SeaSplash

A mobile application developed using Flutter SDK promoting safe swim locations.

Name: **John McDonald**

Student ID: **20006166**

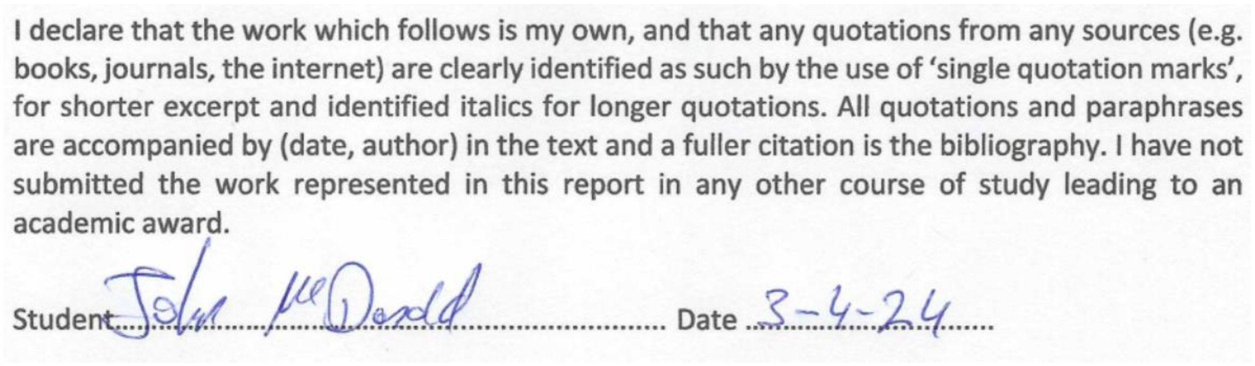
Course Leader: **Colm Dunphy**

Supervisor: **Catherine Fitzpatrick**

Course: **Higher Diploma in Computer Science**

Module: **Final Project**

Declaration of Authenticity


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 [illegible] Date 3-4-24

Acknowledgements

I wish to thank my project supervisor Catherine Fitzpatrick for her encouragement and guidance throughout this project. A huge thanks goes to course coordinators Colm Dunphy, Eamonn de Leastar and Pete Windle for all the work they do helping students get to the finish line. Also, to Laura McGibney and Joan Mangan for their constant support and encouragement. A very huge thank you to my wonderful family and especially my very understanding wife Sharon for pushing me to do this course in the first place and supporting me throughout the past two years.

Titles

Commercial: SeaSplash
Academic: Mobile App Promoting Safe Swim Locations for Swimmers

GitHub Link

https://github.com/jmcied/SeaSplash

Introduction

Sea swimming has become increasingly popular around Ireland in recent years. Although there are lots of places where you should never swim with dangerous currents. People don’t know if you are not from that area to avoid them. Also, you should never swim alone no matter what your swimming ability.
My project vision was to develop a mobile application named SeaSplash. The primary aim of SeaSplash is to provide users, especially those new to an area or on holidays, with a convenient tool to discover and share the best and safest sea swimming spots. The application will enable users to log their favorite sea swimming locations, incorporating images, and map coordinates. Additionally, real-time sea conditions will be displayed through integration with weather APIs. A community element will be incorporated allowing users to organize swim meetups and enjoy the beauty Irish coastlines has to offer.

Table of Contents

Introduction....2
Research, Analysis, Modeling + Planning....4
Features....4
Research & Analysis....5
Weather Apps....5
Tides Near Me....5
HighTide....6
UK & Ireland Beaches....6
Weather API....8
IRISH WEATHER API....8
OpenWeather....9
StormGlass....10
Frontend....11
Flutter....11
Backend....12
Firebase....12
Methodology & Design....13
User Story....13
Agile....15
Planning Sprints....15
Project Tracking....16
Color Theme....17
Technology....18
Programming Language....18
Integrated Development Environment....19
State Management....20
File Structure....20
Implementation....22
Critical Self-Review....31
Bibliography....32

Research, Analysis, Modeling + Planning

Features

The main features I hope to implement are the following -

- User Profiles: Users can create profiles to log and manage their favorite sea swimming locations.
- Location Logging: Users can add new sea swimming locations, complete with detailed descriptions, images, and precise map coordinates.
- Weather Integration: The application will pull in weather data through APIs to provide current sea conditions for each location.
- Privacy Settings: Users can choose to make their swimming locations either public or private based on their preferences.
- Favorites: Users can save and organize their favorite sea swimming spots for quick and easy access.
- Community Feature: A community-driven feature will allow users to share their public sea swimming locations, fostering a sense of community, and facilitating the organization of swim meetups.

Research & Analysis

From my initial research, there are some mobile apps available currently on Google Play Store that are mainly focused on weather conditions like tides and waves, but none offer any community features.

Weather Apps

Tides Near Me

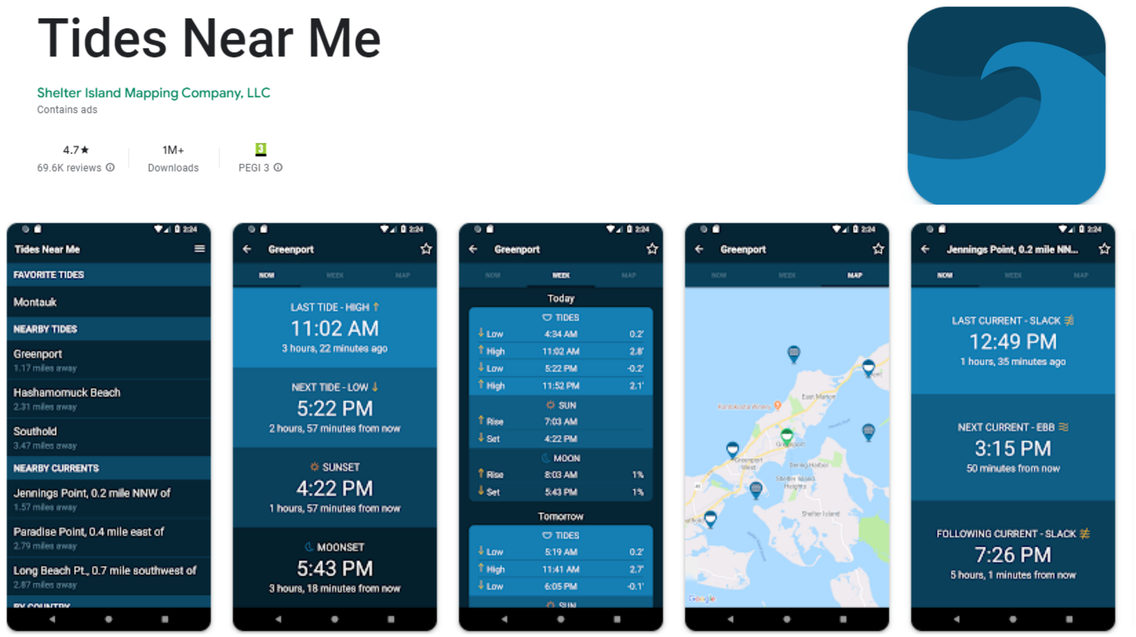


Figure 1. ‘Tides Near Me’ (Google, 2024)

HighTide

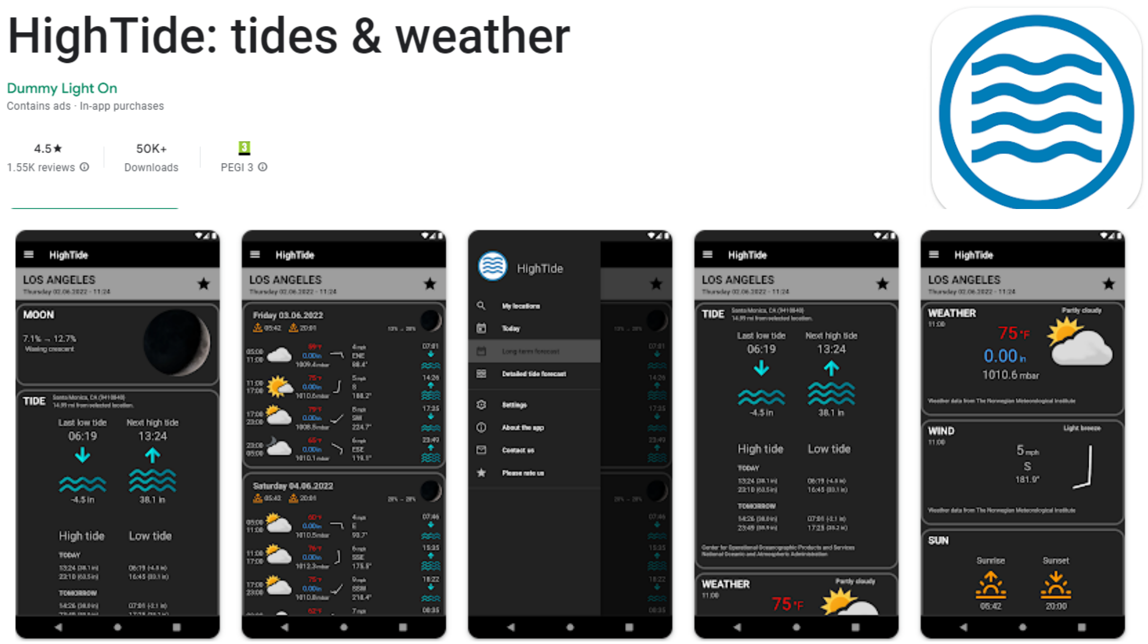


Figure 2. ‘HighTide’ (Google, 2024)

UK & Ireland Beaches

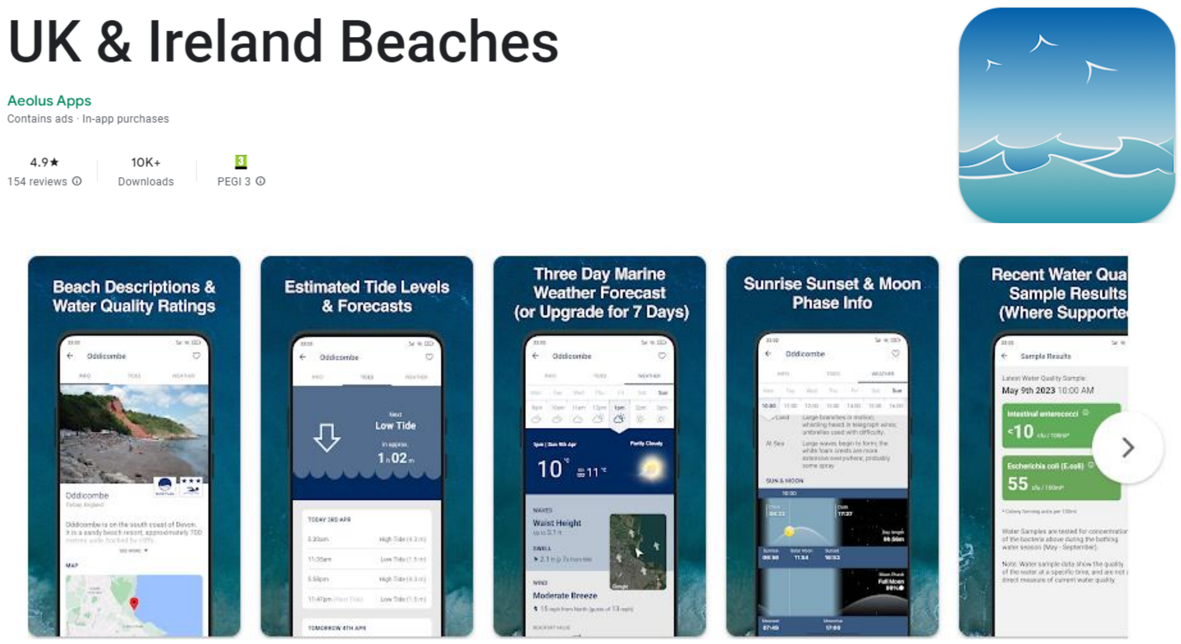

UK & Ireland Beaches
Aeolus Apps
Contains ads · In-app purchases
4.9★
154 reviews
10K+
Downloads
PEGI 3
Beach Descriptions & Water Quality Ratings
Estimated Tide Levels & Forecasts
Three Day Marine Weather Forecast (or Upgrade for 7 Days)
Sunrise Sunset & Moon Phase Info
Recent Water Qua Sample Results (Where Supporte
Oddicombe
Next Low Tide

