A Web-based Multi-criteria Bicycle Route Planning Application

Jake Bailey

2002753

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Science
of the
University of Portsmouth.

School of Computing
Engineering Project

2024

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Date: 2024

Abstract

Cyclists regularly use route planning applications to determine the route they take for their journey, whether to commute, complete a short-circuit ride or a long-distance ride. There is a range of pre-existing applications to allow cyclists to do this; however, most of these have a limited range of customisable criteria considered in the routing algorithm.

This dissertation will investigate how the current solutions consider user-customisable criteria in their routing algorithms and will provide an open-source prototype system. The proposed system will consider pre-existing data, such as weather forecasts and hazard data, whilst catering for custom criteria chosen by the user and location of accommodation for long-distance routes.

Acknowledgements

Much stuff borrowed from elsewhere

Consent to Share

I consent for this project to be archived by the University Library and potentially used as an example project for future students.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

Route planning is essential to all cyclists, from casual to professional, as it's important to know where they are going and what to expect on their journey. A casual rider may wish to find the quickest and safest route to their destination. In contrast, a professional rider would want to find the most challenging route, not necessarily the quickest.

Currently, there are multiple route planning applications which range in their flexibility for different types of cyclists and their needs. No one application effectively caters for all cyclists, meaning multiple applications would be required depending on the type of route the user wants to plan.

The proposed solution is a web application enabling cyclists to plan a route ahead of their ride and choose which criteria they want to affect the routing algorithm and, therefore, the final route. Some examples of these criteria are accommodation, road type and elevation. Other criteria will be used in determining the route, such as weather conditions and hazards along the route.

1.2 Aims and Objectives

The overall aim of the project is to develop and build a prototype which can be further developed in the future to plan a range of routes for cyclists with a range of data-driven criteria to cater to the requirements of that specific user. Enabling customisable criteria for each user will allow them to understand the decisions made to plan their route and improve the user's experience when they are cycling, and give more members of the public the incentive to begin cycling.

1.3 Proposed Solution

The solution is a prototype web application designed for cyclists of all levels (casual to professional). It will plan cycling routes based on a range of data-driven, user-customisable and fixed criteria through a user-friendly, modern user interface (UI). Furthermore, the solution will provide key information on the planned route, such as elevation, average speed and time to complete the route.

1.4 Scope

There is a potentially large scope that could be attributed to this project whereby the application could be developed to consider a range of different sports that could benefit from route planning. These sports could be, running, walking and mountain cycling, which would increase the overall scope as there would be many other factors to consider for these sports. This would expand the current scope of the project to a great scale.

To mitigate the risk of the scope becoming unfeasible, it's vital to focus on the key functionality planned for the application. Therefore, the decision has been made to narrow the scope and focus explicitly on catering the application to route-planning Road Cyclists. Focusing on this user-base will ensure focus on the key functionality to provide effective route planning on roads catering to the user's needs.

1.5 Deliverables

This project will consist of two deliverables:

- The project report
- The artefact (web-based bicycle route planner)

1.6 Resources and Facilities used

The section demonstrates all resources and facilities used to complete this project. It includes developing the prototype and writing this document; the list should aid anyone wishing to re-create this project. Some decisions on tools used are personal; for instance, the Integrated Development Environment (IDE), which is predominantly down to preference. On the other hand, the programming languages chosen are specific to the features and performance benefits they provide in developing such applications.

- Applications/websites used to complete research and literature review
  - Google Scholar
  - EBSCO Database
- Programming Languages

  - Javascript
  - HTML5
  - CSS
  - Go
  - Structured Query Language (SQL)
- Integrated Development Environment/Text Editor
  - Visual Studio Code
- Database Management System (DBMS)
  - PostgreSQL
- Wireframes and Prototyping
  - Figma
- Libraries/Frameworks
  - React.js
  - Vite
  - Bootstrap.js
  - Gin
  - Auth0
  - Open Route Service (ORM)
  - OpenStreetMap (OSM)
  - Leaflet
  - Leaflet Routing Machine
- Testing
  - Jest
  - Go Test Framework
- Browser
  - Google Chromium
- Project Management and Version Control
  - GitHub

- GitHub Interactive Kanban Board
- GitHub Issues
- Git

1.7 Risk Assessment

It is vital to understand the potential risks at the beginning of the project to establish a way to mitigate those risks should they occur. The risks have been identified when considering this project (see Table 1.1, p15).

Table 1.1: Project Risk Assessment

Type	**Risk**	**Likelihood**	**Severity**	**Description and Mitigation**
Personal	Illness	Low	Medium	There is the chance that I may fall ill unexpectedly during the project. To mitigate the risk of this visit a doctor if unwell and allocate time for rest around university work.
	Supervisor illness	Medium	Medium	The supervisor becoming unwell in any instance could impact the completion of tasks due to the lack of technical guidance and may delay the project's timeline. If this occurs, we will conduct online meetings and plan time in preparation for the potential delay.
Management	Changing requirements	Medium	Medium	During the project, the requirements initially suggested may change. To mitigate this, communicate regularly with the supervisor to discuss the feasibility of the current requirements to prevent any future changes.

	Time Availability	Medium	Medium	The time to dedicate to the project may become more limited as I have more work to complete for other modules. To prevent this, plan tasks ahead of time to consider potential future setbacks.
Technical	Data loss/corruption	Low	High	Data loss would mean the prototype must be developed again from scratch. To mitigate this issue, create regular backups on GitHub and locally in case either fails.
	Hardware failure	Low	High	PC/Server/Laptop being used for development/hosting crashes/hardware gets damaged. Ensure multiple devices are being used so development can continue even whilst one device is down.
	Documentation availability	Low	Medium	Documentation for frameworks, languages or libraries is unavailable. There is a range of readily available online for all the required technologies.

1.8 Legal, Ethical, Social and Professional issues

The key legislation to consider for this project is the Data Protection Act 2018 (DPA 2018). The project should not store personal information. However, the user's current location will be requested upon the launch of the application; when the application no longer uses this data, it will be deleted and re-requested when the user enters the application again. Future iterations could implement accounts, storing a small amount of sensitive user information to include more features. However, the submitted artefact shouldn't contain this data; regardless of this fact, it will be ensured the artefact abides by all principles of the DPA 2018 due to it handling the current location data of the user.

One social issue that could arise is that the artefact may entice more public members to start cycling more frequently; whilst this result will be a great incentive for protecting the environment, some road users are cautious, with many cyclists riding unsafely on the roads. The artefact will push users to ride safely and abide by all road safety laws, just as vehicles do; there will also be the option only to use cycle routes/lanes when plotting a route to ensure those cyclists who are less comfortable on roads still feel safe on the routes planned by the artefact.

Chapter 2

Literature Review

This chapter presents the analysis and review of relevant literature conducted throughout this project. The research comprises of literature surrounding different routing algorithms and their current usage within current route planner applications. It also explores other relevant technologies which will be utilised in developing the artefact, such as web frameworks and programming languages.

2.1 Background

This project has a high level of complexity. It utilises custom, user and system inputs into a data-driven route planning algorithm, displaying the output on an OpenStreetMap-based web application.

This review begins with researching literature on different web technologies and programming languages to develop an understanding of what technologies were available to build the proposed system. Once an understanding of the technologies was developed, a review of current competing products was reviewed. This allowed for gaps in the current market to be identified, therefore developing an idea of requirements for the proposed system (see Section 2.3, p18).

Furthermore, research into how cyclists consider different risk factors when planning a ride was key in understanding what routing algorithm was implemented to fit the project's requirements best. An in-depth review of existing literature was conducted to understand two primary risk factors for cyclists: cycling infrastructure (see Section 2.4.1, p19) and weather conditions (see Section 2.4.2, p21).

2.2 Research Methods

An initial exploration of sources and subject areas was conducted using Google Chrome to comprehensively understand the topic at hand. Certain regions of interest were also highlighted through crowd-sourcing ideas from knowledgeable individuals. After these areas were outlined, in-depth research was conducted primarily through Google Scholar and the University of Portsmouth EBSCO database. The Zotero Chrome plugin and app managed citations and bibliography items (“Zotero | Your personal research assistant”, n.d.). All bibliography items have also been stored in a CSV file, utilising Zotero’s export functionality; doing so ensures that all relevant sources can be re-visited at any time and are not lost after this project has concluded.

2.3 Competing Products

There are many different route planners available with a range of different features, some common between applications and others specific to one. Applications like Plotaroute.com (“Free Route Planner for Outdoor Pursuits - plotaroute.com”, n.d.), Komoot (“Komoot | Find, plan and share your adventures”, n.d.) and Google Maps (“Google Maps”, n.d.) have some commonalities, however serve slightly different purposes (see Figure 2.1, p18).

		Application Name										
Desirability (1-5)	Features	Plotaroute.com	Routeyou.com (Paid)	Komoot.com	Cycle.travel	Cyclestreets.net	Maps.google.com	Web.bikemap.net (Paid)	Ridewithgps.com	OSMaps(Paid)	National Cycle Network	Number of Applications with Feature
5	Cycling Infrastructure Issue Waypoints											0
3	Hotels/Campsites on Rt				X							1
3	Measure Section	X										1
2	Annotate	X										1
2	Repeat Route (laps)	X										1
2	Make Route (auto generate)	X										1
1	Repeat Loop (add loops to route)	X										1
1	Plot Radius from Pt	X										1
1	Combine Route	X										1
5	Key Tourist Pts		X	X								2
3	Weather Forecast									X	X	2
3	Shorten Route	X	X									2
2	Replot Section	X										2
2	Split Route	X							X			2
1	Tabletop 3D Augmented Reality (iOS only)									X	X	2
4	Fitness Level		X	X		X						3
3	Social Sharing	X	X	X			X					4
5	Different Vehicles	X	X	X			X	X				5
4	Import Route	X	X						X	X	X	5
4	Round Trip	X	X	X					X	X		5
3	Alt Routes		X	X	X	X	X					5
5	Route Terrain Summary		X	X	X				X	X	X	6
5	Instructions	X	X	X	X	X	X	X				7
4	Reshape Route (Drag Anchor Pts)	X	X	X	X	X	X	X				7
3	Trace Route	X	X	X	X	X		X	X			7
2	Print		X	X	X		X	X		X	X	7
5	Export Route GPX/KML/TCS/FIT	X	X	X	X	X		X		X	X	8
5	Snap to Map (fix GPX Errors)	X	X			X	X	X	X	X	X	8
5	Change Start/End Pt	X	X	X	X	X	X	X	X			8
4	Export Route (save online)	X	X	X	X	X	X	X	X			8
4	Reverse Route	X	X	X	X	X	X	X	X			8
4	Delete Section	X	X	X	X	X	X	X	X			8
3	Manual Route Planning	X	X	X	X	X		X	X	X		8
5	Elevation Plot	X	X	X	X	X	X		X	X	X	9
4	Map Layers	X	X	X	X	X	X	X		X	X	9
	Total Number of Features	26	22	19	15	14	13	13	12	11	10	

Figure 2.1: Table of current solutions and their included features

2.3.1 Plotaroute.com

Plotaroute.com (“Free Route Planner for Outdoor Pursuits - plotaroute.com”, n.d.), now referred to as Plotaroute, contains a wide range of features, including nearly all features highlighted in Figure 2.1 (see Figure 2.1, p18). The main shortfall of Plotaroute was identified as its UI (User Interface) rather than the features included. The UI of Plotaroute is very cluttered due to the number of features present in the application (see Figure A.1, p59). Unless the user is an expert and has used the application before, it is initially confusing what each part of the application does. Due to this, at first glace, it's unclear what type of route planner Plotaroute is, which will fundamentally affect a user's initial decision on whether or not to use the application.

2.3.2 Komoot

Komoot ("Komoot | Find, plan and share your adventures", n.d.) offers a simpler-looking yet feature-rich application for planning and discovering routes (see Figure A.2, p60). The application has a key focus on community, whereby a user doesn't necessarily need to plan a route, they can simply discover a route or even share a route of their own. This functionality allows the user to require minimal effort when planning location-specific rides, however doesn't offer discovery of longer routes, for example, Land's End to John O'Groats. When compared to Plotaroute, Komoot is clearly more user-friendly without sacrificing the core features needed by users. One primary setback with Komoot, however, is that some functionality for route planning is part of a paid-for service, therefore locking certain user bases out of some key desirable functionality.

2.3.3 Google Maps

Google Maps ("Google Maps", n.d.) further reduces the specific functionality and offers a very simple, multi-functional route planning and location finding application (see Figure A.3, p60). Google Maps is likely the most user-friendly out of all the applications, simply due to its simplicity and consistency across other Google applications ("Material Design", n.d.). With this simplicity, however, most cycling-specific functionality is not present; the only option the user has when calculating a route is what the Google routing algorithm calculates with potentially a few alternate routes. Therefore, limiting how much the user can customise their route.

2.4 Risk Factors in Route Planning

Risk-based cycling route planning requires extensive knowledge of the impact of cycling and transportation infrastructure currently in place. It is also critical to understand how other external factors impact the risk of a route on an ever-changing basis. Within this section, a range of risk factors were explored to understand how multiple risk factors can be implemented into route planning algorithms.

2.4.1 Cycling Infrastructure

The cycling infrastructure along a route must be understood because it is common for cyclists to share the same infrastructure as motorised vehicles. However, a cyclist has no physical protection if a crash occurs (Reynolds et al., 2009). There is often purpose-built infrastructure for cyclists, whether bike lanes alongside shared roads or off-road bike paths and this segregated infrastructure can help improve the safety of a route for a cyclist.

		Application Name										
Order of Preference	Features	Plotaroute.com	Routeyou.com (Paid)	Komoot.com	Cycle.travel	Cyclestreets.net	Maps.google.com	Web.bikemap.net (Paid)	Ridewithgps.com	OSMaps(Paid)	National Cycle Network	Number of Applications with Feature
1	Round Trip	X	X	X					X	X		5
2	Export Route (save online)	X	X	X	X	X	X	X	X			8
3	Hotels/Campsites on Rt				X							1
4	Different Vehicles	X	X	X			X	X				5
5	Manual Route Planning	X	X	X	X	X		X	X	X		8
6	Elevation Plot	X	X	X	X	X	X		X	X	X	9
7	Cycling Infrastructure Issue Waypoints											0
8	Measure Section	X										1
9	Map Layers	X	X	X	X	X	X	X		X	X	9
10	Navigation	X	X	X	X	X	X	X	X			8
11	Annotate	X										1
12	Modify Route (Drag Anchor Pts)	X	X	X	X	X	X	X				7
13	Repeat Loop (add loops to route)	X										1
14	Split Route	X							X			2
15	Replot Section	X										2
16	Alt Routes		X	X	X	X	X					5
17	Export Route GPX/KML/TCS/FIT	X	X	X	X	X		X		X	X	8
18	Import Route	X	X						X	X	X	5
19	Shorten Route	X	X									2
20	Social Sharing	X	X	X			X					4
21	Make Route (auto generate)	X										1
22	Print		X	X	X		X	X		X	X	7
23	Snap to Map (fix GPX Errors)	X	X			X	X	X	X	X	X	8
24	Repeat Route (laps)	X										1
25	Weather Forecast									X	X	2
26	Delete Section	X	X	X	X	X	X	X	X			8
27	Route Terrain Summary		X	X	X				X	X	X	6
28	Combine Route	X										1
29	Fitness Level		X	X		X						3
30	Plot Radius from Pt	X										1
31	Reverse Route	X	X	X	X	X	X	X	X			8
32	Key Tourist Pts		X	X								2
33	Trace Route	X	X	X	X	X		X	X			7
34	Instructions	X	X	X	X	X	X	X				7
35	Tabletop 3D Augmented Reality (iOS only)									X	X	2
	Total Number of Features	26	22	19	15	14	13	13	12	11	10	

Page

Figure 2.2: Features in priority order based on user feedback

Furthermore, Hong states how investing in effective cycling infrastructure "mitigates the negative effects of poor weather conditions" (Hong et al., 2020), which further demonstrates that good, known infrastructure is key to improving the physical and perceived safety of a route in a range of different weather conditions.

Furthermore, crowd-sourced data from route planners, cyclists and fitness applications such as Strava Routes (“Strava | Running, Cycling & Hiking App - Train, Track & Share”, n.d.) have been key in developing new infrastructure. Boettge states how the most accurate assessment of a cycle network would come directly from the cyclists who use the network (Boettge et al., 2017). Cyclists who use the network are the most familiar with the quality of each route and how traffic conditions improve the safety of the route. Utilising the GPS information from route planners and fitness tracking applications alongside direct input from cyclists can help build new routes and improve pre-existing routes, therefore preventing injuries and high-risk situations by modifying transportation infrastructure (Reynolds et al., 2009).

Areas with little to no cycling infrastructure, such as busy roads and roundabouts, force cyclists to have a heightened attentiveness that other road users don’t have to consider due to not only the physical danger but the cyclists’ perceived danger (Doorley et al., 2015). These risks should be considered within route planning to decrease the number of ’risk’ areas along a cyclist’s route whilst also giving local areas the incentive to make

Measure	Netherlands	Edinburgh	Cambridge
Consistency	4.54	2.17	2.83
Directness	4.29	2.17	3.83
Attractiveness	4.28	2.22	3.56
Road safety	4.41	2.38	3.5
Comfort	4.46	2.33	3.33
Spatial integration	4.43	3.29	3.86
Experience	4.50	2.20	4.40
Socio-economic value	4.67	3.33	3.67
Overall Score	4.45	2.51	3.62

Figure 2.3: Comparison of the bicycle infrastructure scores (Hull and O'Holleran, 2014).

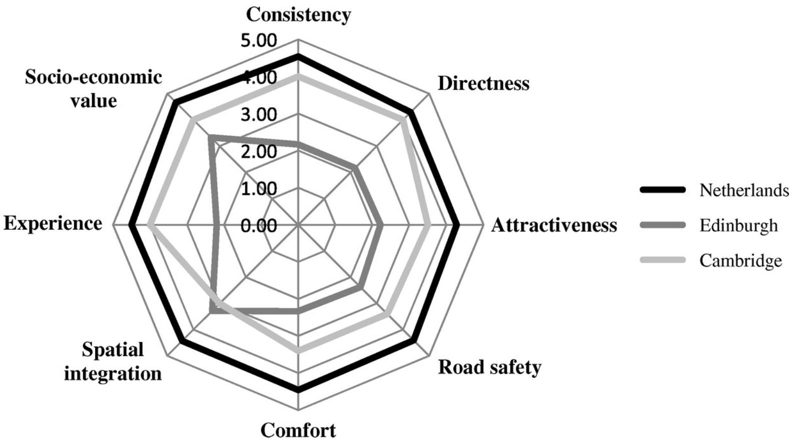


Figure 2.4: Spider web diagram comparing the Bicycle Infrastructure Scores (Hull and O'Holleran, 2014).

infrastructure modifications to decrease the number of 'high-risk' points. Therefore, in the long-term, it will mitigate the need for constant action by cyclists to ensure their safety, which will, in turn, influence individuals' decisions to cycle (Reynolds et al., 2009).

