SETU

Ollscoil
Teicneolaíochta
an Oirdheiscirt

South East
Technological
University

Interactive Map of SETU Carlow

Final Report

28th April 2025

Supervisor: Dr Oisin Cawley
Student: Gábor Major – C00271548

Table of Contents

Table of Figures

Abstract

The purpose of the Interactive Map of SETU (South East Technological University) Carlow is to provide an easy to use and lightweight application for any new students or guest visitors coming into the college. It is a map application that not only shows the map of the campus but also provides a pathfinding capability instructing on which route to take inside the college buildings.

The application is available as both a web application for convenient access, and as an Android application for using the map on the go. This allows people with Android devices to use the native application with extra features, and for other people to use the web version from any device with a modern web browser.

Another key aspect of the application is that it includes a fully graphical editor for the map data, such as the details and location of rooms and buildings, built into the application itself. This allows for editing the map without the hassle of manually editing database entries.

Introduction

This document is the final report on the Interactive Map of SETU Carlow project. The report lays out an overview of the project, the overall structure of the application, a description of the data structures along with a short explanation of the algorithms used.

Next the achievements of the project are shown, what was and was not achieved and the difference between the initial design and the final application. The learning outcomes of the project are listed with the challenges that have been encountered in the project and the differences that would be done on a start over.

Finally the two types of testing done with the application is explained, along with a comparison with the FURPS metrics set out at the start of the project, and a conclusion is given to finish off the report.

Overview of Project

The Interactive Map of SETU Carlow is an map application for students and lecturers for finding their way around the Carlow campus, both inside and outside the buildings. It is similar in functionality to other map applications, such as showing points of interest, searching for places and routing between locations. The main difference however is the addition of the ability to view the insides of buildings and show a path to take through the various buildings of the campus.

Users

The map application is primarily aimed at new students coming into the college campus at the start of their studies. Second, any guest visitors would appreciate having access to an interactive map where they can specifically search up a room or lecturer’s office location. The application also has an admin editing mode where the map can be updated in real time without the hassle of having users update their applications as all is done in the background.

Platform

The application targets two separate platforms, the Android mobile phone for users with location tracking capabilities and offline mode, and a web application for both everyday users and for admin users as this platform offers the functionality to update the Firebase database.

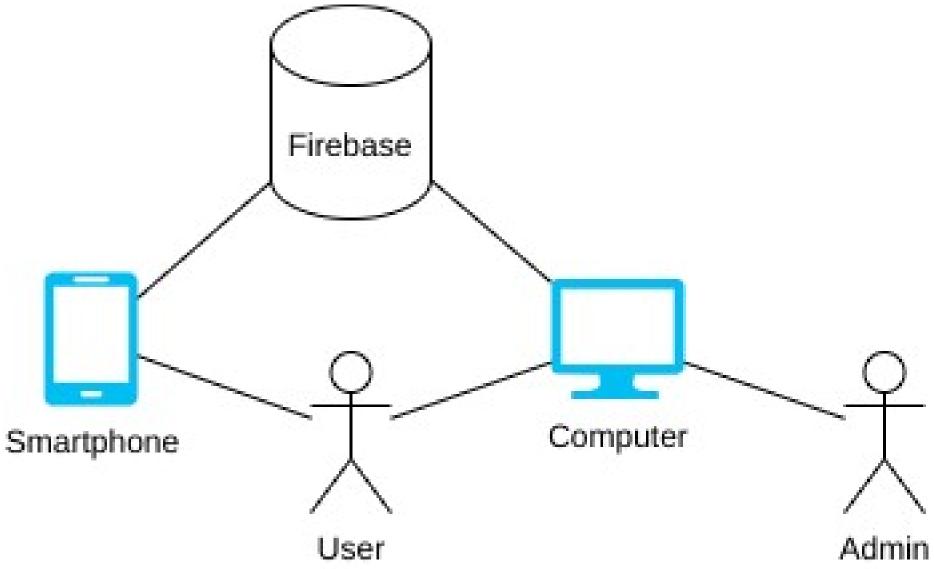


Figure 1: System Architecture Diagram

Technology

The following list is the selection of technologies that was used in the making of the map application system.

Godot

The Godot game engine was used to develop the core of the system, the application itself. The application was written in GDScript and exported to both the mobile and web platforms. (Godot Project, 2025)

GodotGPSPlugin

The GodotGPSPlugin is the plugin that allows Android exported applications to use the Android geolocation API to get the location of the mobile phone. This feature is not available for the web application. (PraxisMapper, 2025)

GodotFirebase

The GodotFirebase plugin is used to communicate with the Firebase database. The plugin gives only very basic functionality so an extra set of functions was created to hook onto the plugin. (GodotNuts, 2025)

Google Firebase

Google Firebase was used as the storage medium for the map data that defines all of the building, room and waypoint locations. It is a NoSQL (Structured Query Language) database that uses documents which are like associative arrays, and collections which are used to store the document. (Google, 2025)

Data Structures

The data structure layout used for storing the map data in Firebase is as follows in the entity relationship diagram. Each structure is a different document stored which has a connection to other documents by referencing the collection they are stored in.

The data structures consist of:

- Base map, which holds waypoints and buildings.
- Buildings, which hold waypoints and rooms.
- Rooms, which hold waypoints.
- Waypoints, which have connection references to other waypoints.