Figure 3. ‘UK & Ireland Beaches’ (Google, 2024)

	Tidal Info	Images	Map	Community Chat	Beach Safety
Tides Near Me	✓	✗	✗	✗	✗
High Tides	✓	✗	✗	✗	✗
UK & Ireland Beaches	✓	✓	✓	✗	✗ Water Quality Only

Table 1 – Mobile App Comparison

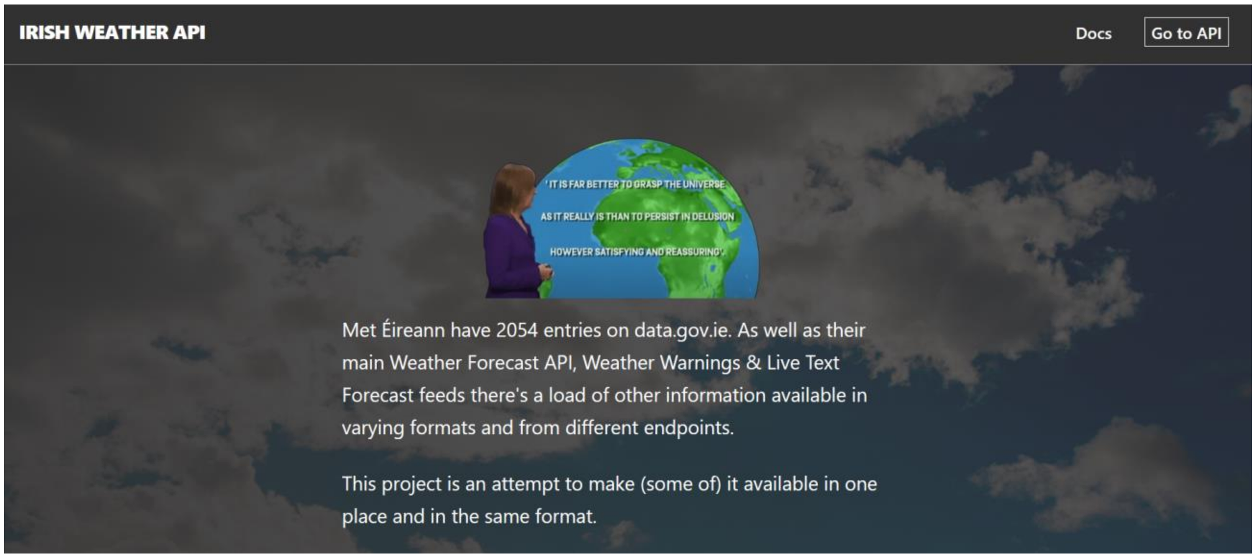
The most popular app is Tides Near Me which had 1 million+ downloads. All the apps contain general tidal information around main peninsulas, nothing specific to a smaller lesser-known beach. The App UK and Ireland Beaches is an aesthetically pleasing app with images, map locations and a nice water quality feature.
From my research I believe there is a gap in the market for an app that offers local knowledge about safe swim locations and a friendly engaging community connecting feature.

Weather API

Up to date weather information will be used within the application to give the user current weather conditions such as sea & air temperature, tidal conditions such as high & low tide times etc.

IRISH WEATHER API

‘*Met Éireann has 2054 entries on data.gov.ie. As well as their main Weather Forecast API, Weather Warnings & Live Text Forecast feeds there's a load of other information available in varying formats and from different endpoints.*’ (MetEireann, 2024)


IRISH WEATHER API
Docs
Go to API
‘IT IS FAR BETTER TO GRASP THE UNIVERSE
AS IT REALLY IS THAN TO PERSIST IN DELUSION
HOWEVER SATISFYING AND REASSURING’.

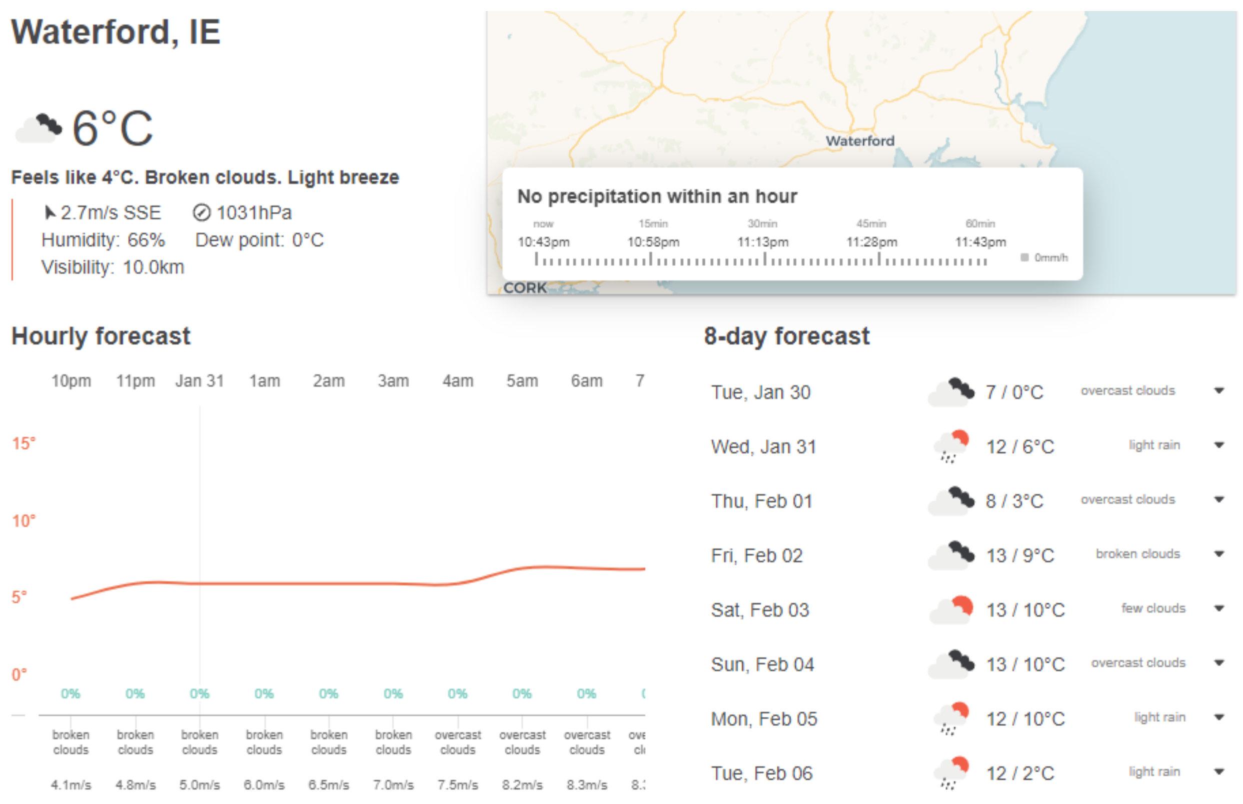
Met Éireann have 2054 entries on data.gov.ie. As well as their main Weather Forecast API, Weather Warnings & Live Text Forecast feeds there's a load of other information available in varying formats and from different endpoints.

This project is an attempt to make (some of) it available in one place and in the same format.

Figure 4 ‘Irish Weather API’ (MetEireann, 2024)

OpenWeather

'Access current weather data for any location. We collect and process weather data from different sources such as global and local weather models, satellites, radars and a vast network of weather stations. JSON, XML, and HTML formats. Included in both free and paid subscriptions.' (openweather, 2024)



Waterford, IE

6°C

Feels like 4°C. Broken clouds. Light breeze

2.7m/s SSE   1031hPa
Humidity: 66%   Dew point: 0°C
Visibility: 10.0km

No precipitation within an hour

now	15min	30min	45min	60min
10:43pm	10:58pm	11:13pm	11:28pm	11:43pm

0mm/h

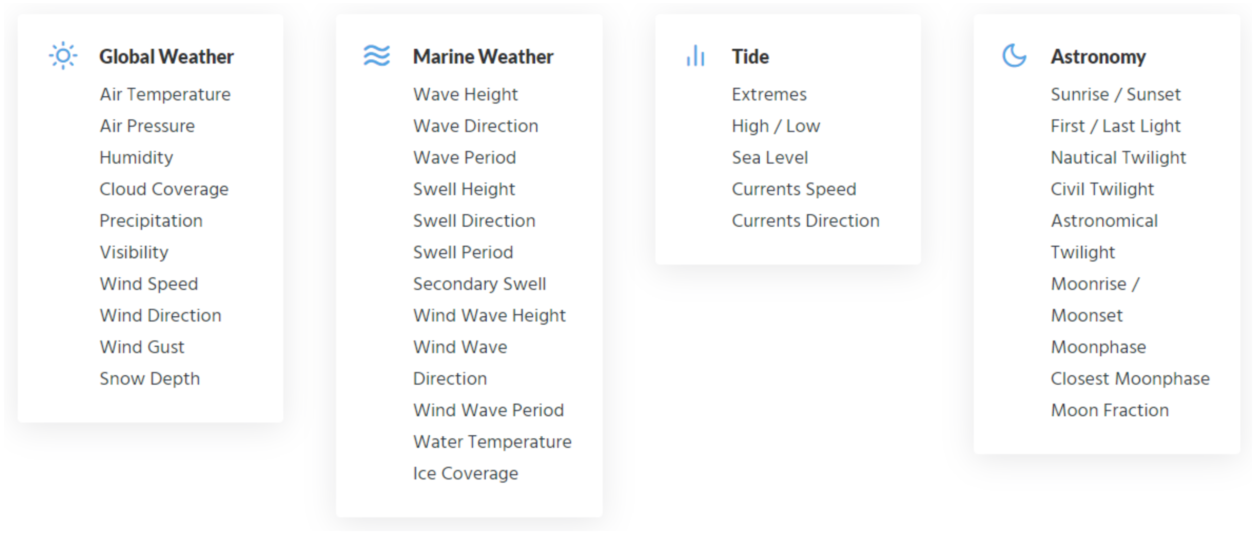
8-day forecast

Tue, Jan 30	7 / 0°C	overcast clouds
Wed, Jan 31	12 / 6°C	light rain
Thu, Feb 01	8 / 3°C	overcast clouds
Fri, Feb 02	13 / 9°C	broken clouds
Sat, Feb 03	13 / 10°C	few clouds
Sun, Feb 04	13 / 10°C	overcast clouds
Mon, Feb 05	12 / 10°C	light rain
Tue, Feb 06	12 / 2°C	light rain

Figure 5. 'OpenWeather' (openweather, 2024)

StormGlass

‘*Weather forecasts & historical data from the world’s most trusted meteorological institutions in one single API*.’ (stormglass.io, 2024)



Global Weather
- Air Temperature
- Air Pressure
- Humidity
- Cloud Coverage
- Precipitation
- Visibility
- Wind Speed
- Wind Direction
- Wind Gust
- Snow Depth

Marine Weather
- Wave Height
- Wave Direction
- Wave Period
- Swell Height
- Swell Direction
- Swell Period
- Secondary Swell
- Wind Wave Height
- Wind Wave Direction
- Wind Wave Period
- Water Temperature
- Ice Coverage

Tide
- Extremes
- High / Low
- Sea Level
- Currents Speed
- Currents Direction

Astronomy
- Sunrise / Sunset
- First / Last Light
- Nautical Twilight
- Civil Twilight
- Astronomical Twilight
- Moonrise / Moonset
- Moonphase
- Closest Moonphase
- Moon Fraction

Figure 6. ‘StormGlass.io’ (stormglass.io, 2024)

	Air temperature	Water Temperature	Tides	Wave Height
Irish Weather API	✓	✗	✗	✗
OpenWeather API	✓	✗	✗	✗
StormGlass API	✓	✓	✓	✓

Table 2 – Weather API Comparison

Irish Weather lists most of the Met Éireann open data. However, results are very limited and don’t contain information regarding tidal conditions.
OpenWeather gives a good detail of the weather, however no info on tide conditions.
StormGlass gives the best overall comprehensive weather conditions including sea conditions. Both OpenWeather and StormGlass offer a free basic account that allows a limited number of requests to be made each day. A paid version would have to be taken into consideration for more intensive testing and development and for future application development into a commercially viable solution.