Hull and O'Holleran also demonstrate how cities with a high reputation among cyclists also have safer roads and more attractive infrastructure. The Netherlands scored relatively equally amongst all categories, in comparison to cities with less of a reputation and, therefore, a lower standard of cycling infrastructure (see Figure 2.3, p21) (see Figure 2.4, p21) (Hull and O'Holleran, 2014). This supports how Reynolds et al. further illustrate how investing in cycling infrastructure will greatly incentivise individuals to cycle due to the decreased risk.

2.4.2 Weather Conditions

g Weather conditions will also have a pivotal effect on how a route planner will calculate the safest route for a cyclist. Following on from Cycling Infrastructure (see Section 2.4.1, p19), it is demonstrated how a lack of good infrastructure goes hand-in-hand in creating an unsafe route alongside the weather. To ensure the safety of cyclists, all routes and road surfaces must be maintained to withstand different weather conditions (Shoman et al., 2023).

The weather also impacts a cyclist's likelihood to ride; Flynn states that cyclists 'were nearly twice as likely to commute by bicycle when there was no morning precipitation' (Flynn et al., 2012). It is clear that even minor changes in the weather can drastically affect a cyclist's decision to ride, further demonstrating how vital the perceived safety of cycling is in deciding whether to ride.

Contrasting this, Hull and O'Holleran state that the main environmental barriers included too much traffic, too many hills, no bike lanes/trails, no safe place to cycle and badly maintained streets (Hull and O'Holleran, 2014). Therefore suggests that the weather should have a minimal impact on a cyclist's decision to ride if the infrastructure is sufficient. Despite the findings of Hull and O'Holleran, it seems to be a common finding that the perceived safety of cycling, both in regard to the changing weather conditions and cycling infrastructure, is the primary factor in choosing cycling over an alternative method of transport. Miranda-Moreno and Nosal have shown how when infrastructure is implemented, there is generally an increase in total bicycle usage and diversion of cyclist flows away from roads to purpose-built infrastructure even in less ideal weather conditions (Miranda-Moreno and Nosal, 2011).

2.5 Conclusion

To conclude, route planning with different customisable preferences has been implemented by a range of different existing organisations; however, focusing on a risk-based routing approach has not been addressed by these existing solutions. Utilising pre-existing routing algorithms such as Open Route Service ("Openrouteservice", n.d.) or Open Source Routing Machine ("Project OSRM", n.d.) and integrating custom, weather and infrastructure data alongside the usual user-preferences has not been implemented within existing solutions. Therefore, this enables a unique system to be developed whereby crowd-sourced infrastructure data alongside weather data provided by OpenWeatherMap combined form a risk index utilised in a customised routing algorithm.

Furthermore, in order to develop this system, React.js was chosen to develop the front end and Go for the back end. Next.js was initially considered for the front-end. However, it was later found that Next's server-side rendering was not supported by Leaflet; used for Mapping with OpenCycleMap; ("Leaflet — an open-source JavaScript library for interactive maps", n.d.;"OpenCycleMap.org - the OpenStreetMap Cycle Map", n.d.) due to it requiring direct interaction with the DOM. Go with the Gin Web Framework ("Gin Web Framework", n.d.) was chosen to develop the API and back-end due to its increased performance benefits over alternative languages such as Node.js with Express.js.

Chapter 3

Methodology

Choosing which Software Development Life Cycle (SDLC) methodology is a key decision at the beginning of any software development project; the methodology demonstrates the expected route that development will take during the project's lifetime. I have decided to use the Incremental methodology whilst integrating key Project Management methods in other team-focused methodologies, such as Agile (see Chapter 4, p24).

3.1 An Incremental Development Methodology

I have chosen the Incremental Development Model for this project since the Waterfall Model cannot precisely and completely describe the real software development life cycle (Dapeng Liu et al., 2011). Each iteration will represent a full software life cycle vaguely following the waterfall methodology's structure: Requirements analysis, Design, Development, Testing, and Release. Incremental development allows for more flexibility during the software development process. It is feasible for a solo-development project as it does not require collaboration with other team members as the Agile methodology does.

The Incremental model breaks larger tasks into smaller, more achievable sub-tasks/increments. Each task can be broken down into all or some of the stages mentioned earlier. Therefore, if some stages are unnecessary for an increment, they don't need to be followed. Development of the increments can be managed using a Kanban board. It can also manage progress in completing this document. To see the added benefits of using Kanban, see Section 4.2.

There are some downsides to using the Incremental model, with one key failure of the model being merging changes between increments. This downfall of the model introduces a discontinuity of design purpose where the user interface and programming interfaces become discontinuous between increments (Dapeng Liu et al., 2011). To mitigate this issue, a consistent programming interface be implemented to aid in developing easy-to-read source code throughout the development of increments.

Chapter 4

Project Management

4.1 Methodology

As stated throughout (see Chapter 3, p23), the chosen methodology for this project is the Incremental SDLC Model. The project will be split into increments to break down larger tasks into smaller, more manageable subtasks. Each subtask will have its deadline to contribute to the changes within that increment. There will be regular incremental changes during the development and implementation of the artefact. All tasks to be completed will be managed with GitHub projects utilising its Kanban, Table and Milestones functionality (see Section 4.2.1, p24).

4.2 GitHub

4.2.1 Kanban Board and Projects

From the beginning of the project, a GitHub project was created alongside the GitHub Repository to manage all tasks to be completed for this document and the development of the artefact. All tasks to be completed were added to the backlog at the beginning of development and allocated a milestone and label to categorise each task.

Milestones were created for each Increment of the prototype and for each section of this document to be completed. This would allow tasks to be assigned to each milestone to better manage which task belongs to which overall objective. Each task would then be assigned a 'To Do' status when selected for development/writing and would progress throughout the other Kanban Board stages until it is marked as 'Done' and closed.

An extra status has been added to the Kanban board with the name 'In Review'; tasks placed in this status were awaiting guidance from the project supervisor (see Section 4.3, p25).

4.2.2 Version Control

GitHub will host the project source code with Git Software Version Management (SVM). Using GitHub and Git enables branches to be created in the repository to manage code changes and link each change to a pull request which enables incremental development. Branches will allow for branch-level testing and pull requests for automated merges once the increment has been completed. These merges will also throw merge conflicts if issues in the code arise between increments, ensuring the code base remains accurate and forcing the conflict to be resolved before changes are made. Commits are also highly-valuable during development as they allow code changes to be managed and reversed if needed.

4.3 Supervisor Meetings

Meetings will be scheduled on a weekly or fortnightly basis; my supervisor allocates a range of slots each week where I can book a time which works for me. The aim is to have all meetings face-to-face where possible. However, if the supervisor or I become unwell, we will conduct a video conference where possible. If the supervisor has planned leave, or I will be away for some time, we will communicate ahead of time to devise a plan for the project while one or the other is away and unable to meet regularly. Doing so will further allow me to plan my workload ahead of time and effectively keep track of the progress throughout the project.

[[INSERT LIST OF MEETINGS]]

Chapter 5

Requirements

The identification of requirements was critical to the success of the project. Primary research was undertaken to gather an understanding of real user preferences from a list of proposed functionalities as to understand their wants and needs of the application. It was critical to declare clear and concise requirements to ensure absolute clarity throughout the project. This chapter discusses all methods undertaken to gather requirements, including the final requirements found as a result of this process.

5.1 Requirements Gathering

Requirements were gathered via multiple channels, initially research was undertaken to understand the offerings of current solutions to gather a list of potential functionalities. These were then taken to the client to set out a base set of user requirements to be used in primary research. A simple research application was developed to collect the opinions of cyclists in the general public, enabling participants to order a subset of a wider list in order of preference (see Figure 5.2, p28).

5.2 Identifying Users

The aims highlighted the need to expand the route planning functionalities available to cyclists (see Section 1.2, p11). Multiple user groups were identified from this:

- Commuter - A commuter would require a simple round trip route, likely focused on ease of ride and speed taken to commute.

- Regular Cyclists - A cyclist who focuses on planning a ride tailored for their skill level. Wants multiple route options with varying levels of intensity.

- Pro Cyclists - A pro cyclist would require a deeper level insight into each planned route, including a range of customisability and integration with fitness applications.

5.3 Requirements Specification

User stories were established through various methods: regular face-to-face meetings were conducted with the client and research was conducted through a web application built to understand the users' preferred functionality (see Figure 5.2, p28). Each user story was broken down into specific system requirements. The requirements have been ranked using the MoSCoW model; this demonstrated each requirements' level of importance aiming to prioritise the necessary requirements, limiting the time spent on the less-necessary requirements. The MoSCoW model seemed the most appropriate choice in prioritising requirements because of the research undertaken with real-world users, utilising the data collected on user preferences (see Figure 5.1, p27)(see Figure 2.2, p20).

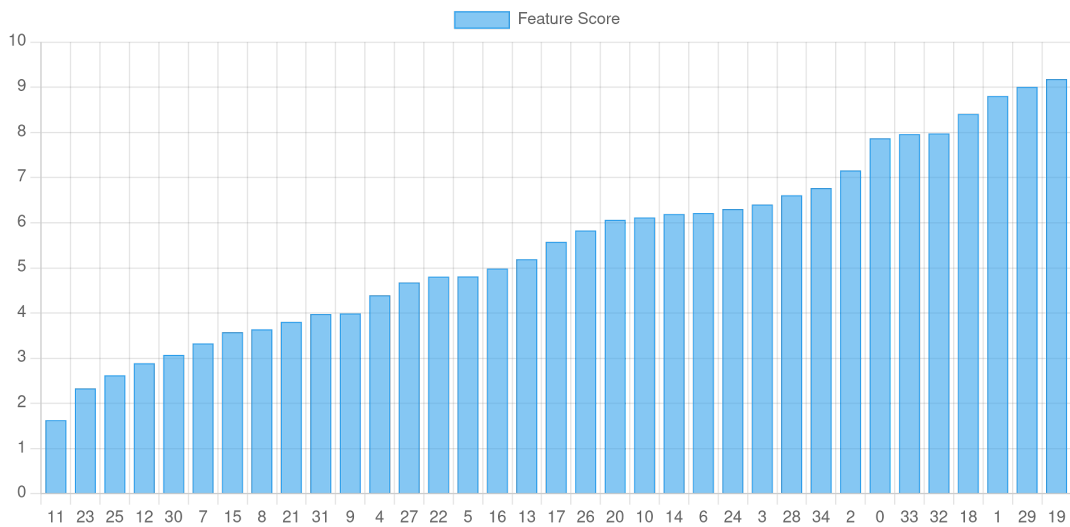


Figure 5.1: User Feedback

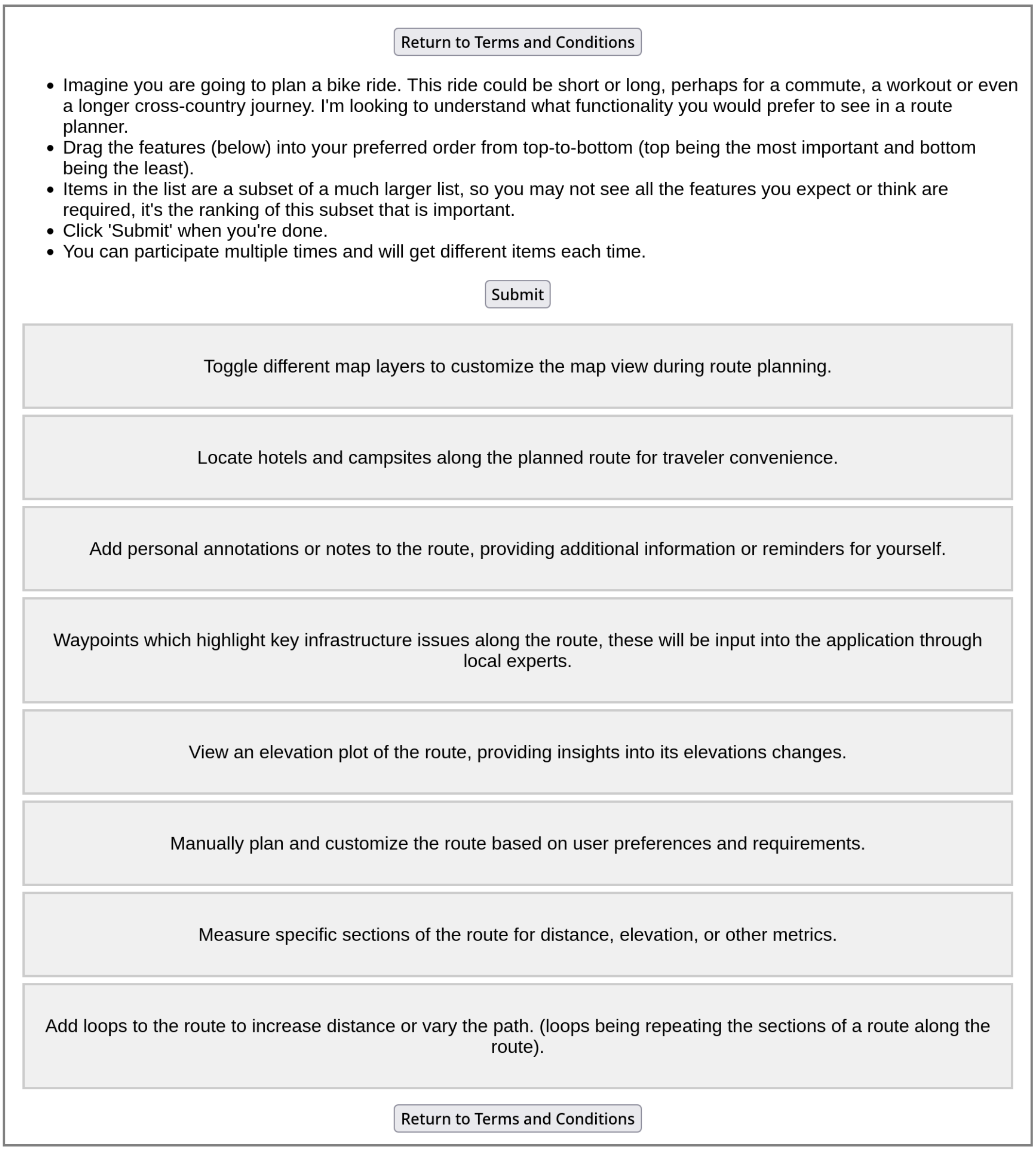


Figure 5.2: Firebase Research Application

5.4 Elicited Requirements

The elicited requirements were a direct result of discussions with the client and the primary research undertaken (see Section 5.3, p27). Each user story was divided into multiple acceptance criteria/system requirements and were segregated into a table per user story.

Table 5.1: User Story 01

As a user, I want a page that allows me to configure my starting and destination location to plan a route.		
ID	Acceptance Criteria / System Requirements	Priority
SR1	The system must provide a route configuration panel.	Must
SR2	The route configuration page must provide a starting and destination location input field.	Must
SR3	The route configuration page should suggest accurate geolocations based on the location inputs.	Must
SR4	The route configuration page must determine the geolocation based on the user input.	Must
SR5	The route configuration page must plan the route once two or more locations are input.	Must

Table 5.2: User Story 02

As a user, I want to change preferences to allow me to customise the route, including avoiding certain road types and road altitudes.		
ID	Acceptance Criteria / System Requirements	Priority
SR6	The system must provide an overlay window to allow the user to update routing preferences.	Must
SR7	The update preferences overlay must provide options to ’avoid’ along the route.	Must
SR8	The update preferences overlay must provide a ’via’ user input field.	Should
SR9	The update preferences overlay must provide a ’leave time’ user input field.	Should
SR10	The update preferences overlay must provide a ’arrive time’ user input field.	Should
SR11	The update preferences overlay must provide a ’round trip’ user input field.	Could

Table 5.3: User Story 03

As a user, I want to be able to export the planned route for use on my mobile phone or GPS device.		
ID	Acceptance Criteria / System Requirements	Priority
SR12	The system must provide an option to export the planned route.	Must
SR13	The system must provide an export feature to export the route to the 'GPX' file format.	Must
SR14	The system must provide an export feature to export the route to the 'GeoJSON' file format.	Should
SR15	The system must provide an export online (to Google Drive, OneDrive and/or other cloud services)	Must

Table 5.4: User Story 04

As a user, I want to share my route with other people.		
ID	Acceptance Criteria / System Requirements	Priority
SR16	The system must provide a share functionality overlay.	Should
SR17	The share overlay must provide an option to share direct over email.	Should
SR18	The system must provide an option to share the route direct to Strava.	Could

Table 5.5: User Story 05

As a user, I want to be provided with route suggestions based on predicted weather conditions over the week.		
ID	Acceptance Criteria / System Requirements	Priority
SR19	The system must provide the user with a weather condition overlay.	Must
SR20	The weather condition overlay must provide the user with the weather for the current day.	Must
SR21	The weather condition overlay must provide the user with the weather for the next week.	Should
SR22	The weather condition overlay must provide the user with the option to enable weather conditions in the route planning algorithm.	Could
SR23	The weather condition overlay must provide the user with suggestions on the best days to cycle.	Could
SR24	An option to include weather in route planning should be provided, ensuring the user enters the planned day to ride	Could

Table 5.6: User Story 06

As a user, I want to view the route in detail and get information about parts of the route.		
ID	Acceptance Criteria / System Requirements	Priority
SR25	The system must provide the user with an interactive map to display the planned route.	Must
SR26	The interactive map must allow the user to zoom into parts of the planned route.	Must
SR27	The interactive map must allow the user to select parts of the route and receive detailed information about that subsection of the route.	Should
SR28	The interactive map must allow the user to select and drag the planned route to modify its path.	Should
SR29	The system must display an elevation graph for the planned route beneath the interactive map.	Should
SR30	The system must allow the user to measure chosen sections of the route	Must
SR31	The system must provide multiple map layers to give users the greater options when viewing the route	Must

Table 5.7: User Story 07

As a user, I want the ability to add and utilise hazard/infrastructure waypoints to be considered in route planning.		
ID	Acceptance Criteria / System Requirements	Priority
SR32	The system must provide a user input modal to input Hazard and Infrastructure Data.	Must
SR33	The hazard input modal must provide a Type drop-down menu based on the OSM Hazard Types.	Must
SR34	The hazard input modal must provide a date entry point to specify the date the hazard was seen.	Should
SR35	The hazard input modal must provide a submit button to add the hazard to the hazard index.	Must
SR36	The infrastructure input modal must provide a Type drop-down menu with different types of cycling/road infrastructure	Must
SR37	The infrastructure input modal must provide a date entry point to specify when the bad infrastructure was found	Must
SR38	The infrastructure input modal must provide an input box providing the user with the option to supply more detail	Should
SR39	Both Hazard and Infrastructure data should be displayed on the map, with an option to toggle on/off, and report errors	Should

Table 5.8: User Story 08

As a user, I want to be able to view key waypoints along the journey such as, accommodation and key tourist points.		
ID	Acceptance Criteria / System Requirements	Priority
SR40	The system must provide a map layer to include key waypoints.	Must
SR41	The waypoint layer must provide locations of accommodation along the route.	Must
SR42	The waypoint layer must provide locations of tourist points along the route	Should
SR43	The waypoint layers must be able to be toggled on and off	Must
SR44	Each waypoint must be clickable to provide extra detail on each point	Must
SR45	Each waypoint must have a button to add stop along the route and the route will be re-plotted via the waypoint.	Should

Table 5.9: User Story 09

As a user, I want select what type of cyclist I am, for example road cyclist and commuter cyclist.		
ID	Acceptance Criteria / System Requirements	Priority
SR46	The system must provide an option within the route planning modal to select the rider type.	Must
SR47	The system must provide an option within the route planning modal to select the fitness level of the rider.	Should

Table 5.10: User Story 10

As a user, I want to be able pre-planned routes from other applications.		
ID	Acceptance Criteria / System Requirements	Priority
SR48	The system must provide an import option for GPX files.	Should
SR49	The system must provide an import option for GeoJSON files.	Should

Table 5.11: User Story 11

As a user, I want to be able to share my route to other services and social media.		
ID	Acceptance Criteria / System Requirements	Priority
SR50	The system must provide an export to Garmin Routes.	Should
SR51	The system must provide an option to share to different social media platforms.	Should

Chapter 6

Design

This chapter explains the design process undertaken considering both the research found within the literature review (see Chapter 2, p17) and the established requirements (see Chapter 5, p26).

