

Interim Report

Quartermaster

TU856

BSc in Computer Science

**Ethan Holohan**

**C20322553**

**Colette Kirwan**

School of Computer Science

Technological University, Dublin

**29/11/24**

Abstract

Airsoft is a popular and growing hobby with new people playing trying the hobby every year, however Airsoft is a complex game, with each site having its own sets of rules and games and as a result issues arise from players not fully knowing the rules and site layouts, leading to rule breaking and potential injury.

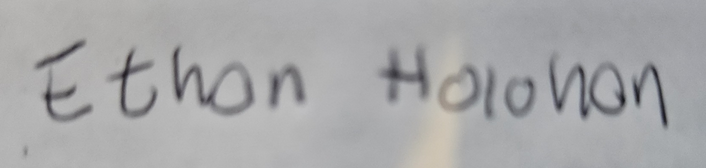
The Goal of this application is to help both newer players and experienced ones better understand the rules of each site and the games played there, as well as help staff enforce rules and maintain safety for players.

The way this app intends to do this is by making sure that all the information that players need for each game is available at all times during the game, by adding in a tracking feature that allows players to see where each other are for gameplay and for staff to see where all players are on the site, to help rule enforcement by using geofencing technology.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Ethan Holohan**

**11/11/2024**

Acknowledgements

I would like to thank my project supervisor Colette Kirwan for all the help and stellar guidance they offered on this report.

Id also like to thank the FYP course co-ordinator Jack O’Neill for all his work organising the FYP course this year.

Id also like to thank the Players at the Airsoft site I work with and my boss John for helping with testing and requirements gathering.

Table of Contents

[1. Introduction 7](#_Toc119925914)

[1.1. Project Background 7](#_Toc119925915)

[1.2. Project Description 7](#_Toc119925916)

[1.3. Project Aims and Objectives 7](#_Toc119925917)

[1.4. Project Scope 7](#_Toc119925918)

[1.5. Thesis Roadmap 7](#_Toc119925919)

[2. Literature Review 8](#_Toc119925920)

[2.1. Introduction 8](#_Toc119925921)

[2.2. Alternative Existing Solutions to Your Problem 8](#_Toc119925922)

[2.3. Technologies you’ve researched 8](#_Toc119925923)

[2.4. Other Research you’ve done 8](#_Toc119925924)

[2.5. Existing Final Year Projects 8](#_Toc119925925)

[2.6. Conclusions 8](#_Toc119925926)

[3. System Design 9](#_Toc119925927)

[3.1. Introduction 9](#_Toc119925928)

[3.2. Software Methodology 9](#_Toc119925929)

[3.3. Overview of System 9](#_Toc119925930)

[3.4. Other Sections 9](#_Toc119925931)

[3.5. Conclusions 9](#_Toc119925932)

[5. Prototype Development 11](#_Toc119925938)

[5.1. Introduction 11](#_Toc119925939)

[5.2. Prototype Development 11](#_Toc119925940)

[5.3. Other Sections 11](#_Toc119925941)

[5.4. Conclusions 11](#_Toc119925942)

[6. Issues and Future Work 12](#_Toc119925943)

[6.1. Introduction 12](#_Toc119925944)

[6.2. Issues and Risks 12](#_Toc119925945)

[6.3. Plans and Future Work 12](#_Toc119925946)

[6.3.1. GANTT Chart 12](#_Toc119925947)

[Bibliography 13](#_Toc119925948)

# 1. Introduction

## Project Background

Airsoft is a sport/hobby that is played all around the world, it is very similar to paintball with two teams fighting over an objective decided upon by game marshals, who’s responsibility it is to make sure players are safe, and that they are following the rules of the game, however this is quite the challenge many players don't know the site well and some sites are very large and many players get lost or confused through out the game which leads to a bad experience, and since there are not that many marshals compared to the number of players, its hard to keep track of all players, and this can lead to safety issues.

Additionally despite the fact Airsoft has been around for a while now, there is very little digital integration until recent years.  
  
The goal of this app is to try address these issues, by giving marshals more tools to monitor the status of the game and its players, and to improve player experience by easing navigation and providing more information.

## Project Description

Quartermaster is a Mobile application that is designed for Airsoft players and Airsoft Marshals, its purpose is to give airsoft players more information so they can play the game better with less confusion over rules, and to help marshals make sure that players remain safe throughout the game day, it intends to do this using new technology on smartphones to track users on the site and display accurate site maps of each site.

## Project Aims and Objectives

The main aims of the quartermaster project are:

* To enhance players understanding of the game and reduce confusion
* To provide a way of improving team coordination and encouraging team play
* To ensure that players always have a way of communicating with a marshal in an emergency.
* To Provide players with a easy method of tracking their stats.

The way i will know that these goal have been achieved is, Live User testing, multiple field tests will be carried out using the app and if users report that the app achieves its main goals well, then i will know that the goal has been achieved.

The main objectives of Quartermaster are:

* To implement a method of users being able to contact marshals across the site
* To have all the information about a game and the site being easily accessible through the application
* To have a way of tracking players stats through geofencing and GPS tracking.

To ensure that I achieve these objectives, I will use a robust set of requirements gathered from requirements gathering to have a set success criteria for this application.

## Project Scope

Quartermaster's Scope is focused on enhancing the Airsoft experience for both players and marshals, however for this project it is also important to understand what this app doesn't intend to do

In Scope:

* this app is intended to act as a tracking app during an Airsoft event
* this app is intended to help people navigate in a large Airsoft event
* this app is intended to help people understand the rules and objectives of Airsoft
* this app is intended to help enforcement of these rules

Out of Scope:

* this app is not intended to get rid of marshals or their role in the Airsoft experience
* this app is not intended to be used as a tracking app outside of the Airsoft arena

## Thesis Roadmap

In the following chapters of this report we will go through firstly the literature review, where we will go through the research i have done surrounding my project and the technologies that will be used

secondly in the system design we will go through the preliminary design of my app and the reasoning behind it.

Thirdly we will go through my preliminary design work and my reasoning behind it.

Fourthly we will then go through my prototype i produced for this report and my developing of it

Fifthly we will go through the issues i encountered and how i tried to resolve them and my future plans.

# Literature Review

## 2.1. Introduction

In this chapter we will go through the research that was done to make sure that the app could work and to try select the best tool for the job for the app, as well as reviewing some other solutions to the problem and seeing where they failed and succeed, and finally reviewing other students work and seeing what they went with for technology how they designed their app.

## 2.2. Alternative Existing Solutions to My Problem

**Ares Alpha**

Ares alpha is a pre-existing cross platform application which allows users to create or join a session created by a user, then that user can create teams and assign players to teams and squads within those teams, users can see where everyone on their squad is located via GPS functionality and can see marks created on the map by their squad leader, this is to facilitate the co-ordination of Milsim Airsoft games.

The main strength of this app is its ability to assist team management and its ability to show where squad mates are at all times, this is due to the complex team management system which helps split up players into squads for the purposes of the Milsim with squad leaders able to see the names and who is in the squad allowing for easier co-ordination out in the field, secondly its ability to show where squad mates are at all times is another strength as in Milsim type games players get lost easily due to the large field.

The main weakness of this app, is it is hard to understand and use, when trying to create games on the app, nowhere dose it explain how to do it, or that a desktop is required, is UI is also difficult to use and hard to understand what is happening, the app has a lot of features they are just buried under confusing options menus, another weakness is its unreliability, it only supports certain versions of android and as many as half of the players where not able to connect to the app on the game-day due to these issues.

**Airsoft Force Tracking**

Airsoft Force Tracking is a pre-existing application which allows users to create or join a session created my an admin, then players can join a team and see where their team members are and put down markers on the map and see what game is being played.

The Main strength of this app is the ability for the players to interact with the map in a simple way by being able to place markers on the map to co-ordinate their team more effectively, and it being simple to create new sessions on the app allowing for it to be used easily on a skirmish day, another strength of the app is the ability to download after the game, the paths and tracking of players to see where players went during the game, this would be very helpful to the marshal allowing them to improve game modes via data gained from this.

The main weakness of this app is that their are no safety features on this app, there are no ways for players to contact the marshals in-case someone gets injured in the field, which is very important in larger Milsim games where the field is large and the marshals are spread out over the entire area, this could lead to people being seriously hurt with no contact with a marshal or team-mate, a worst case scenario in Milsim games.

What I’ve learned from analysing these previous solutions are that while there are some good solutions and ideas from these apps they tend to be missing important parts of the whole solution or have serious reliability issues that are encountered often throughout, I need when developing my solution, to keep in mind where these apps went right and how to include it in my project, and where they went wrong and how to avoid the issues that these apps have.

## 2.3. Technologies I’ve researched

**Native App vs Web app**

When creating this app there are 2 routes i could go down, either a native app using either Java or koltin with android studio, or i could create a web-app using react.js, both have their advantages over each other

a Web app is easily multiplatform, because web apps run on the web and do not need a local installation they are very easy to make compatible with a wide range of devices with only as single codebase, secondly web-apps have a more straight forward development process.

A native app has a lot more functionality, with greater access to sensors on the phone, most importantly for this project GPS, and have better performance versus web apps, which is important in my app for preserving battery life over long periods in Milsim games[1]

For the reasons listed above, in my project i will use a native app due to the fact that my app needs access to accurate GPS to work correctly and the improved performance will make the battery last longer in the field.

**Java vs Koltin**

Java is a popular programming language for developing native apps on an android system, it is used on over 3 billion devices and has extensive support and nearly all android devices, it also has extensive online documentation due to its many years in use.

Koltin is a newer android focused programming language for developing apps using android studio, it has many new features compared to Java such as Null Safety and a lot less boiler plate code, its is compatible with Java being able to use the same library's designed for Java and can run on devices that have Java installed [2]

Due to these factors i will use Koltin over Java in my project, despite the fact i am more familiar with Java overall, the cross compatibility and the ability to use the same library's as Java with the additions of new features and less code needed makes me feel like koltin is the best choice for this project

**No SQL vs SQL**

There are 2 directions that i can go for a database in this project, either using No SQL with Firebase or a MySQL database with GCP

No SQL is a type of database that stores data differently from standard SQL databases, instead of storing data using rows and tables in a relational model like SQL, it instead uses other ways based on the model; either document, key value, wide-column or graph.