Frontend

Flutter

Having gotten a feel for Android Development in semester 4 where we used Kotlin. I researched other programming languages associated with Mobile Application Development to further expand my knowledge. It was here that I discovered Dart which uses the Flutter SDK framework. Flutter, developed by Google, is an open-source framework that crafts appealing, cross-platform applications using a single codebase, allowing deployment to mobile, web and desktop. This would ideally suit me as my app was focused on the android and iOS user market.

- Using the same codebase allows simultaneous development for both android and iOS applications, instead of writing code independently using languages such as Kotlin and Swift.
- Flutter allows the developer to be more productive by building apps quickly with Hot Reload, updating code and seeing the changes almost instantly.
- Every pixel can be managed to produce personalized, adaptable designs to suit any screen size.
- Dart is a descriptive programming language which allows for fast and efficient development.
- Integrates easily with Firebase and Google Maps.

For this project I will be focused on the development of an Android app since I already have a windows laptop and android smartphone. To test iOS applications, I would need a macOS device.

Flutter
Develop for Mobile, Web and Desktop
Native Mobile Apps:
Loved and used in production by thousands of Developers, Startups and Enterprises
Flutter for Web:
In Beta now, but has strong support from the developer community
Desktop Apps:
MacOS Support: In Alpha
Windows and Linux: Under Development

Figure 7. ‘Flutter Framework’ (Bing, 2024)

Backend

Firebase

For the backend storage I will be using Firebase. Firebase integrates and scales very easily with any android application. It is a comprehensive platform developed by Google that provides a wide range of tools and services to help developers build, improve, and grow their apps.

- Firebase offers a real-time NoSQL database that allows users to sync data across devices in real-time without the need for any extra server code.
- Firebase provides authentication solutions, including email/password authentication and social logins such as Google.
- Firebase allows for the storage of user generated content such as images, videos, and files.
- Firebase Cloud Messaging is a cross-platform messaging service that reliably allows you to send instant messages at no cost.

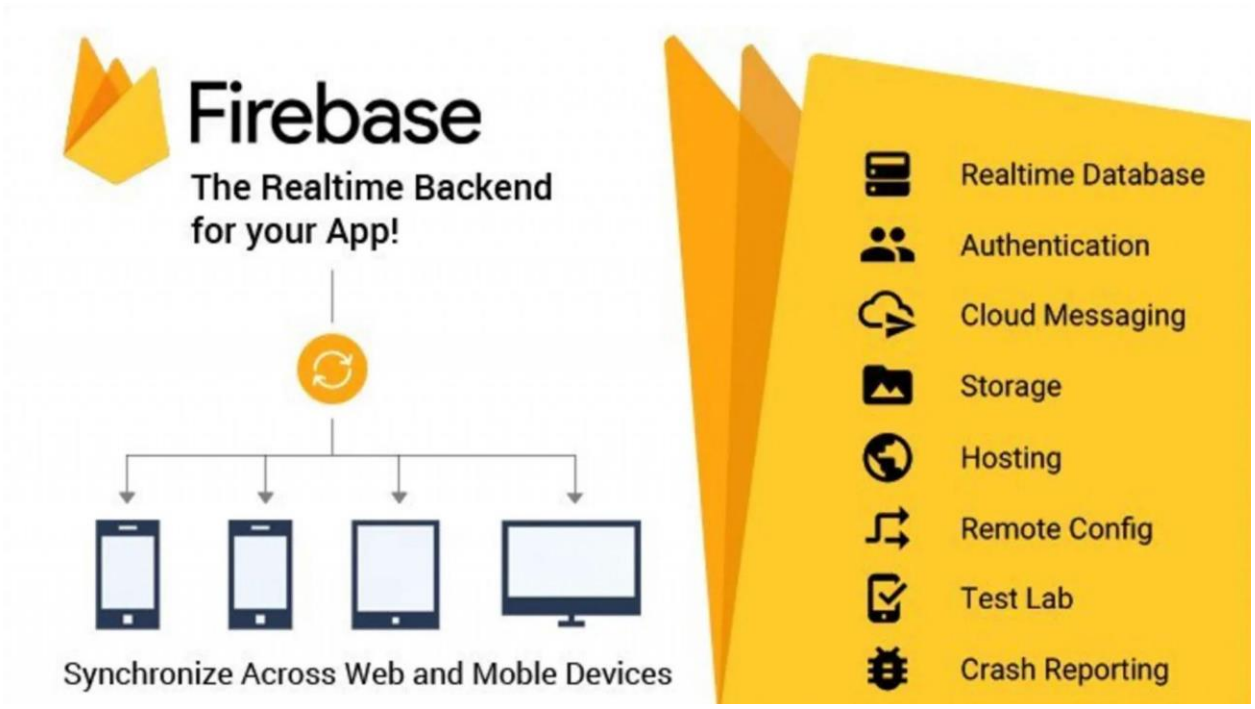


Figure 8 ‘Firebase’ (Bing, 2024)

Methodology & Design

User Story

There will be two main users of the SeaSplash app.

- New User

Users who are new to the area and are looking for a safe swimming spot. They will be able to sign up and login and view a list of swimming locations close to them. Each swim location will have information such as title, images, google map's location and tidal information. They will also the able to see if there are any planned swim meetups with other like-minded swimmers.

- Casual User

Local users who frequently go swimming and are knowledgeable of the area. These users will be able to sign up and login and view a list of swimming locations. These types of users might wish to add lesser-known swimming spots to the list, they can choose to make these locations public or private. Swimming locations can be saved as favorites. Casual users can decide to organize a swim meetup on a particular day & time.

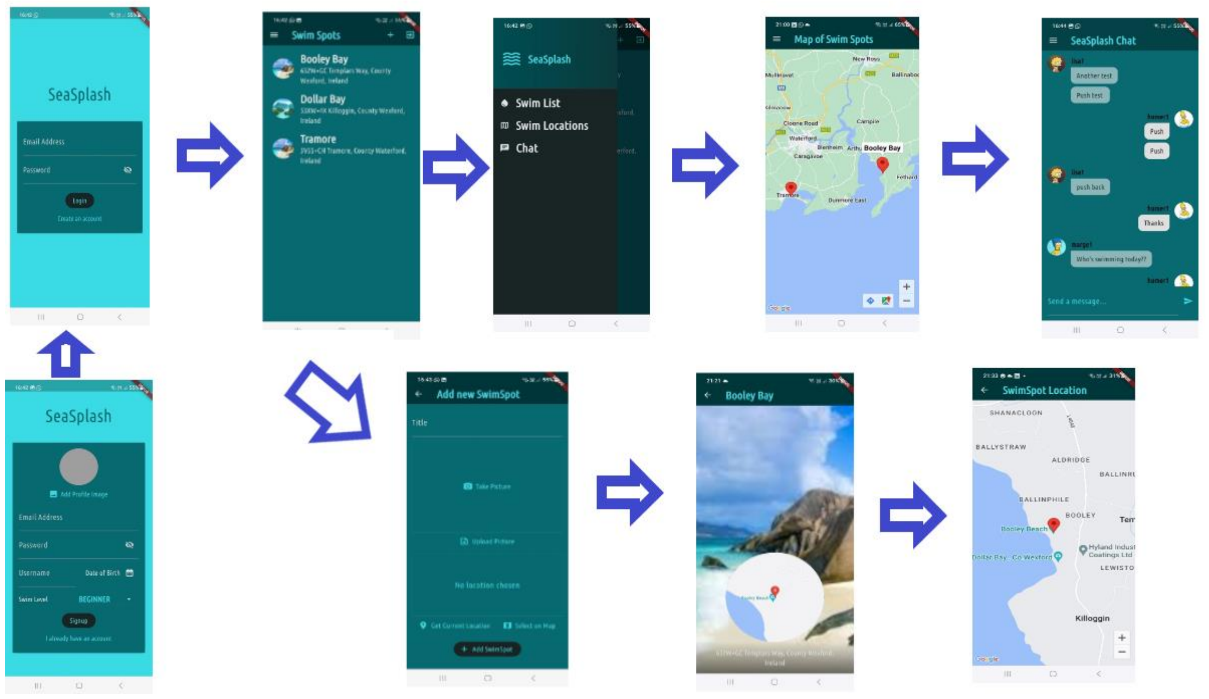


Figure 25. Screen Flow Diagram.

Figure 26 below gives a broad high level overview of the navigation throughout seaSplash application.