6.1 Assumptions and Decisions Taken

Discussions with the client required specific assumptions and decisions to be made to assit in the project's overall development and progression.

A decision was made to build the application as a web app, primarily to aid with accessibility and device compatibility of the artefact, this type of application also opens the artefact up to a large pre-existing userbase with greater ease. Different screen sizes have been acommodated for with the use of CSS media queries, however the artefact is aimed at desktop route planning, with the potential to be expanded to support mobile devices in the future.

It was assumed that the target users would have a preexisting knolwedge of route planning and had used similar systems in the past. Due to this, instructions to use the application were not provided as a part of the artefact. Futhermore, a minimal user interface approach was used when developing the artefact to ensure the apps ease of use, whilst enabling key custom route planning functionality (see Section 6.2, p34).

Research was undertaken into different available open source routing algorithms, finally deciding to implement Open Route Service ("Openrouteservice", n.d.) due to its customisability and native support of round trip route planning.

Specific data sets were stored within the browser LocalStorage, this was chosen for certain data such as user authentication tokens, last route plotted and other data which needed storing between browser sessions. Data sets such as the previous planned route (and related data sets) enabled the user to load the previous route when opening the application after having closed the tab/browser previously. Other data sets such as user tokens, enabled the application to remain logged into external services such as Strava ("Strava | Running, Cycling & Hiking App - Train, Track & Share", n.d.), Garmin Connect (International, n.d.) and Google Drive ("Home - Google Drive", n.d.), only requiring the user to reauthenticate once these tokens had expired.

6.2 User Interface

Initially, multiple hand-drawn low-fidelity protoype was created to demonstrate two iterations of the artefact (see Figure 6.1, p35), demonstrating some design changes and new features in the second low-fidelity prototype (see Figure 6.2, p35). After the low-fidelity prototypes were complete, a high fidelity prototype was created based on the second design using MarvelApp ("Marvel - The design platform for digital products. Get started for free." n.d.).

The following 5 components form the overall User Interface (UI): Map, Elevation Chart, Router, Route Preferences Panel and the Weather/Side Panel. The primary component which most other components were dependant on was the Map component; each other component would add enhanced functionality to the artefact.

6.2.1 Colour Scheme

The plain and simple colour scheme was chosen for simplicity and ease-of-use of the artefact. Despite dark mode being considered for the artefact, it was chosen against after conducting research on pre-existing route planners, it was found that a light map, and surrounding UI was generally considered more usable to the average user. A dark mode could be implemented in the future, enabling the user to switch modes should they choose, also altering the map layer to the corresponding dark/light theme.

6.2.2 Low Fidelity Prototype

Low-fidelity protoyping was vital in the project's early stages, ensuring that all UI elements would meet the user expectations and requirements. The project's clear focus on enhanced routing features, it was critical to ensure that the initialy designs were kept simple, to maximise usability, whilst still including all the required, key functionality. Two iterations were divided into two separate low-fidelity prototypes, where the first demonstrates the core functionality (see Figure 6.1, p35), whereas the second also includes some extra functionality, for example the Hazard Index (see Figure 6.2, p35).

6.2.3 High-Fidelity Prototype

The high-fidelity prototype was made after low-fidelity prototyping has concluded. The second low-fidelity prototype was expanded upon at this stage, keeping the design simple, whilst ensuring the required functionality could be implemented. It was critical to ensure tthe high-fidelity protype wouldn't overbare the user with too much information at once, therefore the necessary functionality is available in the main view (see Figure 6.3, p36), whereas extra options are hidden behind dropdown menus (see Figure 6.4, p36).

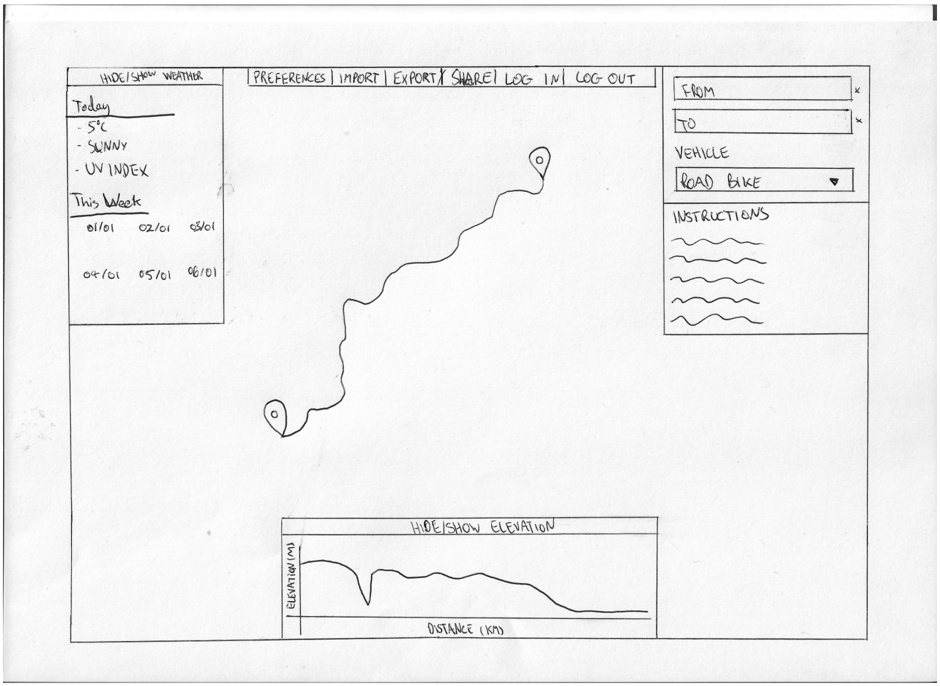


Figure 6.1: Low-Fidelity Prototype 1

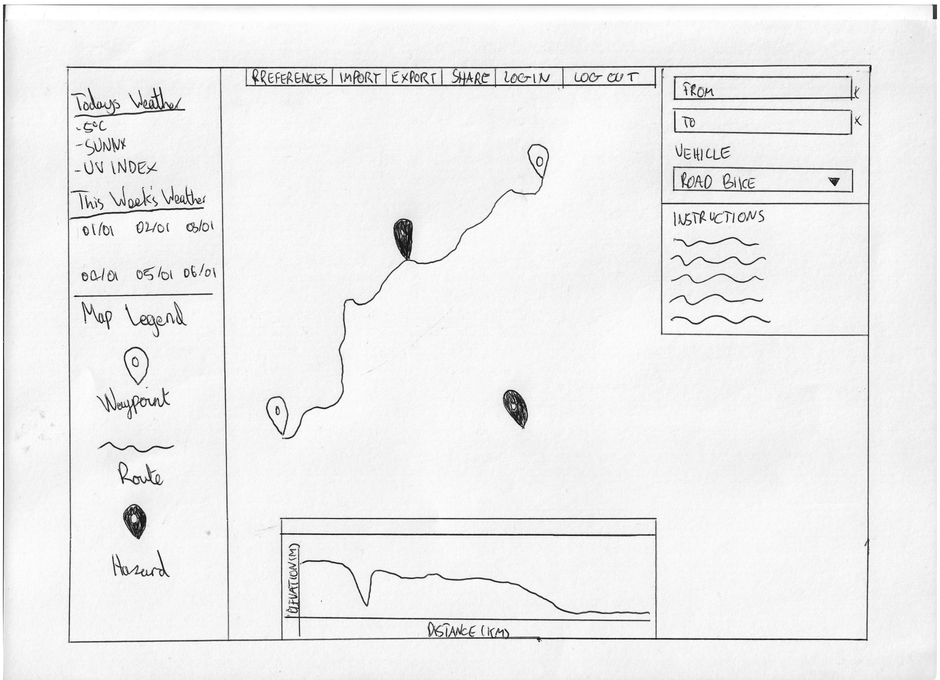


Figure 6.2: Low-Fidelity Prototype 2

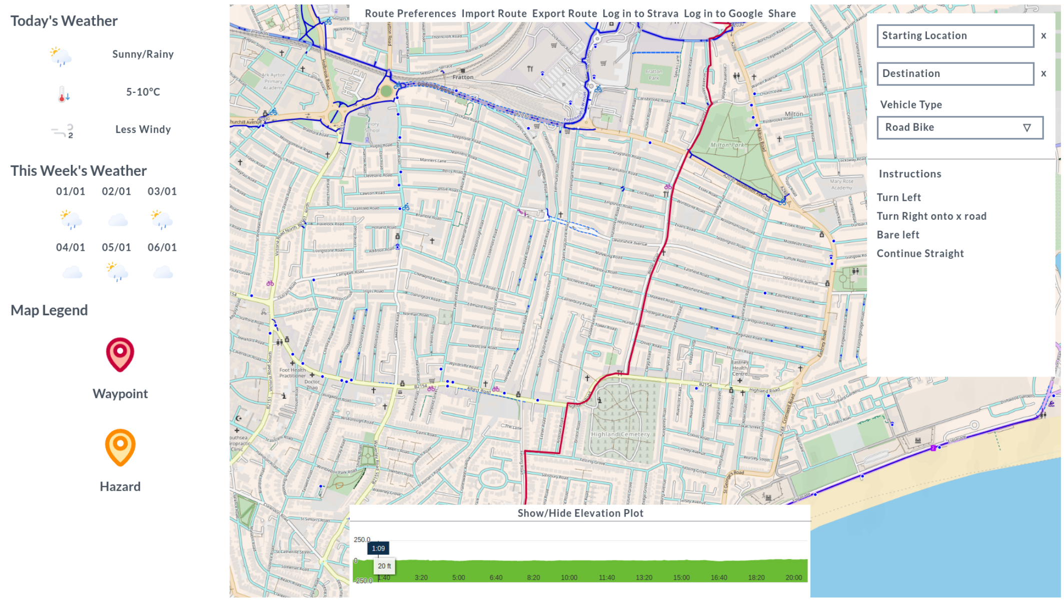


Figure 6.3: High-Fidelity Prototype 1

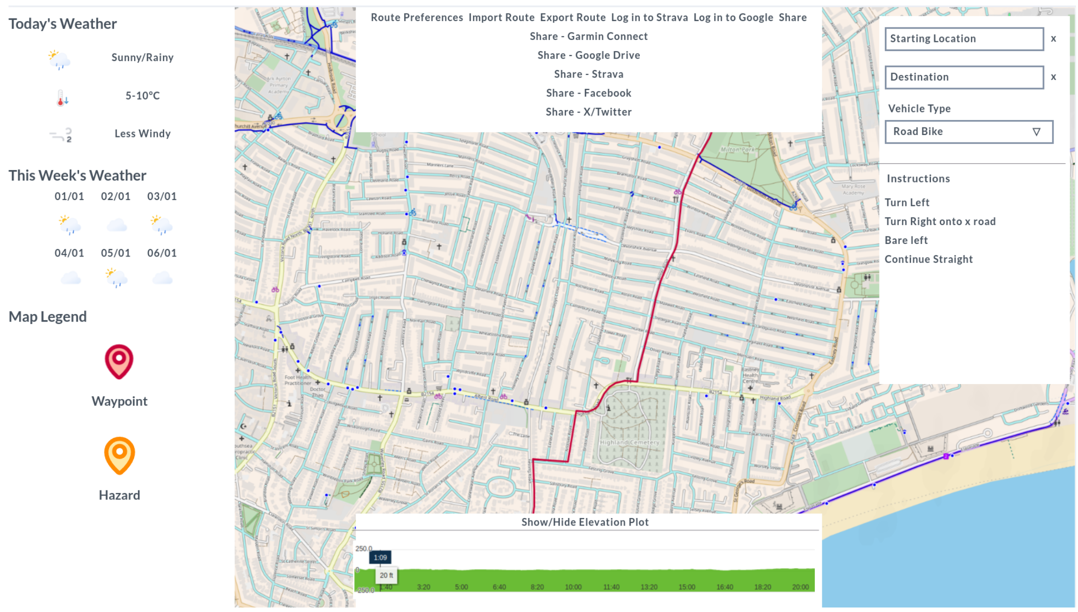


Figure 6.4: High-Fidelity Prototype 2

6.3 Use Cases

Use cases were determined by discerning potential uses of the artefact, subsequently a use case diagram was then created to visualise the artefacts different uses. The diagram is formed of actors, tasks and system components which are requred to response when each task is performed. Visual Paradigm was used to create the use case diagram ("Ideal Modeling & Diagramming Tool for Agile Team Collaboration", n.d.).

6.3.1 Use Case Diagram

During the requirements gathering process, the target user groups were identified, each user group would access identical functionality (see Chapter 5, p26), therefore all users were grouped into one single actor. Doing so enables all users to access all functionalities of the artefact, where they can use as little or as many of the extended features as they require (see Figure 6.5, p38).

6.4 System Design

It is vital to design not only the viewable client-side of the application, but to also consider the internal system, comprising of the Go Gin Web Server, PostgreSQL database as well as the React.js client. This chapter delves further into the internal design of the artefact.

6.4.1 Architecture Design

The client-server architecture was used to build the artefact (see Figure 6.6, p39). This architecture was selected to enable easy expansion of the artefact, with the aim for the web application to be hosted centrally, along with the PostgreSQL (PSQL) database (see Section 6.4.2, p39). For security reasons, it was also necessary to implement this architecture due to different services requiring oauth, these services make enforce authentication through the server, rather than completely on the client-side application. Furthermore, this architecture enabled different processes to be handled via the server, reducing the overall load the client application requires to run.

Client-Server architectures do however, add the risk of a single point of failure, whereby the application is wholly dependant on the web server. Overall the risk was deemed minimal due to the lack of sensitive information being stored (see Table 1.1, p15). In the case of a system failure, the application would remain unaccessable until fixed, however there is a low chance of this occurring as the appropriate measures such as thorough testing have been undertaken to reduce the risk (see Chapter 8, p53).

