the App if I am currently logged in.
So That	1) When I click Login, I will be presented with a screen which will allow me to enter an email and password, 2) If the email and password combination do not match an existing user, then I will be given a message to say that login credentials are not valid 3) If successful then I will be authenticated to the App. <b>Note:</b> Authentication will be managed using Firebase Auth services.	So That	1) When I click Logout, I will be logged out of the app and taken to the main screen with options to List, Map and Login. <b>Note:</b> Authentication will be managed using Firebase Auth services.
Priority	9	Priority	10
Estimate	4 hours	Estimate	2 hours

Title	(11) Limit Access until User Logged-in	Title	(12) Display image on New Screen
As a	Ordinary User (Existing)	As a	Ordinary User (Existing)
I Want	To disable certain functionality until a user has been authenticated.	I Want	To be able to display a saved image on a new screen and map zoomable
So That	<p>The following functionality will not be allowed for <b>non</b>-logged-in users, and they will be taken to the login screen</p> <ul style="list-style-type: none"> <li>1) Deleting a Ringfort</li> <li>2) Updating a Ringfort</li> <li>3) Creating a Ringfort</li> <li>4) View Ringfort Details</li> </ul> <p><b>Note:</b> Authentication will be managed using Firebase Auth services.</p>	So That	<p>1) When editing a Ringfort, allow the user to click an existing saved image which would open on a new screen.</p> <p>2) Allow the image to be zoomable on the new screen.</p> <p>3) Allow for an attractive transition from one screen to the other.</p>
Priority	11	Priority	12
Estimate	5 hours	Estimate	5 hours

Title	(13) Searching the Ringfort List	Title	(14) Searching the Ringfort Map
As a	Ordinary User (Existing)	As a	Ordinary User (Existing)
I Want	To be able to search the list of Ringforts and also filter by Province	I Want	To be able to search the Map of Ringforts and display only marker for matching Ringforts
So That	<p>1) On the Ringfort List screen there will be a search bar, that when typed into will immediately start filtering Ringforts. The search will be in all these fields</p> <ul style="list-style-type: none"> <li>- Name and Description</li> <li>- Access</li> <li>- Address and Province</li> </ul> <p>2) There are 4 province buttons showing on the list screen. And pressing one will limit the Ringforts displayed to those located in that province.</p> <p>3) The province selection can be used in conjunction with the search bar.</p> <p>4) There is an option to revert to all Ringforts displayed.</p>	So That	<p>1) On the Ringfort Map screen there will be a search bar, that when typed into will immediately start filtering Ringforts. This will limit the markers displayed to those which match the search criteria The search will be in all these fields</p> <ul style="list-style-type: none"> <li>- Name and Description</li> <li>- Access</li> <li>- Address and Province</li> </ul> <p>2) There is an option to revert to all Ringforts displayed on the Map.</p>
Priority	13	Priority	14
Estimate	8 hours	Estimate	5 hours

Title	<b>(15) Add a Ringfort as a Favourite</b>	Title	<b>(16) Make Photos Available to all</b>
As a	Ordinary User (Logged-in)	As a	Ordinary User (Logged-in)
I Want	To be able set a Ringfort as a favourite	I Want	To be able to view images taken by other users
So That	1) On the Ringfort List screen there is an option to set one as a favourite 2) On the update Ringfort screen there is an option to set it as favourite 3) There is an option to filter ringforts by users' favourites 4) There is an option to revert to all Ringforts displayed.	So That	1) Make the images taken by a user available to other users. i.e., on the Edit screen and on the list screen  NOTE: Image storage will be handled using Firebase Storage
Priority	15	Priority	16
Estimate	7 hours	Estimate	5 hours

Title	<b>(17) Set up Approval System</b>	Title	<b>(18) Load and Display NMS Data</b>
As a	Ordinary User (Logged-in)(Admin-user)	As a	Ordinary User (Logged-in)(Admin-user)
I Want	To make any Add/Update Deletes subject to approval by an Admin User	I Want	To make the NMS data available to the app and display on a new map. With the ability to click this data to update and add to live database.
So That	1) When an Ordinary user adds a new Ringfort it is sent to an Admin user for approval before it will appear to other users.  2) When an Ordinary user deletes an existing Ringfort it is sent to an Admin user for approval before it will disappear for other users.  3) When an Ordinary user updates an existing Ringfort it is sent to an Admin user for approval before it will appear updated to other users.  4) An Admin user will see an new screen of changes awaiting approval which they can approve or reject	So That	1) Download National Monument Service data from their website and load to a new Firebase collection.  2) When a user chooses NMS Uploaded Data from the Nav Drawer they would be sent to a map screen of all the NMS sites  3) When a user clicks on one of these Ringfort sites they would be brought to an update screen where additional info and images can be added.  4) When the data is saved, it is added to the live collection of Ringforts and removed from the NMS collection on Firebase.
Priority	17	Priority	18
Estimate	10 hours	Estimate	12 hours

## Appendix D – Original Sprint Plan

This is the initial 3 sprint plan. These should complete successfully on the 13.02.2022