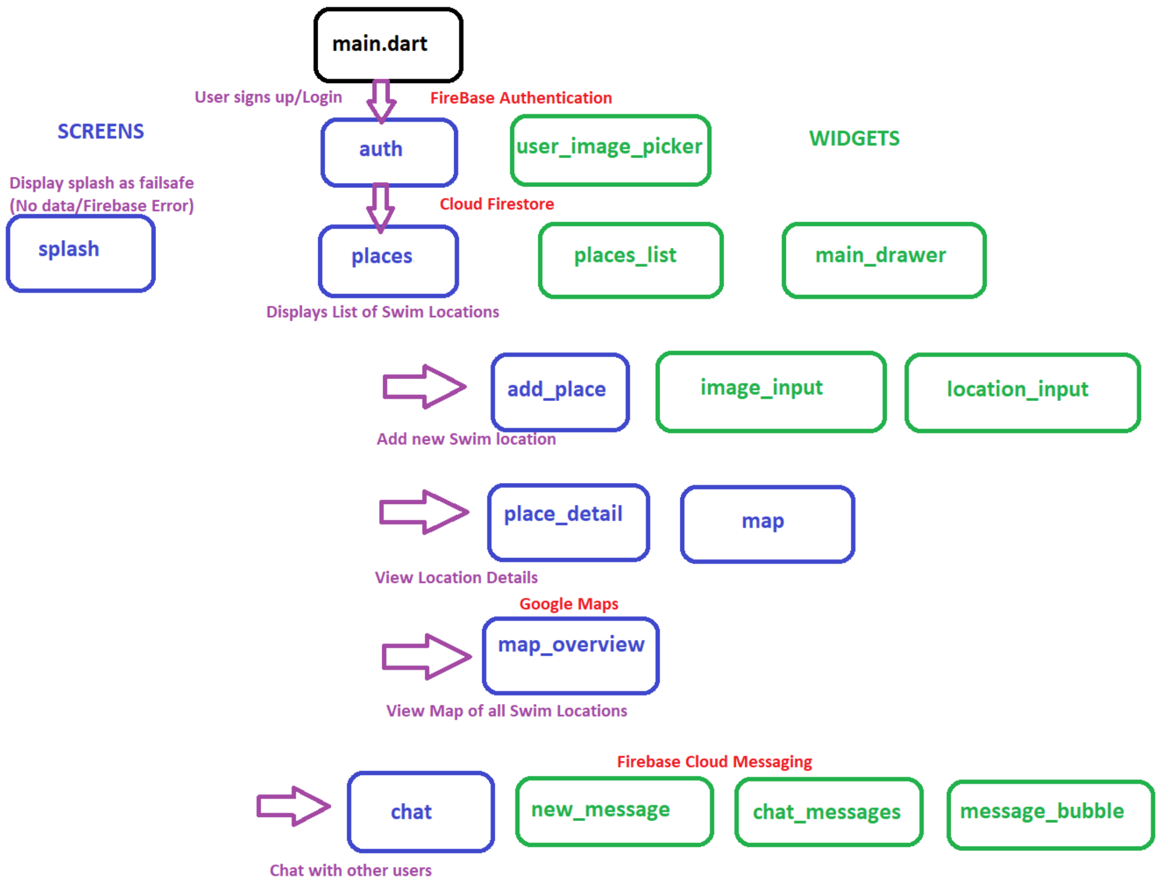


Figure 26. App Overview Flow Diagram

Agile

The Agile methodology is a project management framework that divides the project into multiple phases, often referred to as sprints.
It is an approach where teams reflect and review their progress after each sprint, adjusting their strategy accordingly for the next sprint.
I chose this method because this was how the course was structured throughout my studies and would allow me to break up the project features into sprints, where a single feature could be focused on and developed during a particular sprint.

Since GitHub was used as the project's version control, GitHub projects were used to keep account of the project's tracking and progression. This was convenient to keep everything in the same environment. The project's landing page was also deployed using GitHub pages.

Planning Sprints

Project Setup (Week 1-2):
- Set up the development environment.
- Create basic project structure.

User Profiles and Location Logging (Week 3-4):
- Implement user profile creation and management.
- Enable users to log new sea swimming locations.

Weather Integration (Week 5-6):
- Integrate weather APIs to display real-time sea conditions.

Privacy Settings and Favorites (Week 7-8):
- Implement privacy settings for swimming locations.
- Enable users to save and organize their favorite locations.

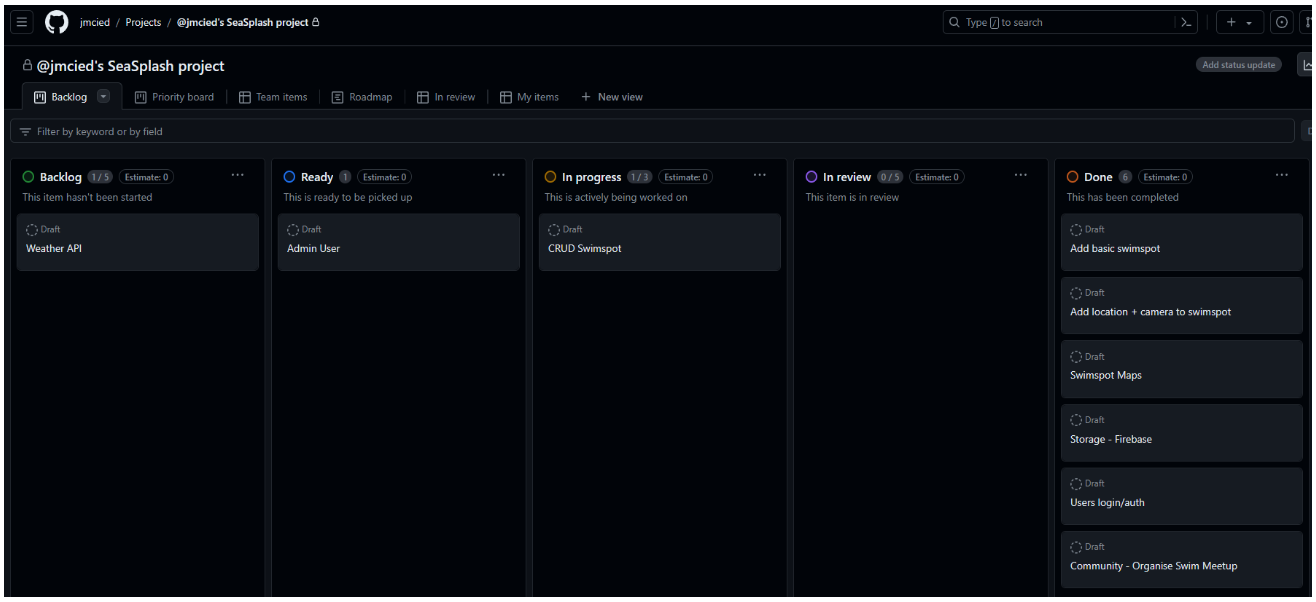
Community Feature (Week 9-10):
- Implement the community-driven feature for sharing sea swimming locations.
- Facilitate the organization of swim meetups.

Testing and Debugging (Week 11):
- Conduct thorough testing to ensure the application's functionality and reliability.
- Address any identified bugs or issues.

Documentation and Submission (Week 12):
- Compile comprehensive documentation for the project.
- Prepare for the final submission.

Project Tracking

GitHub provides project boards which allow the visualization of tasks and can be organized into columns filtered by statuses such as Ready, In Progress and Done.


jmcied / Projects / @jmcied's SeaSplash project
Type / to search
@jmcied's SeaSplash project
Add status update
Backlog
Priority board
Team items
Roadmap
In review
My items
New view
Filter by keyword or by field

Backlog 1 / 5 Estimate: 0
This item hasn't been started
Draft
Weather API

Ready 1 Estimate: 0
This is ready to be picked up
Draft
Admin User

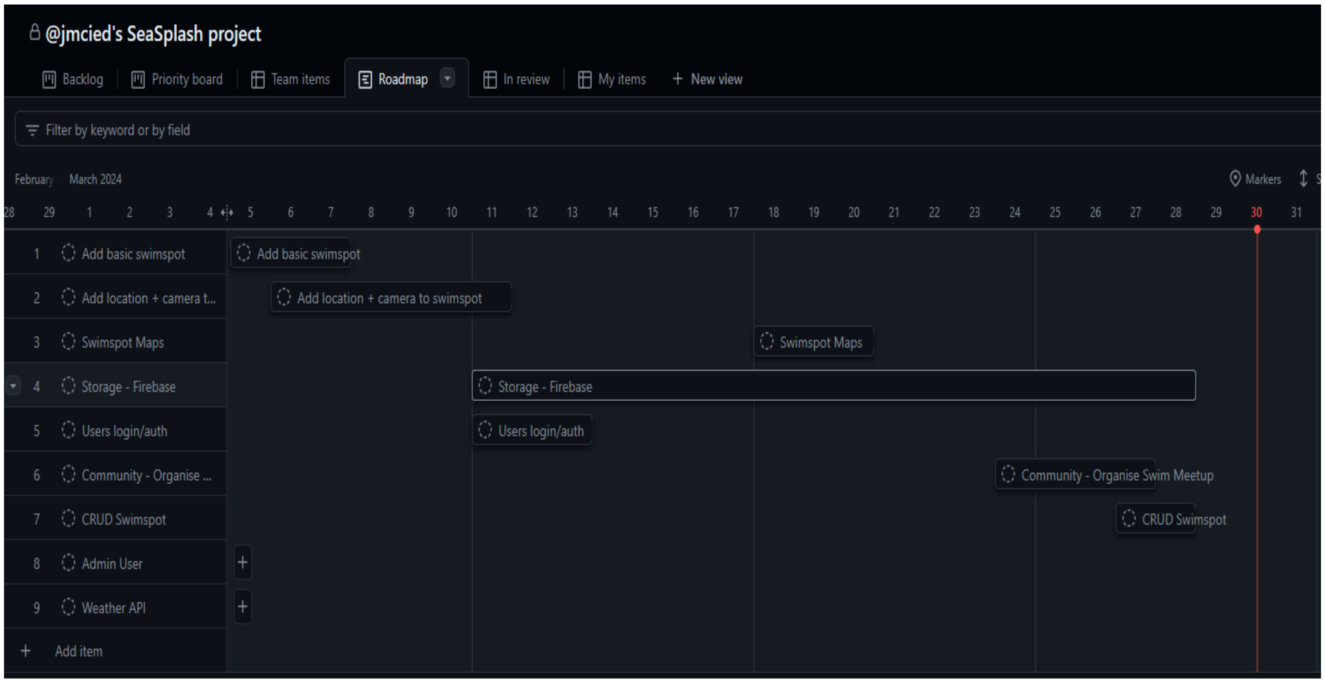
In progress 1 / 3 Estimate: 0
This is actively being worked on
Draft
CRUD Swimspot

In review 0 / 5 Estimate: 0
This item is in review

Done 6 Estimate: 0
This has been completed
Draft
Add basic swimspot
Draft
Add location + camera to swimspot
Draft
Swimspot Maps
Draft
Storage - Firebase
Draft
Users login/auth
Draft
Community - Organise Swim Meetup

Figure 9. GitHub Project

A roadmap was created to keep track of task duration.



1 Add basic swimspot — Add basic swimspot
2 Add location + camera t... — Add location + camera to swimspot
3 Swimspot Maps — Swimspot Maps
4 Storage - Firebase — Storage - Firebase
5 Users login/auth — Users login/auth
6 Community - Organise ... — Community - Organise Swim Meetup
7 CRUD Swimspot — CRUD Swimspot
8 Admin User
9 Weather API

Figure 10. GitHub Roadmap

Color Theme

Creating visually appealing applications is very important in mobile development. I wanted a clean looking app with little clutter whose color theme was inspired by the sea and was trendy and up to date. Upon researching color trends, I found that ‘*For 2025, trend forecasting company WGSN and Coloro named* A.I. *Aqua, a tech-inspired shade of blue, their color of the year for 2025.’* (WebFX, 2024)

This shade was ideally suited to my sea swimming inspired app as it paired perfectly with the blue green shades of ocean water and reminded me of being by the sea. I wanted to incorporate a theme from the start instead of leaving it to the end as an afterthought. These shades were used throughout the application and incorporated into backgrounds, text colors, app bars and navigation drawers.

Technology

Utilizing Flutter as the front-end SDK platform and Firebase as the Backend-as-a-Service (BaaS) platform is a powerful combination. Since both are supported by Google, they offer seamless integration possibilities and a host of documentation and tutorials for support. Flutter applications can easily connect to Firebase services using FlutterFire, a set of plugins that provide access to Firebase APIs. This allows integrating features such as real-time data synchronization, user authentication, cloud storage and push messaging with minimal configuration. Flutter also integrates smoothly with Google Maps, enhancing apps with geolocation and mapping abilities. In Flutter layouts/screens are built using widgets which inherit properties from their parent. Widgets can be nested within other widgets.