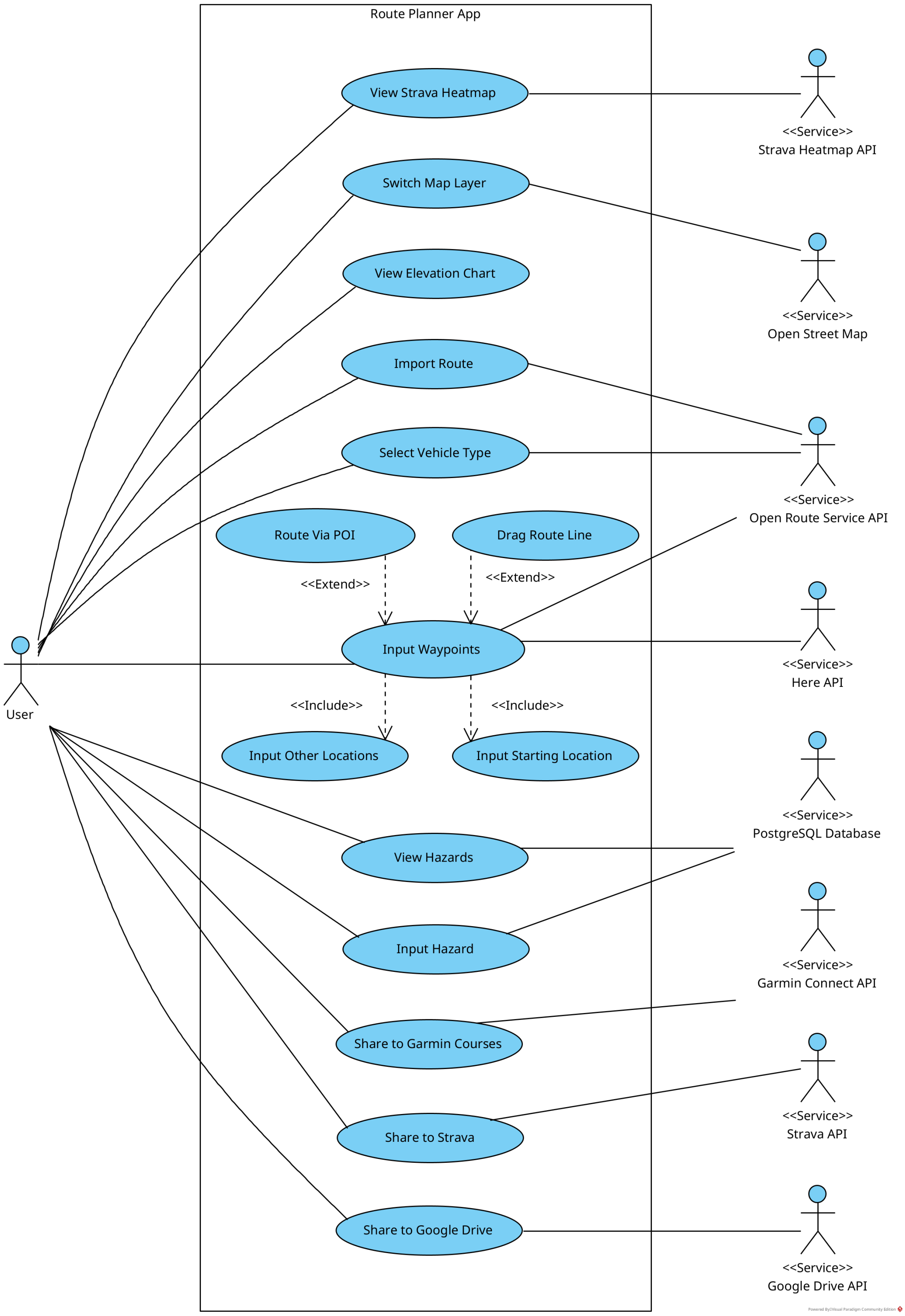


Figure 6.5: Use Case Diagram

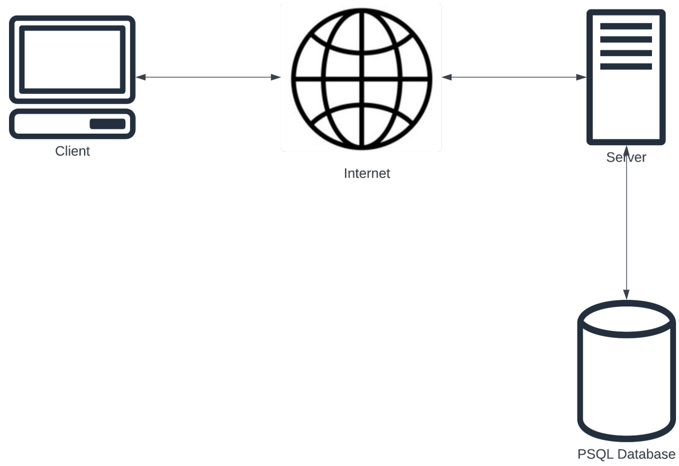


Figure 6.6: Client-Server Architecture

6.4.2 Database Design

The database is a pivitol part of the artefact's extended functionality, it primarily servers as a hazard index database, whilst also storing different areas of concerning cycling infrastructure. PostgreSQL was chosen as the Relational Database Management System (RDBMS), as PostGIS was available to use to add the functionality for storing, indexing and querying of geospatial data ("PostGIS", n.d.). This data is used to display concerning areas on the map for which users may want to avoid, with the potential to include native routing support to avoid these hazards, futher highlighted under future work (see Section 9.5, p54) (see Figure 6.7, p39).

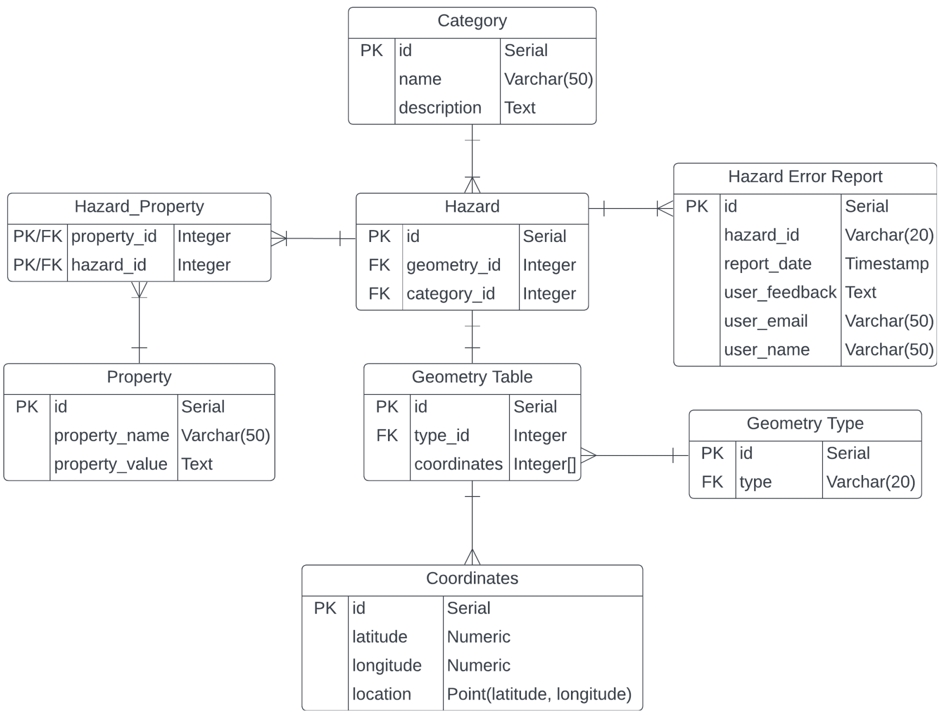


Figure 6.7: Entity Relationship Diagram

6.4.3 Modular Architecture

React.js allows for a modular design approach utilising the component-based framework and storing variables in the form of states to re-render the application depending on the states of different variables. All the necessary data, functions and states can be passed to and from different components using the props functionality built into React, whereby a component inherits data from its parent component (see Figure 6.8, p40).

While the App.jsx file is the starting point of the React application, the Map component acts as the primary parent for most of the artefact's key components. Multiple states and variables are declared within Map and are transferred across the rest of its children via props, this is necessary, because all the key functionality of the route planner is wholly dependent on the Map component. However, not all child components of Map are required for the application to function, demonstrating the modular functionality of the application. The LeftPanel component and its children remain standalone from the rest of the application as these components act to display information, rarely updating.

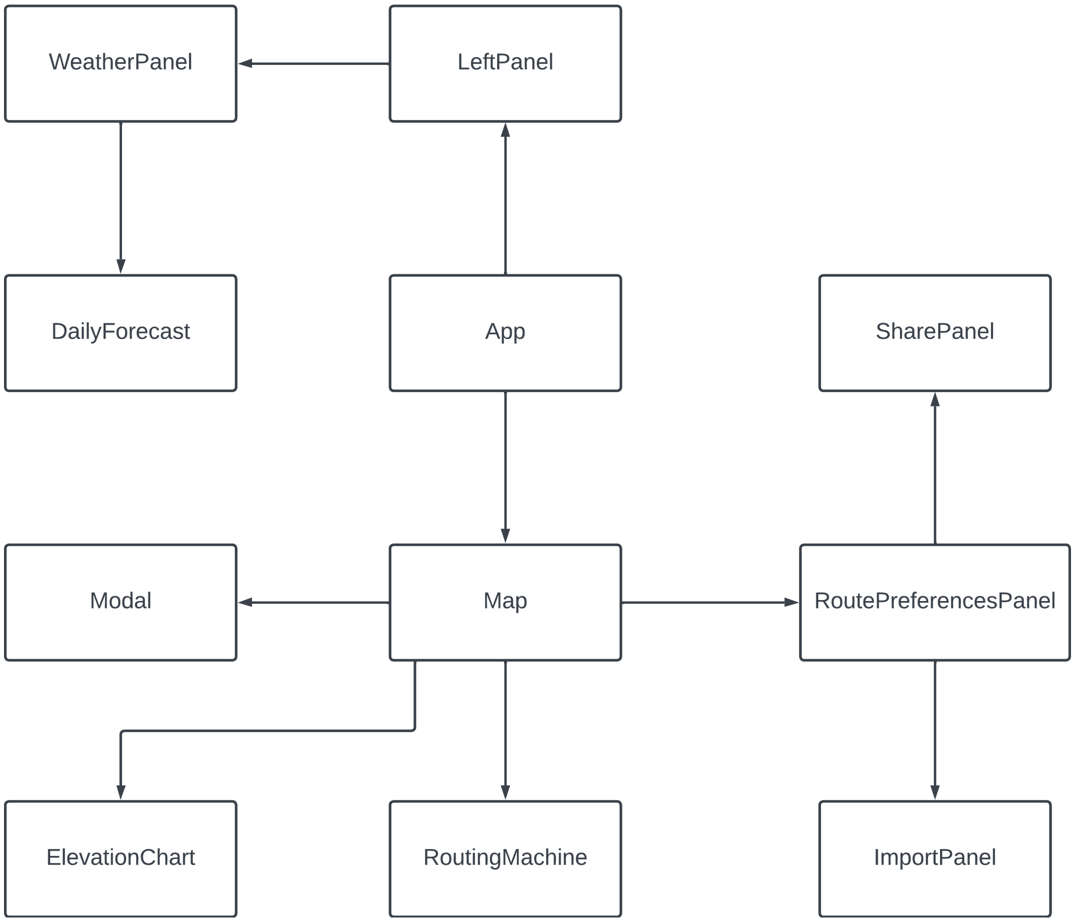


Figure 6.8: Modular Components

6.5 Critique of Designs

The progressive approach was used to develop the UI, and whilst beneficial to the overall project and its aims, mobile devices were not catered for. While this can be seen as bad software development practice, the artefact is aimed at users on a desktop environment, exporting routes to external mobile navigation/fitness applications. Therefore, the designs were catered for a desktop experience, over a mobile-first design.

Both high-fidelity designs only demonstrate a light mode colour scheme. Due to this, all users are forced to use the artefact in a brighter, light mode with no option of customisability. In hindsight, it would have been beneficial to develop multiple high-fidelity designs to include both a light and dark mode for the artefact.

The database was designed to only store hazard and cycling infrastructure information, ideally, the database would store more than just this data. To expand some of the artefact's basic functionality, the database would also store user oAuth tokens, as well as their planned routes. Implementing such tables would enable the user to load more than just the last planned route into the application, without the requirement to store GPX/GeoJSON files locally, instead, they could save routes to their account. These tables would also enable a more expansive sharing functionality to social media, enabling users to share links to routes, rather than map screenshots of their routes (see Section 9.5, p54).

Finally, the modular architecture approach, was ideal for developing the artefact, however, the modules could have been better structured to maximise the benefits of the modular approach. The artefact should be customisable, whereby specific modules can be disabled entirely if the user doesn't require them. At this stage, the design only caters to the manual disabling of each module, with certain modules being necessary for the artefact to function (see Section 9.4, p54).

Chapter 7

Implementation

This section describes the artefact development process. Despite multiple stumbling blocks during development, these were overcome through problem-solving, thorough investigation and reading of software documentation.

7.1 Development Environment

Visual Studio Code (VS Code) was chosen as the development environment for development ("Visual Studio Code - Code Editing. Redefined", n.d.). VS Code was chosen due to its flexibility in working with multiple programming languages through the use of extensions. The React.js Framework, JavaScript and Go Language extensions were installed to ensure seamless development of the backend and frontend of the artefact. The ESLint extension was also used using Go and JavaScript configurations to ensure the consistency and readability of the codebase.

7.2 Iteration 1 - Basic UI and Core Routing

The initial stages of development began using a first draft of elicited requirements (see Appendix B, p61) where later in the project, a new set of user requirements was formed based on primary research. This method proved effective in not delaying initial development, where the core requirements were unlikely to change greatly. Furthermore, Iteration 1 consisted of implementing the building blocks of the artefact, including setting up a React.js App, Gin Webserver, PostgreSQL database and the core routing functionalities.

7.2.1 Leaflet - Open Street Map (OSM)

The Map component was the first to be developed because the rest of the artefact would be dependent on its mapping functionality. The component contained a div element containing the MapContainer component imported from React Leaflet ("React Leaflet | React Leaflet", n.d.). The component includes a tile layer to make API requests to Open Street Map ("OpenStreetMap", n.d.) to get the tile images for the map, and then display these tiles to form the full-screen map element. Temporary variables were set up to store the latitude and longitude values representing the start and end of a route, then drawn on the map using a polyline and marker points (see Figure 7.1, p43).

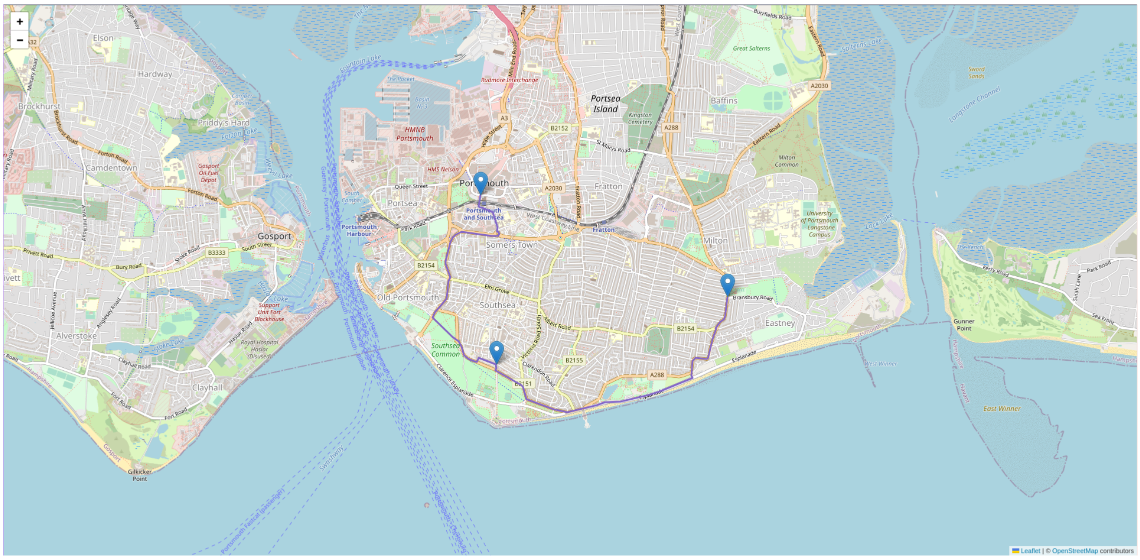


Figure 7.1: Basic Map with Route

7.2.2 Basic Route Planning

After conducting research, the most effective way to implement route planning with Leaflet was to use Leaflet Routing Machine ("Leaflet Routing Machine", n.d.). This library enabled a RoutingMachine component to be natively added on top of the React Leaflet map component, whilst providing a basic, customisable route planning UI (see Figure 7.2, p44). The routing API in use at the beginning was Open Source Routing Machine (OSRM) ("Project OSRM", n.d.) which appeared to meet all requirements of a routing algorithm until round trip routing was required (see Section 7.3, p46). This new routing functionality enabled the user to select a start, destination and any intermediate locations, a route would then be planned. At this stage, the route could be altered through the use of waypoints, but the artefact did not provide any further custom routing features to the user.

7.2.3 Elevation Chart

To create an elevation chart, the component was declared, with the plan to use Chart.js ("Chart.js", n.d.) to draw the plot on a canvas element. A div was created to hold the chart component, where the elevation data was gathered from a state variable called 'coordinates' which contained the route latitude, longitude points and the altitude/elevations for each point (see Figure 7.3, p44). The distance along the route for each elevation point is calculated by dividing the total route distance (taken from a state variable called 'summary') by the length of the 'coordinates' array. To ensure continuity between the map and the elevation plot, a simple feature was added to allow the user to hover over the elevation