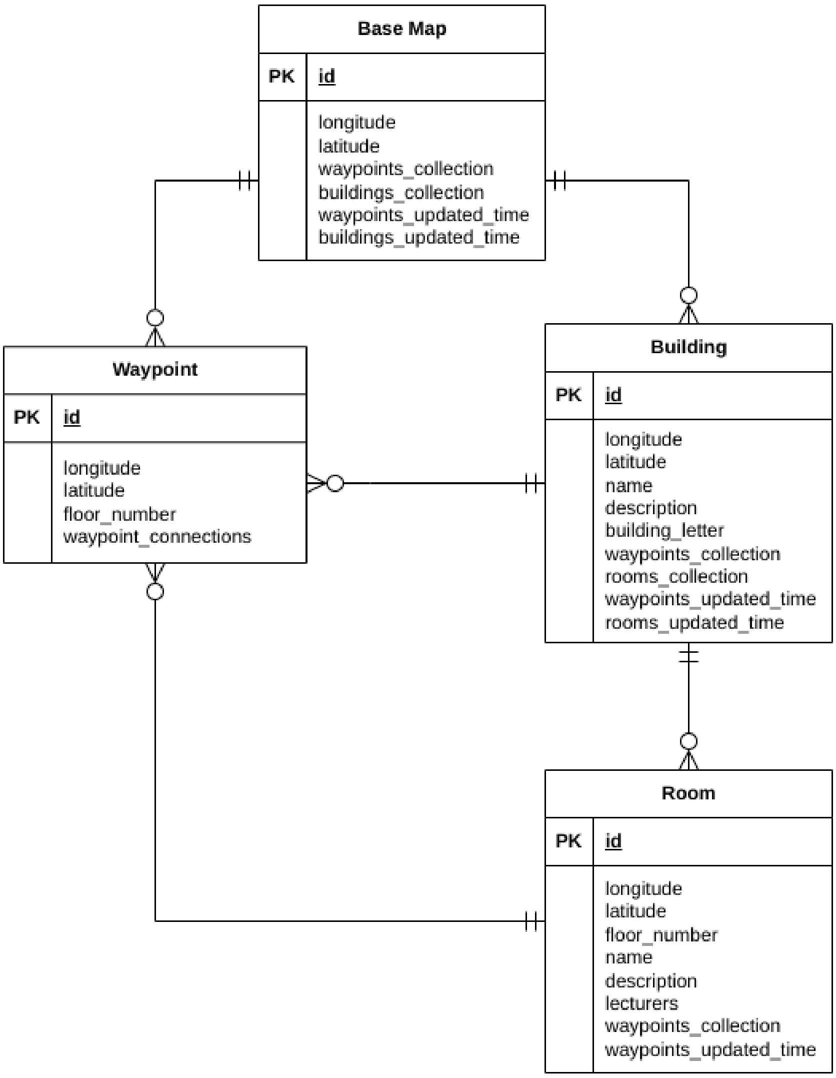


Figure 2: Entity Relation Diagram

Algorithms

Here are the top three most important algorithms that are used to create the functionality of the map application.

A Pathfinding*

The A* (star) pathfinding algorithm is used to create the shortest routes between the two locations the user wishes to travel between. (Brilliant.org, 2025)

Equirectangular Projection

Equirectangular projection is used to map the latitude and longitude values of the locations of structures to an XY plane. This is a necessary step in order to avoid skewing the data.

Searching

The searching functionality was facilitated by a custom algorithm which has two separate modes, a short and long search, and has weighting applied to the information of rooms and buildings searched.

Achievements

These achievements for the application include the core and non-core features along with what was not achieved which would be features I would keep working on with more time available. A comparison of the differences between the concept of the application and the finished application is show as well.

Achieved

The following features were achieved and incorporated into the application, including the core and non-core features set out at the start of the project, and also some additional features that were added during development.

Core Features

- Collection of rooms and buildings labelled with information such as, location, name, description, building letter and lecturers present in the room.
- Display of the floor plan of campus along with rooms and buildings.
- Move, zoom and rotate the map, and change floors.
- Search through all rooms and buildings by name, building letter or lecturer name, and search shows more details about found results and location.
- Pathfinder uses the start and end structures to create a shortest route to show.
- Pathfinder takes into account the floors of the structures.
- Admin panel available for editing the information of all structures, which is synced to the Firebase database.

Non-Core Features

- User is able to bookmark rooms or buildings for quick access.
- Latitude and longitude coordinates are used for the positioning of structures.
- Ability to use the Android Geolocation API to show the user their location around the campus.
- Connections between waypoints can be set to open or closed which the pathfinder takes into account when creating routes.
- Disabled and wheelchair users are taken into account when creating a route with the ability to avoid stairs.

Other Features

- Calculate and show the walking distance between two locations.
- Show the last modified timestamps for structures in admin panel.
- Loading icons and information screens present.
- Map data cache is saved for faster startup times and offline use.
- UI colours change depending on current floor.
- Show start and end locations as green and red icons, with arrows along the route to take.
- Save active route on closing the application.

Not Achieved

The features that are listed next are the non-core features set out at the start of the project, and any other features that have been thought up during development. Some of these features are small enough that they could be added in if an extra month of development time was available, whereas others like the 3D rendering and editing history would take a bit more time to develop.

Non-Core Features

- Pathfinder shows a selection of potential routes for the start and end locations.
- 3D rendering of rooms and buildings of the campus.
- Ability to edit the floor plans, or shapes of rooms and buildings.

Other Features

- Simplify the way waypoints are added and the connections are created.
- Ability to change the parent of a waypoint.
- Improve Android location tracking indoors.
- Use Android location together with the route for better pathfinding UI and walking mode.
- Use Android location as a start or end point for pathfinding.
- Create an isometric map mode.
- Create an editing history for admin mode to allow to undo changes, see history and simplify saving to Firebase.

Differences

From the start of the project the design of the application has been present both in a metal image and as a concept in the design document. The following compares the similarities and differences in which the application turned out visually.

Concept

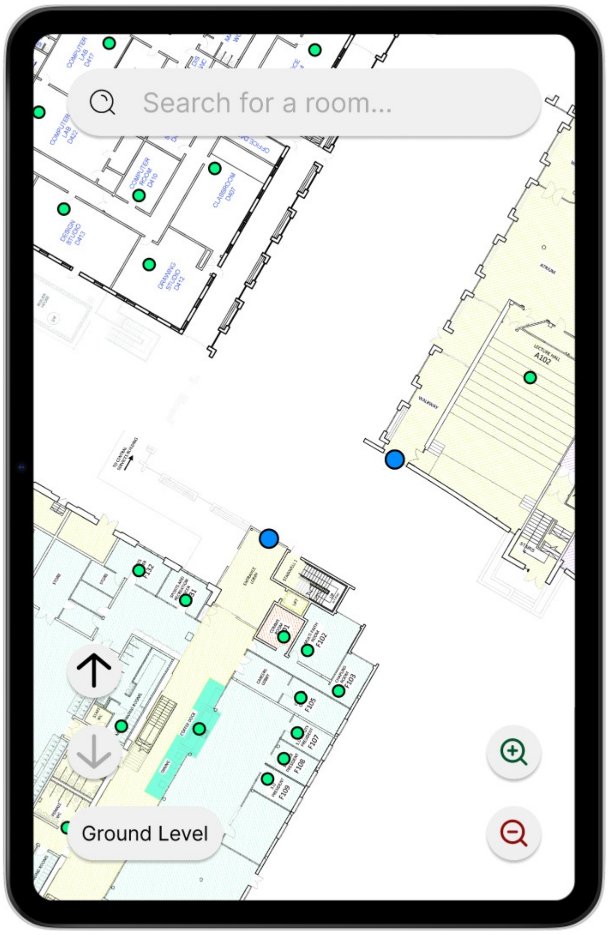


Figure 3: Map Overview Screen Concept

The layout of all of the UI is quite similar with the small differences being the different colouring of the UI, the location of the floor indicator, the presence of the current location button on the right and the addition of the navigation and stairs buttons.

The most obvious and biggest change is the fact that there are two search bars, one for the starting structure and one for the end structure.