The advantages of No SQL databases are that are quite flexible and can deal better with high user loads, there are also the different types of No SQL database as said before that can be used individually or together in a multi-model database, this gives No SQL its flexibility[3].

The advantages of a SQL database are that the are more secure than a normal NO SQL database and are more likely to be used for government or secure industry as the meet the criteria of certain standards. They also tend to be better for transactional databases and enterprise resource planning systems.[4]

Due to these factors i will be going with a No Sql database with firebase over a SQL one, this is mainly due to performance as my database will be accesses and updated often, however I will need to keep an eye on my usage limits and make sure that my app doesn’t use to much data

**Nearby Connections API**

The nearby connections API is a API for android that allows nearby devices to connect to other nearby devices regardless of internet connection, by allowing devices to perform NDP and establish direct offline wireless links to exchange data [5].

This can be used in my app by allowing devices with outdated data to be able to exchange data with a nearby device that has the updated data allowing it to still receive and send data allowing phones that are out of connection to be able to still communicate with marshals in an emergency and still receive GPS data for player location [6].

**Maps**

in my app the map that tracks and shows other players location is a vital part of my project functionality from my research there are a few different library’s that i could use for my project, these where the google maps API and the Map box API, both of these API offer similar services, another option to to consider is just directly using Open Street maps through a library like OSM droid.

Map Box is an alternate mapping solution to larger providers such as google maps, it provides a large amount of customisation options and it has a slick modern design to it maps, it uses the data off OpenStreetMaps and has a good SDK and API as well, it also has a generous free tier.

The advantages of Map box are firstly that it offers a lot of customisation compared to google, you can edit every layer of the map and that could be useful for adding different map styles for accessibility and personal preference in the app, also Map Box is cheaper to use with it only charging once you have 25000+ users as well as 28,000 web loads per month, google however is $7 per 1,000 loads which is expensive. [7]

Google Maps API Is an API offering from google which allows developers to integrate google maps into their product, it is a well built piece of software with plenty of data and easy integration with android native apps thorough its API, it has extensive route finding technology, being one of the most convenient ways to find and get to a place.[8]

The advantages of Google maps is it has a lot more map data and features, with satellite images and street view being some of the services that map box lacks, however i don't think that these are too relevant to my project, though having satellite images would be nice.

Open Street maps is an open source mapping project that uses volunteers to map out the world, it is entirely customisable due to its open source nature and offers itself as an alternative to other more authoritative mapping services[9]

The advantages of Open Street Map over the rest of the option are, firstly its is entirely free to use and wont have any overhead costs for me to deal with, it is also extremely customisable with many layers styles to chose from when making the app, one of the weaknesses of directly using openstreetmaps is that the OSM droid library which now is no longer being updated any more, last update august 19 2024.

From my research i have concluded that i will be using Open Street maps as it is a cheaper more customisable SDK that offers a good library (OSM Droid) of functions for android, despite the fact it is no longer updated, it hasn't gotten old enough yet for me to not want to use it.

## 2.4. Other Research

**Cross platform considerations**

For this app there are 2 platforms i was considering developing on, either iOS or android, both have an almost equal market share in Ireland, and creating an app for both platforms would be to much work for the scope of the project, so a platform had to be chosen

from my research android became the obvious choice, to develop in iOS I would need to either use a VM running macOS or a mac itself, which is not ideal for development, additionally I have alot more experience with Android development, so android was chosen.[10]

**GPS**

GPS or global positioning system is one of the major technologies that has enabled a lot of the utility of phones, the way GPS works is by having satellites in the GNSS network continuously broadcasting their location, once your phone has the location of four of these satellites it can locate your position anywhere in the world, to a degree of accuracy, typically 5-16 feet, influenced by your surroundings, generally the more of the sky that is visible, the better.[11]

GPS will be a corner stone of my project a good amount of the functionality of the app will be based around it, one challenge i will need to overcome when it comes to GPS is the accuracy issue, since Airsoft happens often in forests or inside buildings, the accuracy will be affected.

**Geofencing**

Geofencing is a technology that allows GPS enabled devices to do something when they enter or leave a geo fenced area, these areas can be as large as they want and can be any polygonal shape, due to GPS not being entirely accurate, it can give many false positives or negatives when near the edges of a geo-fenced area.[12]

Geofencing is used for many things, its commonly used in marketing, giving location based advert to better appeal to a user and get more value out of their advertisement. [13].

what i want to use it for in my project is to implement areas that are “out of bounds”, areas that are not in play for either gameplay reasons or safety reasons, a lot of the time newer players will go into these areas due to a lack of experience on the site, so from my research, geofencing would be the way to-do this.

**GDPR Considerations**

GDPR is a set of rules and regulations implemented by the EU in 2018 that had huge impact on how data was handled across the entire world, it is hugely important for my project since i gather sensitive information that i must abide by these rules.

In broad strokes the data protection principles can be broken up into 7 points

1. Lawfulness, fairness and transparency — Processing must be lawful, fair, and transparent to the data subject.
2. Purpose limitation — You must process data for the legitimate purposes specified explicitly to the data subject when you collected it.
3. Data minimization — You should collect and process only as much data as absolutely necessary for the purposes specified.
4. Accuracy — You must keep personal data accurate and up to date.
5. Storage limitation — You may only store personally identifying data for as long as necessary for the specified purpose.
6. Integrity and confidentiality — Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality (e.g. by using encryption).
7. Accountability — The data controller is responsible for being able to demonstrate GDPR compliance with all of these principles.

[14]

My main concern with my app will be Data minimization and Integrity, my reasoning for this is that I’m already collecting sensitive information like location, i don't want to store identifiable personal data in case of a data breach, secondly having good data security is kind of just a good best practice, so sticking to this as a principle will be beneficial. The rest of the principles will also be kept in mind as well.

**Battery Life**

Quartermaster will be using GPS as a core feature of its design, however GPS and location based services do come with some downfalls, one of the major ones, especially for this project is the increased power drain that will limit battery life on the phone that my application is running on, this is a problem as most Airsoft games can last anywhere from a few hours to multiple days, so battery life must be considered.

In indoor areas battery consumption from GPS can be increased by as much as 75% from the phone using more battery to find a better signal strength[15], one way to perhaps try keep battery life costs down would be to use less accurate location finding methods such as using coarse network location which drains only a tenth of what using fine location may take using GPS[16].

Hopefully this will help offset issues of using this app in areas with bad signal, it might be more inaccurate but i feel that having there phone with a workable amount of battery is more important.

## 2.5. Existing Final Year Projects

**Anseo!**

**Author: Jonathan Hew**

Anseo is a web app that is designed to be used by lectures to keep attendance to their lectures in a more convenient way than a paper sign in sheet, it does this by creating a session and allowing students to join it only if they are physically present within a geo-fenced area.

Jonathan used a three tier model for the project with a presentation layer, application layer and data layer, the user opens the web app, which uses react for its function, which then connects using node.js to a PostgreSQL server to store data.

They also used the Agile development method feature driven development for their project, their reason was because of the numerous features of the project, this is similar to my project and the many features i will also have to make so they might be a good example of what to follow.

Anseo! Uses a lot of similar technology to my app as they use geo-fencing and gps technology while one user hosts a central session that all other users connect to.

**Nitelite**

**Author: Sean Breen**

Nitelite is a native mobile app that is designed to help people stay safe in nightlife environments, it intends to do this by creating an app that allows people to keep track of their friends in a busy night life environment and if needed request urgent help from friends in a dangerous situation.

They also had a 3 tier system, with a react native front end and django back end and a SQL database.

They also used an agile development methodology, though they did state that due to the nature of agile it would be hard to use in a single person project, but it would be ideal to stick to the 6 key principles to ensure future best practice.

Nitelite interested me because of how they used Bluetooth beacons to get accurate positions when inside a building which could be useful for my app.

**Conclusions**

From these project I have examined above, I have learned some important lessons, firstly the agile development method seems to be quite popular and for good reason, it leads to better outcomes for the project overall. Secondly, both projects gave me a good idea of what a finished project looks like and what my requirements will have to be like to achieve a similar result to the above projects.

## 2.6. Conclusions

The conclusion of this literature review conducted for my final year project are, firstly the research into different tech away from the stuff i would usually use led to some interesting and beneficial insight into some of the tech i am going to use for this app and made me change my mind on certain choices i had already made up in my head about the development of this app.

Secondly from my examination of previous solutions there is defiantly some existing solutions, however both of those examined have flaws i also need to overcome and strengths i should try put in my project, and it proved that what i wanted to do was feasible.

Thirdly reading how previous students approached similar problems showed me some examples of what was expected in terms of complexity from this project and ideas on how to move forward with it.