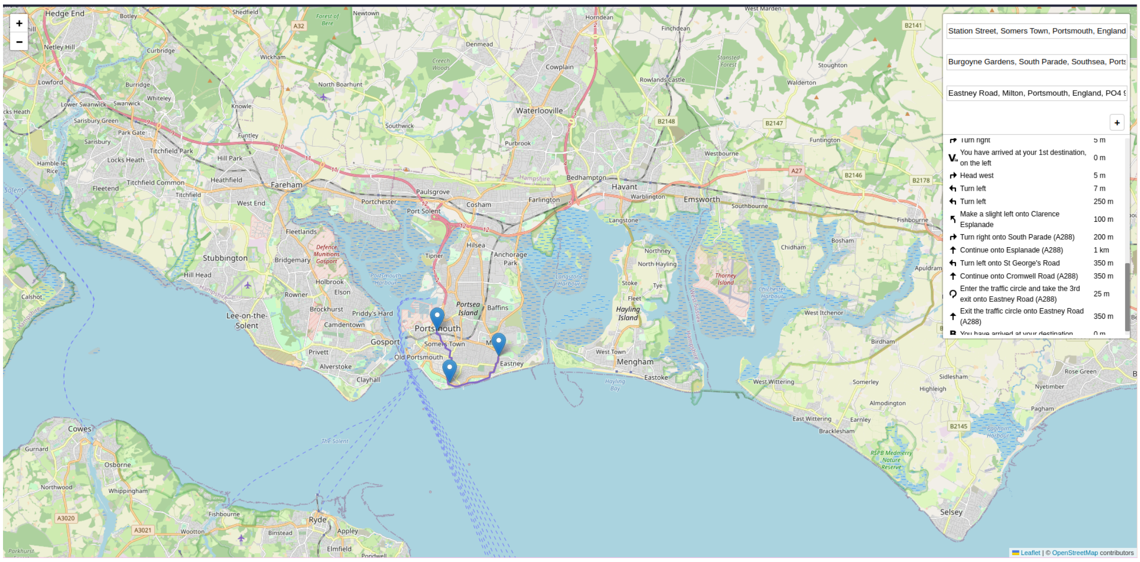


Figure 7.2: Routing UI

plot, with the matching point along the route being highlighted. This required the hover point on the chart's canvas element to be found, matching the hover value to the latitude and longitude with the respective point and drawing a circle element on the Leaflet map (see Figure 7.4, p45).

```
▼ Array(33) i
  ▶ 0: LatLng {lat: 50.79892, lng: -1.09116, alt: 7}
  ▶ 1: LatLng {lat: 50.79892, lng: -1.09119, alt: 7}
  ▶ 2: LatLng {lat: 50.79896, lng: -1.09148, alt: 7}
  ▶ 3: LatLng {lat: 50.799, lng: -1.09177, alt: 7}
  ▶ 4: LatLng {lat: 50.79909, lng: -1.0921, alt: 7}
  ▶ 5: LatLng {lat: 50.79922, lng: -1.09233, alt: 7.01}
  ▶ 6: LatLng {lat: 50.79933, lng: -1.09242, alt: 7.02}
  ▶ 7: LatLng {lat: 50.79956, lng: -1.09244, alt: 7.05}
  ▶ 8: LatLng {lat: 50.79989, lng: -1.09238, alt: 7.17}
  ▶ 9: LatLng {lat: 50.80027, lng: -1.09236, alt: 7.34}
  ▶ 10: LatLng {lat: 50.80025, lng: -1.09268, alt: 7.44}
  ▶ 11: LatLng {lat: 50.80011, lng: -1.09442, alt: 7.66}
  ▶ 12: LatLng {lat: 50.80004, lng: -1.09527, alt: 7.37}
  ▶ 13: LatLng {lat: 50.80003, lng: -1.09544, alt: 7.32}
  ▶ 14: LatLng {lat: 50.80002, lng: -1.09562, alt: 7.27}
  ▶ 15: LatLng {lat: 50.79998, lng: -1.09611, alt: 7.15}
  ▶ 16: LatLng {lat: 50.79995, lng: -1.09648, alt: 7.08}
  ▶ 17: LatLng {lat: 50.79993, lng: -1.09676, alt: 7.06}
  ▶ 18: LatLng {lat: 50.7999, lng: -1.09706, alt: 7.02}
  ▶ 19: LatLng {lat: 50.79986, lng: -1.0977, alt: 6.87}
  ▶ 20: LatLng {lat: 50.79983, lng: -1.09847, alt: 6.47}
  ▶ 21: LatLng {lat: 50.79981, lng: -1.09939, alt: 5.87}
  ▶ 22: LatLng {lat: 50.7998, lng: -1.09961, alt: 5.77}
  ▶ 23: LatLng {lat: 50.7998, lng: -1.09998, alt: 5.57}
  ▶ 24: LatLng {lat: 50.7998, lng: -1.10033, alt: 5.37}
  ▶ 25: LatLng {lat: 50.79979, lng: -1.10082, alt: 5.15}
  ▶ 26: LatLng {lat: 50.79978, lng: -1.10159, alt: 5}
  ▶ 27: LatLng {lat: 50.79977, lng: -1.10205, alt: 5}
  ▶ 28: LatLng {lat: 50.79977, lng: -1.10248, alt: 5}
  ▶ 29: LatLng {lat: 50.79975, lng: -1.103, alt: 5}
  ▶ 30: LatLng {lat: 50.79974, lng: -1.10311, alt: 5}
  ▶ 31: LatLng {lat: 50.79972, lng: -1.10406, alt: 5}
  ▶ 32: LatLng {lat: 50.79972, lng: -1.1043, alt: 5.05}
    length: 33
```

Figure 7.3: Route Waypoint/Elevation Array

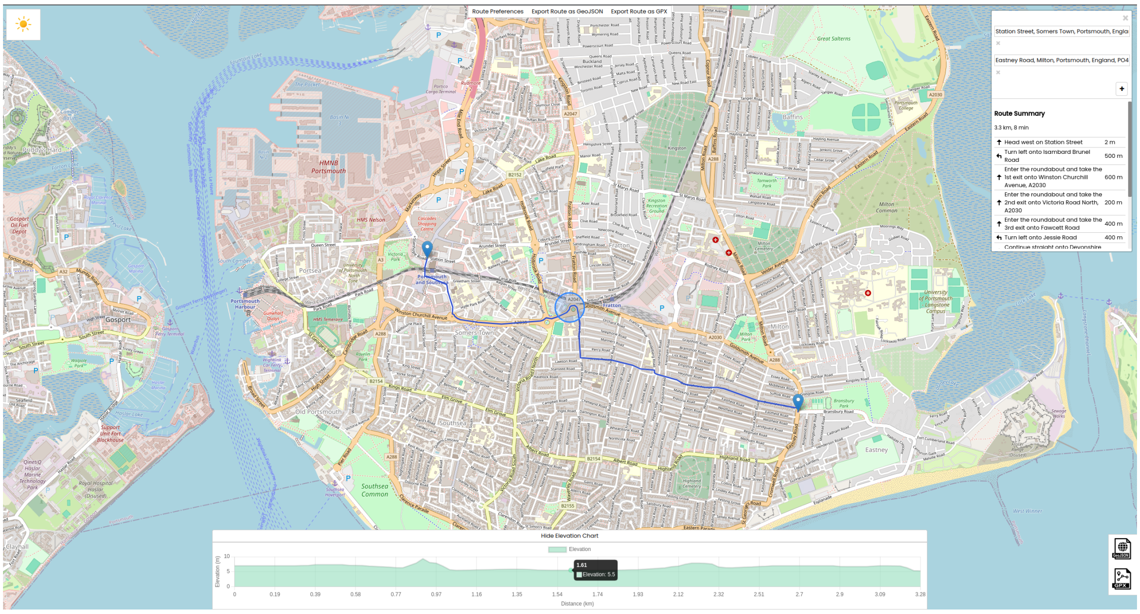


Figure 7.4: Elevation Plot Hover Functionality

7.2.4 Weather Information Panel

A generic weather panel was created within iteration 1, displaying weather information for the current day at the user's current location. The weather panel used the browser's geolocation API called within a React useEffect to get the user's general location, asking for permission via a pop-up window, then storing the geolocation in a state variable. The approximate coordinates returned from the API were then passed to Open Weather Map to gather all weather data for the current day. The Meteocons icons ("Weather Icons by Bas", n.d.) were used to display images demonstrating the current weather conditions (see Figure 7.5, p45). The visibility of the weather panel was determined by a State variable, if the variable was true, the height and width of the panel expand, and when false, reduce to the button size.

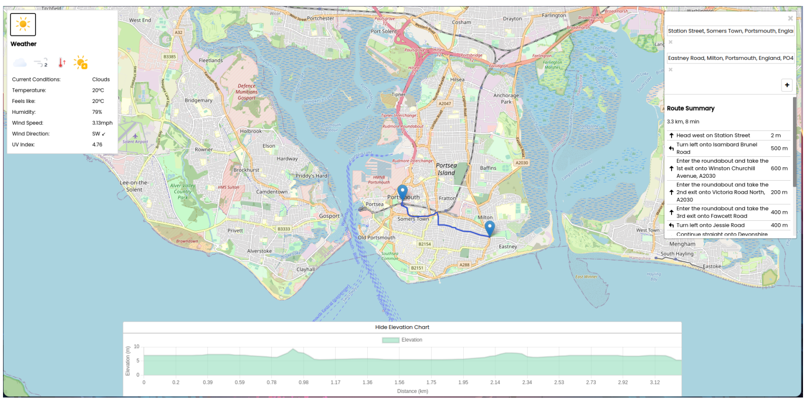


Figure 7.5: Weather Panel

7.2.5 Main Challenges

The main challenge of Iteration 1 (i1), was primarily integrating multiple services to communicate seamlessly through the React.js frontend. Syncing hovering over the elevation plot, with a matching point along the route polyline, proved difficult. Despite the calculations being simple to determine at what latitude and longitude to draw the circle marker on the map when hovering on the plot, getting the map instance reference was more difficult than expected. Initially, the reference would return undefined, therefore no marker was drawn on the map, however after implementing a useState and declaring a local copy of the map instance, the hover functionality worked seamlessly.

Furthermore, the only other challenge faced was retrieving the user's geolocation from the browser. A useEffect was required to access the geolocation API, however, the useState was at first missed to store the geolocation value once the API had returned the data. Once the state variable was added, however, the weather panel would re-render once the data was retrieved.

7.3 Iteration 2 - Route Sharing, Hazard Index and POI Integration

Unlike i1, Iteration 2 (i2) development was conducted after the primary research and requirements review had been completed. The remaining updated requirements were prioritised and distributed between i2 and Iteration 3 (i3) (see Section 7.4, p52). I2 builds on i1, enhancing existing functionality and introducing new features throughout the iteration.

7.3.1 Sharing Route Functionality

Route sharing was implemented with i2, when routes were found using Open Route Service, GPX ("GPX: the GPS Exchange Format", n.d.) and GeoJSON ("GeoJSON", n.d.) strings were generated. These files were then used to create a share to email feature using the Twilio SENDGRID API ("Email API", n.d.). Initially, it was planned to send emails direct to the SENDGRID directly from the client, to mitigate the middle-man required to send the request, however it was found that SENDGRID would only accept incoming requests from the server. Therefore, an API endpoint was created with Gin to handle the POST request from the frontend.

Each email would contain a simple amount of predefined text with the included GPX and/or GeoJSON files selected for export. A modal was created to display the email form, where the user could input the recipient's email address, route name and choose what type of file to share (see Figure 7.5, p45).

Initially, it was planned to use the Strava API to upload GPX files directly to the user's

Strava Routes account, however, it was found that the Strava API had no such endpoint to integrate with routes. It was decided to still implement Strava integration, but with activities, enabling route export as an activity for users without dedicated fitness devices.

A modal was created (see Figure 7.8, p48) and a POST request endpoint set up to handle the request to the Strava API. The user would be required to log in to their Strava account, to enable the artefact access to upload activities. Once logged in, the user could select the route to upload, then the artefact would send the GPX file to the Strava API, where it would be converted to an activity and stored in the user's account (see Figure 7.7, p48).

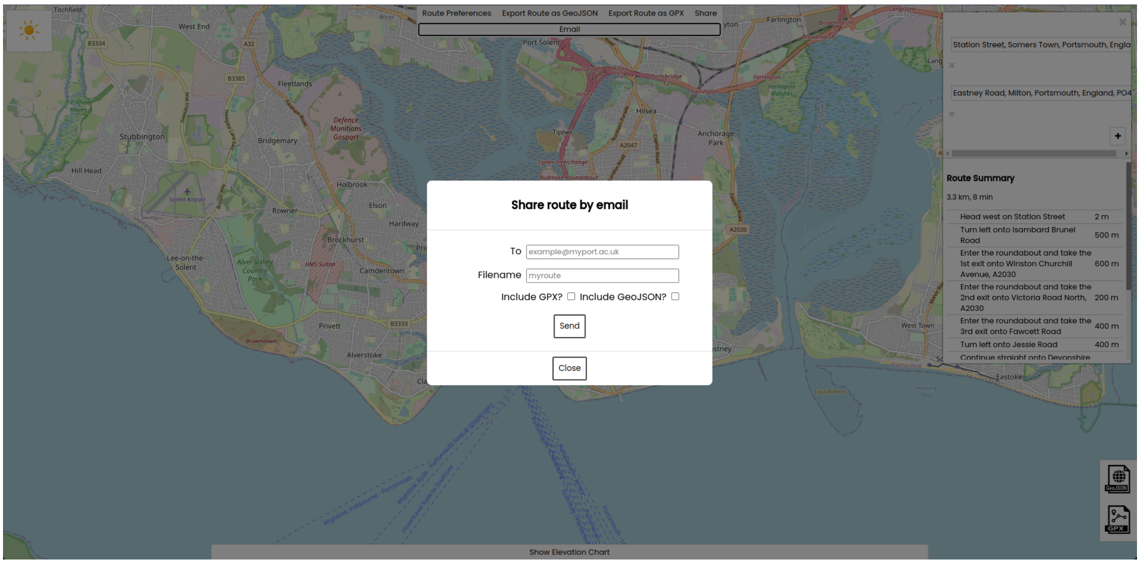


Figure 7.6: Share to Email Modal

7.3.2 Hazard Index Database Creation and Integration

Creation and integration of the hazard index database was the largest task of i2. The database was created using PostgreSQL using PostGIS. The database was split into eight tables representing a hazard, related details and its geospatial data (see Section 6.4.2, p39). Hazard types were chosen based on the most relevant hazards defined on the Open Street Map wiki ("Key:hazard - OpenStreetMap Wiki", n.d.) with the addition of a 'Cycling Infrastructure' hazard. The 'Cycling Infrastructure' hazard type was added to allow users to report and view bad cycling infrastructure in their area, with the intention of routing algorithms avoiding these areas in the future.

To integrate the database into the artefact, API endpoints were created to handle GET and POST requests to the database. The database was connected to the Go backend using the 'github.com/lib/pq' package ("pq package - github.com/lib/pq - Go Packages", n.d.), with Gin handling the routing of requests to the database. Using the fetch API, the frontend could make API calls, to add and retrieve hazards from the database, supplying coordinates for the hazard location/area and the hazard type (see Figure 7.9, p49).

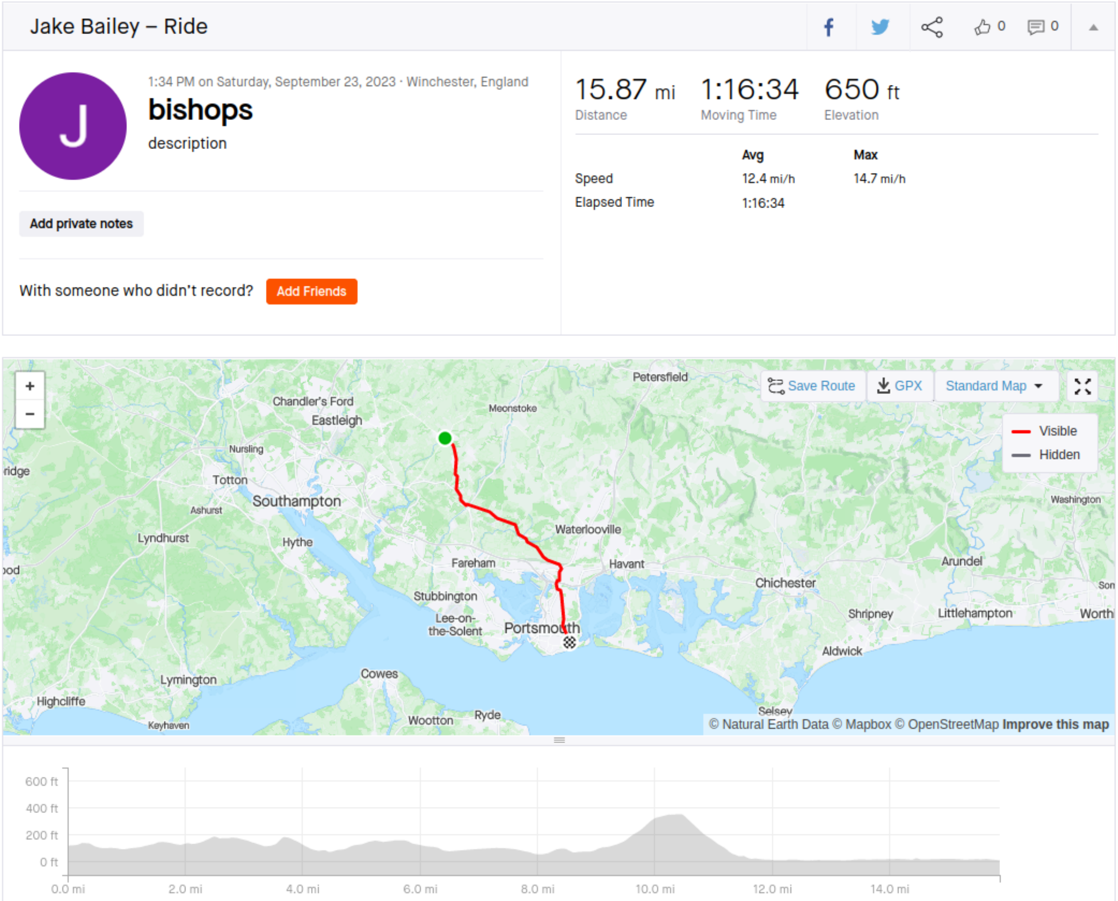


Figure 7.7: Strava Activity

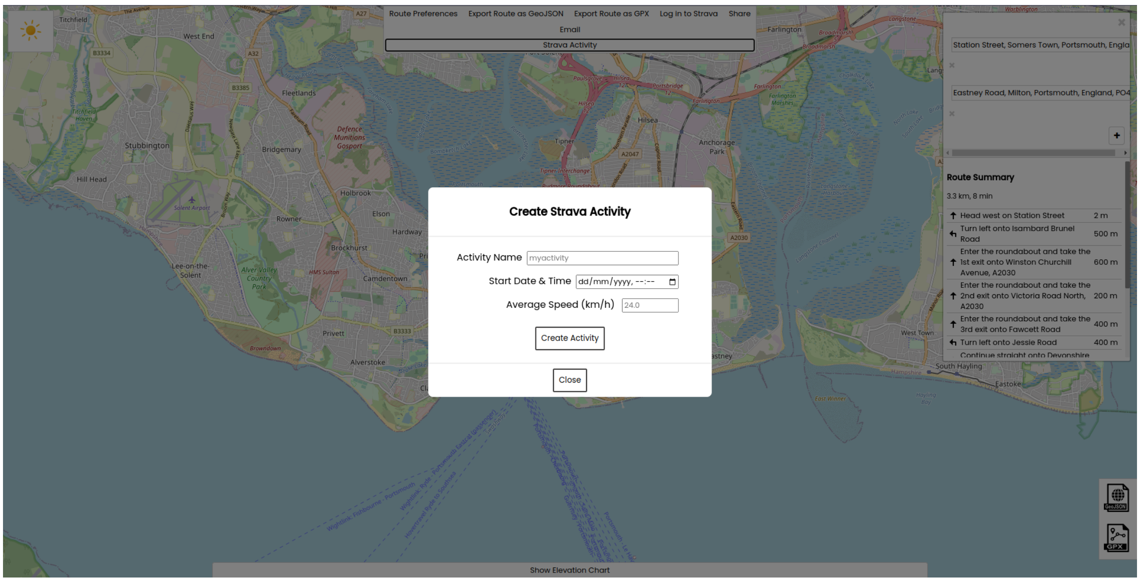


Figure 7.8: Share to Strava Modal

Furthermore, it was vital to make the hazards visible on the map, therefore a new map layer was created to display the hazards. The layer was added to the map using the React Leaflet library, where the hazards were retrieved from the database and drawn on the

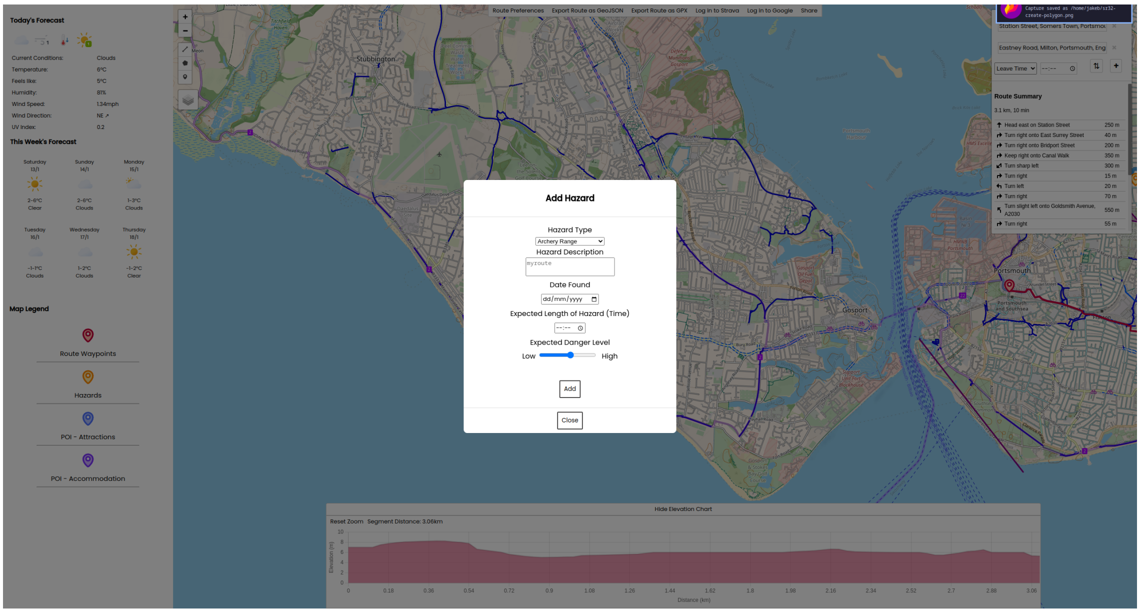


Figure 7.9: Hazard Creation Modal

map. These hazards were shown as both markers and polygons to represent a point or area hazard (see Figure 7.10, p49). The API endpoint was set up to allow the user to retrieve all hazards within a five mile radius of a point, to enable the user to see hazards in their area, as the map was panned, the hazard layer would update (see Figure 7.11, p50).