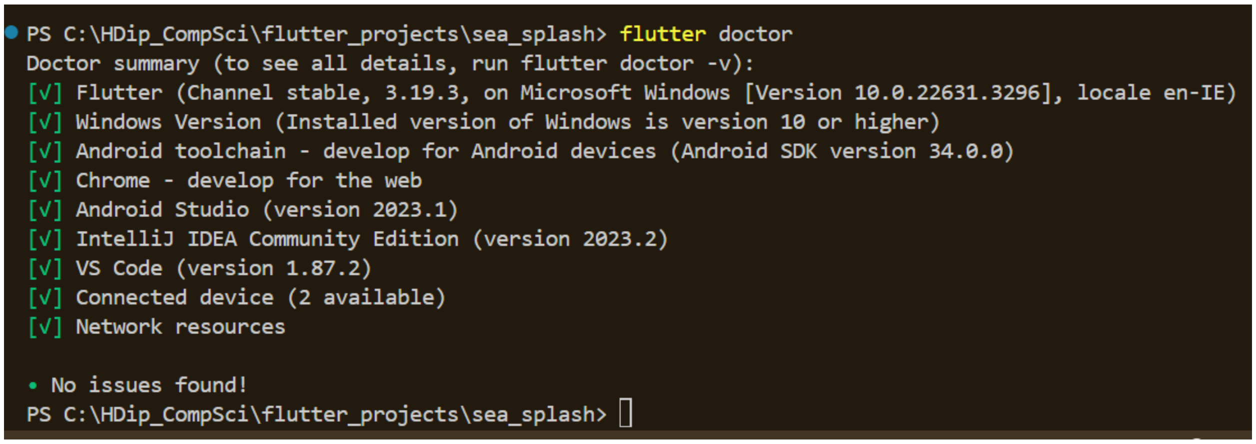
Programming Language

The language used for flutter development is Dart. It is an open-source programming language developed by Google. It is widely used in the development of mobile and web applications. In order to get a proper understanding, extensive research and study was achieved through the "Flutter documentation" (Flutter, 2024) and also through the completion of Udemy course '*Flutter & Dart - The Complete Guide [2024 Edition]*' (Schwarzmüller, 2024)

Integrated Development Environment

For this project I chose to use Visual Studio Code as the IDE. I found it to be a very nice enjoyable environment to work in from previous assignments. It is also recommended by the flutter team '*The Flutter team recommends installing Visual Studio Code 1.77 or later and the Flutter extension for VS Code. This combination simplifies installing the Flutter SDK*.' (Flutter, 2024)

VS code integrates easily with flutter through some simple configuration, installing the correct packages and setting path variables. Once all these are configured the command flutter doctor can be run within the terminal producing an output shown below verifying configuration.



```
PS C:\HDip_CompSci\flutter_projects\sea_splash> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, 3.19.3, on Microsoft Windows [Version 10.0.22631.3296], locale en-IE)
[√] Windows Version (Installed version of Windows is version 10 or higher)
[√] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[√] Chrome - develop for the web
[√] Android Studio (version 2023.1)
[√] IntelliJ IDEA Community Edition (version 2023.2)
[√] VS Code (version 1.87.2)
[√] Connected device (2 available)
[√] Network resources

• No issues found!
PS C:\HDip_CompSci\flutter_projects\sea_splash>
```

Figure 11. Flutter doctor terminal output.

By using flutter in Visual Studio code, errors are highlighted as you type which speeds up development and reduces errors.

State Management

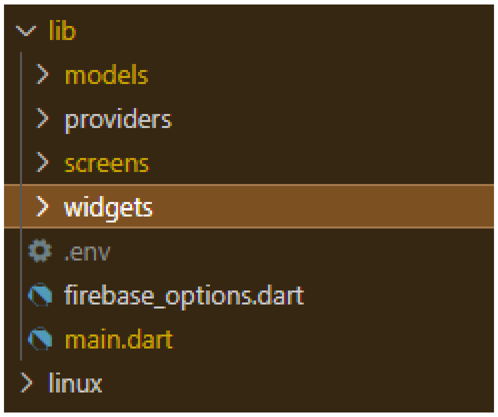
Riverpod is a reactive caching framework for Flutter and will be used for binding data and managing state changes.

'A reactive caching and data-binding framework.
Riverpod makes working with asynchronous code a breeze by:

- *handling errors/loading states by default. No need to manually catch errors.*
- *natively supporting advanced scenarios, such as pull-to-refresh.*
- *separating the logic from your UI*
- *ensuring your code is testable, scalable, and reusable*' (Flutter, 2024)

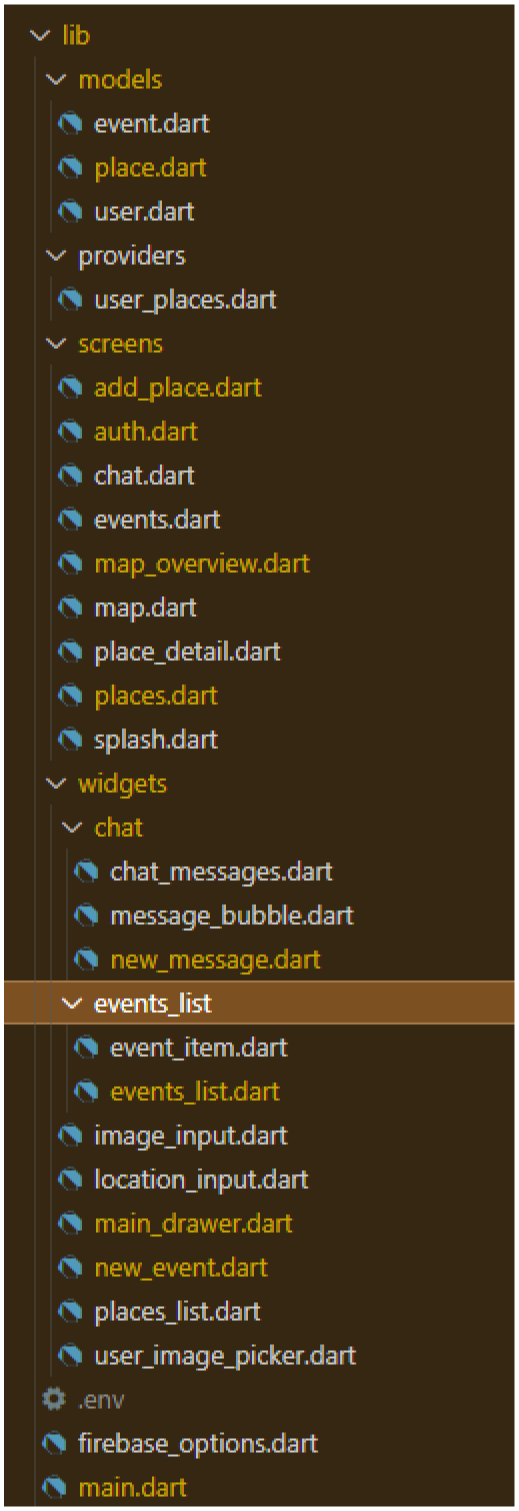
File Structure

A simple layer-first approach was adopted, whereby all files were separated into folders related to app layers such as models, providers, screens, widgets.



```
lib
  models
  providers
  screens
  widgets
  .env
  firebase_options.dart
  main.dart
linux
```

Figure 12. File Structure



```
lib
  models
    event.dart
    place.dart
    user.dart
  providers
    user_places.dart
  screens
    add_place.dart
    auth.dart
    chat.dart
    events.dart
    map_overview.dart
    map.dart
    place_detail.dart
    places.dart
    splash.dart
  widgets
    chat
      chat_messages.dart
      message_bubble.dart
      new_message.dart
    events_list
      event_item.dart
      events_list.dart
    image_input.dart
    location_input.dart
    main_drawer.dart
    new_event.dart
    places_list.dart
    user_image_picker.dart
  .env
  firebase_options.dart
  main.dart
```

Figure 13. File structure breakdown.

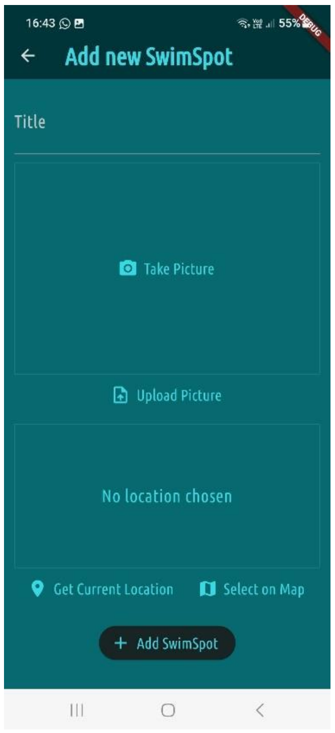
Implementation

First thing that was done to get familiar with flutter and dart was to learn with help from a Udemy course, '*Flutter & Dart - The Complete Guide [2024 Edition]'* (Schwarzmüller, 2024).
This course was chosen as it had the highest rating and the highest user downloads, as it was the 2024 edition it would be the most up to date. Learning flutter had a steep learning curve, and it took longer than planned to become familiar with it, however it would be crucial to any project to have a good understanding before development could commence.

To get started, basic screens were created to add a place which initially just took in a title and another to display a list of places. Next a place details screen was created to display details about individual places. Riverpod was used for handling state and data binding. A place model was created initially just to handle a title input of type string and a unique id assignment for each place.

I wanted to make use of the device's native features like the camera and user location which would be beneficial to a mobile application. A section from the Udemy course mentioned earlier '*Using Native Device Features*' (Udemy, 2024) was adapted to take pictures using the devices camera and add them to place model which was updated to an image of type File. This image would then be used as a background for a place details screen and in the places list screen. Uploading an image from a device's gallery was also implemented.

Flutter has many package plugins available which makes taking full advantage of devices features very easy. Like the camera, device location can be used. This was done by adding '*Flutter location package*' (pub.dev, 2024) and adapting '*Getting the User's Current Location*' (Udemy, 2024). This allowed retrieving of a user's location. Google Maps API was configured by enabling several APIs such as Places, Geolocation, Directions, Maps from Google Cloud. Initially a free account was created but I found some map features stopped working as was only rectified once I upgraded to a pay-as-you-use version. User location was added to the place model as a location class which was made up of latitude, longitude and a human readable address equivalent of latitude and longitude using Google Reverse Geocoding API. A preview of the location was displayed in the add place form. A manual option to create a location by dropping a pin was also added. The details screen was then further developed into displaying a location snapshot and an address. When snapshot is pressed a full map screen will be opened with help from '*Adding a Map Screen*' (Udemy, 2024), utilizing Google Maps full feature experience.



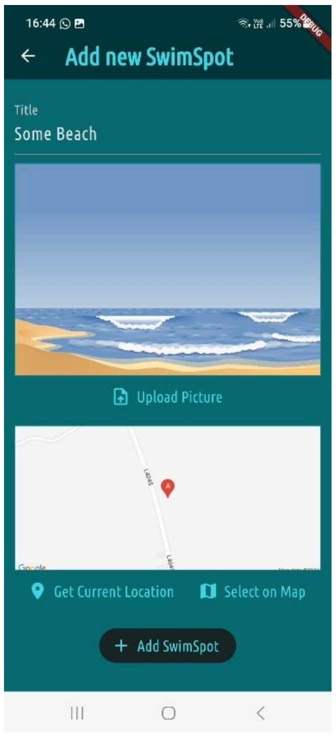


Figure 14. Add place form.