Result

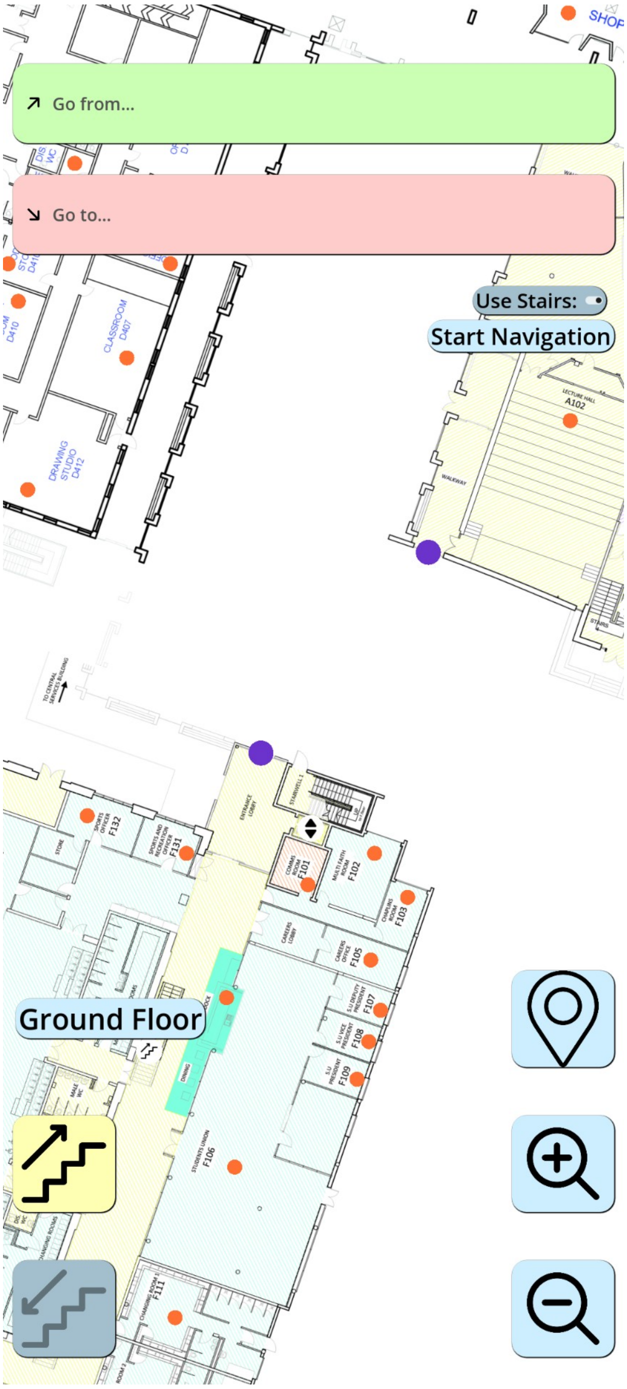
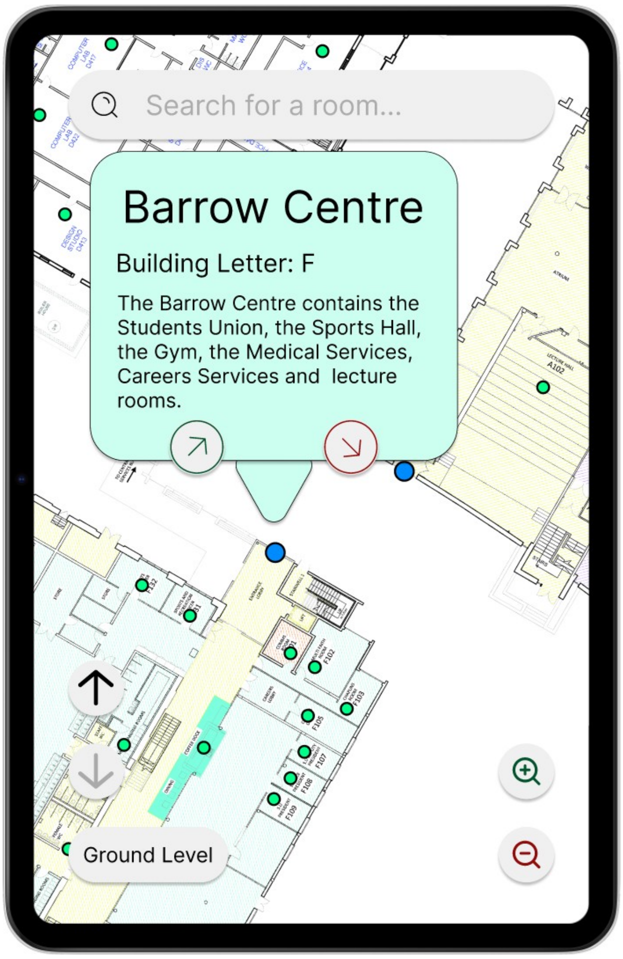


Figure 4: Map Overview Screen Result

Concept



Barrow Centre

Building Letter: F

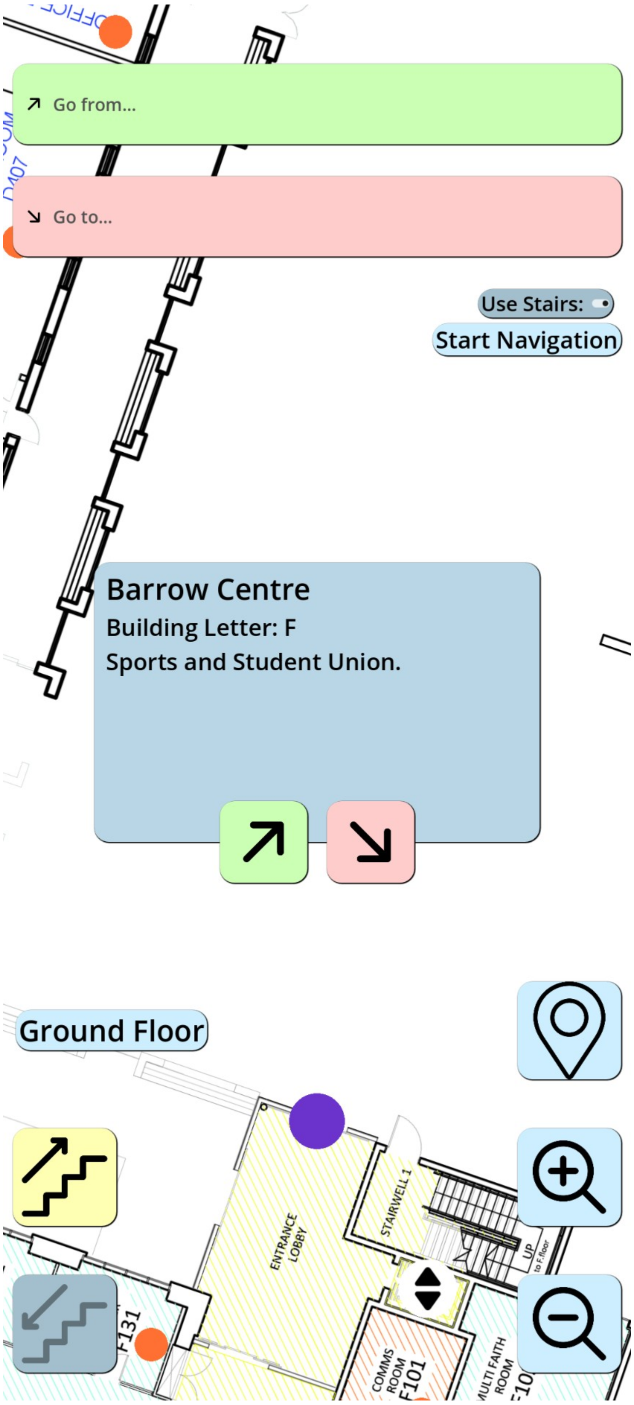
The Barrow Centre contains the Students Union, the Sports Hall, the Gym, the Medical Services, Careers Services and lecture rooms.