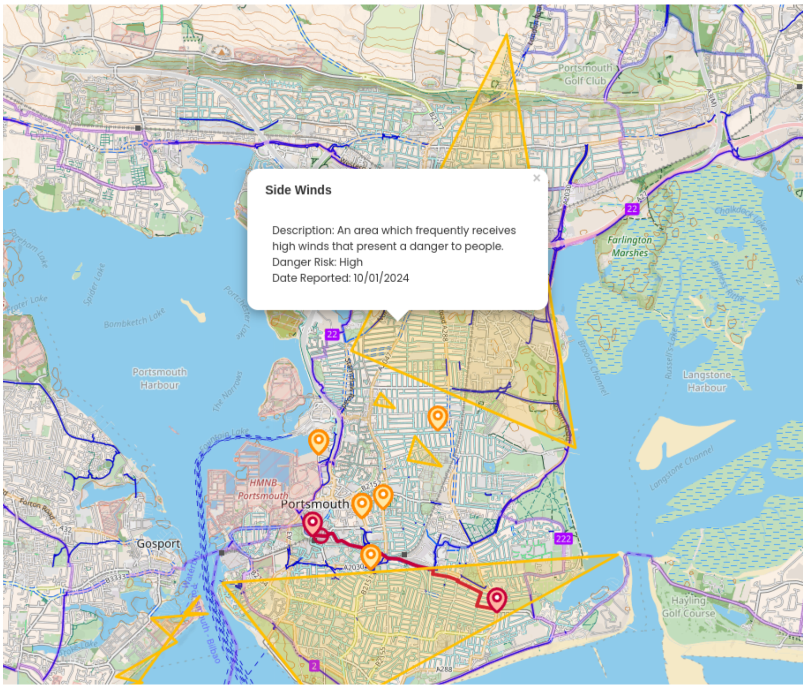


Figure 7.10: Hazard Map Layer

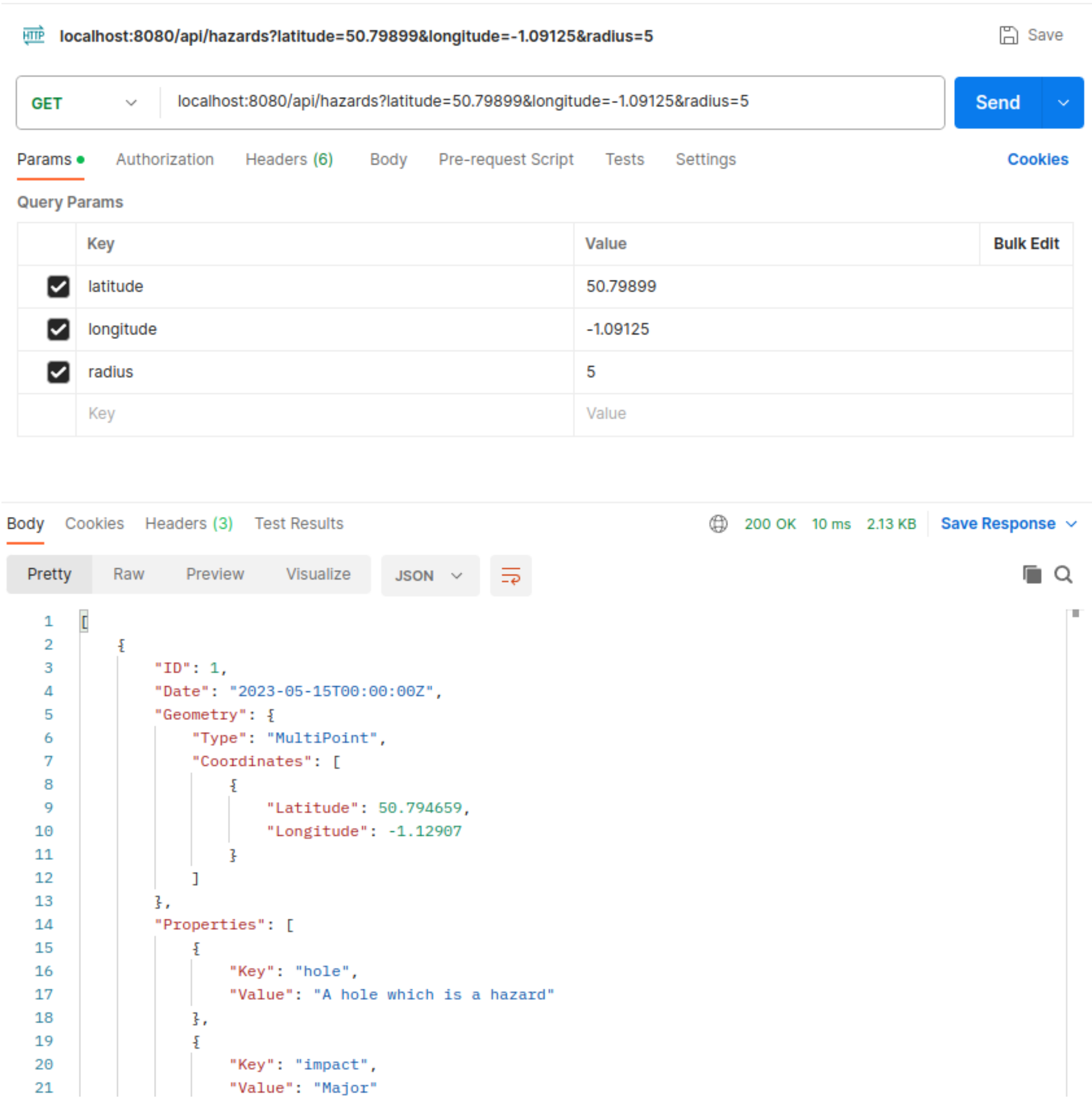


Figure 7.11: Hazard API Query

7.3.3 Key Point-of-Interest (POI) Integration

The POI integration was the final feature of i2, where the user could view points of interest in their area. The POI data was retrieved from the Foursquare Places API ("Places API", n.d.). Two separate map layers were created similar to the hazard layer (see Section 7.3.2, p47), one to display accommodation POI and another to display the attractions/leisure POI (see Figure 7.12, p51). The layer also allowed the user to click on each POI marker to view more information about the location, such as the name and address.

After the core POI layers were developed, two buttons were then added to each POI popup window. These were to allow a user to either route to or via the POI. If the user chose to route to, the last route waypoint would be updated to the POI's latitude and longitude. Whereas to route via, the route waypoints would initially be taken to find which route waypoint was closest to the POI, it would then be inserted into waypoints before the closest waypoint. The route would then be recalculated to include the POI as a waypoint (see Figure 7.13, p51).

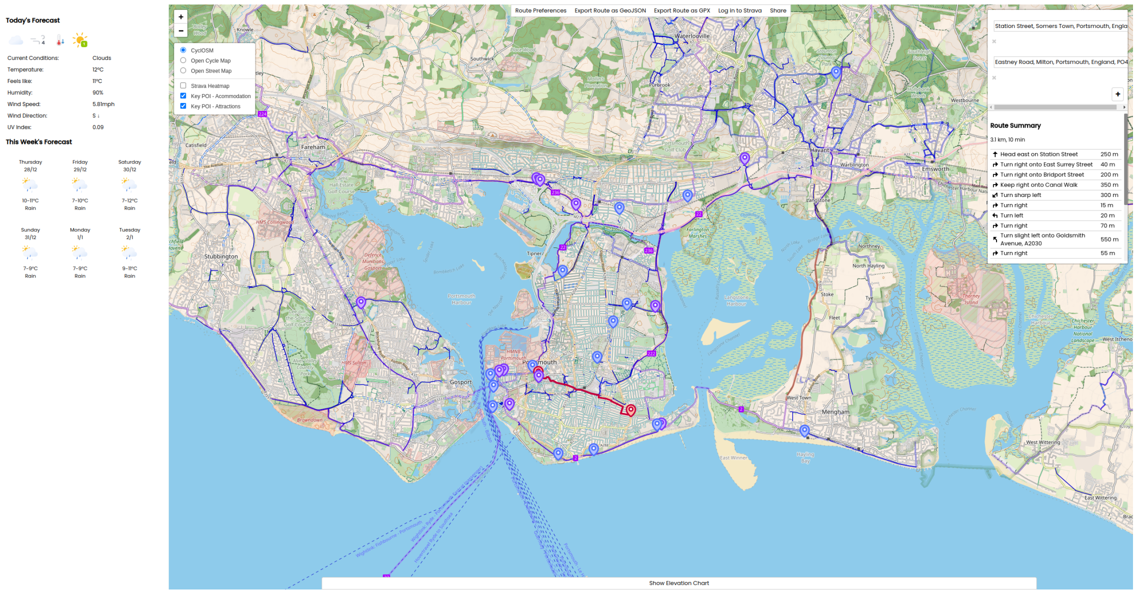


Figure 7.12: Attractions and Accommodation Layers

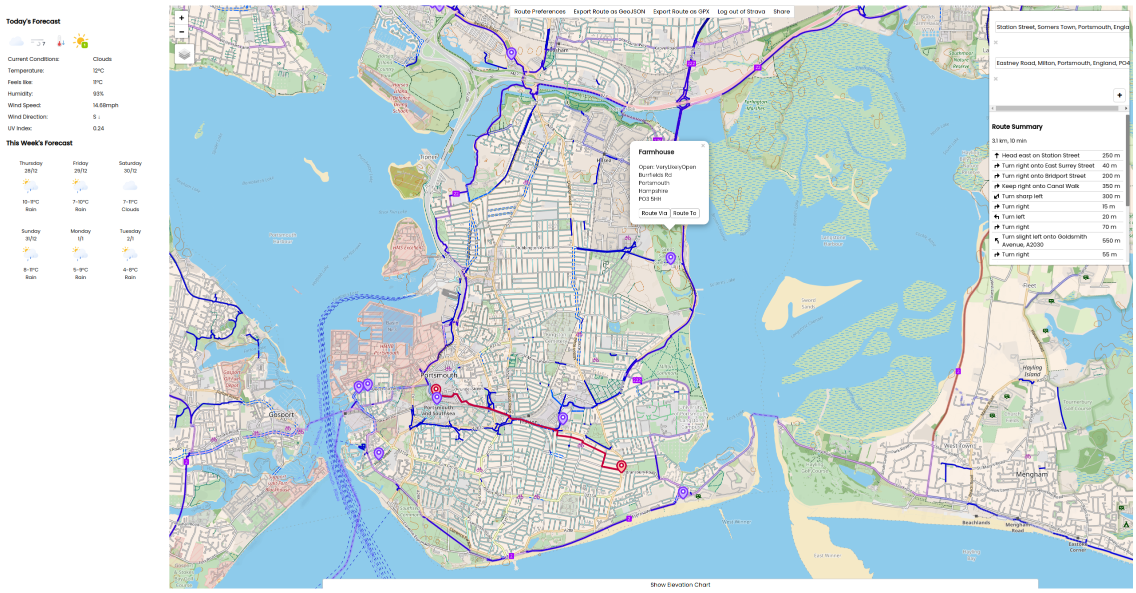


Figure 7.13: Route To/Via POI

7.3.4 Main Challenges

The main challenges for i2 resided in setting up the SQL queries to handle the hazard data requests as well as oauth authentication with the Strava API.

The INSERT hazard query was the most complex. The PLpgSQL language was used to create a function to handle this logic as loops were required when inserting multiple PostGIS Point data types (coordinates) for each hazard. A loop was also required to insert the properties for each hazard, where each hazard could have one or more. The function was then called from the API endpoint, passing the json string input directly into the PLpgSQL function as type JSONB.

The Strava API was also challenging as it was required to implement oAuth 2.0 manually to authorise the application to upload activities to the user's account. The Strava API documentation was relatively clear on how to implement this process, however, having not manually implemented oAuth before, without using libraries such as Auth0, it was a steep learning curve. This experience proved great later on however when implementing oAuth 1.0 for use with the Garmin Connect API (see Section 7.4.3, p52).

7.4 Iteration 3 - Round Trip, Route Import and Garmin Integration

7.4.1 Round Trip Routing

7.4.2 Route Import

7.4.3 Garmin Connect Integration

7.4.4 Social Media Sharing

7.4.5 Main Challenges

Chapter 8

Testing

8.1 Unit Testing

8.1.1 Testing here

8.2 Postman

8.3 Chrome Developer Tools

Chapter 9

Evaluation

9.1 Evaluation Methods and Techniques

9.2 Evaluation Process

9.3 Evaluating Requirements

9.4 Limitations

9.5 Future Work

Chapter 10

Reflection and Conclusion

References

Boettge, B., Hall, D. M., & Crawford, T. (2017). Assessing the bicycle network in st. louis: A PlaceBased user-centered approach [Number: 2 Publisher: Multidisciplinary Digital Publishing Institute]. *Sustainability*, *9*(2), 241. https://doi.org/10.3390/su9020241

Chart.js. (n.d.). Retrieved February 3, 2024, from https://www.chartjs.org/

Dapeng Liu, Shaochun Xu, & Wencai Du. (2011). Case study on incremental software development. *2011 Ninth International Conference on Software Engineering Research, Management and Applications, Software Engineering Research, Management and Applications (SERA), 2011 9th International Conference on*, 227–234. https://doi.org/10.1109/SERA.2011.43

Doorley, R., Pakrashi, V., Byrne, E., Comerford, S., Ghosh, B., & Groeger, J. A. (2015). Analysis of heart rate variability amongst cyclists under perceived variations of risk exposure. *Transportation Research Part F: Traffic Psychology and Behaviour*, *28*, 40–54. https://doi.org/10.1016/j.trf.2014.11.004

Email API [Twilio]. (n.d.). Retrieved February 16, 2024, from https://www.twilio.com/en-us/sendgrid/email-api

Flynn, B. S., Dana, G. S., Sears, J., & Aultman-Hall, L. (2012). Weather factor impacts on commuting to work by bicycle. *Preventive Medicine*, *54*(2), 122–124. https://doi.org/10.1016/j.ypmed.2011.11.002

Free route planner for outdoor pursuits - plotaroute.com. (n.d.). Retrieved November 15, 2023, from https://www.plotaroute.com

GeoJSON. (n.d.). Retrieved February 16, 2024, from https://geojson.org/

Gin web framework [Gin web framework]. (n.d.). Retrieved November 16, 2023, from https://gin-gonic.com/

Google maps [Google maps]. (n.d.). Retrieved November 15, 2023, from https://www.google.com/maps

GPX: The GPS exchange format. (n.d.). Retrieved February 16, 2024, from https://www.topografix.com/gpx.asp

Home - google drive. (n.d.). Retrieved February 2, 2024, from https://drive.google.com/drive/u/0/home

Hong, J., Philip McArthur, D., & Stewart, J. L. (2020). Can providing safe cycling infrastructure encourage people to cycle more when it rains? the use of crowdsourced cycling data (strava). *Transportation Research Part A: Policy and Practice*, *133*, 109–121. https://doi.org/10.1016/j.tra.2020.01.008

Hull, A., & O'Holleran, C. (2014). Bicycle infrastructure: Can good design encourage cycling? [Publisher: Routledge _eprint: https://doi.org/10.1080/21650020.2014.955210]. *Urban, Planning and Transport Research*, *2*(1), 369–406. https://doi.org/10.1080/21650020.2014.955210

Ideal modeling & diagramming tool for agile team collaboration. (n.d.). Retrieved February 2, 2024, from https://www.visual-paradigm.com/

International, G. (n.d.). *Garmin connect | free online fitness community* [Garmin connect]. Retrieved February 2, 2024, from https://connect.garmin.com/

Key:hazard - OpenStreetMap wiki. (n.d.). Retrieved February 16, 2024, from https://wiki.openstreetmap.org/wiki/Key:hazard