**Requirements**

From what I have learned, I have built a set of requirements using the ideas gathered from above and using some requirements gathering from the expected user base using interviews and a survey.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Req ID | Requirement Desc | Category | Priority | Source | Criteria | Notes |
| UR-001 | Realtime Tracking of players and teammates | Functional | High | Survey | Tracking works well | Completed in Prototype |
| UR-002 | Stats Tracking of players | Functional | Medium | Survey | Be able to track your win loss and other stats |  |
| UR-003 | Accurate Map of site | Functional | High | Survey | Have the ability to create an accurate map of the site |  |
| UR-004 | Reliability of functionality | Non-functional | High | Interview | Have all the functions work reliably | This is a common issue with other apps |
| UR-004.1 | Tracking Accuracy | Non-functional | High | Interview | Have tracking be accurate to 5-10 feet |  |
| UR-004.2 | Session Reliability | Non-functional | High | Interview | Have session be easy to join and not close randomly |  |
| UR-005 | Safety SOS features | Functional | High | Interview | Have a way for players to contact marshals in case of an emergency |  |
| UR-006 | Be able to tell how many people are at the site/ are booked in remotely | Functional | Low | Survey | Have the ability to see different sites and how many people are planning to be there | Would require a lot of work and a whole site system, do if time permits but otherwise ignore |

# 3. System Design

## 3.1. Introduction

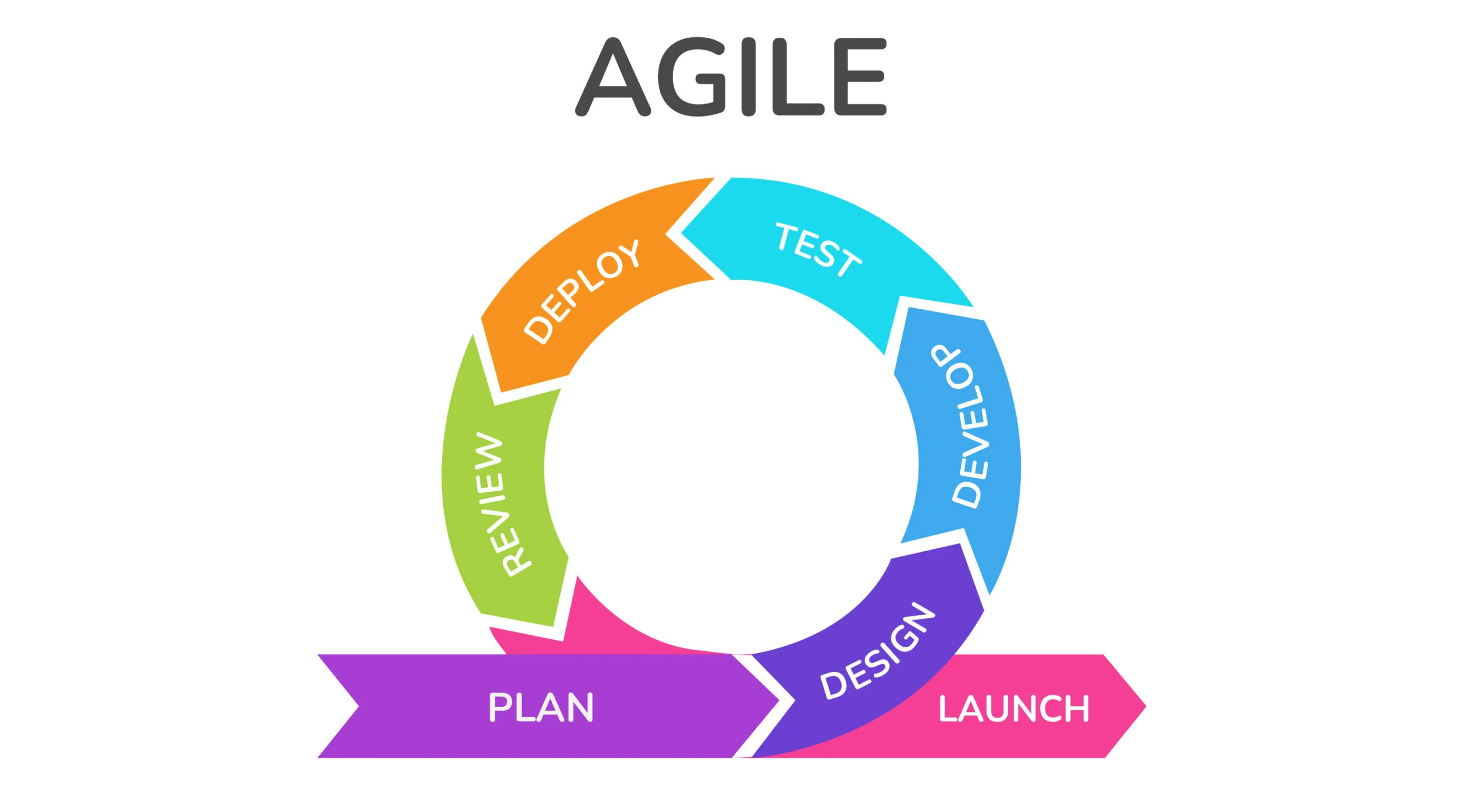
System design is a vital part of the development process, as it sets in place guidelines and the design for the project, to ensure that the project is done both to a good standard and in a good time frame as well. In this chapter we will go through some software methodology’s and the one chosen for the project, then we will go through an overview of the system

## 3.2. Software Methodology

In my research i found 2 software methodologies that i thought where suitable for this project

**Agile development methodology**

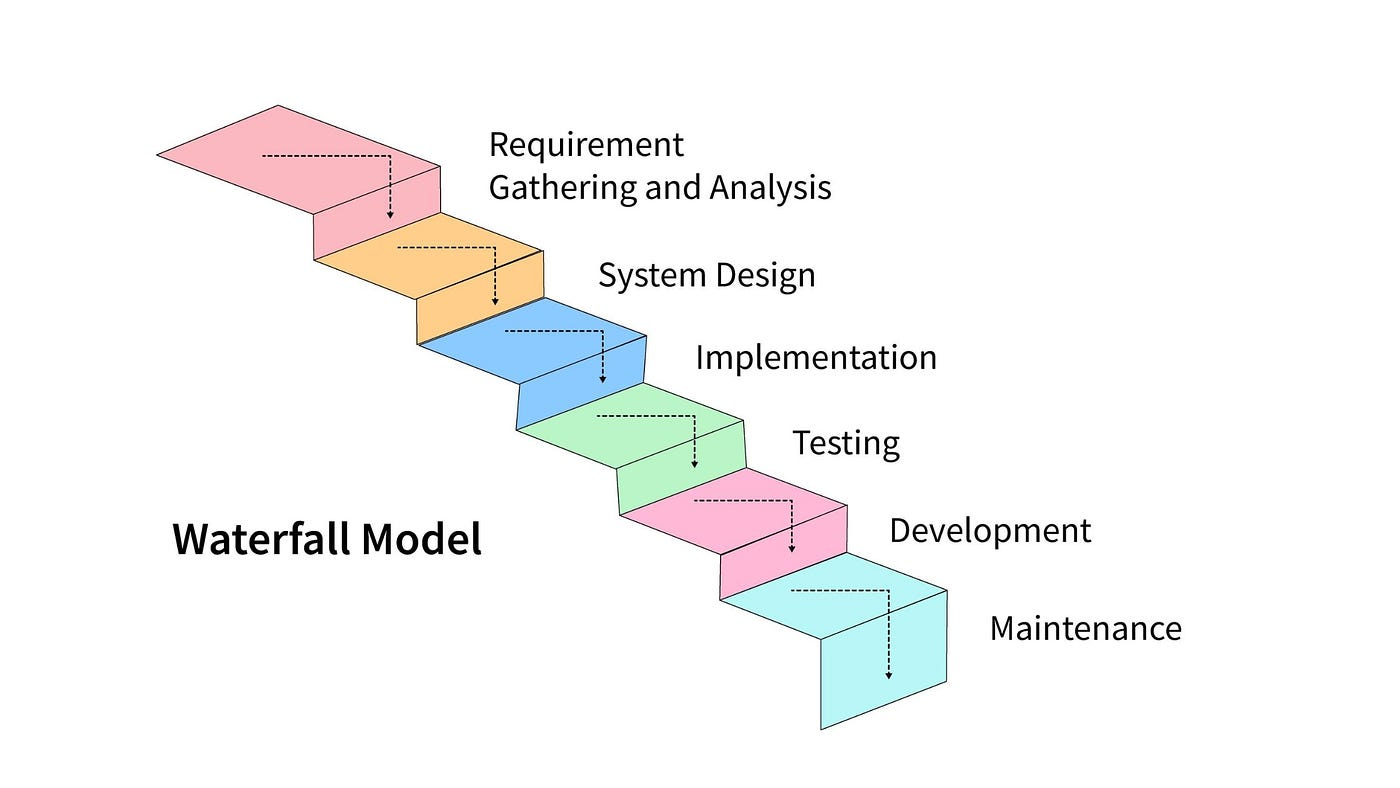
The agile methodology is a methodology framework that is based around breaking down the project into multiple stages and iterating on the development cycle, it is designed to be reactive and be able to changed based on changing requirements and take in shareholder or project owner feedback and incorporate it into the project with minimal issues, there are a few way to implement an agile approach, examples include scrum and kanban[17][18]

Figure 1: example of a agile development cycle

This approach appeals to me due to my circumstances around this project, i am a solo developer for this so a lot of the collaborative benefits of agile will be lost however since i work at an Airsoft site it puts me in a good position to receive feedback on each iteration of the project.

**Waterfall development methodology**

the waterfall development methodology is a linear sequential life cycle model, where planning designing, development and launching are handled in sequential order, due to this highly structured approach, the waterfall methodology is highly inflexible and doesn't react well to changing requirements.

Figure 2: an example of a waterfall development cycle

However since this project is a solo created project, its highly structured nature and easy to use nature might be beneficial to the project and its focus on getting the project done rather that re-iterating over and over could help get the project done quicker[19]

**The Chosen Method: Agile Kanban**

In the end i decided that using an agile Kanban method would be good for this project, i feel that with the ability to be able to regularly test my app on a weekly basis that an agile method would produce good results, as i will be able get feedback on each sprint and actually use that feedback to improve the project.

The main reason i am going to be using the kanban method is because it is very good at visualisation, with each task being represented on a physical board as a card, it will help me keep track of what work i need to do and what work i have already done. Additionally a way to visualise the progress i will be making will help me stay motivated and productive for this project.