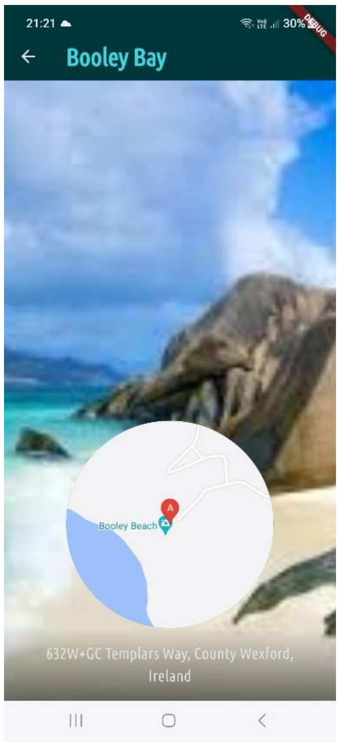


Figure 15. Place details screen.

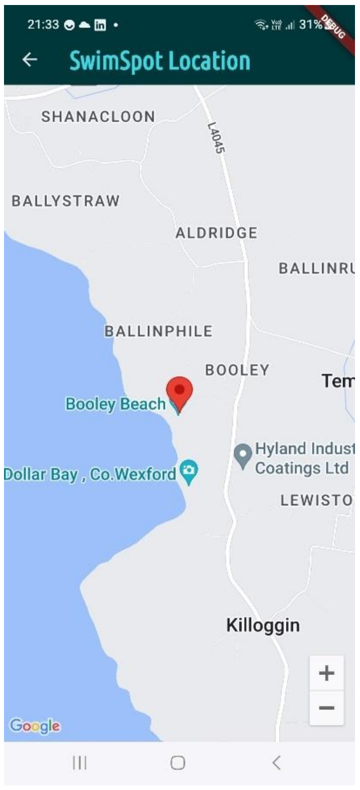


Figure 16. Place details Map Screen.

Sensitive API keys were protected from being accidentally pushed to GitHub by adding .env file. ‘*How to add .env File in Flutter’* (geeksforgeeks, 2024)

At this stage in the project nothing was being saved when the app restarted so persistence had to be addressed. This was implemented by using devices local storage first. This was done through using another handy package from flutter ‘*sqflite*’ (pub.dev, 2024). This allowed added places to be stored on the device’s memory.

Authentication was tackled next. A single authentication screen was used where certain widgets were displayed depending on whether the user chose to create an account see Figure 17 or if they already had an account Figure 18. This is a nice feature of flutter where a single screen can be re-used for multiple functions by breaking it up into multiple widgets. Firebase Authentication was setup using an email password method ‘*App & Firebase Setup’* (Udemy, 2024)

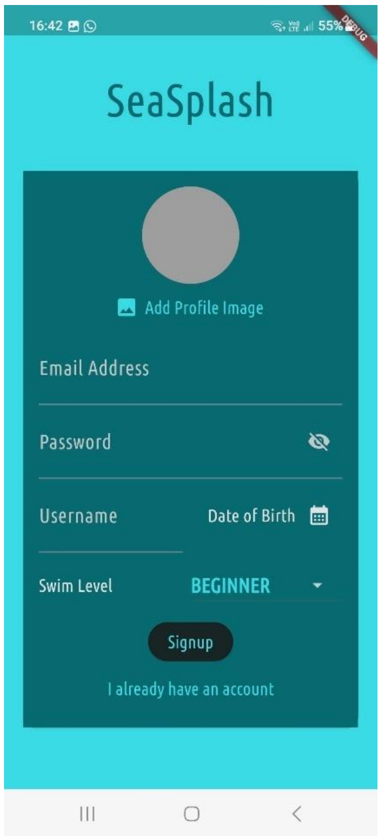


Figure 17. Create User

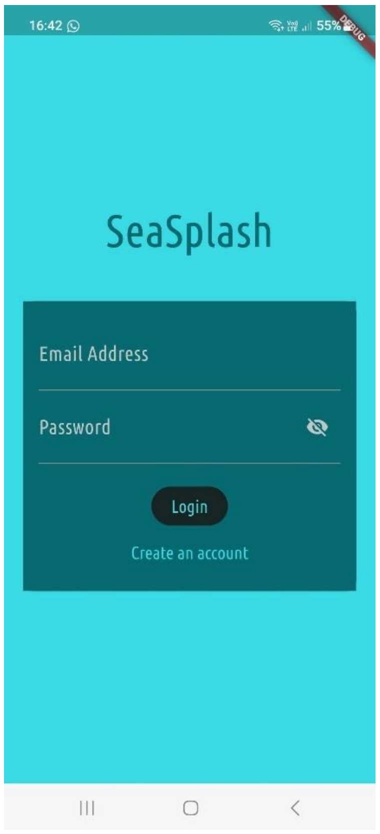
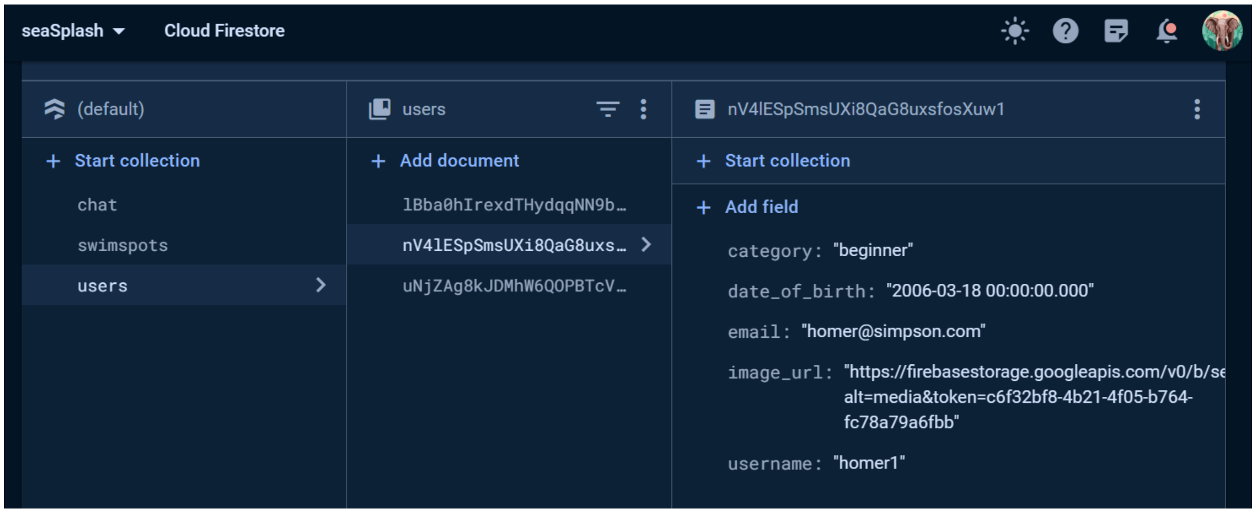


Figure 18. User Sign in

A new user is required to enter a profile picture which is stored using Firebase cloud storage, valid email, password, username, date of birth and swim level which is selected from a drop-down menu. Date of birth is selected from an easily customized date-picker widget from Flutter.
All fields are required to be entered when the user creates an account within a snackbar pop-up highlighting any missing fields. User information is stored with Firestore database, by creating a model shown below in Figure 19.



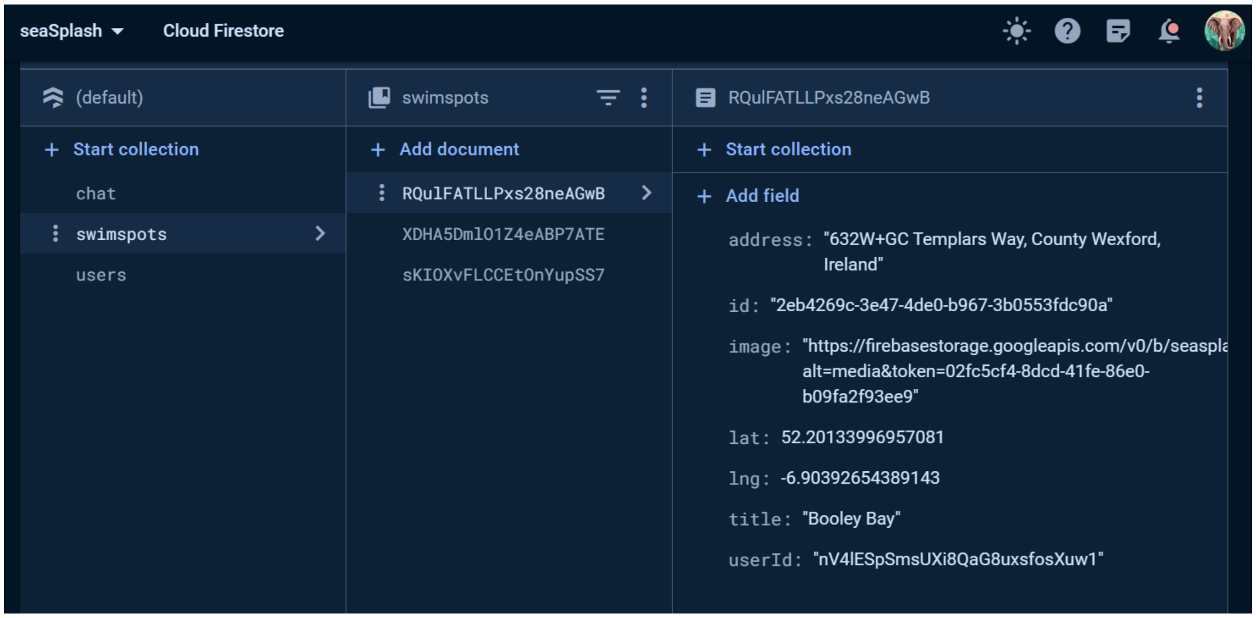
```
chat
swimspots
users

1Bba0hIrexdTHydqqNN9b…
nV41ESpSmsUXi8QaG8uxs…
uNjZAg8kJDMhW6QOPBTcV…

category: "beginner"
date_of_birth: "2006-03-18 00:00:00.000"
email: "homer@simpson.com"
image_url: "https://firebasestorage.googleapis.com/v0/b/se
           alt=media&token=c6f32bf8-4b21-4f05-b764-
           fc78a79a6fbb"
username: "homer1"
```

Figure 19. Users Cloud Firestore

A swim spots database was also created in Firestore see Figure 20, storing the added place from our application, storing address, id, image URL, latitude, longitude, title, and user ID.



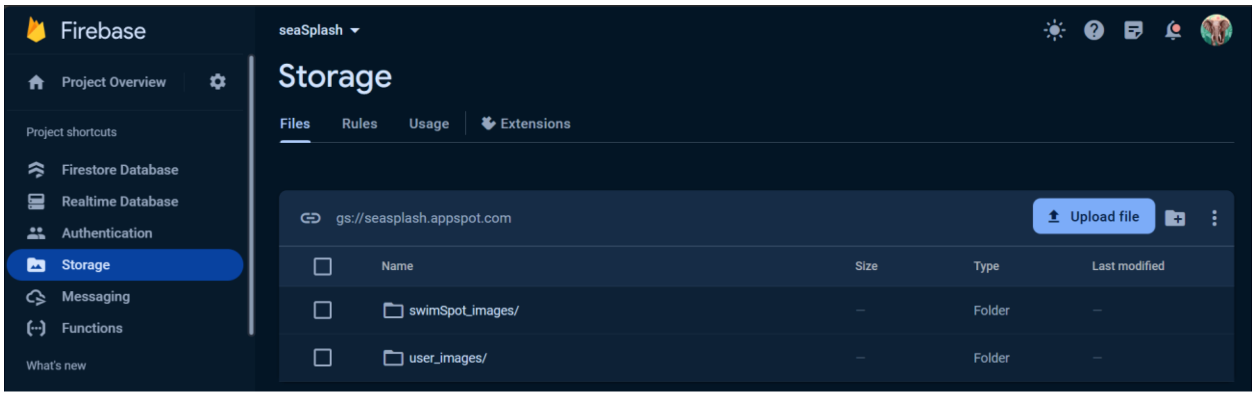
```
chat
swimspots
users

RQulFATLLPxs28neAGwB
XDHA5Dml01Z4eABP7ATE
sKIOXvFLCCEtOnYupSS7

address: "632W+GC Templars Way, County Wexford,
         Ireland"
id: "2eb4269c-3e47-4de0-b967-3b0553fdc90a"
image: "https://firebasestorage.googleapis.com/v0/b/seaspla
       alt=media&token=02fc5cf4-8dcd-41fe-86e0-
       b09fa2f93ee9"
lat: 52.20133996957081
lng: -6.90392654389143
title: "Booley Bay"
userId: "nV4lESpSmsUXi8QaG8uxsfosXuw1"
```

Figure 20. Swim Spots Cloud Firestore

The image URL from the swimspots database links to the cloud storage location of the image see figure 21, which stores swim spots images and user profile images.



gs://seasplash.appspot.com

Name	Size	Type	Last modified
swimSpot_images/	–	Folder	–
user_images/	–	Folder	–

Figure 21 Image Cloud Storage

A maps screen was created which displayed locations of all added swim spots, with the title of each swim spot displayed once tapped see Figure 21 below.