Figure 5: Information View Screen Concept

Here the information panel is styled a little differently and the colour is changed to match the current floor’s colour.

The to and from buttons are also styled as a rounded square instead of circles, like the other buttons.

Result



Barrow Centre

Building Letter: F

Sports and Student Union.

Figure 6: Information View Screen Result

Concept

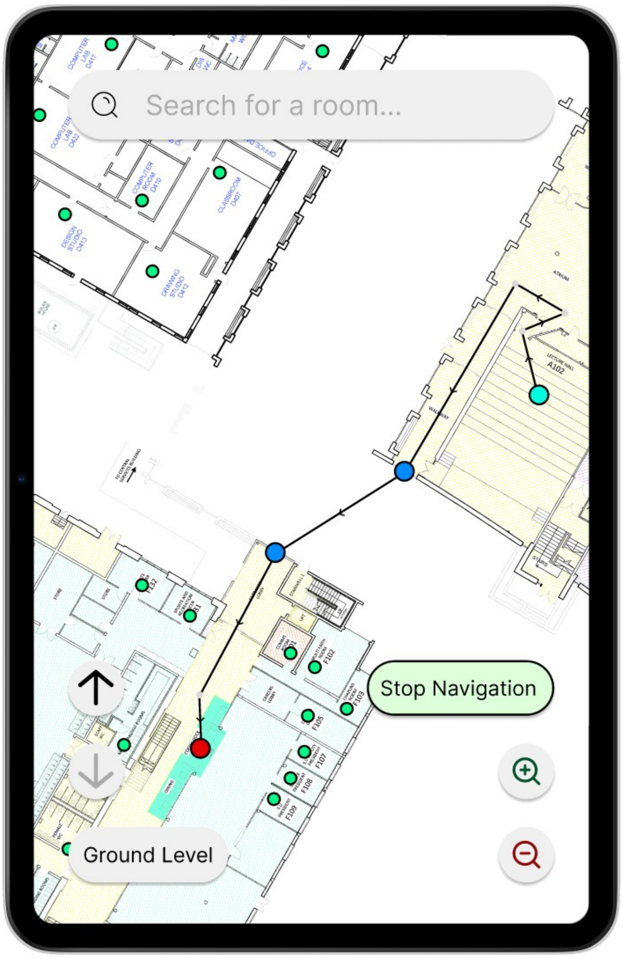


Figure 7: Navigation Screen Concept

As said in the first comparison the navigation button is moved under the search bars, along with a distance indicator showing the walking distance.

The colour of the route is changed to light blue to better show the dark blue arrows.

The icons for rooms and buildings and start and end structures is also changed.

Result

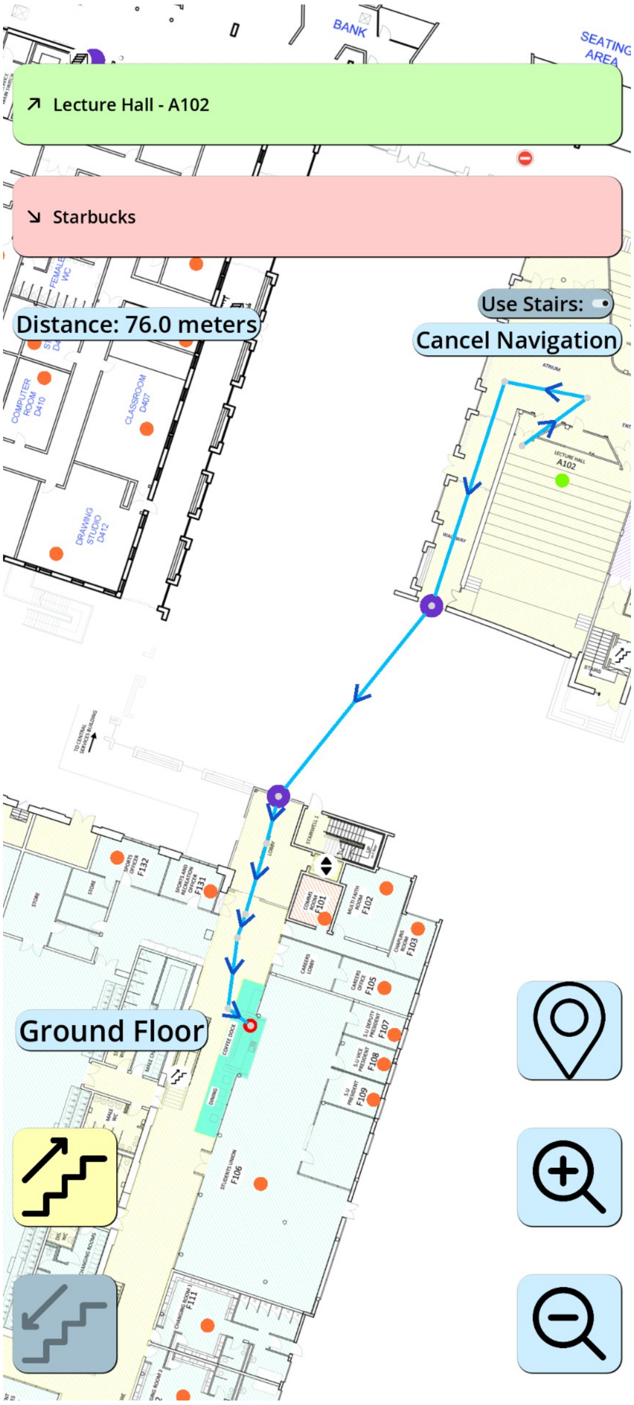
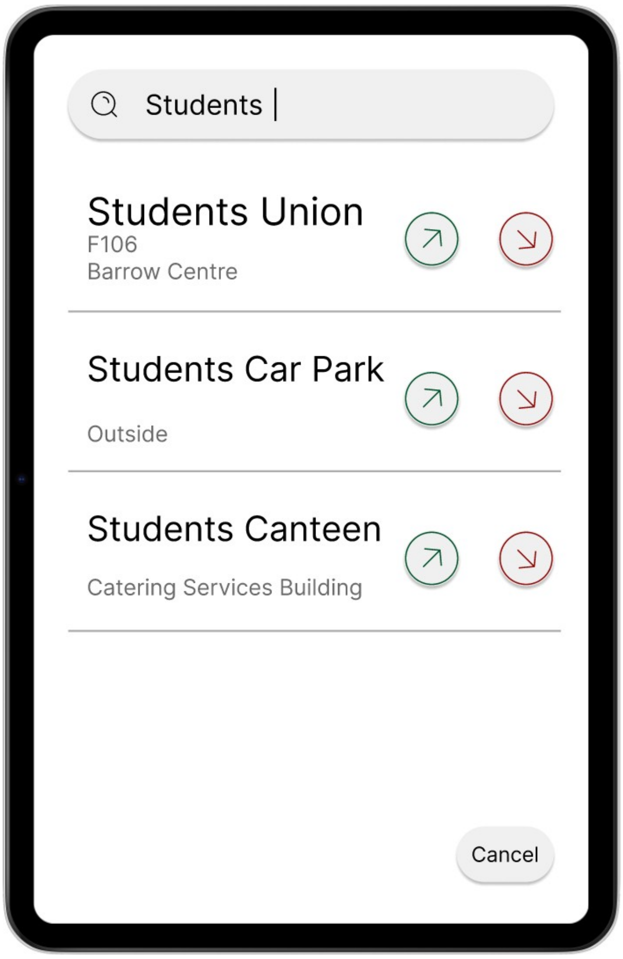