## 3.3 Requirements gathering

For the app quartermaster to be a useful app we need to gather some requirements from the intended user bases, in order to do this I conducted informal interviews with players on site and talked to fellow marshals about what they would want from such an app, I also used a small survey to try get more general info and from these I have come up with this set of requirements

## 3.4. Overview of System

**Front End**

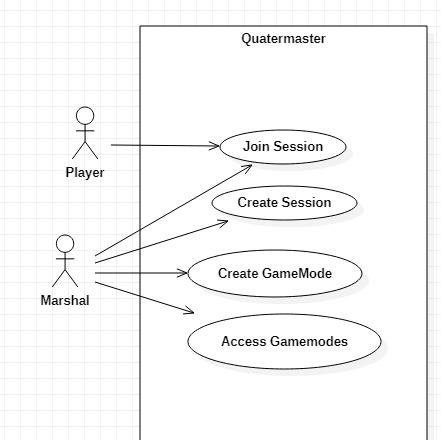
For the Frontend of the app, since i am doing android native, will be using Jetpack Compose for designing UI elements and the structure of each activity within the app, Kotlin will be used to add interactivity to these UI elements as well, this allows me to get the best performance out of my app, using the most up to date methods for developing and creating my UI elements.

**Back end**

For the Back end i am using a Firebase Real-time database as my No SQL database, Koltin will be used to communicate with it from the native app, firebase allows for quick expansion of services in case i experience large amounts of traffic to the real-time database, in addition firebases real-time data Synchronization will ensure that all user will have access to the same data throughout the game.

**Use Cases**

in my app there a 2 different types of User Players and Marshals, players only need of the app is to be able to join sessions , marshals act as the admins they create the sessions the games etc. and they also need to be able to join other marshals sessions as there can be more than one marshal out there at once.

Figure 3: Use case

**Use Case Diagrams**

In this part I will elaborate on the use cases set out previously

|  |  |  |
| --- | --- | --- |
| **UC-0x** | **Title** | |
| **Dependencies** |  | |
| **Description** |  | |
| **Preconditions** |  | |
| **Ordinary Sequence** | **Step** | **Action** |
|  |  |
| **Postcondition** |  | |
| **Exceptions** | **Step** | **Action** |
|  |  |
| **Comments** |  | |

|  |  |  |
| --- | --- | --- |
| **UC-01** | **Create Session** | |
| **Dependencies** |  | |
| **Description** | The app will behave in this way when the create session option is selected | |
| **Preconditions** | The user has opened the app and isn’t already In a session | |
| **Ordinary Sequence** | **Step** | **Action** |
| **1** | User selects the Create session button |
| **2** | System connects to the firebase server |
| **3** | System creates a new session with a unique ID |
| **4** | System creates a new user in that session and moves the user to the Map screen |
| **Postcondition** | The user is in a session and can have other users join it | |
| **Exceptions** | **Step** | **Action** |
| **2** | if the system cannot connect throw an exception |
| **Comments** |  | |

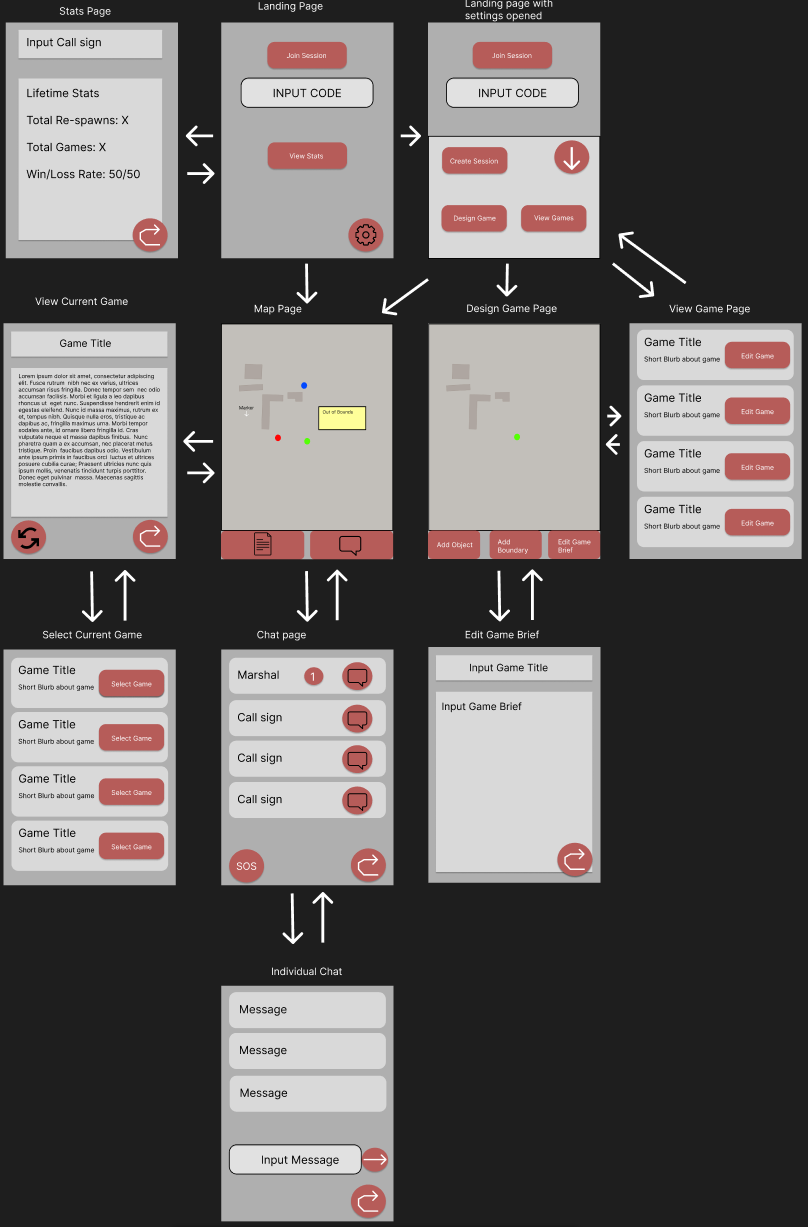
|  |  |  |
| --- | --- | --- |
| **UC-02** | **Join Session** | |
| **Dependencies** |  | |
| **Description** | The app will behave this way when a user attempts to join a session | |
| **Preconditions** | The User has opened the app and isn’t already in a session. | |
| **Ordinary Sequence** | **Step** | **Action** |
| **1** | User inputs a session code |
|  | **2** | User presses the join session button |
|  | **3** | System connects to the firebase |
|  | **4** | System checks to see if there is a session that exists with that code |
|  | **5** | If the session exists it adds them to it and puts them in the map screen |
| **Postcondition** | The user is in a session | |
| **Exceptions** | **Step** | **Action** |
| **3** | Cannot connect, throw an error |
| **5** | Session doesn’t exist throw an error |
| **Comments** |  | |

|  |  |  |
| --- | --- | --- |
| **UC-03** | **Create Game mode** | |
| **Dependencies** |  | |
| **Description** | The app will behave this way when the user creates a gamemode | |
| **Preconditions** | The user has opened the app and has navigated to the | |
| **Ordinary Sequence** | **Step** | **Action** |
|  |  |
| **Postcondition** |  | |
| **Exceptions** | **Step** | **Action** |
|  |  |
| **Comments** |  | |

|  |  |  |
| --- | --- | --- |
| **UC-04** | **Call SOS** | |
| **Dependencies** |  | |
| **Description** |  | |
| **Preconditions** |  | |
| **Ordinary Sequence** | **Step** | **Action** |
|  |  |
| **Postcondition** |  | |
| **Exceptions** | **Step** | **Action** |
|  |  |
| **Comments** |  | |

**App Structure**

the structure and navigation of Quartermaster is important, especially since it is going to be used in a high stress environment, therefore it is important to get down a good design for the main structure of the app early, to do this i created a map and initial UI design for my app

Figure 4: UI Mockup of Pages in App

**Landing Page**

This is the first page that will open when the app is launched for players they only have to care about joining a session, that is why the join session button and its input field are kept centre screen and prominent, the stats button is kept underneath it and isn't as important, and therefore doesn't need as much space, the marshal options are hidden in a menu that slides up when the settings button is clicked, this slide up menu has all the options needed to create game for use later and to create sessions, the reason i have done this is to not confuse users with a ton of options as most users will never use the marshal side of the app.

The stats page is where users will input their “call sign” this will be used to identify who they are in the app and to message them using the message function, if they try to join a session before they input a name, it will bring them to this page and ask them to set their name, the stats are there purely for the benefit of the user, i have chosen to only have respawns, total games and win loss rate as the only tracked stats as these can all be tracked by the phone via geofencing and the results of games.

When a user joins or creates a session they are brought straight to the map page.

**Map Page**

This page contains the map page, this is the central page that a lot of other pages such as game brief and the messaging system connect to, the map displays the users current location updated live on the screen, and it will also contain other areas marked out in the currently loaded game, such as re-spawns, areas that are out of play and other gameplay objects and objectives, these will be displayed in contrasting colours with labels on the map, from this page the user can view the game brief or they can view their messages.

When a game ends it will display which team won and the score, this will be tracked in stats.

**Game Brief**

This page contains the game brief it contains a summary of the current game its objectives and the name of the game, also if the user is the admin (player who created the session) there will be an additional option to select a game which lists the games created on the apps game creation section, once selected it will change the brief on every connected device to the selected brief and load and game objects onto the map, and clear old ones.

**Game Creation**

This page contains the game creation tools, these are designed to be used by marshals to create games for loading later, marshals can add objects, game objectives , re-spawns and edit the text brief here, these are going to be saved locally to the device and uploaded to the firebase only when selected by the marshal and not be permanently saved there.