Sprint No:	1	2	3
Start Date	04.01.2022	17.01.2022	31.01.2022
End Date	16.01.2022	30.01.2022	13.02.2022
Tasks	1) Compare and Select Front End Technologies (4 hours)  2) Study selected Front End Technology to become competent (25 hours)  TOTAL: 29 hrs	1) Contact User (Local Historian) and discuss requirements (1 hour)  2) Mock up sample screens by manually drawing and confirm with user (4 hours)  3) Study selected Front End Technology to become competent (12 hours)  4) Start project report layout and structure (3 hours)  5) Document design of Data model (3 hours)  TOTAL: 23	1) Write user Stories for highest priority tasks (3 hour)  2) Turn screen drawings into Wireframes (4 hours)  3) Write project report to interim report level (10 hours)  4) User Story 1 Dev (7 hours) (Add a New Ringfort)  5) User Story 2 Dev (5 hours) (Add New Ringfort Location)  6) User Story 3 Dev (4 hours) (Add New Ringfort Images)  TOTAL: 33
Status	Complete	Complete	In progress

Sprint No:	<b>4</b>	<b>5</b>	<b>6</b>
Start Date	14.02.2022	28.02.2022	14.03.2022
End Date	27.02.2022	13.03.2022	27.03.2022
Tasks	1) Continue Project Report (5 hour) 2) User Story 4 Dev (5 hours) (List all Ringforts) 3) Wireframe Login/Sign-up Screens (2 hours) 4) User Story 5 Dev (5 hours) (Edit Ringfort Details) 5) User Story 6 Dev (4 hours) (Delete Ringfort Details) 6) User Story 7 Dev (8 hours) (Make Data Available to All Users)	1) Continue Project Report (5 hour) 2) User Story 8 Dev (7 hours) (Allow User to Signup) 3) User Story 9 Dev (4 hours) (Allow User to Login) 4) User Story 10 Dev (2 hours) (Allow User to Logout) 5) User Story 11 Dev (5 hours) (Limit Access until User Logged-in) 6) User Story 12 Dev (5 hours) (Display saved Image on new screen)	1) Get Handbook Entries ready (3 hours) 2) Complete self-reflection in report (6 hours) 3) User Story 16 Dev (5 hours) (Make Photos Available to all) 4) User Story 13 Dev (8 hours) (Searching the Ringfort List) 5) User Story 14 Dev (5 hours) (Searching the Ringfort Map) 6) User Story 15 Dev (7 hours) (Setting a Ringfort as a favourite)
	TOTAL: 29	TOTAL: 28	TOTAL: 34
Status	Awaiting	Awaiting	Awaiting

Sprint No:	<b>7</b>	<b>8</b>	<b>9</b>
Start Date	28.03.2022	04.04.2022	11.04.2022
End Date	03.04.2022	10.04.2022	17.04.2022
Tasks	NOTE: 1 Week Sprint 1) Complete Project Report 2) Fix any code bugs and see if any small additions possible.. 3) Complete README	NOTE: 1 Week Sprint 1) Prepare Demo Video	NOTE: 1 Week Sprint 1) Prepare Presentation
Status	Awaiting	Awaiting	Awaiting

## Appendix E – Updated Sprint Plan as at 01.03.2022

It became apparent after the 1<sup>st</sup> three sprints that progress was being made quicker than originally planned, largely due to more time being spent than original envisaged. Based on this I decided to do some sprint re-planning to get a more accurate picture.

Sprint No:	1	2	3
Start Date	04.01.2022	17.01.2022	31.01.2022
End Date	16.01.2022	30.01.2022	13.02.2022
Tasks	1) Compare and Select Front End Technologies (4 hours)  2) Study selected Front End Technology to become competent (25 hours)  TOTAL: 29 hrs	1) Contact User (Local Historian) and discuss requirements (1 hour)  2) Mock up sample screens by manually drawing and confirm with user (4 hours)  3) Study selected Front End Technology to become competent (12 hours)  4) Start project report layout and structure (3 hours)  5) Document design of Data model (3 hours)  TOTAL: 23	1) Write user Stories for highest priority tasks (3 hour)  2) Turn screen drawings into Wireframes (4 hours)  3) Write project report to interim report level (10 hours)  4) User Story 1 Dev (7 hours) (Add a New Ringfort)  5) User Story 2 Dev (5 hours) (Add New Ringfort Location)  6) User Story 3 Dev (4 hours) (Add New Ringfort Images)  7) User Story 4 Dev (5 hours) (List all Ringforts)  TOTAL: 38
Status	Complete	Complete	Complete

The next 3 sprints have been totally re-planned. Based on spending more time per sprint and tasks taking less time than originally planned.

Sprint No:	4	5	6
Start Date	14.02.2022	28.02.2022	14.03.2022
End Date	27.02.2022	13.03.2022	27.03.2022
Tasks	1) Continue Project Report (2 hour)  2) User Story 5 Dev (5 hours) (Edit Ringfort Details)  3) User Story 6 Dev (4 hours) (Delete Ringfort Details)  4) User Story 7 Dev (5 hours) (Make Data Available to All Users)  5) User Story 8 Dev (5 hours) (Allow User to Signup)  6) User Story 9 Dev (4 hours) (Allow User to Login)  7) User Story 10 Dev (2 hours) (Allow User to Logout)  8) User Story 11 Dev (5 hours) (Limit Access until User Logged-in)  TOTAL: 32	1) Continue Project Report (5 hour)  2) User Story 16 Dev (5 hours) (Make Photos Available to all)  3) User Story 13 Dev (6 hours) (Searching the Ringfort List)  4) User Story 14 Dev (4 hours) (Searching the Ringfort Map)  5) User Story 15 Dev (7 hours) (Setting Ringfort as User Favourite)  6) User Story 18 Dev (12 hours) (Adding NMS data and displaying on Map and allow to update)  7) Install Xcode on apple IMAC and build the project for IOS. (4hrs)	1) Get Handbook Entries ready (10 hours)  2) Complete self-reflection in report (6 hours)  3) User Story 12 Dev (8 hours) (Allow images to be displayed on new screen and zoomable)  4) Bug fixes and commenting (4 hours)  TOTAL: 28
Status	Complete	In-progress	Awaiting