Figure 8: Navigation View Screen Result

Concept



Students |

Students Union
F106
Barrow Centre

Students Car Park
Outside

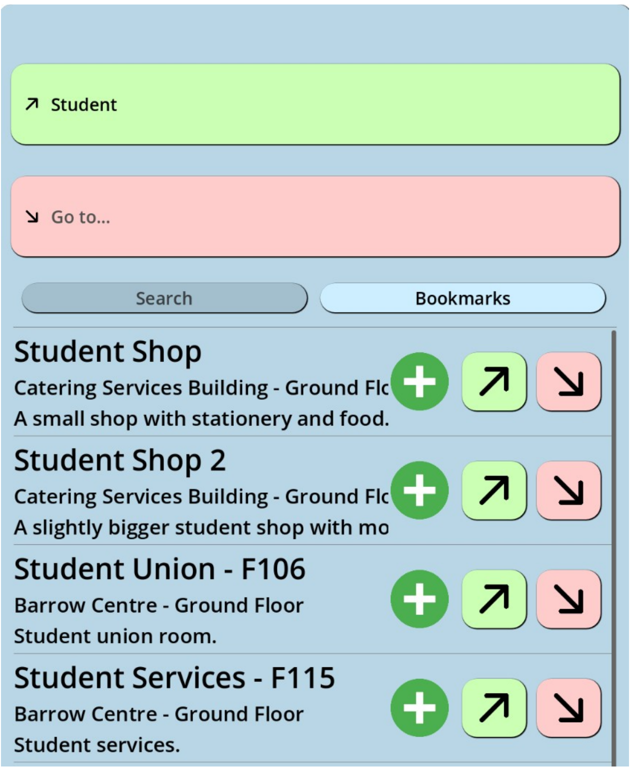
Students Canteen
Catering Services Building

Figure 9: Search Screen Concept

The main difference in the search screen is the addition of the bookmark buttons for adding and deleting bookmarks, and the bookmarks tab.

The cancel button was removed as the screen automatically closes once a structure is selected or the search bar is cleared.

Result



↗ Student

↘ Go to...

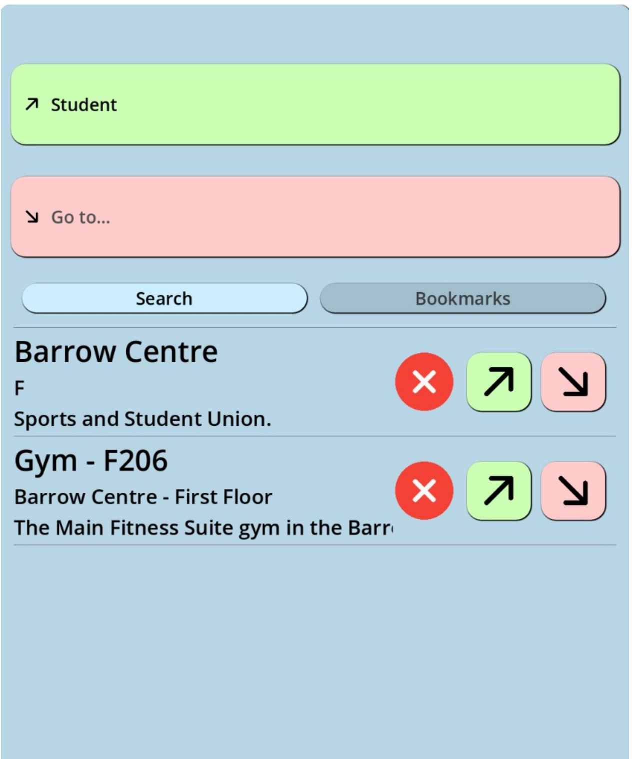
Student Shop
Catering Services Building - Ground Flc
A small shop with stationery and food.

Student Shop 2
Catering Services Building - Ground Flc
A slightly bigger student shop with mo

Student Union - F106
Barrow Centre - Ground Floor
Student union room.

Student Services - F115
Barrow Centre - Ground Floor
Student services.