**Messaging**

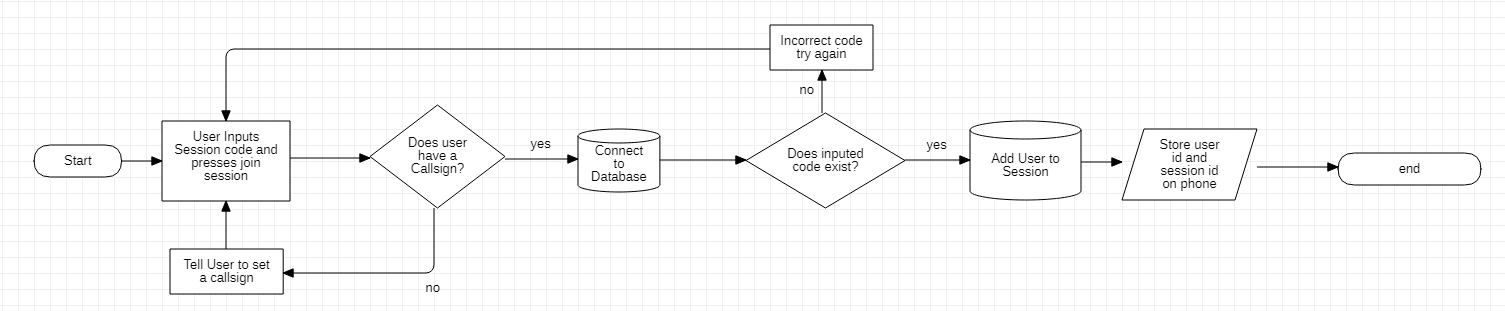
This page contains the messaging service part of the app, it allows players to communicate directly with the marshals and other players on their team, players will be identified by their call-sign on the app, this is to avoid saving personal information such as real names to the database, additionally if the player is in distress the SOS button can be pressed which will immediately notify everyone in the session that they need help and ping their location on the map, showing where they are to all players.

**Activity Diagrams**

in this section i will go though some activity diagrams showing how i intend the features of my app to be implemented

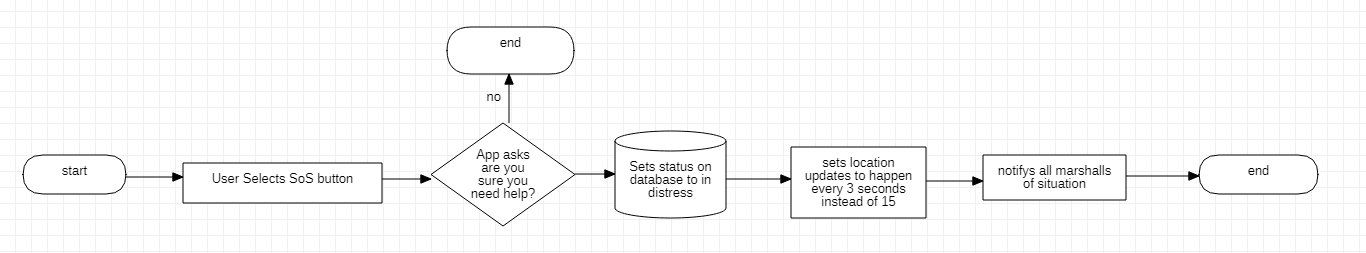
**Joining Session**

A user opens the app and presses the join session button after inputting a code for a session, if the user hasn't set a call-sign on the stats page it will tell them to do so, if that is done it will then connect to the firebase and query the database to see if the session exists, if it doesn’t it will tell them their code is wrong and keep them on the home page, other wise it will generate a user id and add it to the session then it will store both the session id and the user id locally and then send the user to the map page.

Figure 5: joining session activity diagram

**Sending a SOS**

A user navigates to the messaging menu and presses the SOS button, it will ask them are they sure, this is in-case this gets pocket dialled as when an emergency like this happens the games is usually ended, it will then update the status of the user on the server to in-distress which will immediately notify the marshal and any nearby player of the problem, then hopefully the user can get help even if they are alone.

Figure 6: sending a SOS

## 3.5. Other Sections

**Balancing battery life with location updates**

Due to Quartermasters intended design is that the app is to be used throughout the course of a game-day (4-24 hours) battery life is very important, however a lot of the technologies i use in this app tend to consume a lot of battery, especially since i want the location tracking and geofencing to be running while the app is in the background or the app is closed.

One way i will try to be managing this is to only have location be updated every 15 seconds to minimize the amount of battery that is used by the application, this will save battery at the cost of live information.[20]

Another way i can reduce battery consumption is via lowering the accuracy of the GPS as GPS will take up more power to get a better signal in bad connection areas, however i am loath to do this as i feel that it would reduce the usefulness of my app.

**UI Design**

**See Figure 4 for UI mock-up**

UI Design is a vital part of having a good application, for my app especially since it is designed to be used in a high stress environment in an area where you do not want to have you phone out for too long, so good and easily navigable UI is essential, in my design i have tried to do this in a few ways

firstly i made it so that all intractable elements within the app have the same colour and style to them, this makes it so that once the user opens the app the intractable options are easy to identify quickly

secondly i used a high contrast design so that all important information and buttons stand out from the rest of the page, this makes it easy to identify important part of the app for the user and increases readability when out in the field.

However i do think my UI mock-up could be improved, firstly its fairly flat its hard to identify what elements are atop each other in some cases, and there is a lot of white space in the app that makes it feel kinda empty, i will have to overcome these issues when implementing this design into practice on my app.

## 3.6. Conclusions

From my design work, i have come to the conclusion that in order for my design to be successful when implemented it needs to make sure that the user can understand what is going on inside the app quickly and easily, due to the nature of the environment that the app is designed to be used in, the agile framework i have chosen will help me especially from the integration of user feedback on every sprint, and the UI design work will ensure that the application UI will be elegant at the end of the development.

# 4. Prototype Development

## 4.1. Introduction

For my prototype i wanted to try get the main functionality of the app down, in my mind the main thing that everything needed to be done first to even start working on anything else was being able to join and create sessions and then actually being able to see the location of other users within that session.

So with that as my goal i began prototype development

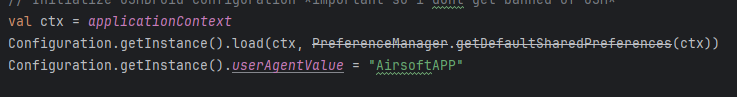
## 4.2. Prototype Development

**Preliminary learning**

to start off with since i haven't done android development in a while and i was using a new language and UI design method with Koltin and jetpack compose, i had to read a little documentation and what the syntax was of both languages and how they are intended to be used, once that was done i then moved onto trying to get the map view with OSMDroid working

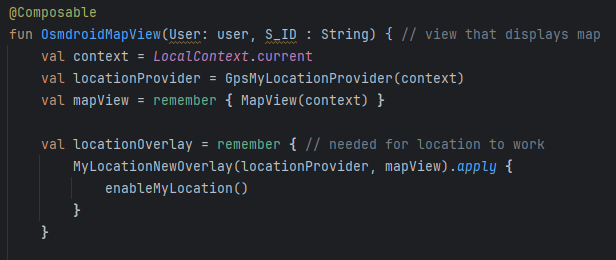
**Setting Up The Map View**

So when i started on the map i had some issues with setting up OSM droid, mainly because the documentation for it was not great and it was all using XML instead of what i was doing, so it took a bit of effort to setup, but in the end i got it running.

Figure 7: setting up the config for OSMDroid

This is setting up the config so when i get information off the OSM servers i dont get banned as and default apps with this not setup can get your app banned from the service.

In order for the current location to be marked on the map an location overlay had to be added to the map view, overlays are how OSM droid can display additional information over the map.

Figure 8: OSMDroid Map view

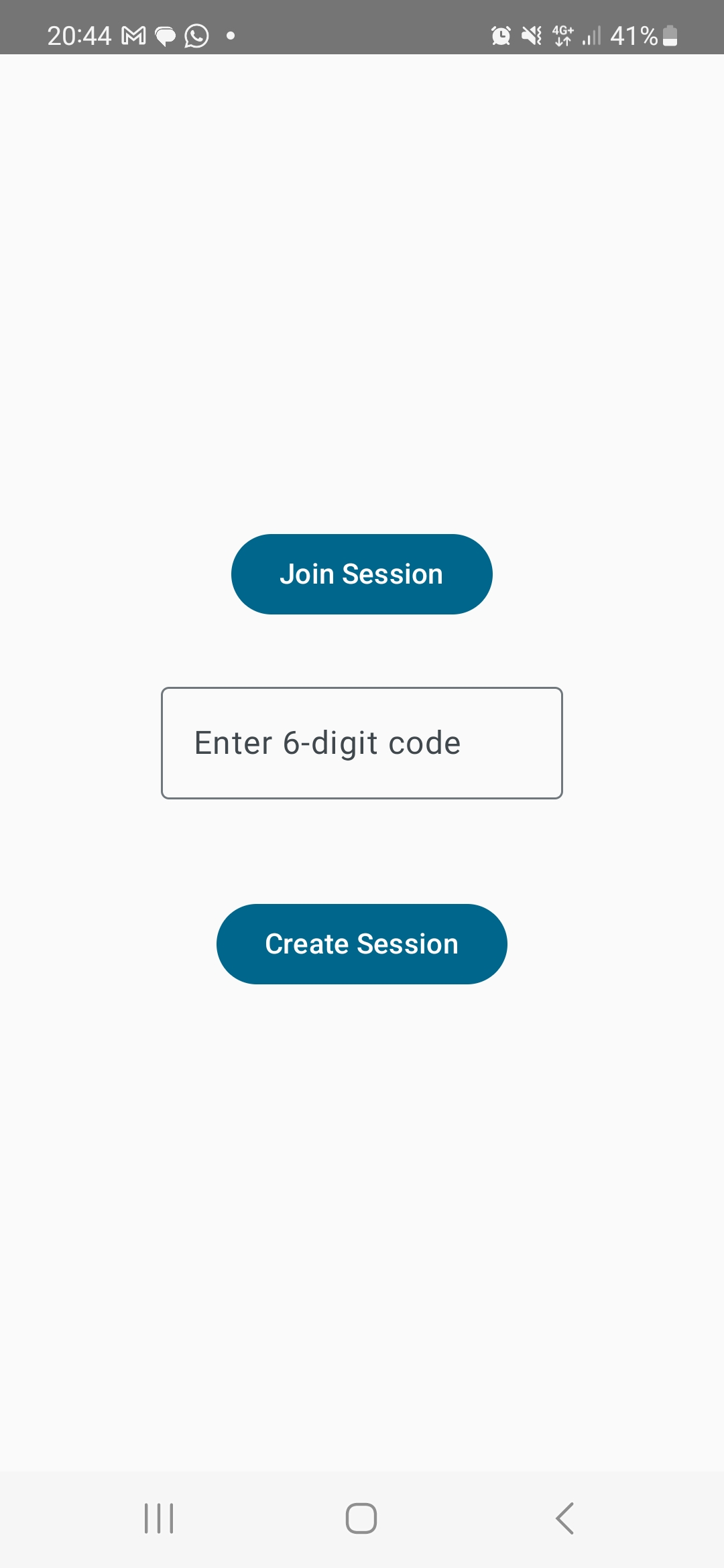
**Setting up Joining / Creating Sessions**