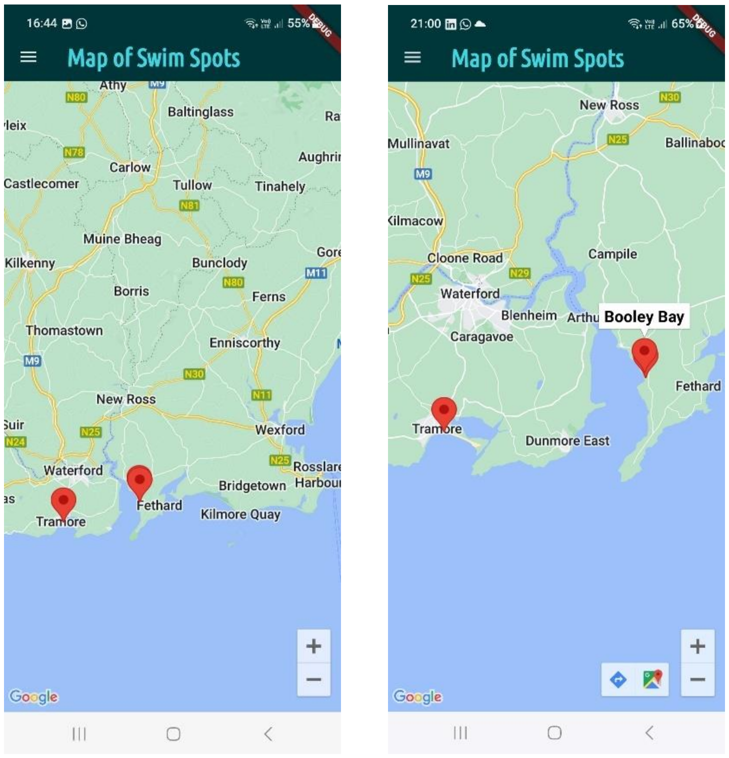


Figure 21. Map of Swim Spots

Next I focused on retrieving places from Firestore and displaying them in our app see Figure 22. Pressing on a swim spot would bring up a details screen for that swimspot see Figure 15. There was a bug that images weren't being retrieved from Firestore. After a lot of debugging I found that the way I was treating images had to be changed, instead of trying to display a file like I was doing when images were saved to local storage, I had to display a NetworkImage since I was now retriving a URL path in the form of a string to my image stored in Firebase cloud.

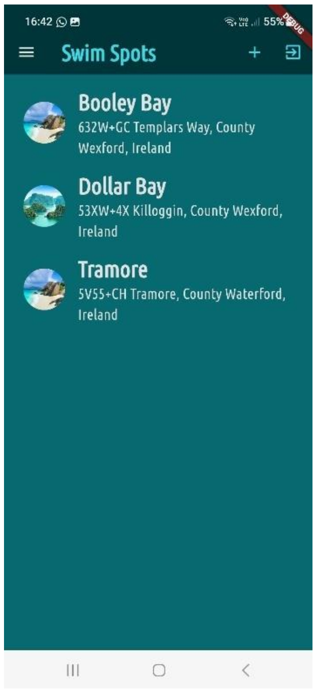

16:42
55%
DEBUG
Swim Spots
Booley Bay
632W+GC Templars Way, County Wexford, Ireland
Dollar Bay
53XW+4X Killoggin, County Wexford, Ireland
Tramore
5V55+CH Tramore, County Waterford, Ireland

Figure 22. Swim Spots List View

Since the number of screens had increased significantly, navigation needed to be improved. A side navbar was then created through using yet another widget from flutter, the drawer widget. This creates a complimentary user experience that allows smooth transitions to different screens Figure 23.

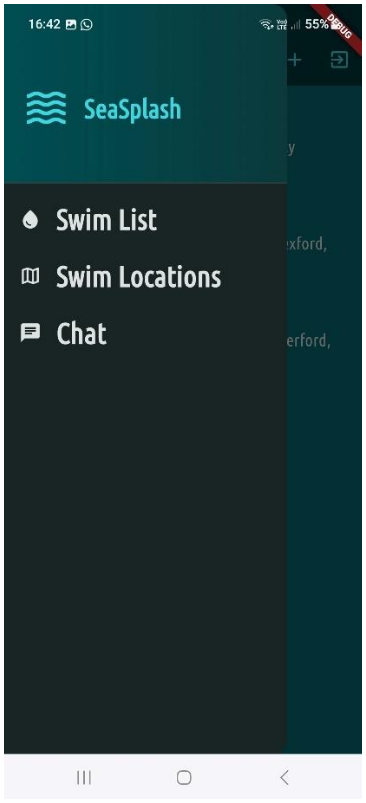
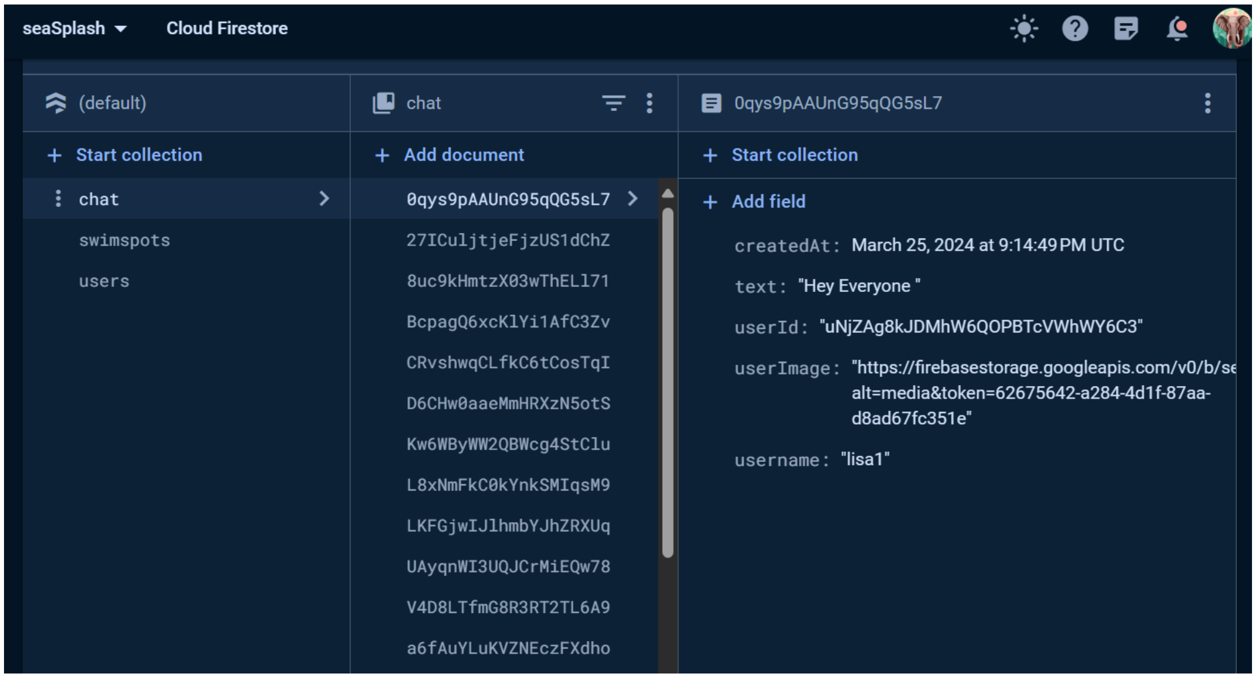


Figure 23. Side Drawer Navigation

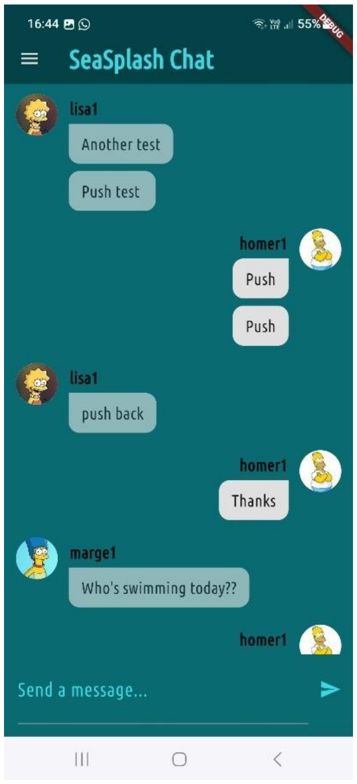
As I neared the project deadline and with a week left, I was at a crossroads. I still had two main features I needed to implement, Weather APIs that would display real-time weather conditions and create a community feature that would encourage users to engage with each other. I knew realistically I could only achieve completing one. Since from my initial research what I found missing from other similar style swimming applications was the community feature I decided to go with that. Also, I found that Weather APIs have become reliant on purchasing a paid version for successful development. My initial idea was to create an events type feature where users could see swim meetups that were happening at certain beaches at certain times and click that they were attending.

This proved to be troublesome approach and I think more complicated than it needed to be so instead I created a chat feature utilizing Firebase Cloud Messaging where users could send messages to each other and organize swim meetups. A cloud chat Firestore was created storing user ID, user image, username, text and created at fields see Figure 24. Firebase incorporates a push notification feature that allows message notifications to be sent to users of the app even if the app isn't being used see Figure 24. This feature was implemented using the module '*Building a Chat App*' (Udemy, 2024). The styling of the chat bubbles was taken from '*flutter-complete-guide-course-resources*' (Schwarzmüller, 2024)



```
createdAt: March 25, 2024 at 9:14:49 PM UTC
text: "Hey Everyone "
userId: "uNjZAg8kJDMhW6QOPBTcVWhWY6C3"
userImage: "https://firebasestorage.googleapis.com/v0/b/se
           alt=media&token=62675642-a284-4d1f-87aa-
           d8ad67fc351e"
username: "lisa1"
```

Figure 24. Chat Cloud Firestore



lisa1: Another test
lisa1: Push test
homer1: Push
homer1: Push
lisa1: push back
homer1: Thanks
marge1: Who's swimming today??
homer1

Figure 24. Chat screen.

Critical Self-Review

Overall, I enjoyed this project and found Mobile Application Development an enjoyable journey. I broaden my knowledge into a programming language that I had no experience before this project in using Dart and gained vast experience using the Flutter framework that I believe will become even more popular and be incorporated by universities in their education deliverance. I gained more experience using cloud storage through Firebase and have become comfortable utilizing APIs like Google Maps for projects.