Figure 10: Search Screen Result



↗ Student

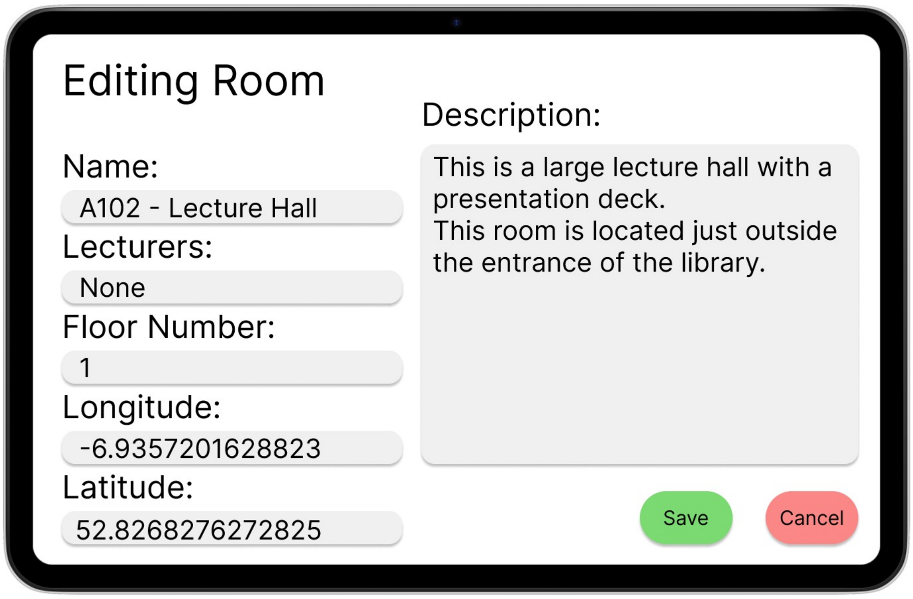
↘ Go to...

Barrow Centre
F
Sports and Student Union.

Gym - F206
Barrow Centre - First Floor
The Main Fitness Suite gym in the Barr

Figure 11: Bookmarks Screen Result

Concept



Editing Room

Name: A102 - Lecture Hall

Lecturers: None

Floor Number: 1

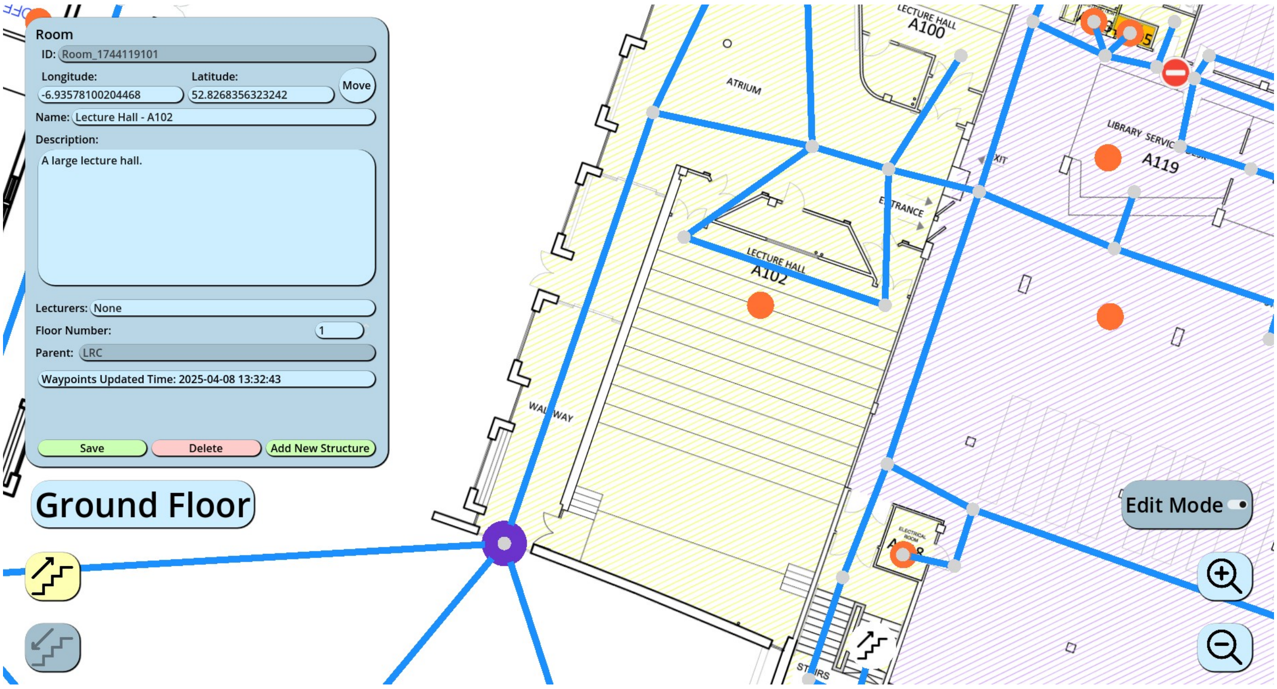
Longitude: -6.9357201628823

Latitude: 52.8268276272825

Description: This is a large lecture hall with a presentation deck.
This room is located just outside the entrance of the library.

Figure 12: Admin Room Editing Screen Concept

Result



Room

ID: Room_1744119101

Longitude: -6.93578100204468

Latitude: 52.8268356323242

Name: Lecture Hall - A102

Description: A large lecture hall.

Lecturers: None

Floor Number: 1

Parent: LRC

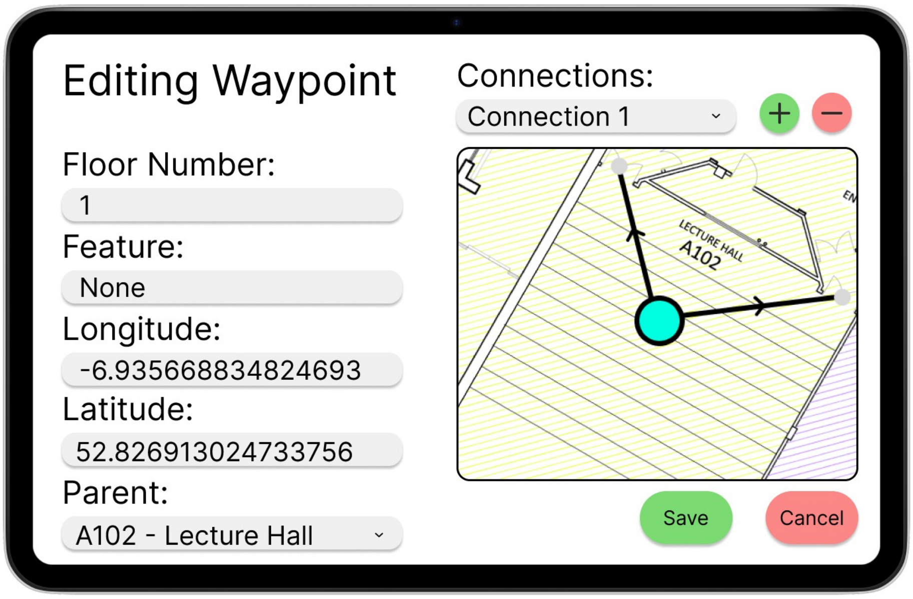
Waypoints Updated Time: 2025-04-08 13:32:43

Figure 13: Admin Room Editing Screen Result

The admin editing panel looks quite different with it being to the side allowing view onto the map while editing.

Other additions include showing the last modified timestamp, a move button using the mouse, a delete and add new structure buttons.