## Appendix F – Updated Sprint Plan as at 14.03.2022

The replan on the 01.03.2022 proved to be accurate. An additional task was added into sprint 5, this was User story 17 – adding approval system for normal users. Here is the current status as at the 14.03.2022 which is the start of sprint 6.

Sprint No:	1	2	3
Start Date	04.01.2022	17.01.2022	31.01.2022
End Date	16.01.2022	30.01.2022	13.02.2022
Tasks	1) Compare and Select Front End Technologies (4 hours)  2) Study selected Front End Technology to become competent (20 hours)  TOTAL: 24 hrs	1) Contact User (Local Historian) and discuss requirements (1 hour)  2) Mock up sample screens by manually drawing and confirm with user (4 hours)  3) Study selected Front End Technology to become competent (12 hours)  4) Start project report layout and structure (3 hours)  5) Document design of Data model (3 hours)  TOTAL: 23	1) Write user Stories for highest priority tasks (3 hour)  2) Turn screen drawings into Wireframes (4 hours)  3) Write project report to interim report level (10 hours)  4) User Story 1 Dev (7 hours) (Add a New Ringfort)  5) User Story 2 Dev (5 hours) (Add New Ringfort Location)  6) User Story 3 Dev (4 hours) (Add New Ringfort Images)  7) User Story 4 Dev (5 hours) (List all Ringforts)
Status	Complete	Complete	Complete

Sprint No:	4	5	6
Start Date	14.02.2022	28.02.2022	14.03.2022
End Date	27.02.2022	13.03.2022	27.03.2022
Tasks	1) Continue Project Report (2 hour)  2) User Story 5 Dev (5 hours) (Edit Ringfort Details)  3) User Story 6 Dev (4 hours) (Delete Ringfort Details)  4) User Story 7 Dev (5 hours) (Make Data Available to All Users)  5) User Story 8 Dev (5 hours) (Allow User to Signup)  6) User Story 9 Dev (4 hours) (Allow User to Login)  7) User Story 10 Dev (2 hours) (Allow User to Logout)  8) User Story 11 Dev (5 hours) (Limit Access until User Logged-in)  TOTAL: 32	1) Continue Project Report (5 hour)  2) User Story 16 Dev (5 hours) (Make Photos Available to all)  3) User Story 13 Dev (6 hours) (Searching the Ringfort List)  4) User Story 14 Dev (4 hours) (Searching the Ringfort Map)  5) User Story 15 Dev (7 hours) (Setting Ringfort as User Favourite)  6) User Story 18 Dev (12 hours) (Adding NMS data and displaying on Map and allow to update)  7) User Story 17 Dev (10 hours) (Approval System for Normal users)  8) Install Xcode on apple IMAC and build the project for IOS. (4hrs)	1) Get Handbook Entries ready (10 hours)  2) Complete self-reflection in report (6 hours)  3) Complete report (7 hours)  3) User Story 12 Dev (8 hours) (Allow images to be displayed on new screen and zoomable)  4) Bug fixes and commenting (4 hours)  TOTAL: 35
Status	Complete	Complete	In-progress

## Appendix G – Firebase Pricing

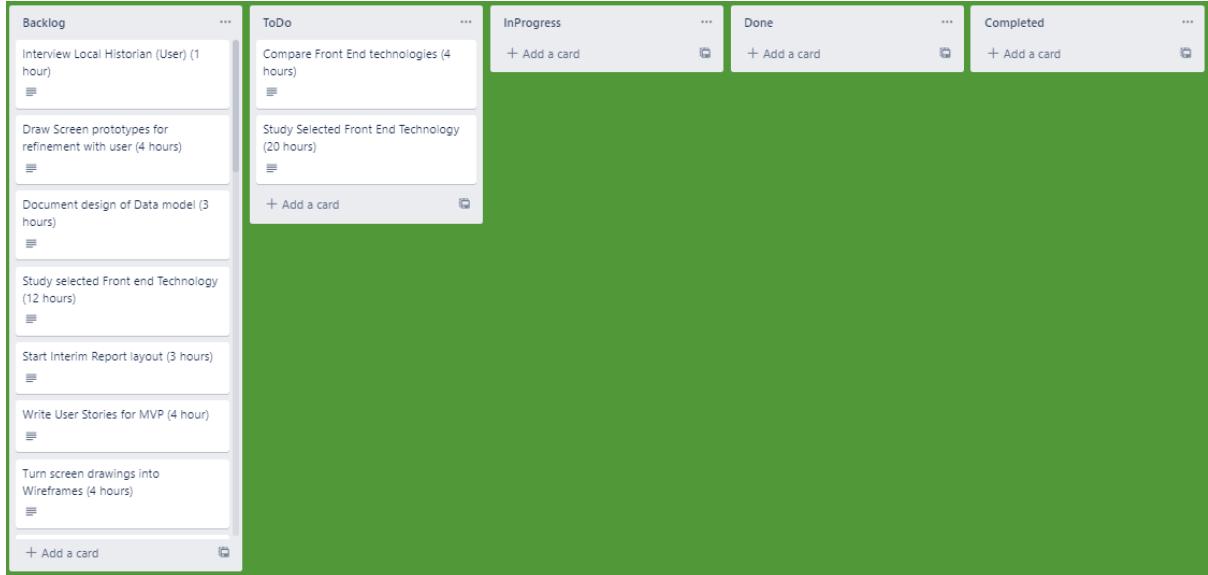
These are the latest Firebase prices from their website.