Before i could start coding the creating and joining a session, i had to create a firebase realtime database to host the data, thankfully the process of doing this is very easy due to the easy to use starting guides and the good documentation, once that was setup, all i had to do was setup my app to use firebase by adding the dependencies and setting up the con fig file, the i had to decide on a structure for my data, i ended up using this as my document structure for the database

Figure 9: Database document example

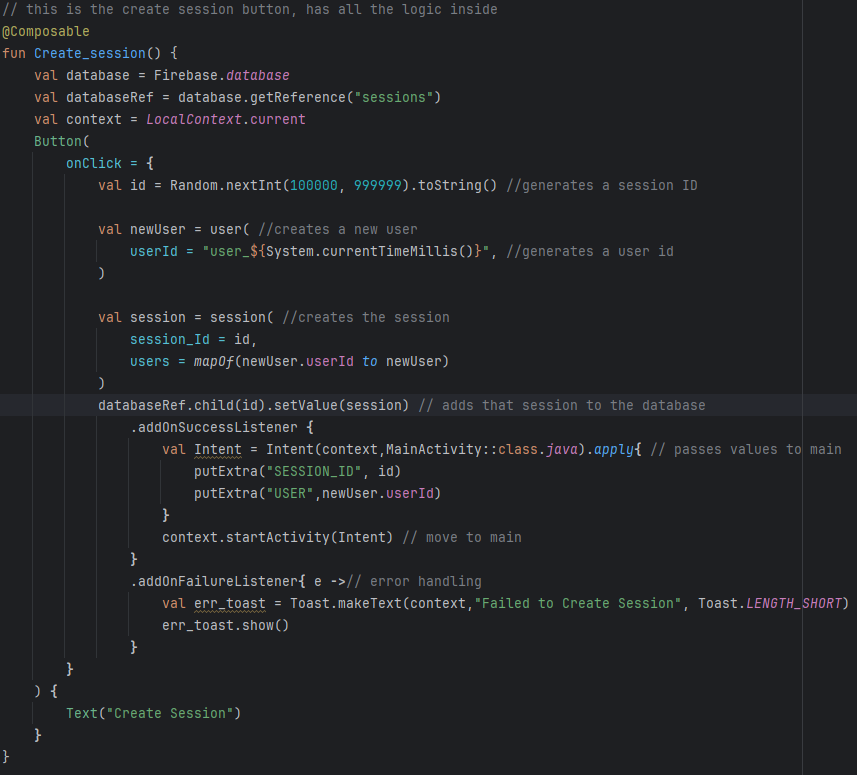
for the prototype i kept the structure simple only adding what i really needed in, in the future i will need to add more fields for stuff like the SOS and messaging features but until then this fulfils my needs.

Then i moved onto the harder part, implementing first creating a session and then joining a session, firstly i created a new activity and set it as the new startup page, this became landing\_page.kt, then i added the create session and join session button with a text input field for a basic UI.

Figure 10: prototype UI for landing page

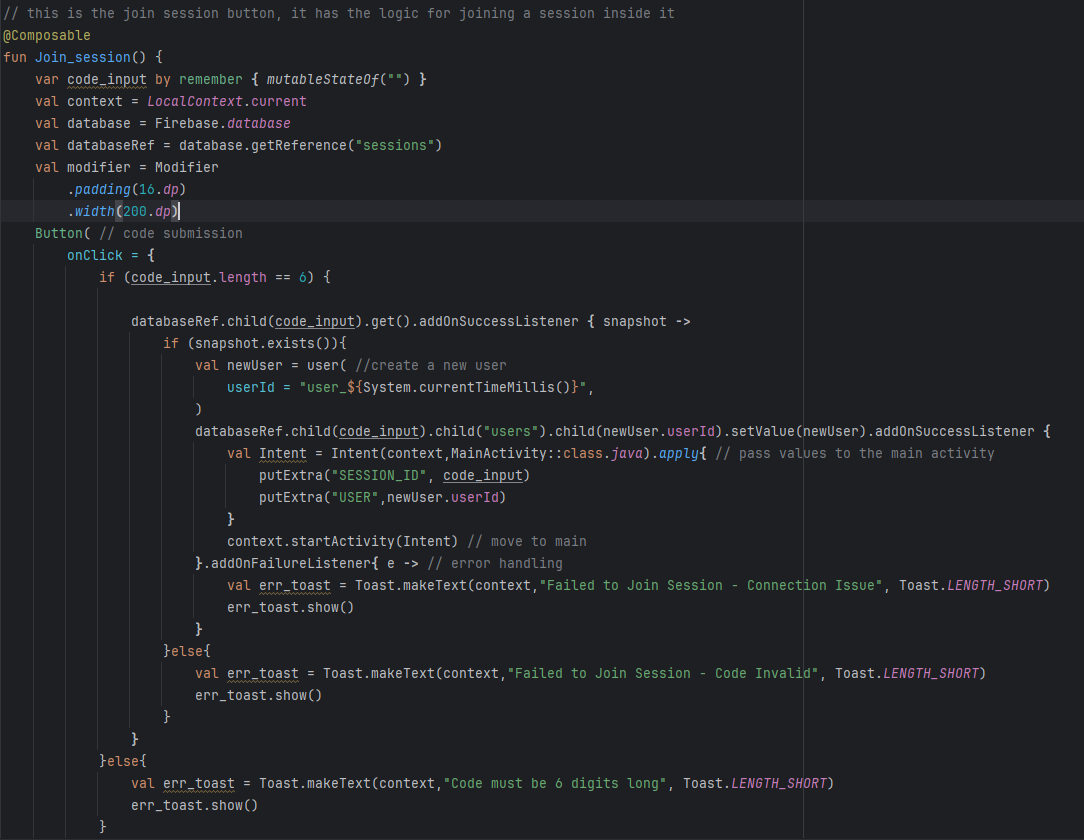
Next i had setup how sessions where created, i created the User and session classes to contain the information about users and the current session, they have the same parameters as the database document, this is in an attempt to ensure that the database functions correctly, as i had issues in development with documents being created outside of the “sessions” index due to bugs.

so the way creating a session works is, firstly it creates a user, using the phones current time in milliseconds, hardly an id that would get repeated and cause issues, then it creates the session object with a random id, it then connects to the database and on a successful connection and update, it will pass the session and user id to the main activity for the map.

Figure 11: code for creating a session

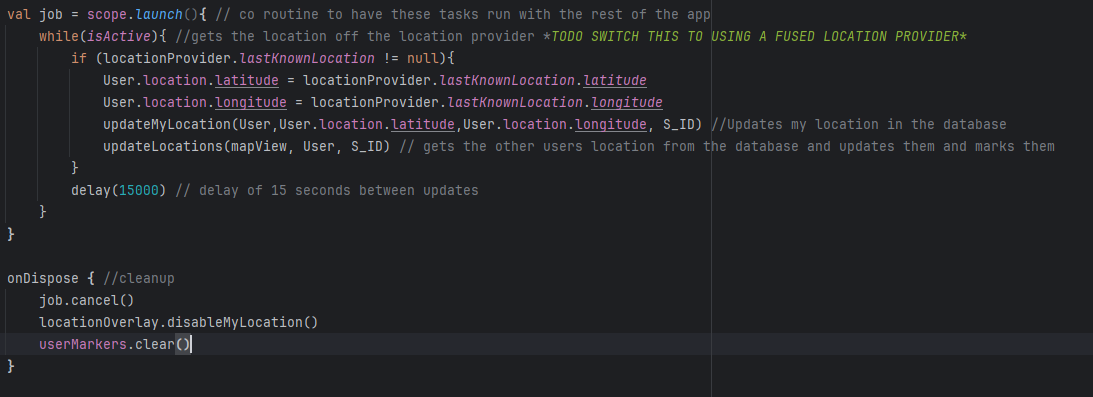
For joining session its a bit more complicated as we have to check the database for the session first before and then add the user to that session.

I implemented joining a session by firstly when the input button is pressed it checks to see if there is a 6 digit code in the code\_input field, if there is one it will check the database by getting any session with the same code as the one inputted, if there is one it then creates a user similarly to create session and then adds that user to the session in the database before passing the user and the session id to the main activity.