Concept



Editing Waypoint

Floor Number: 1

Feature: None

Longitude: -6.935668834824693

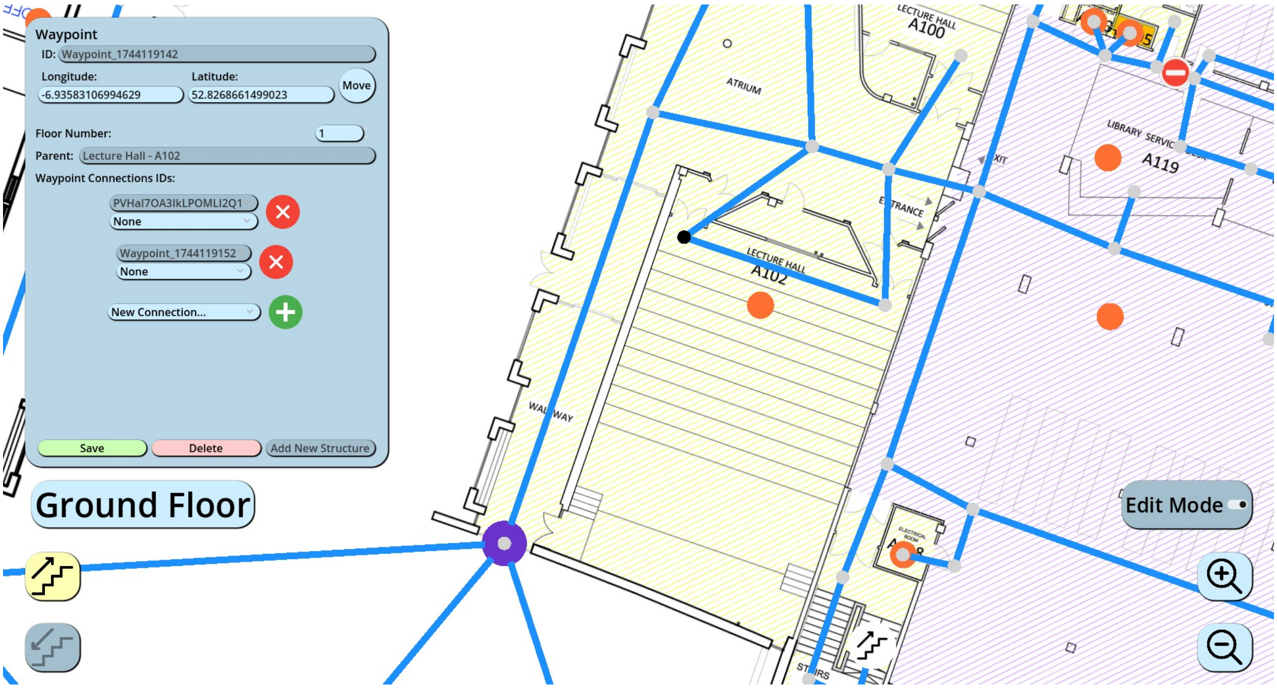
Latitude: 52.826913024733756

Parent: A102 - Lecture Hall

Connections: Connection 1

Figure 14: Admin Waypoint Editing Screen Concept

Result



Waypoint

ID: Waypoint_1744119142

Longitude: -6.93583106994629

Latitude: 52.8268661499023

Floor Number: 1

Parent: Lecture Hall - A102

Waypoint Connections IDs:

PVHal7OA3IkLPOMLI2Q1 — None

Waypoint_1744119152 — None

New Connection...

Figure 15: Admin Waypoint Editing Screen Result

This screen got changed in a similar way as other editing screen, with the addition of one extra change.

Instead the connections of the waypoint being a dropdown it is now a list of connections and features, such as stairs or closed.

Learning Outcomes

The following sections include not just the technical skills that have been acquired over the course of the project work but also the less measurable personal skills as well. The challenges section goes over the challenges encountered and how they were solved, and finally some improvements that could be made if the project was started again.

Technical

Godot Game Engine

The biggest part of learning was Godot itself. I have used it before for smaller programs, however learning about how to handle asynchronous calls in the engine, which the Firebase API (Application Programming Interface) used, was a new undertaking for me.

Another part of learning the engine was the use of platform dependant code and the exporting to different platforms. As the application developed was to be used on two separate platforms, web and Android, there were some differences in features and UI (User Interface) design. This allowed me to write code where the core functionality and methods were platform agnostic, reducing the need for development time and testing, and other sections where the functionality was specified for a platform allowing for extra features for the target platform.

Godot UI

Another big part of learning was the creation of the UI in the engine. I have not used the UI elements available in Godot before. This project allowed me to work on a UI that was a bit more complex and aimed at the two platforms.

I also got to work on two separate UIs, one for the everyday user of the map application, and the other one for the admin users to change and edit the map data. I learned to create the two UIs with ease of use and functionality in mind for both.

Google Firebase

Working with Firebase allowed me to learn about NoSQL databases in a practical hands on approach, about how Firebase stores data as documents and collections. I also got to learn about user credential management and access rules, as the users needed read only access whereas the admins needed read and write access to the database.

Pathfinding

As the core functionality of this application was the interactive pathfinding, learning about A* pathfinding, and weighted routing and implementing it was a learning experience for me.

Geographic Coordinate System

From the very start a core feature of the application was the use of the latitude and longitude values of waypoints, rooms and buildings. I have previously had an understanding of the system, however I learned much about how to represent the global coordinates on a 2D (2 Dimensional) plane.

Personal

Decomposition

As I have not worked on such a big project before on my own, planning and time management was a key role in creating a successful application. However the biggest benefit that I found was the use of decomposition for the features of the project. I found it much easier and enjoyable to work on the project when the tasks where broken down into small and easily manageable sections.

Organisation

My organisation skills were also improved as a result of meeting with my supervisor every week. I had to properly reflect on the work that I had done the previous week, and give an account of the progress made and any challenges that I have encountered.

Public Speaking

I also got to work on my speaking skills as a result of not just the weekly meetings with my supervisor, where I had to explain the work progress, but also at the presentations that we had to give on our projects.

Challenges

Firebase

The biggest challenge I had with Firebase from the very start is the basic nature of the add document feature. Currently in Firebase when a new document is added it is always empty, however as this project had four separate types of structures, the ability to have four separate template would have been great to have in Firebase. This challenge was overcome once the admin editing interface was functional and allowed me to speed up the editing of the map data.

Firebase API

The second challenge I had was not directly with Firebase itself but with the API for the Godot game engine. The problem was that any sort of errors that happened with adding, editing or deleting data on Firebase just printed out saying that an error has occurred with no indication on what the problem was, the data, authentication or network issues. However it is not known whether this is a limitation of the Godot code or the Firebase platform. This was all cleared up at the start of the project with the code written for all of the database access and modification, which allowed me to focus on other aspects later.