Komoot | find, plan and share your adventures [Komoot]. (n.d.). Retrieved November 15, 2023, from https://www.komoot.com

Leaflet — an open-source JavaScript library for interactive maps. (n.d.). Retrieved November 16, 2023, from https://leafletjs.com/

Leaflet routing machine. (n.d.). Retrieved February 3, 2024, from https://www.liedman.net/leaflet-routing-machine/

Marvel - the design platform for digital products. get started for free. (n.d.). Retrieved February 2, 2024, from https://marvelapp.com/

Material design [Material design]. (n.d.). Retrieved November 15, 2023, from https://m3.material.io/

Miranda-Moreno, L. F., & Nosal, T. (2011). Weather or not to cycle: Temporal trends and impact of weather on cycling in an urban environment [Publisher: SAGE Publications Inc]. *Transportation Research Record*, *2247*(1), 42–52. https://doi.org/10.3141/2247-06

OpenCycleMap.org - the OpenStreetMap cycle map. (n.d.). Retrieved November 16, 2023, from https://www.opencyclemap.org/

Openrouteservice. (n.d.). Retrieved November 16, 2023, from https://openrouteservice.org/

OpenStreetMap [OpenStreetMap]. (n.d.). Retrieved February 3, 2024, from https://www.openstreetmap.org/

Places API [Foursquare]. (n.d.). Retrieved February 16, 2024, from https://location.foursquare.com/products/places-api/

PostGIS [PostGIS]. (n.d.). Retrieved February 2, 2024, from https://postgis.net/

Pq package - github.com/lib/pq - go packages. (n.d.). Retrieved February 16, 2024, from https://pkg.go.dev/github.com/lib/pq@v1.10.9

Project OSRM. (n.d.). Retrieved November 16, 2023, from https://project-osrm.org/

React leaflet | react leaflet. (n.d.). Retrieved February 3, 2024, from https://react-leaflet.js.org/

Reynolds, C. C., Harris, M. A., Teschke, K., Cripton, P. A., & Winters, M. (2009). The impact of transportation infrastructure on bicycling injuries and crashes: A review of the literature. *Environmental Health*, *8*(1), 47. https://doi.org/10.1186/1476-069X-8-47

Shoman, M. M., Imine, H., Acerra, E. M., & Lantieri, C. (2023). Evaluation of cycling safety and comfort in bad weather and surface conditions using an instrumented bicycle [Conference Name: IEEE Access]. *IEEE Access*, *11*, 15096–15108. https://doi.org/10.1109/ACCESS.2023.3242583

Strava | running, cycling & hiking app - train, track & share [Strava]. (n.d.). Retrieved October 28, 2023, from https://www.strava.com/

Visual studio code - code editing. redefined. (n.d.). Retrieved February 3, 2024, from https://code.visualstudio.com/

Weather icons by bas. (n.d.). Retrieved February 3, 2024, from https://basmilius.github.io/weather-icons/index-fill.html

Zotero | your personal research assistant. (n.d.). Retrieved October 18, 2023, from https://www.zotero.org/

Appendix A

Current Solutions

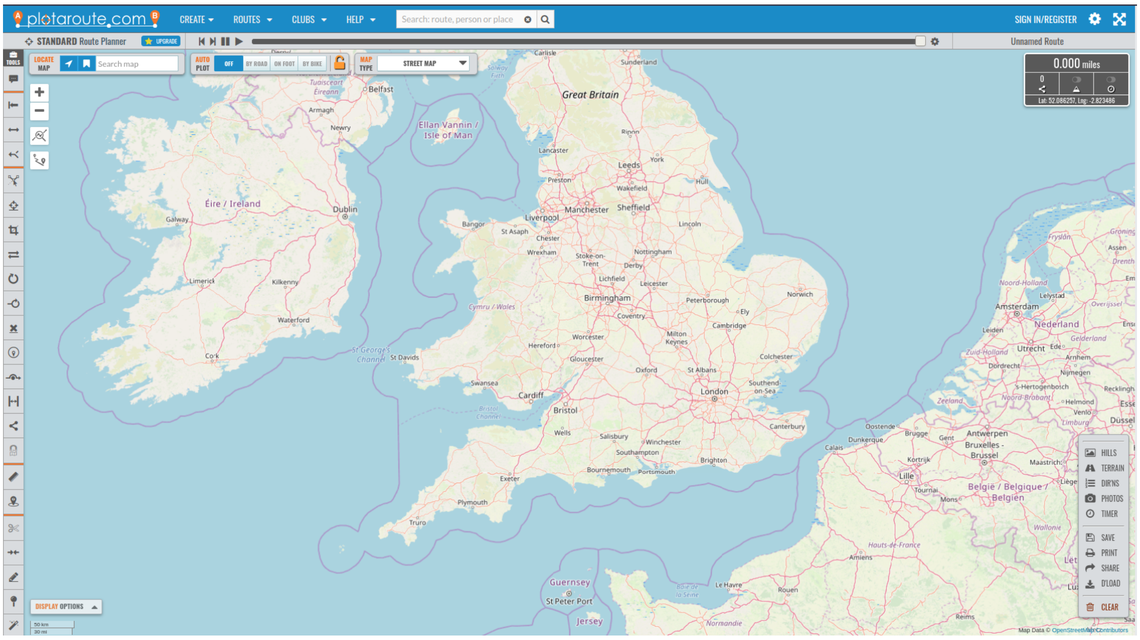


Figure A.1: Plotaroute.com UI

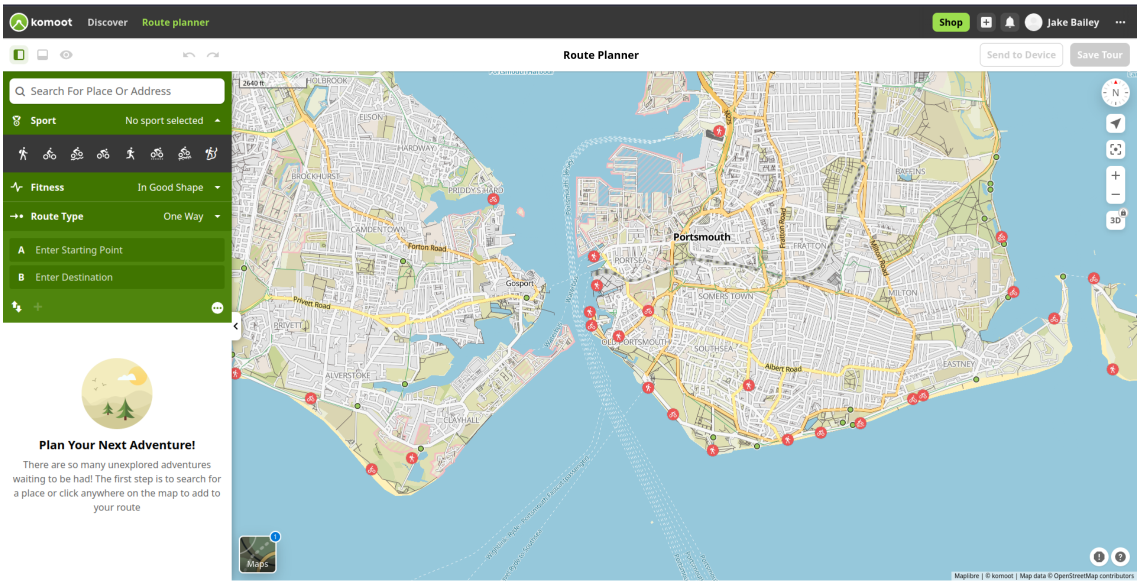


Figure A.2: Komoot UI

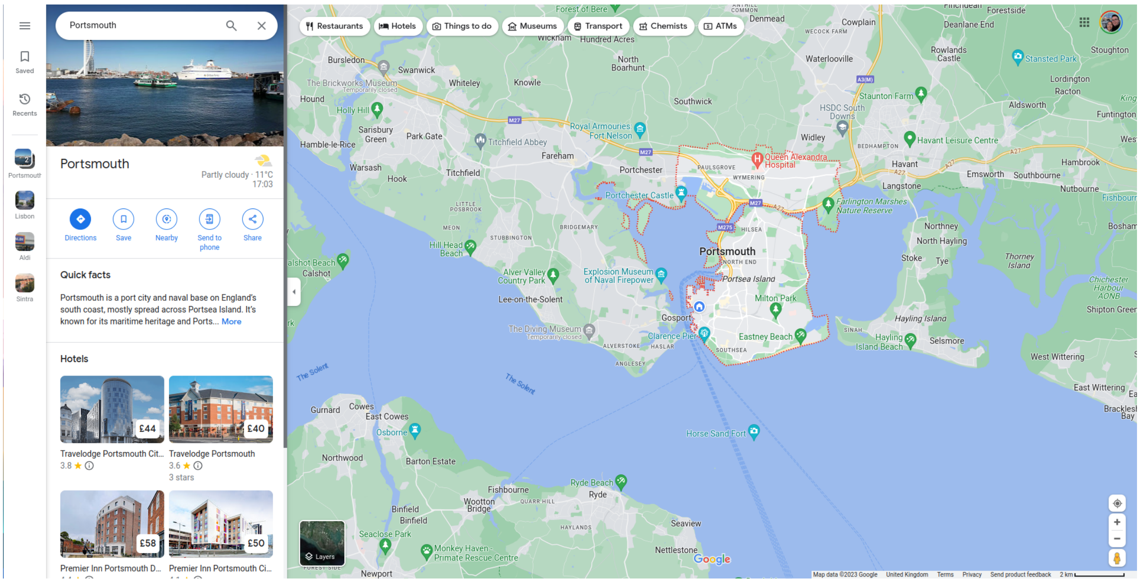


Figure A.3: Google Maps UI

Appendix B

Initial User Requirements

Table B.1: User Story 01

As a user, I want a page that allows me to configure my starting and destination location to plan a route.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide a route configuration page.	Must	SR1
The route configuration page must provide a starting location input field.	Must	SR2
The route configuration page must provide a destination location input field.	Must	SR3
The route configuration page should suggest locations based on the user input fields.	Should	SR4
The route configuration page must find the location position based on the user input.	Must	SR5
The route configuration page must verify both locations are correct before the user can continue.	Must	SR6
The route configuration page must provide a ‘Plan’ button to initiate the route planning algorithm.	Must	SR7

Table B.2: User Story 02

As a user, I want to change preferences to allow me to customise the route further, including avoiding certain road types and road altitudes.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide an overlay window to allow the user to update routing preferences.	Must	SR8
The update preferences overlay must provide an ’avoid’ user input field.	Must	SR9
The update preferences overlay must provide a ’via’ user input field.	Should	SR10
The update preferences overlay must provide a ’leave time’ user input field.	Should	SR11
The update preferences overlay must provide a ’arrive time’ user input field.	Should	SR12
The update preferences overlay must provide a ’round trip’ user input field.	Could	SR13

Table B.3: User Story 03

As a user, I want to be able to export the planned route for use on my mobile phone or GPS device.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide an option to export the planned route.	Must	SR14
The system must provide an export feature to export the route to the ’GPX’ file format.	Must	SR15
The system must provide an export feature to export the route to the ’GeoJSON’ file format.	Should	SR16
The system must provide an export feature to export the route direct to Strava.	Could	SR17

Table B.4: User Story 04

As a user, I want to share my route with other people.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide a share functionality overlay.	Should	SR18
The share overlay must provide the user with the option to share direct over email.	Should	SR19
The share overlay must provide the user with the option to share direct over Google Drive.	Could	SR20
The share overlay must provide the user with the option to share direct over OneDrive.	Could	SR21
The share overlay must provide the user with the option to share direct over Dropbox.	Could	SR22

Table B.5: User Story 05

As a user, I want to be provided with route suggestions based on predicted weather conditions over the week.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide the user with a weather condition overlay.	Must	SR23
The weather condition overlay must provide the user with the weather for the current day.	Must	SR24
The weather condition overlay must provide the user with the weather for the next week.	Should	SR25
The weather condition overlay must provide the user with the option to enable weather conditions in the route planning algorithm.	Could	SR26
The weather condition overlay must provide the user with suggestions on the best days to cycle.	Could	SR27

Table B.6: User Story 06

As a user, I want to view the route in detail and get information about parts of the route.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide the user with an interactive map to display the planned route.	Must	SR28
The interactive map must allow the user to zoom into parts of the planned route.	Must	SR29
The interactive map must allow the user to select parts of the route and receive detailed information about that subsection of the route.	Should	SR30
The interactive map must allow the user to select and drag the planned route to modify its path.	Should	SR31
The system must display an altitude graph for the planned route beneath the interactive map.	Should	SR32

Table B.7: User Story 07

As a user, I want to input hazards from routes I have cycled so the next route planned would attempt to avoid that area.		
Acceptance Criteria / System Requirements	Priority	ID
The system must provide a user input modal to input Hazard Data.	Must	SR33
The user input modal must provide a Type drop-down menu based on the OSM Hazard Types.	Must	SR34
The user input modal must provide a date entry point to specify the date the hazard was seen.	Should	SR35
The user input modal must provide a submit button to add the hazard to the hazard index.	Must	SR35