Products	No-cost Spark Plan	Pay as you go Blaze Plan
<b>Cloud Firestore</b>		
Stored data	1 GiB total	No-cost up to 1 GiB total Then \$0.108 per additional GiB
Network egress	10 GiB/month	No-cost up to 10 GiB/month Then <a href="#">Google Cloud pricing</a>
Document writes	20K writes/day	No-cost up to 20K writes/day Then <a href="#">Google Cloud pricing</a>
Document reads	50K reads/day	No-cost up to 50K reads/day Then <a href="#">Google Cloud pricing</a>
Document deletes	20K deletes/day	No-cost up to 20K deletes/day Then <a href="#">Google Cloud pricing</a>
<b>Cloud Storage</b> ⓘ		
GB stored	5 GB	\$0.026/GB
GB downloaded	1 GB/day	\$0.12/GB
Upload operations	20K/day	\$0.05/10k
Download operations	50K/day	\$0.004/10k
Multiple buckets per project	✗	✓

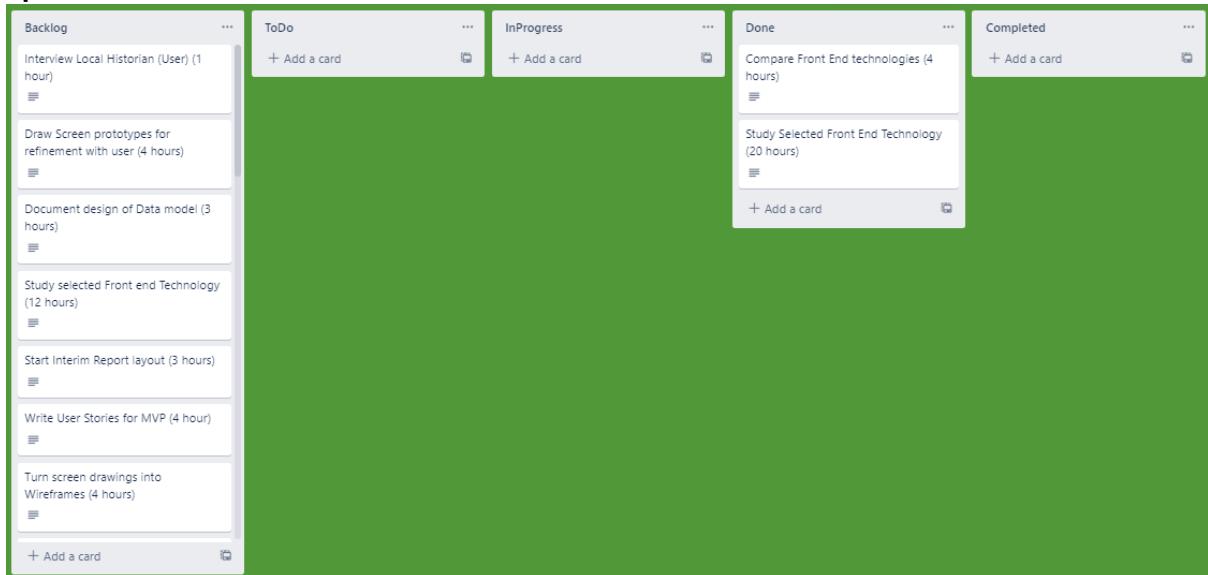
## Appendix H – Trello Sprint Tracking

This section shows the sprint progress, with a snap shot of the Trello board before and after each sprint.