Considering the time constraints with juggling working full-time, family life and studying, I was very pleased with developing a working application within 3 months. SeaSplash is both visually appealing and functional. The application makes use of the devices camera and location and incorporates modern technologies like cloud storage and Google Maps.

Future development I would like to further enhance user experience by bringing in weather conditions for each location, allowing the user to see tidal heights along with weather conditions. I would like to have an admin user verify swim spots and only allow swim location to be added once verified. I would also like to add additional information about each swimming spot like parking, lifeguard on duty, water quality. Given more time I would like to use Google login also for user authentication.

Time management was one of the main problems for this project. Looking back, I would have started learning Flutter and Dart a lot sooner so I could give more time to development. Ideally learning a new programming language and framework would be done before commencement of the project. Implementation was delayed until a good knowledge base was built and once started development flowed relatively smoothly.

Bibliography

Bing, 2024. *Bing Images Firebase.* [Online]
Available at: https://www.bing.com/images/search?q=firebase&form=HDRSC4&first=1
[Accessed 1 Febuary 2024].
Bing, 2024. *Bing Images Flutter Framework.* [Online]
Available at:
https://www.bing.com/images/search?view=detailV2&ccid=2%2F6H9iOv&id=8AC6E27EF24D40472E508AA46AA5A96476A8128C&thid=OIP.2_6H9iOvbDxd6SyFWO5rmQHaEK&mediaurl=https%3A%2F%2Fblog-cdn.domaincer.com%2Fblog%2Fwp-content%2Fuploads%2F2021%2F11%2F05113302%2FFlutter-F
[Accessed 1 Febuary 2024].
Flutter, 2024. *Flutter documentation.* [Online]
Available at: https://docs.flutter.dev
[Accessed 1 Febuary 2024].
Flutter, 2024. *flutter_riverpod 2.5.1.* [Online]
Available at: https://pub.dev/packages/flutter_riverpod
[Accessed 1 March 2024].
geeksforgeeks, 2024. *geeksforgeeks.* [Online]
Available at: https://www.geeksforgeeks.org/how-to-add-env-file-in-flutter/
[Accessed 1 March 2024].
Google, 2024. *Google Play Store.* [Online]
Available at: https://play.google.com/store/apps/details?id=me.tidesnear.free
[Accessed 1 January 2024].
Google, 2024. *Google Play Store.* [Online]
Available at: https://play.google.com/store/apps/details?id=dummy.lighton.floogfjaere
[Accessed 1 January 2024].
Google, 2024. *Google Play Store.* [Online]
Available at: https://play.google.com/store/apps/details?id=com.gmail.edhesketh.ukbeaches
[Accessed 1 January 2024].
MetEireann, 2024. *IRISH WEATHER API.* [Online]
Available at: https://weather.apis.ie/
[Accessed 1 Febuary 2024].
openweather, 2024. *openweather.org.* [Online]
Available at: https://openweathermap.org/
[Accessed 1 January 2024].
pub.dev, 2024. *location 6.0.0.* [Online]
Available at: https://pub.dev/packages/location
[Accessed 1 March 2024].
pub.dev, 2024. *sqflite 2.3.2.* [Online]
Available at: https://pub.dev/packages/sqflite
[Accessed 1 March 2024].
Schwarzmüller, M., 2024. *academind github.* [Online]
Available at: https://github.com/academind/flutter-complete-guide-course-

resources/blob/main/Lecture%20Attachments/14%20Chat%20App/message_bubble.dart
[Accessed 1 March 2024].
Schwarzmüller, M., 2024. *Flutter & Dart - The Complete Guide [2024 Edition].* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37130436#overview
[Accessed 1 Febuary 2024].
stormglass.io, 2024. *Global Weather API.* [Online]
Available at: https://stormglass.io/
[Accessed 1 January 2024].
Udemy, 2024. *Adding Map Screen.* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37266982#overview
[Accessed 1 March 2024].
Udemy, 2024. *App & Firebase Setup.* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37736590#overview
[Accessed 1 March 2024].
Udemy, 2024. *Building a Chat App.* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37736584#overview
[Accessed 1 March 2024].
Udemy, 2024. *User's Current Location.* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37266946#overview
[Accessed 1 March 2024].
Udemy, 2024. *Using Native Device Features.* [Online]
Available at: https://www.udemy.com/course/learn-flutter-dart-to-build-ios-android-apps/learn/lecture/37214954#overview
[Accessed 1 March 2024].
WebFX, 2024. *10 Modern Web Design Trends for 2025.* [Online]
Available at: https://www.webfx.com/blog/web-design/modern-web-design/#:~:text=For%20a%20website%20to%20have,at%20Chobani's%20sleek%20mobile%20design
[Accessed 1 Febuary 2024].

This Ethics Checklist must be completed for all final year undergraduate, taught postgraduate and research projects in the School of Science and Computing.

View your response(s)

Respondent: **John Mc Donald** (Group: CM-HDIPCS) Submitted on: Saturday, 25 November 2023, 11:51 AM

Ethics Checklist for Undergraduate, Taught Postgraduate and Research Projects in the School of Science and Computing

All students in the School of Science and Computing who are either (1) in the final year of an undergraduate/BSc degree, or (2) on a taught postgraduate/MSc programme **must complete this Ethics Checklist before conducting their project** regardless of the project type or discipline. The Checklist should also be completed by anyone (whether staff member or student) conducting a **research project** (whether programmatic or not) within the School.

The purpose of this Ethics Checklist is to **identify projects that will require formal ethical approval** from the School Research Ethics Committee, or the SETU Research Ethics Committee, before they can proceed.

Students/applicants should note that this Ethics Checklist is a **formal declaration**, and great care must be taken to **answer all questions accurately**. Students should consult with their project supervisors/advisors regarding any aspects or questions that they are unsure of before completing and submitting the Ethics Checklist.

Students/applicants must **answer all questions** presented to them until the Checklist questionnaire is completed.

Feedback Report

No human experimentation issues (UG).

No animal experimentation issues (N/A).

No issues regarding the use of human tissues.

No animal tissue or biological fluids issues.

No ionising radiation issues.

No primary data collection issues (N/A).

No underage/vulnerable people issues (UG).

?

No issues regarding existing/secondary data use (N/A).

No controversial data issues.

No issues related to the collection of rare or protected plants.

No issues regarding the use of genetically modified (GM) plant material.

Instructions:

1. If the above feedback is **entirely** **green** then, based on your answers, there is **<u>no need to apply for ethical approval</u>** for your project.
2. If **any** part of the above feedback is **yellow/amber**, then there is at least one issue with your project that needs to be reviewed and **<u>you must apply for ethical approval</u>** to continue your project.
3. If **any** part of the above feedback is **red** then there is a serious ethical issue and **<u>you cannot continue your project</u>** as currently planned.

It is recommended that you print this Feedback Report to a PDF file for your records. You should also forward and discuss this Feedback Report PDF with your project supervisor. They will be able to advise if you have any further questions or if you need to apply for ethical approval.

1 * Are you a student on a **final year undergraduate** programme, a **taught postgraduate** programme, or are you conducting a **research project**?

[✓ Final Year Undergraduate]
Taught Postgraduate
Postgraduate Research Project
Other Research Project

2 * What is the **working title** of your project?

SeaSwim - WebApp/Android App for sea swimmers

3 * Who are the project **supervisors/advisors/principal investigators**?

Colm Dunphy / TBC

4 * Does your project involve **human experimentation**?

Yes [✓ No]

5 * Does your project involve **live animal experimentation**?

Yes [✓ No]

(6) * Is the planned animal experimentation limited to **non-invasive procedures only** (such as feeding, weighing, or taking naturally voided faecal or hair samples), and does **not** involve any invasive procedures (such as taking rectal faecal samples or blood) from live animals?

Yes No

7 * Does your project involve the use of **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses?

Yes [✓ No]

(8) * Do you intend to only use established **commercial human cell lines**, and no other **human** remains/cadavers/tissues/cells/biological fluids/embryos/foetuses in your project?

Yes No

9 * Does your project involve the use of **animal cells, tissues** or **biological fluids**?

Yes [✓ No]

(10) * Do you intend to only use (1) **established commercial animal cell lines**, or (2) **slaughterhouse-derived tissues/fluids**, or (3) **fluids collected as part of routine animal husbandry** (e.g. milk) and no other animal tissues or biological fluids in your project?

Yes No

11 * Does your project involve the **collection of rare or protected plants**?

Yes [✓ No]

12 * Does your project involve the generation or use of **genetically modified (GM) plant material**?

Yes [✓ No]

(13) * Do you agree to (1) only use **established genetically modified (GM) plant cell lines, seeds, or plant products** in your project, (2) **not generate new plant mutations** using chemical or other means, and (3) follow specified SETU **containment and use protocols** for GM plant materials at all times?

Yes No

14 * Does your project involve the use of **ionising radiation**? (e.g. use of gamma ray spectrometry)

Yes [✓ No]

(15) * Do you agree to carefully **follow the instructions** of the SETU designated **Radiation Protection Officer (RPO)**, and **adhere to all legal requirements** as set out in the Radiological Protection Act 1991 (Ionising Radiation) Regulations (2019), regarding the use of ionising radiation materials and equipment?

Yes No

16 * Does your project involve the **collection of any new (or primary) data** from **individual people or groups**?

Yes [✓ No]

(17) * Does your project involve the **collection of any new (or primary) individual or group data** that is **personally or uniquely identifying**? (e.g. data about people or organisations/companies/groups that could be used to identify those individuals or groups; data collection might take any form, including internet and social media data, etc.)

Yes No

(18) * Will you ensure that participants who you are collecting data from are provided with **fair warning** and must provide **explicit informed consent** for any data collected?

Yes No

(19) * Will you ensure that any project-related data collection, data storage, and data use is in **full compliance** with the **EU General Data Protection Regulation (GDPR)** and the **Data Protection Act (2018)**?

Yes No

(20) * Does any of the data that you intend to collect include **sensitive or private personal information** about individuals, or **commercially sensitive information** about organisations/companies/groups?

Yes No

21 * Does your project involve **persons under the age of 18 years** (i.e. minors), or **any vulnerable groups**? (e.g. prisoners, refugees, those in care, addiction service users, etc.)

Yes [✓ No]

22 * Does your project involve the use of **existing (or secondary) human data**? (i.e. data originally collected for another purpose)

Yes [✓ No]

(23) * Is the existing or secondary human data you intend to use either (1) **anonymous/non-personally identifying** and in the **public domain**, or (2) available with **explicit and specific informed consent or permission** for the data to be **legally** reused in the way you intend?

Yes No

(24) * Are any aspects of the primary/secondary data you intend to use for the project **controversial** in nature?

Yes No

25 * Before you submit the Ethics Checklist, you must **confirm all of the following**:

- [✓] I understand that the Ethics Checklist is a formal declaration.
- [✓] I have answered all questions on the Ethics Checklist carefully and truthfully.
- [✓] The supervisor/advisor (or principal investigator) for the project is present as the Ethics Checklist is being submitted, or they have given me explicit permission to submit it in their absence.
- [✓] I have had adequate ethics training and/or instruction prior to completing the Ethics Checklist.
- [✓] I understand, and agree to abide by, the general ethical principle of "do no harm" for this project.
- [✓] I will follow the instructions given in the Feedback Report.

26 * Authentication Code (ask your project supervisor/advisor for this code)

Enter Student Number: 20006166

Enter the Authentication Code below and click "Verify Code" Verify Code

Note: If an INVALID authentication code is used then this submission is NULL and VOID

4698