Figure 12: code for joining a session

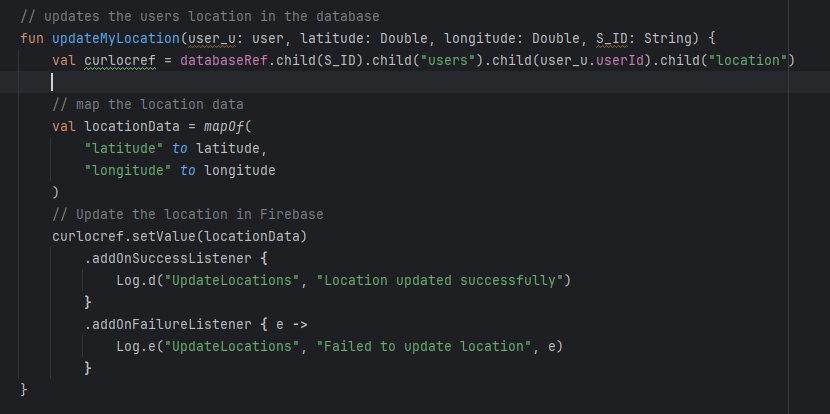
**Tracking the Other Users in the Session and uploading my location to the database**

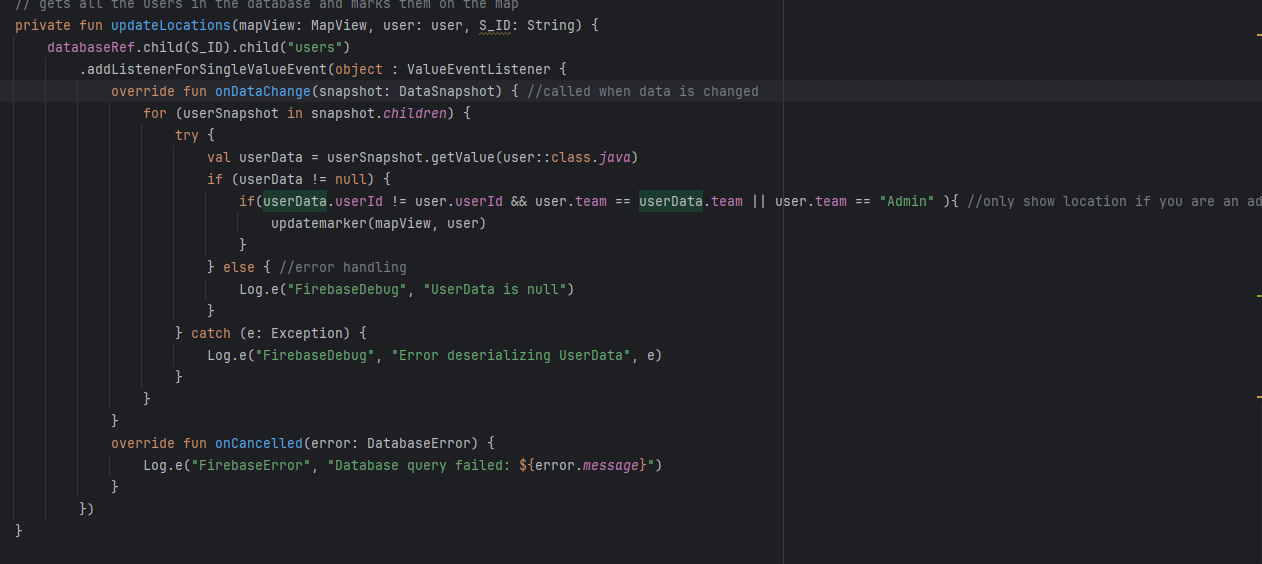
next step in getting my prototype done now that we are connected to the database and able to join a session is to now get all the other users locations from the database and then display them on the map using markers. To-do this without disrupting the rest of the code the use of a co-routine was needed, inside that co-routine it will both update the local list of other users and my current location on the database.

Figure 13: co-routine code

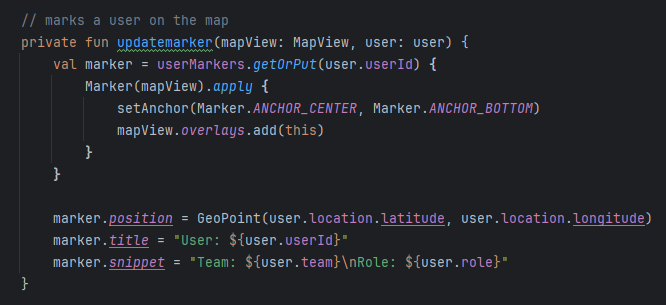
Next i got the update my location function working, i started by just extracting the current location as longitude and latitude and mapping it to a value, i did this because in the database the location is nested inside of each user in the location value, which contains both longitude and latitude, then its as simple as uploading that value to the database.

Next step was to implement the updateLocations function, this functions goal was to retrieve a list of users from the database and then mark on the map where each user was, it does this by getting every user in the session, then converting the raw data back into the user class and marking down every user that is either on your team, or if you are an admin marking everyone.

Figure 14: updateMyLocation code

Figure 15: updatelocations code

Finally last step now, all that was left was to implement marking down on the map where each player was this was done by using the OSMdroid marker function which allow me to put markers where ever i want on the map as long as i have the long and lat of where.

Figure 16: code of update markers

## 4.3. Other Sections

**Resolved Issues**

one major issue i had in developing the prototype was my inexperience with using koltin and jetpack compose, a lot of the bugs i encountered throughout the development of the prototype was mainly down to me not really knowing how to format jetpack compose and koltin, and it was difficult to implement the UI even a basic one when you don't really have a good preview, i know jetpack has the @preview feature that you can use to build previews but android studio just wasn't having it i guess and wouldn't let me load a preview, the way i resolved this issue was just through brute force, i am going to have to get good at using these tools if i want i successful project, so just actually applying them in the prototype has given me a better understanding of them, and to solve the preview issue anytime i made changes to the UI i would run it on my phone to make sure it looked the way i wanted it

Another major issue i was having at the tail end of development was issues with accessing the database properly, for almost an entire day i couldn't figure out why when it was updating the user location in the updatemylocation function it would instead of adding the location to the user it would just put it at the top of the session, i was baffled by this, until i realised that i had made an error in how i did the curlocref to get the right place to put the data and after i changed that it resolved itself

Figure 17: the line in question

**Unresolved Issues**

there are unfortunately defiantly some issues with the prototype, firstly the UI for the map isnt great both the users current location and the markers for other players are the default ones at the moment which doesn't look good and the UI for the landing page is as bare bones as possible, secondly there is no way to change your team or role at the moment and that feature that was planned for the prototype was cut due to time constraints

## 4.4. Conclusions

In conclusion i feel that while my prototype is a good starting point for my project as it develops the core feature of the app, and while it was unfortunate that i didn't have enough time to get all the features i wanted in the end, it has helped me achieve a greater understanding of the technologies i am going to use and will greatly help with the next stage of development.

# 5. Issues and Future Work

## 5.1. Introduction

In the writing of this interim report and the development of the prototype numerous issues where encountered, additionally due to the nature of the report much work on developing Quartermaster into a fully fledged FYP needs to be done, in this chapter i will go though the issues and Risks encountered, and talk about how i will try to solve them in the future, then i will talk about my plans and future work that needs to be done to finish the app.

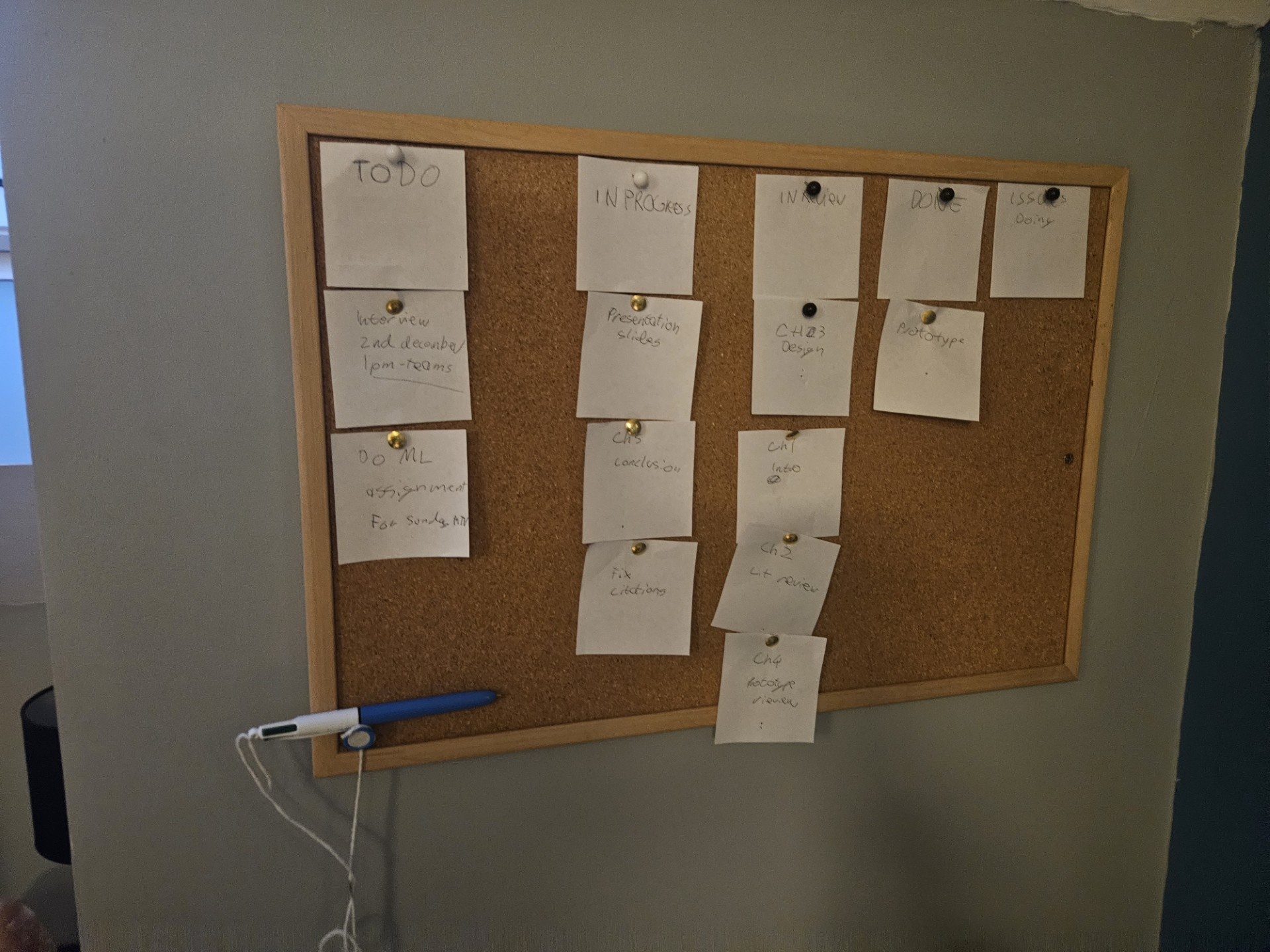
## 5.2. Issues and Risks

**Issue #1: Issues with OSMdroid documentation**

one issue i had was most of the OSMdroid documentation was designed to be used with java and XML not Koltin and Jetpack compose, this caused numerous issues early on when i was trying to get to grips with using OSMdroid in my prototype development, i resolved this issue by learning how jetpack worked properly and then using that to create the map view.