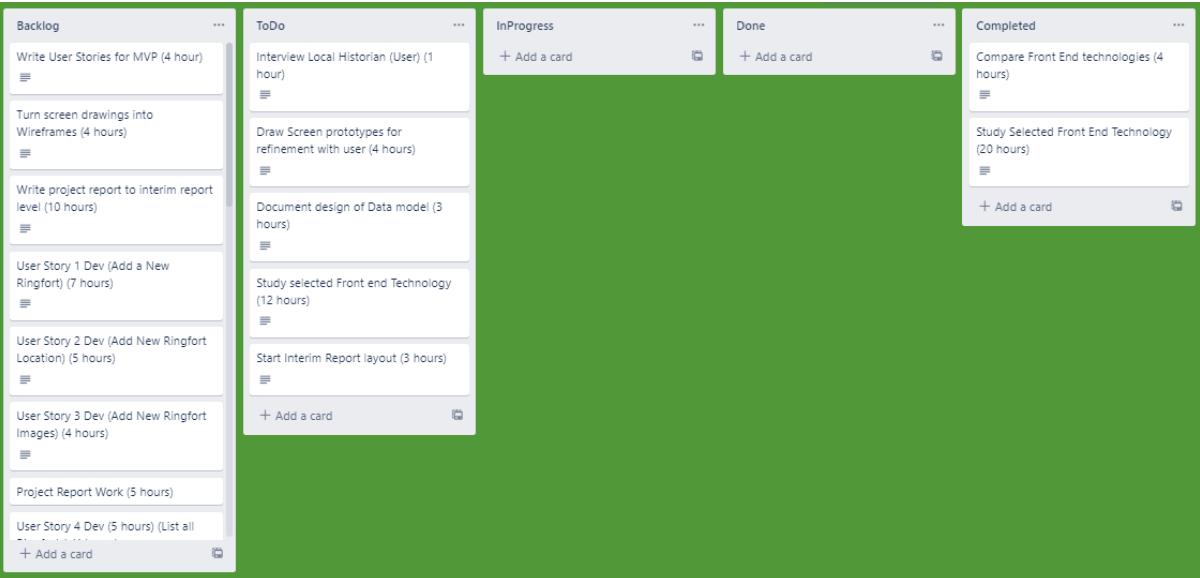
### Sprint 1 Start



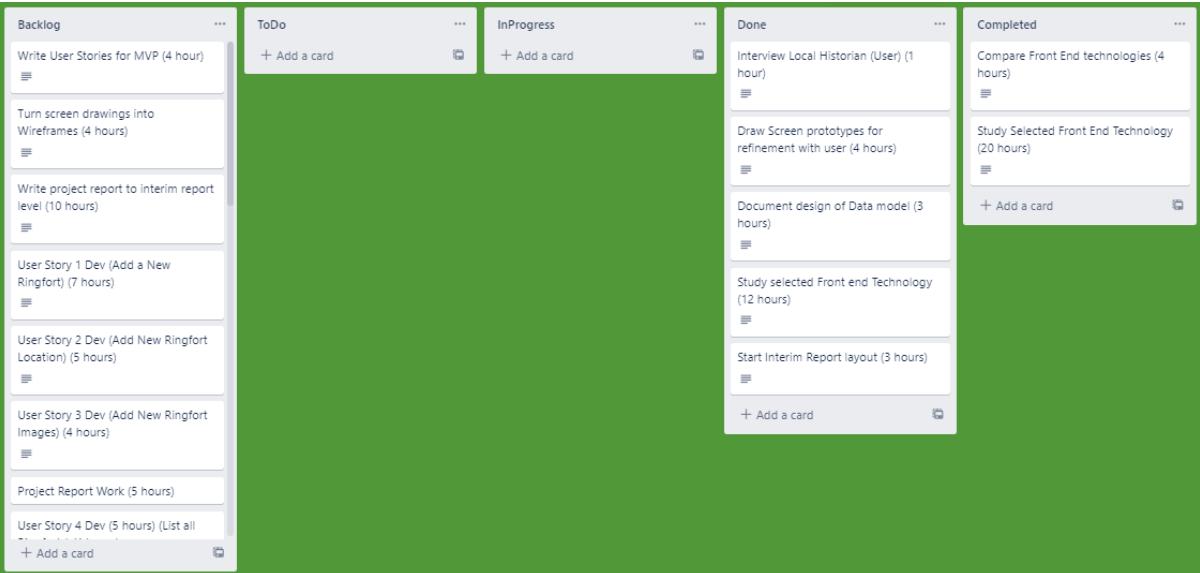
### Sprint 1 End



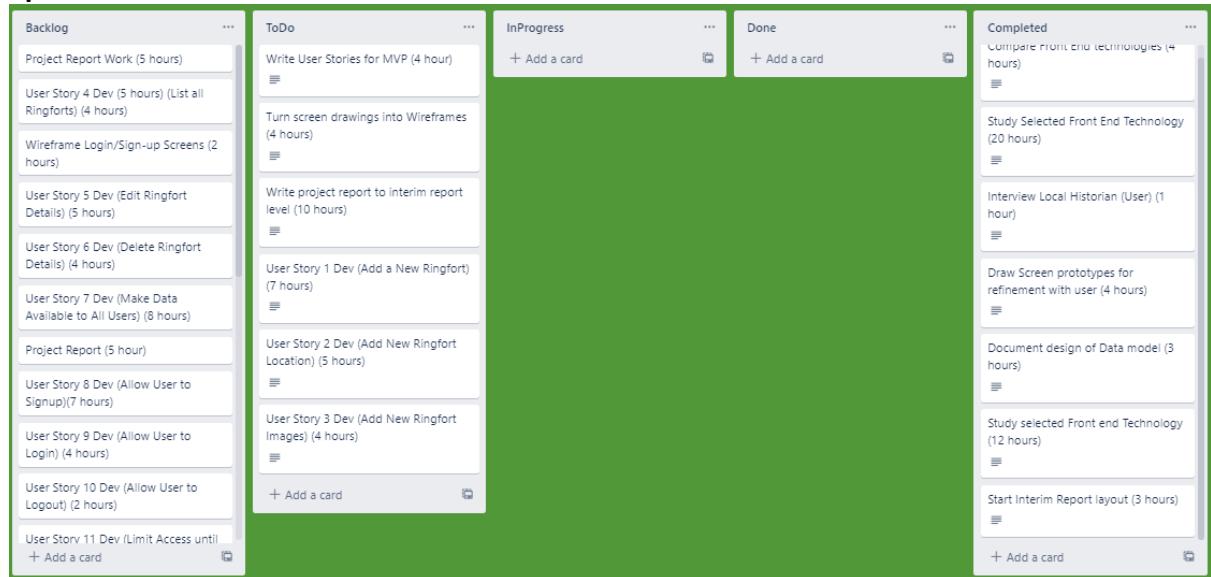
## Sprint2 start



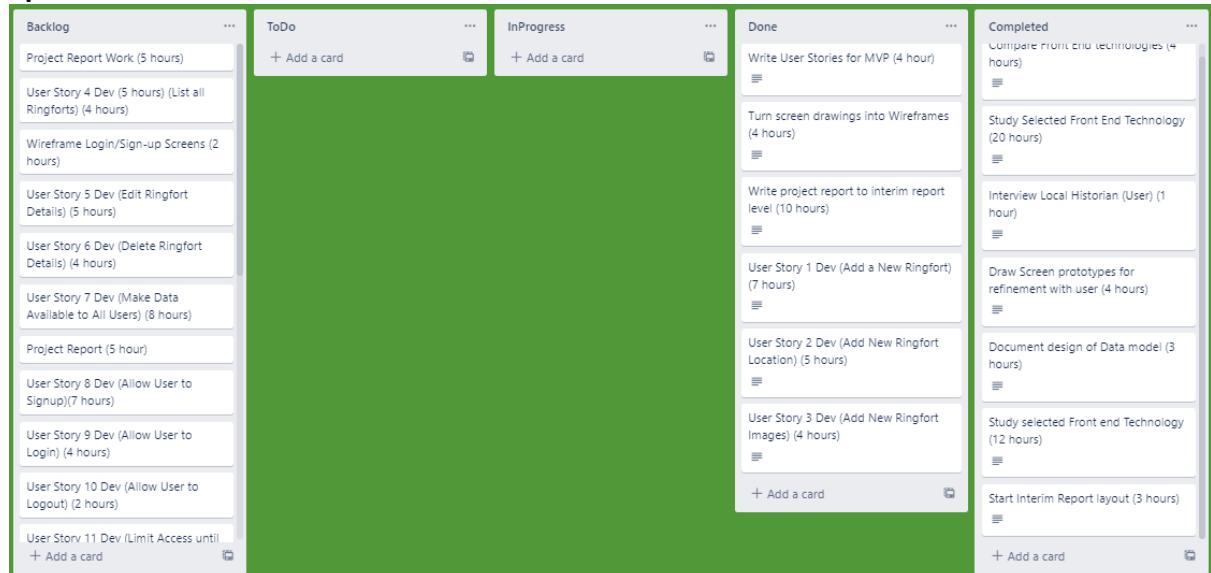
## Sprint2 End



## Sprint 3 Start



## Sprint 3 End



## Final Project Report: Ringforts of Ireland

### Sprint 4 Start

Backlog	ToDo	InProgress	Done	Completed
User Story 4 Dev (5 hours) (List all Ringforts) (4 hours)	Project Report Work (5 hours)	+ Add a card	+ Add a card	Compare Front End technologies (4 hours)
Wireframe Login/Sign-up Screens (2 hours)	User Story 5 Dev (Edit Ringfort Details) (5 hours)			Study Selected Front End Technology (20 hours)
Project Report (5 hour)	User Story 6 Dev (Delete Ringfort Details) (4 hours)			Interview Local Historian (User) (1 hour)
User Story 12 Dev (Add Multiple Images per Ringfort) (8 hours)	User Story 7 Dev (Make Data Available to All Users) (8 hours)			Draw Screen prototypes for refinement with user (4 hours)
Get Handbook Entries ready (3 hours)	User Story 8 Dev (Allow User to Signup) (7 hours)			Document design of Data model (3 hours)
Complete self reflection in report (6 hours)	User Story 9 Dev (Allow User to Login) (4 hours)			Study selected Front end Technology (12 hours)
User Story 16 Dev (Make Photos Available to all) (5 hours)	User Story 10 Dev (Allow User to Logout) (2 hours)			Start Interim Report layout (3 hours)
User Story 13 Dev (Searching the Ringfort List) (8 hours)	User Story 11 Dev (Limit Access until User Logged-in) (5 hours)			Write User Stories for MVP (4 hour)
User Story 14 Dev (Searching the Ringfort Map) (5 hours)	+ Add a card			Turn screen drawings into Wireframes (4 hours)
User Story 15 Dev (Favouriting A Ringfort) (7 hours)				Write project report to interim report level (10 hours)
Complete Project Report				+ Add a card
Fix any code bugs and see if any small additions possible..				
Complete README				
Prepare Demo Video				
Prepare Presentation				
+ Add a card				