Map Projection

The next challenge was the problem of trying to represent latitude longitude coordinate system on a XY plane. A simple latitude to Y and longitude to X conversion wouldn’t work as the results get skewed. To solve this two methods were created to convert to and from latitude and longitude values using equirectangular projection. After this was implemented the waypoints, rooms and buildings were placed in their correct positions.

Android UI

Last major challenge found was when the application was first tested out on an Android phone. At this stage the UI design was close to being finalised however it was not usable on the phone as the font size and the UI elements were too small. This was due to the fact that the phones screen resolution is bigger than the laptop the application was developed on. To solve this the UI got reworked to better suit both platforms.

Differences Next Time

After working with the technologies a bit more in depth there would be a few things that I would do differently if the project was started over.

SQLite

First difference would be to use a SQLite database for storing the map data in the application. Currently all of the map data is stored in an associative array, which works correctly for the application. Throughout my testing of the application I have found that Godot has some sort of limit on the amount of items that can be contained in the associative array. To fix this potential problem I would use a local SQLite database which is able to hold more entries.

Database

As I had quite a lot of trouble working with the Firebase API due to the vague error messages I would choose a different database to work with. There is a drawback to this though as Firebase is the only API available for the Godot game engine. The alternatives to this would be to write a Godot API for another public database provider like MongoDB, or to host an own database and use HTTP requests.

Automated Testing

The third difference would be to make use of some automated testing. Godot natively does not have automated testing built into it. As I had avoided creating long and complicated methods and worked on small tasks one at a time I did not come across many regression bugs. When one was found I was able to quickly track it down due to having a tidy Git history with small commits.

Testing

Automated testing was not setup for this project as Godot natively does not have automated testing built into it. I however still did manual testing on the application throughout it’s development lifetime. During iteration 3 of the project I also handed out the application to other people to get feedback on the abilities of the application.

Manual Testing

While developing the application two types of testing was performed. The first one was testing each new feature or change being put into the application. As the tasks where each small in nature it was easy to test them both from a working and from a trying to break it perspective. Care was also put in if a method was changed that was used by a different functionality to perform regression testing on it.

The second type of testing performed was with using the application itself. As the website interface of Firebase is quite tedious to use and may lead to erroneous data being input, I used the admin editing part of the application. Throughout my usage of the admin editing I found quite a few annoyances and quirks with the application that only came to light after a bit of longer period of use. These user experience changes were then worked on to improve the application.

Public Testing

In iteration 3 of the project the application was in a usable state where the UI was close to finished and safety checks where in place. This allowed me to give out both the Android application and the web application for other people to try out and break for me.

When people were testing it out I took note of the way they use the application without interfering. This allowed me to see what sort of controls or usage was intuitive for people to use. After these tests the following changes where put in place to improve the application:

- Set a limit on how far the camera can be moved away from the centre of the map.
- Improved the touch sensitivity for mobile application.
- Made snap to current location button for mobile application.
- Improved the desktop editing UI and streamlined Firebase syncing.
- Added visual indicators for selecting structures for pathfinding.
- Added ability to select a structure in search bar and camera snaps to it.
- Various bug fixes.

FURPS

The following is a comparison between the FURPS metrics that were set out at the start of the project and the result of the completed application's abilities.

Functionality

Original Metric	Result
User able to select start and end point and route shown according to pathfinding.	User is able to select start and end points from either the map or from the search, the route is then shown correctly.
User able to search through all possible locations.	User is able to search through all room and buildings, for both to and from locations.
User able to see detailed information about a specific location, such as room name and description.	On selecting a room or building the information pop-up is shown.
User able to see whole map of Carlow campus of all buildings and rooms.	Building and room nodes are shown on screen with the floor plans as well.
Application able to update map data from remote server.	On startup the application checks for any updates in the Firebase database.

Usability

Original Metric	Result
UI must be easy to use without any explanations or tutorials.	Mobile controls are the same as other map applications. Desktop controls are similar as well, however there is a short explanation on startup for the map rotation controls.
Visual feedback such as a loading icon must be shown when doing processing such as the pathfinding or searching.	Loading icon is shown on startup for checking updates. Other messages are shown when updating the map data in admin mode.
Moving the map around on screen must be smooth.	Application is fast and smooth on both mobile and desktop.
Text information displayed must have high contrast against its background for better readability.	All text is coloured black with all backgrounds for text being light coloured.

Reliability

Original Metric	Result
The application must be able to work offline.	Application is able to work from download with the in built map data cache.
The application must display waypoints correctly according to the positioning system.	Waypoints, rooms and buildings are positioned correctly using equirectangular projection.
The start and end point of the route must be saved in case the application is closed.	Application saves the configuration of a route i.e. start, end, and whether to use stairs, which is read on startup.

Performance

Original Metric	Result
The start-up time of the application must not be more than 3 seconds.	Godot engine takes about 5 seconds on mobile, and 7 seconds on desktop website to load in the map application. Map application loads under 3 seconds, but may be longer depending on the network connection to the Firebase database.
The searching functionality must show a result in 2-3 seconds.	Searching is near instantaneous on both mobile and desktop.
Pathfinding from start to end point must take no more than 5 seconds.	Pathfinding is near instantaneous on both mobile and desktop.

Supportability

Original Metric	Result
Location and description of waypoints and rooms must be editable.	Full admin editing of all structures.
The extension of new waypoints and rooms must not need changes to the underlying pathfinding algorithm to work.	Upon editing structures and connections the pathfinding algorithm works as expected.

Conclusion

Even though not all of the features that were planned at the start of the project for the map application were implemented, all of the core features were successfully finished and implemented. Out of the non-core features the majority were also implemented, the reason being that not all of it was finished was due to the fact that different ideas and features were thought up during development, which were quicker to develop and increased the functionality or user experience of the application.

In the end even though more time could be put in to continue improving the features, especially the Android location tracking, and more optimising could be done to help speed up the loading of the application, the map application overall is a success at performing its core functionality of helping people be able to search for locations and help them navigate to them.

References

Godot Project (2025) Godot Game Engine. Available at: https://godotengine.org/ (Accessed: 23 April 2025)

PraxisMapper (2025) GodotGPSPlugin. Available at: https://github.com/PraxisMapper/GodotGPSPlugin (Accessed: 23 April 2025)

GodotNuts (2025) GodotFirebase. Available at: https://github.com/GodotNuts/GodotFirebase (Accessed: 23 April 2025)

Google (2025) Cloud Firestore. Available at: https://firebase.google.com/docs/firestore (Accessed: 23 April 2025)

Brilliant.org (2025) A* Search. Available at: https://brilliant.org/wiki/a-star-search/ (Accessed: 23 April 2025)