**Issue #2: Issues with time management**

another issue i had when writing this report was time management i was spending to much time on certain aspects and not enough on others, in some cases i would forget one part of the project and then do no work on it for a week, to fix this issue i started using a kanban board to help visualise the work i needed to do and the progress i was making on the FYP, this approach really helped it motivated me to do the work and since i have started using it my productiveness has gone up considerably as i am able to visualise my progress on the FYP and keep focus.

Figure : my kanban board for the Interim Report

**Issue #3: Issue with report writing**

another issue that i encountered was my reporting writing, i am not great at writing reports and it shows sometimes, this was brought up to me by my FYP supervisor during the writing of the project proposal, to try and resolve this issue i have contacted learning support and i am going to get some help with getting my report writing to a higher standard.

**Risk #1: Risk of feature creep**

with this app and its features there is a large risk of feature creep resulting in me getting new ideas to add stuff as i work on the app, to try address this, i will use the MoSCoW prioritization method, getting the main features done first and then any new ideas will get added on to-do later, not before the core is done.

**Risk #2: Risk of Burnout**

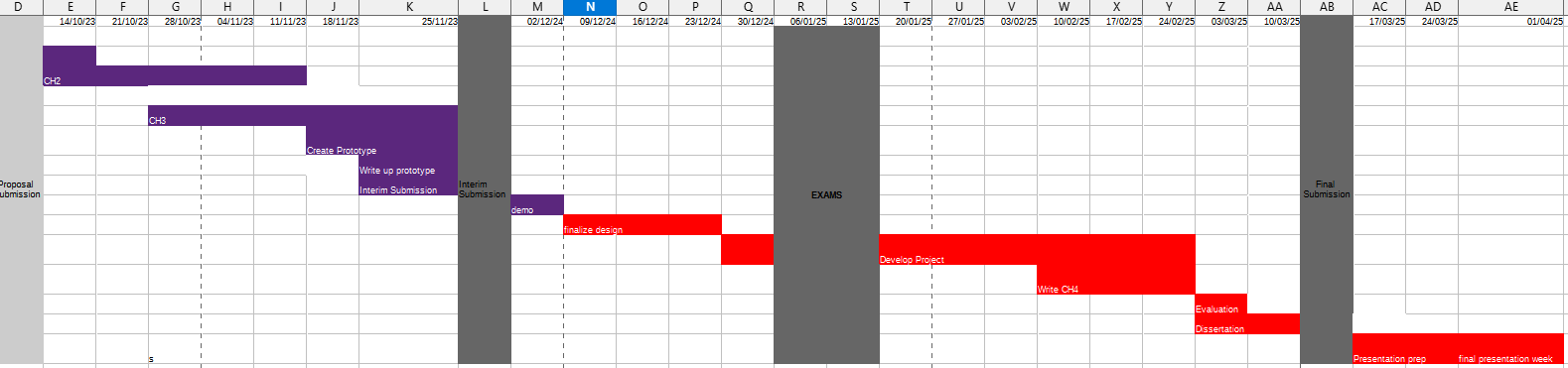
with the amount of work that has to be done there is a large risk of me burning out with this project, which is not to say that i wont get it done, but that i will lose motivation and not put my full energy into this project, i have felt this already while just writing this report due to the poor time management issue from earlier, to try address this i will try to spread the work out more over time, so i have enough time to take breaks in-between sprints on this project to hopefully avoid burnout.

## 5.3. Plans and Future Work

My plans after the interim report is submitted and the demo is done is to immediately get some more design work done, while I’m pretty happy with the structure of the front end of the project more work needs to be done on the back-end to get it up to a good standard of work, additionally i might change over from using a firebase database to one on postgres or mongoDB, the reason behind this is that firebase does have a limit on its free spark tier, and i go over that i will have to pay, however i could set up my own server using one of the aforementioned software and a spare computer i have lying around, this will save me money and also show that i am a capable 4th year student.

### 5.3.1. GANTT Chart

To manage my time till the final submission i have created a gantt chart to show how i will manage time over the coming weeks.



# Bibliography

[1] ‘Web Apps vs. Native Apps vs. Hybrid Apps - Difference Between Types of Web and Mobile Applications - AWS’, Amazon Web Services, Inc. Accessed: Nov. 13, 2024. [Online]. Available: <https://aws.amazon.com/compare/the-difference-between-web-apps-native-apps-and-hybrid-apps/>

[2]‘Kotlin vs Java: the 12 differences you should know’. Accessed: Nov. 13, 2024. [Online]. Available: <https://www.imaginarycloud.com/blog/kotlin-vs-java>

[3] ‘What Is NoSQL? NoSQL Databases Explained’, MongoDB. Accessed: Nov. 16, 2024. [Online]. Available: <https://www.mongodb.com/resources/basics/databases/nosql-explained>

[4]‘NoSQL Vs SQL Databases’, MongoDB. Accessed: Nov. 16, 2024. [Online]. Available: <https://www.mongodb.com/resources/basics/databases/nosql-explained/nosql-vs-sql>

[5] T. Lagos Jenschke, M. Dias de Amorim, and S. Fdida, ‘Nearby connections strategies: Features, usage, and empirical performance evaluation’, *Internet of Things*, vol. 23, p. 100895, Oct. 2023, doi: [10.1016/j.iot.2023.100895](https://doi.org/10.1016/j.iot.2023.100895).

[6] ‘Overview | Nearby Connections’, Google for Developers. Accessed: Nov. 16, 2024. [Online]. Available: <https://developers.google.com/nearby/connections/overview>

[7] ‘Mapbox vs Google Maps — What are the differences?’, SoftKraft. Accessed: Nov. 18, 2024. [Online]. Available: <https://www.softkraft.co/mapbox-vs-google-maps/>

[8]H. Mehta, P. Kanani, and P. Lande, ‘Google Maps’, *IJCA*, vol. 178, no. 8, pp. 41–46, May 2019, doi: [10.5120/ijca2019918791](https://doi.org/10.5120/ijca2019918791).

[9] J. Vargas, S. Srivastava, D. Tuia, and A. Falcao, ‘OpenStreetMap: Challenges and Opportunities in Machine Learning and Remote Sensing’, *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 1, pp. 184–199, Mar. 2021, doi: [10.1109/MGRS.2020.2994107](https://doi.org/10.1109/MGRS.2020.2994107).

[10] ‘Android vs iOS Development: Which Should I Learn First?’ Accessed: Nov. 17, 2024. [Online]. Available: <https://www.upwork.com/resources/android-vs-ios-which-should-i-learn-first>

[11] X. Li, X. Zhang, X. Ren, M. Fritsche, J. Wickert, and H. Schuh, ‘Precise positioning with current multi-constellation Global Navigation Satellite Systems: GPS, GLONASS, Galileo and BeiDou’, *Sci Rep*, vol. 5, no. 1, p. 8328, Feb. 2015, doi: [10.1038/srep08328](https://doi.org/10.1038/srep08328).

[12] Z. Ozdemir and B. Tugrul, ‘Geofencing on the Real-Time GPS Tracking System and Improving GPS Accuracy with Moving Average, Kalman Filter and Logistic Regression Analysis’, in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey: IEEE, Oct. 2019, pp. 1–6. doi: [10.1109/ISMSIT.2019.8932766](https://doi.org/10.1109/ISMSIT.2019.8932766).

[13] M. B. Garcia, ‘Location-based marketing using mobile geofencing: lessons learned from a user-centred application development research’, *International Journal of Technology Marketing*, vol. 17, no. 1, pp. 1–29, Jan. 2023, doi: [10.1504/IJTMKT.2023.127322](https://doi.org/10.1504/IJTMKT.2023.127322).

[14] ‘What is GDPR, the EU’s new data protection law?’, GDPR.eu. Accessed: Nov. 28, 2024. [Online]. Available: https://gdpr.eu/what-is-gdpr/

[15]

L. A. Tawalbeh, A. Basalamah, R. Mehmood, and H. Tawalbeh, ‘Greener and Smarter Phones for Future Cities: Characterizing the Impact of GPS Signal Strength on Power Consumption’, *IEEE Access*, vol. 4, pp. 858–868, 2016, doi: [10.1109/ACCESS.2016.2532745](https://doi.org/10.1109/ACCESS.2016.2532745).

[16] J. Sharkey, ‘Coding for Life--Battery Life, That Is’. http://files.blogjava.net/jicheng687/W\_0300\_CodingforLife-BatteryLifeThatIs.pdf

[17] ‘What is Agile? | Agile 101 | Agile Alliance’. Accessed: Nov. 26, 2024. [Online]. Available: <https://www.agilealliance.org/agile101/>

[18] ‘https://www.idpublications.org/wp-content/uploads/2015/05/Agile-Software-Development-Methodology.pdf’. Accessed: Nov. 26, 2024. [Online]. Available: <https://www.idpublications.org/wp-content/uploads/2015/05/Agile-Software-Development-Methodology.pdf>

[19] ‘https://www.researchgate.net/profile/Udesh-S-Senarath/publication/353324450\_Waterfall\_Methodology\_Prototyping\_and\_Agile\_Development/links/60f41f71fb568a7098b9d035/Waterfall-Methodology-Prototyping-and-Agile-Development.pdf’. Accessed: Nov. 26, 2024. [Online]. Available: <https://www.researchgate.net/profile/Udesh-S-Senarath/publication/353324450_Waterfall_Methodology_Prototyping_and_Agile_Development/links/60f41f71fb568a7098b9d035/Waterfall-Methodology-Prototyping-and-Agile-Development.pdf>

[20] A. T. de Aquino, J. A. L. Barboza Júnior, N. de Araújo Moreira, and P. Peixoto Praça, ‘Impacts of GPS module on energy consumption and machine-learning based battery lifetime estimation’, 2023, doi: [10.48545/ADVANCE2023-FULLPAPERS-3\_3](https://doi.org/10.48545/ADVANCE2023-FULLPAPERS-3_3).