### Sprint 4 End

Backlog	ToDo	InProgress	Done	Completed
User Story 4 Dev (5 hours) (List all Ringforts) (4 hours)	+ Add a card	+ Add a card	Project Report Work (5 hours)	Compare Front End technologies (4 hours)
Wireframe Login/Sign-up Screens (2 hours)			User Story 5 Dev (Edit Ringfort Details) (5 hours)	Study Selected Front End Technology (20 hours)
Project Report (5 hour)			User Story 6 Dev (Delete Ringfort Details) (4 hours)	Interview Local Historian (User) (1 hour)
User Story 12 Dev (Add Multiple Images per Ringfort) (8 hours)			User Story 7 Dev (Make Data Available to All Users) (8 hours)	Draw Screen prototypes for refinement with user (4 hours)
Get Handbook Entries ready (3 hours)			User Story 8 Dev (Allow User to Signup) (7 hours)	Document design of Data model (3 hours)
Complete self reflection in report (6 hours)			User Story 9 Dev (Allow User to Login) (4 hours)	Study selected Front end Technology (12 hours)
User Story 16 Dev (Make Photos Available to all) (5 hours)			User Story 10 Dev (Allow User to Logout) (2 hours)	Start Interim Report layout (3 hours)
User Story 13 Dev (Searching the Ringfort List) (8 hours)			User Story 11 Dev (Limit Access until User Logged-in) (5 hours)	Write User Stories for MVP (4 hour)
User Story 14 Dev (Searching the Ringfort Map) (5 hours)	+ Add a card			Turn screen drawings into Wireframes (4 hours)
User Story 15 Dev (Favouriting A Ringfort) (7 hours)				Write project report to interim report level (10 hours)
Complete Project Report				+ Add a card
Fix any code bugs and see if any small additions possible..				
Complete README				
Prepare Demo Video				
Prepare Presentation				
+ Add a card				

## Final Project Report: Ringforts of Ireland

### Sprint 5 Start

Backlog	ToDo	InProgress	Done	Completed
User Story 12 Dev (Allow images to be displayed on new Screen) (8 hours)	Project Report (5 hour)	+ Add a card	+ Add a card	Compare Front End technologies (4 hours)
Get Handbook Entries ready (3 hours)	User Story 16 Dev (Make Photos Available to all) (5 hours)			User Story 4 Dev (5 hours) (List all Ringforts) (4 hours)
Complete self reflection in report (6 hours)	User Story 13 Dev (Searching the Ringfort List) (8 hours)			Wireframe Login/Sign-up Screens (2 hours)
Complete Project Report	User Story 14 Dev (Searching the Ringfort Map) (5 hours)			Study Selected Front End Technology (20 hours)
Fix any code bugs and see if any small additions possible..	User Story 15 Dev (Favouriting A Ringfort) (7 hours)			Interview Local Historian (User) (1 hour)
Complete README	User Story 18 (Adding NMS data and displaying on Map and allow to update) (12 hours)			Draw Screen prototypes for refinement with user (4 hours)
Prepare Demo Video	User Story 17 (Approval System for normal users) (10 hours)			Document design of Data model (3 hours)
Prepare Presentation	Install Xcode on iMac and build the project for IOS (4 hours)	+ Add a card		Study selected Front end Technology (12 hours)
+ Add a card	+ Add a card	+ Add a card	+ Add a card	Start Interim Report layout (3 hours)
				Write User Stories for MVP (4 hour)
				Turn screen drawings into Wireframes (4 hours)
			+ Add a card	+ Add a card

### Sprint 5 End

Backlog	ToDo	InProgress	Done	Completed
User Story 12 Dev (Allow images to be displayed on new Screen) (8 hours)	+ Add a card	+ Add a card	Project Report (5 hour)	Compare Front End technologies (4 hours)
Get Handbook Entries ready (3 hours)			User Story 16 Dev (Make Photos Available to all) (5 hours)	User Story 4 Dev (5 hours) (List all Ringforts) (4 hours)
Complete self reflection in report (6 hours)			User Story 13 Dev (Searching the Ringfort List) (8 hours)	Wireframe Login/Sign-up Screens (2 hours)
Complete Project Report			User Story 14 Dev (Searching the Ringfort Map) (5 hours)	Study Selected Front End Technology (20 hours)
Fix any code bugs and see if any small additions possible..			User Story 15 Dev (Favouriting A Ringfort) (7 hours)	Interview Local Historian (User) (1 hour)
Complete README			User Story 18 (Adding NMS data and displaying on Map and allow to update) (12 hours)	Draw Screen prototypes for refinement with user (4 hours)
Prepare Demo Video			User Story 17 (Approval System for normal users) (10 hours)	Document design of Data model (3 hours)
Prepare Presentation			Install Xcode on iMac and build the project for IOS (4 hours)	Study selected Front end Technology (12 hours)
+ Add a card			+ Add a card	Start Interim Report layout (3 hours)
				Write User Stories for MVP (4 hour)
				Turn screen drawings into Wireframes (4 hours)
			+ Add a card	+ Add a card

## Sprint 6 Start

Backlog	ToDo	InProgress	Done	Completed
Complete README Prepare Demo Video Prepare Presentation + Add a card	Get Handbook Entries ready (3 hours) Complete self reflection in report (6 hours) Complete Project Report User Story 12 Dev (Allow images to be displayed on new Screen) (8 hours) Fix any code bugs and see if any small additions possible.. + Add a card	+ Add a card	+ Add a card	User Story 6 Dev (Delete Ringfort Details) (4 hours) User Story 7 Dev (Make Data Available to All Users) (8 hours) User Story 8 Dev (Allow User to Signup)(7 hours) User Story 9 Dev (Allow User to Login) (4 hours) User Story 10 Dev (Allow User to Logout) (2 hours) User Story 11 Dev (Limit Access until User Logged-in) (5 hours) Project Report (5 hour) User Story 16 Dev (Make Photos Available to all) (5 hours) User Story 13 Dev (Searching the Ringfort List) (8 hours) User Story 14 Dev (Searching the Ringfort Map) (5 hours) User Story 15 Dev (Favouriting A Ringfort) (7 hours) User Story 18 (Adding NMS data and displaying on Map and allow to update) (12 hours) User Story 17 (Approval System for normal users) (10 hours) Install Xcode on iMac and build the project for IOS (4 hours) + Add a card

## Sprint 6 End

Backlog	ToDo	InProgress	Done	Completed
Complete README Prepare Demo Video Prepare Presentation + Add a card	+ Add a card	+ Add a card	Get Handbook Entries ready (3 hours) Complete self reflection in report (6 hours) Complete Project Report User Story 12 Dev (Allow images to be displayed on new Screen) (8 hours) Fix any code bugs and see if any small additions possible.. + Add a card	User Story 6 Dev (Delete Ringfort Details) (4 hours) User Story 7 Dev (Make Data Available to All Users) (8 hours) User Story 8 Dev (Allow User to Signup)(7 hours) User Story 9 Dev (Allow User to Login) (4 hours) User Story 10 Dev (Allow User to Logout) (2 hours) User Story 11 Dev (Limit Access until User Logged-in) (5 hours) Project Report (5 hour) User Story 16 Dev (Make Photos Available to all) (5 hours) User Story 13 Dev (Searching the Ringfort List) (8 hours) User Story 14 Dev (Searching the Ringfort Map) (5 hours) User Story 15 Dev (Favouriting A Ringfort) (7 hours) User Story 18 (Adding NMS data and displaying on Map and allow to update) (12 hours) User Story 17 (Approval System for normal users) (10 hours) Install Xcode on iMac and build the project for IOS (4 hours) + Add a card

## Bibliography

- Archaeology.ie. (2019). *Archaeological Survey of Ireland | National Monuments Service*. [online] Available at: <https://www.archaeology.ie/archaeological-survey-ireland> [Accessed 4 Dec. 2019].
- Clark, J. (2021). *Top 10 Advantages of Firebase*. [online] Back4App Blog. Available at: <https://blog.back4app.com/advantages-of-firebase/>.
- docs.flutter.dev. (2021). *Flutter architectural overview*. [online] Available at: <https://docs.flutter.dev/resources/architectural-overview> [Accessed 7 Feb. 2022].
- Dziuba, A. (2021). *Choosing a Map API for Your Next App: Mapbox, Google Maps, or OpenStreetMap?* [online] Relevant Software. Available at: <https://relevant.software/blog/choosing-a-map-amapbox-google-maps-openstreetmap/> [Accessed 7 Feb. 2022].
- Firebase. (2019). *Firebase Pricing | Firebase*. [online] Available at: <https://firebase.google.com/pricing>.
- Hendicott, J. (2017). *The Story Behind Ireland's Fairy Forts and Where to See Them*. [online] Culture Trip. Available at: <https://theculturetrip.com/europe/ireland/articles/the-story-behind-irelands-fairy-forts-and-where-to-see-them/>.
- kella (2021). *Firebase V/S REST APIs — A Developer's Guide*. [online] Medium. Available at: <https://medium.com/@sandeepkella23/firebase-v-s-rest-apis-a-developers-guide-54e9b17a4d98> [Accessed 7 Feb. 2022].
- Mapbox. (2021). *Mobile applications | Help*. [online] Available at: <https://docs.mapbox.com/help/getting-started/mobile-apps/> [Accessed 25 Jan. 2022].
- Moqod (2021). *Downsides of Firebase: limitations to be aware of*. [online] Moqod—Software company. Available at: <https://medium.com/moqod-software-company/downsides-of-firebase-limitations-to-be-aware-of-886ade5ae5a2> [Accessed 7 Feb. 2022].
- Statista. (2022). *Cross-platform mobile frameworks used by global developers 2020*. [online] Available at: <https://www.statista.com/statistics/869224/worldwide-software/>

developer-working-hours/ [Accessed 7 Feb. 2022].

Stempniak, A. (2021). *React Native vs. Flutter: A Comparison of Pros and Cons*. [online] www.stxnext.com. Available at: [https://www.stxnext.com/blog/react-native-vs-flutter-comparison/?utm\\_term=&utm\\_campaign=&utm\\_source=adwords&utm\\_medium=ppc&hsa\\_acc=4807594886&hsa\\_cam=15491073565&hsa\\_grp=&hsa\\_ad=&hsa\\_src=x&hsa\\_tgt=&hsa\\_kw=&hsa\\_mt=&hsa\\_net=adwords&hsa\\_ver=3&gclid=Cj0KCQiAu](https://www.stxnext.com/blog/react-native-vs-flutter-comparison/?utm_term=&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=4807594886&hsa_cam=15491073565&hsa_grp=&hsa_ad=&hsa_src=x&hsa_tgt=&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gclid=Cj0KCQiAu) [Accessed 25 Jan. 2022].

Suranga, S. (2021). *You Don't Have to Compare Flutter and React Native Anymore*. [online] Medium. Available at: <https://betterprogramming.pub/you-dont-have-to-compare-flutter-and-react-native-anymore-15ddc4c1342a#:~:text=Flutter%20apps%20come%20with%20different> [Accessed 7 Feb. 2022].

Wikipedia. (2021). *Ringfort*. [online] Available at: <https://en.wikipedia.org/wiki/Ringfort#:~:text=Ringforts%2C%20ring%20forts%20or%20ring> [Accessed 3 Feb. 2022].