Appendix C

Increment 1 Screenshots

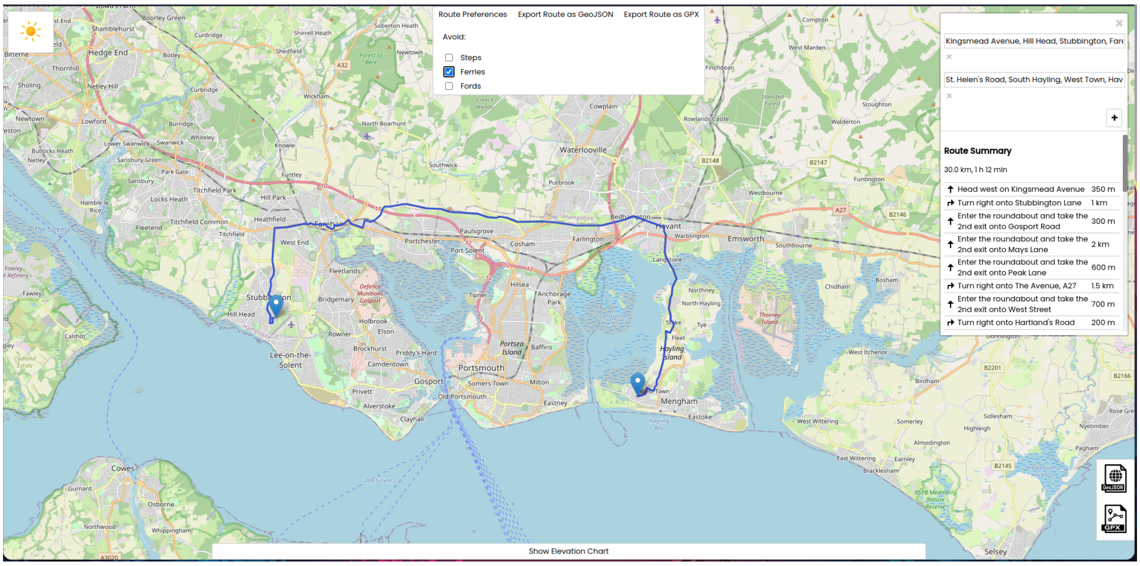


Figure C.1: Top Panel Added

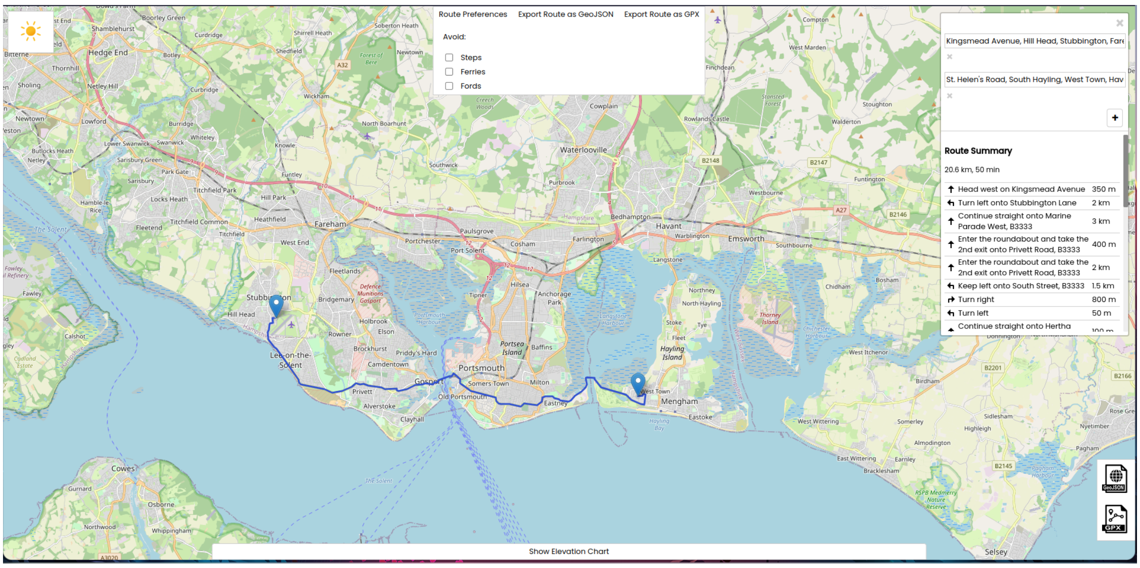


Figure C.2: Top Panel Added 2

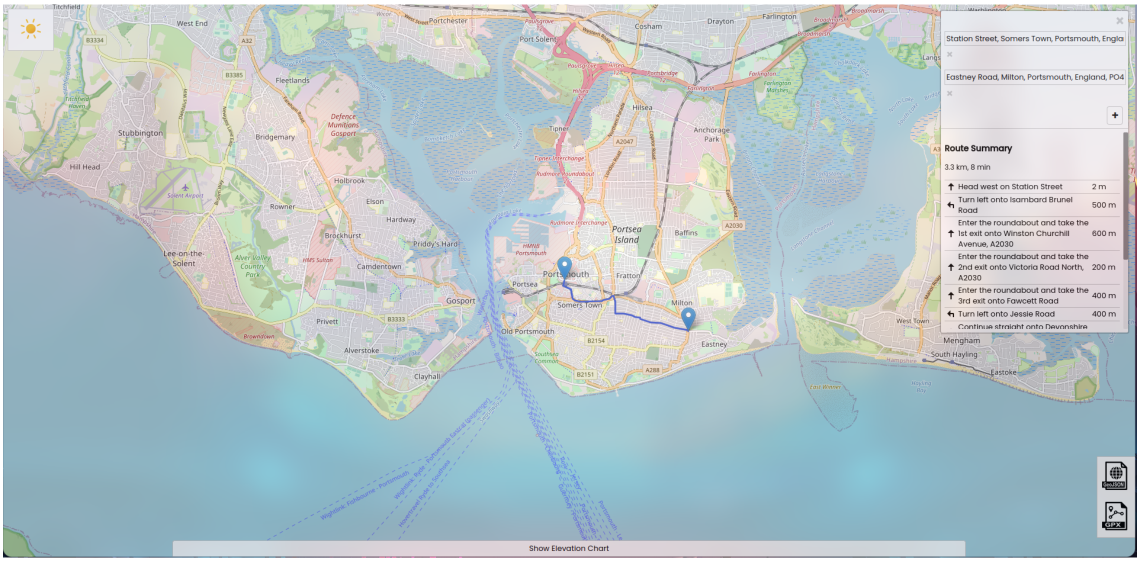


Figure C.3: GPX Icon Added

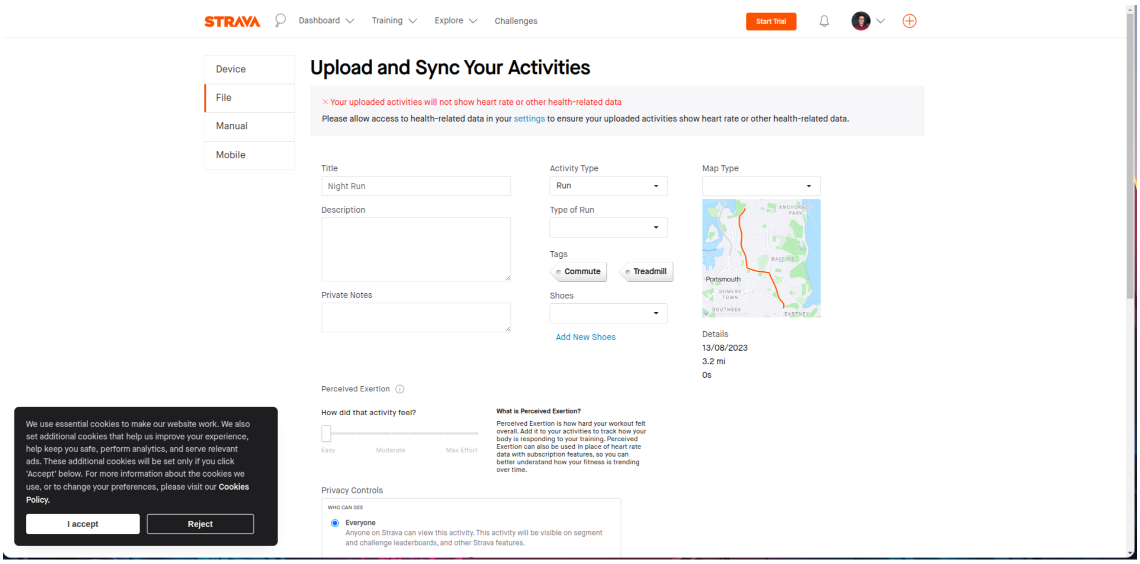


Figure C.4: Manual Strava GPX Upload Test

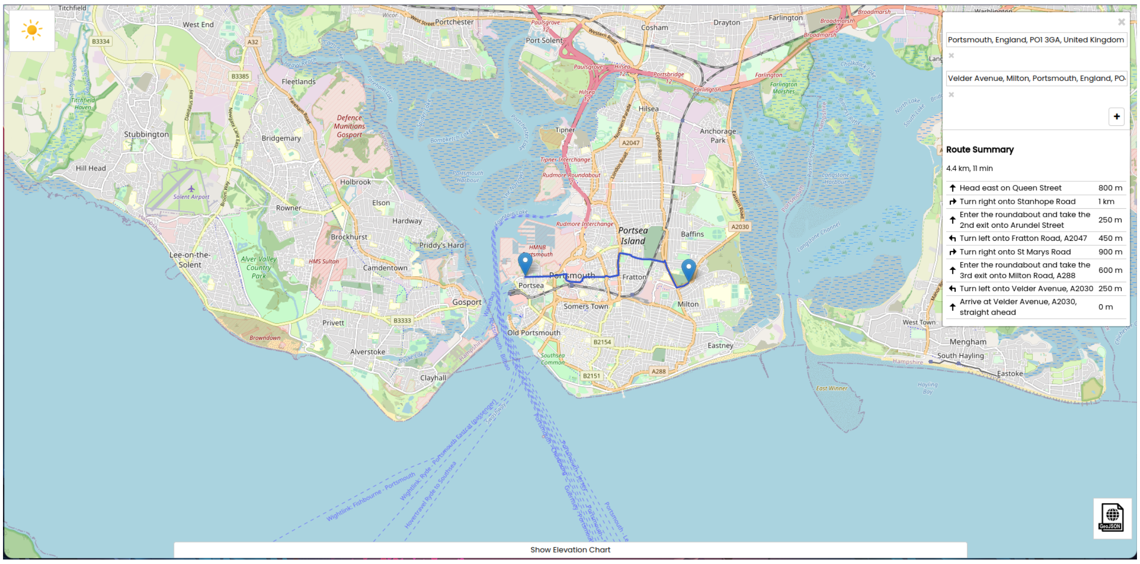


Figure C.5: GeoJSON Export

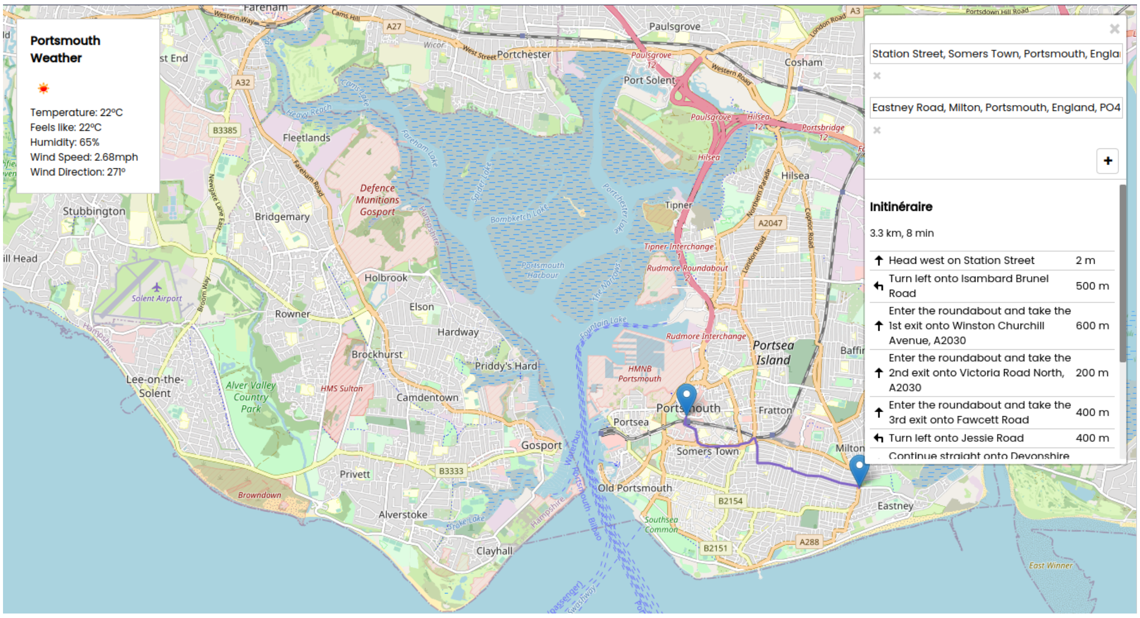


Figure C.6: Old Weather Panel

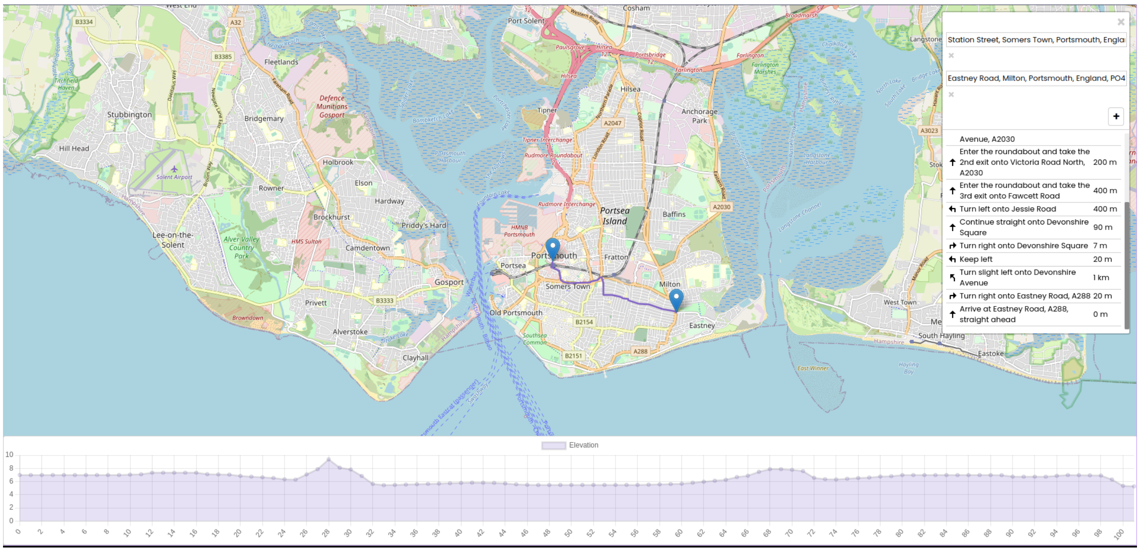


Figure C.7: Basic Elevation Plot

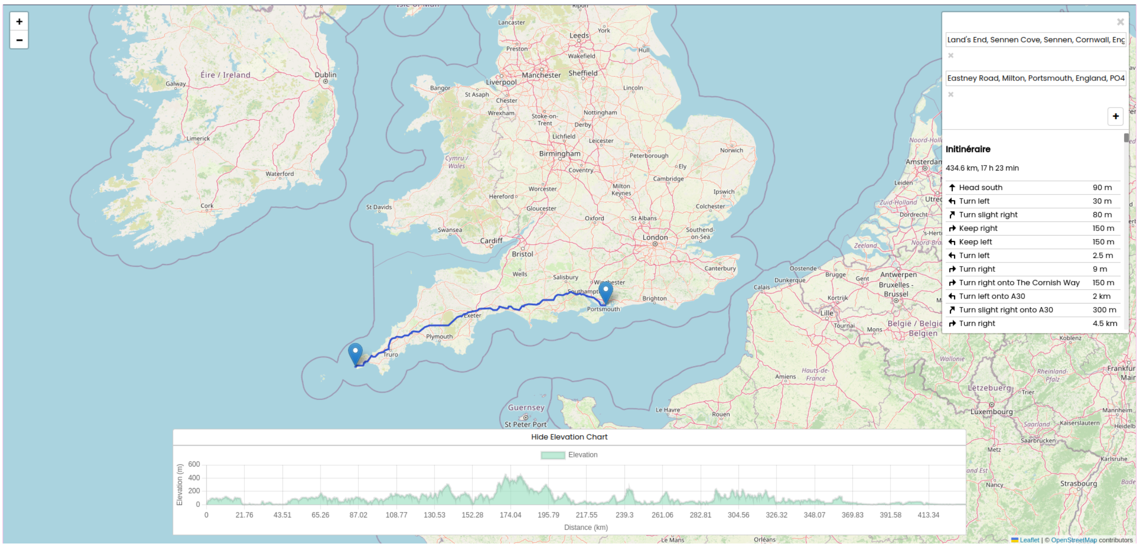


Figure C.8: Basic Elevation Plot Show

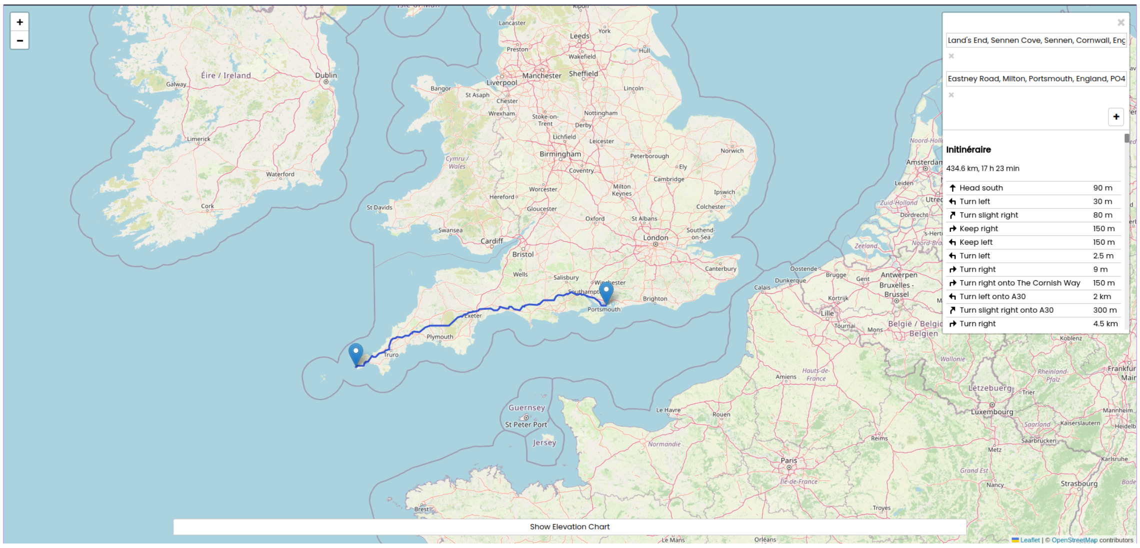


Figure C.9: Basic Elevation Plot Hide

Appendix D

Increment 2 Screenshots

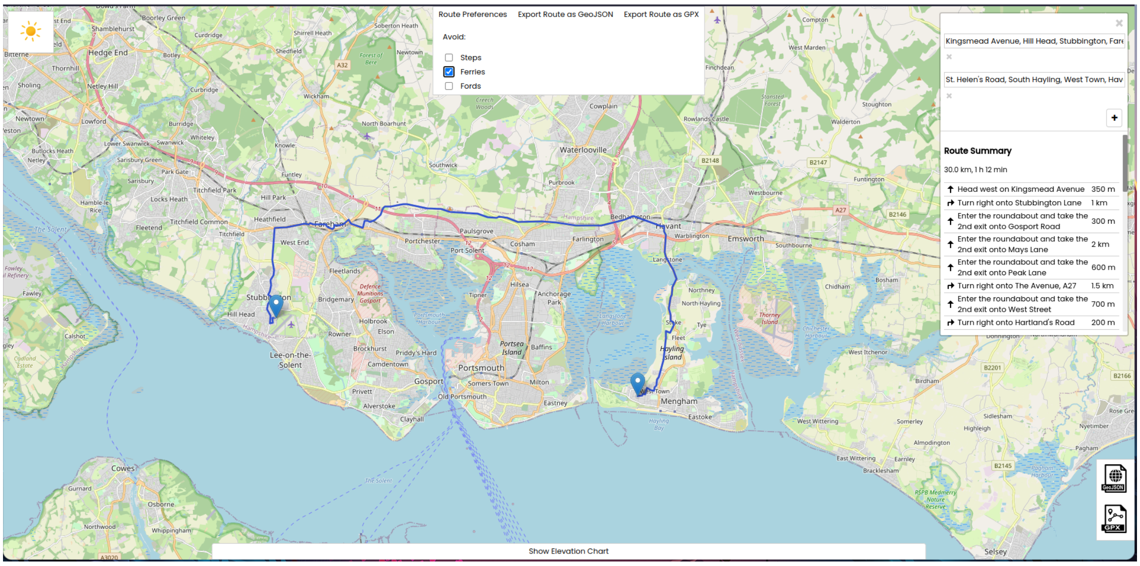


Figure D.1: Top Panel Added

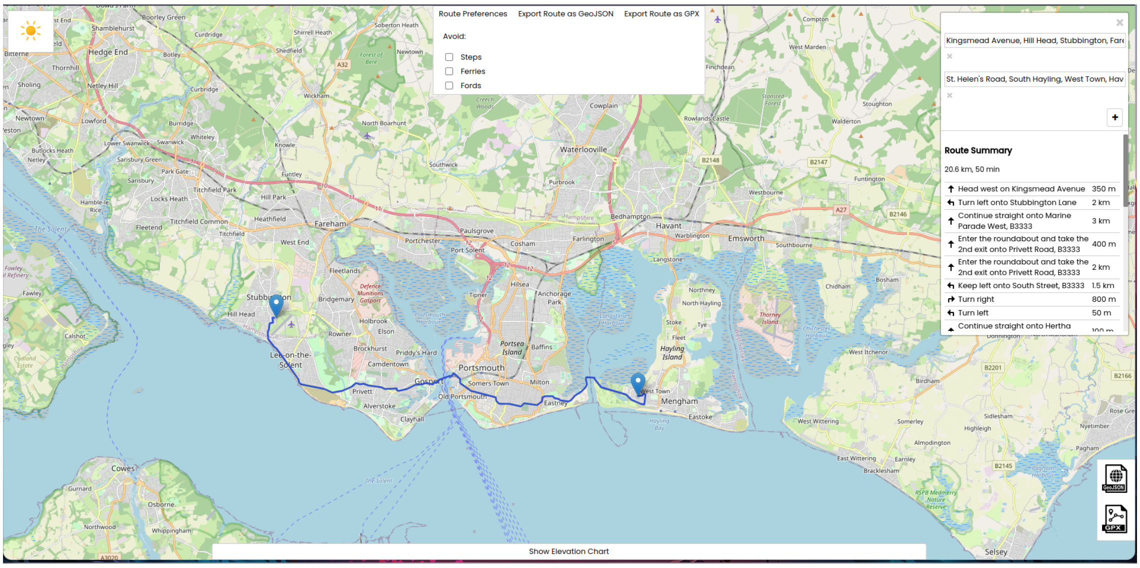


Figure D.2: Top Panel Added 2

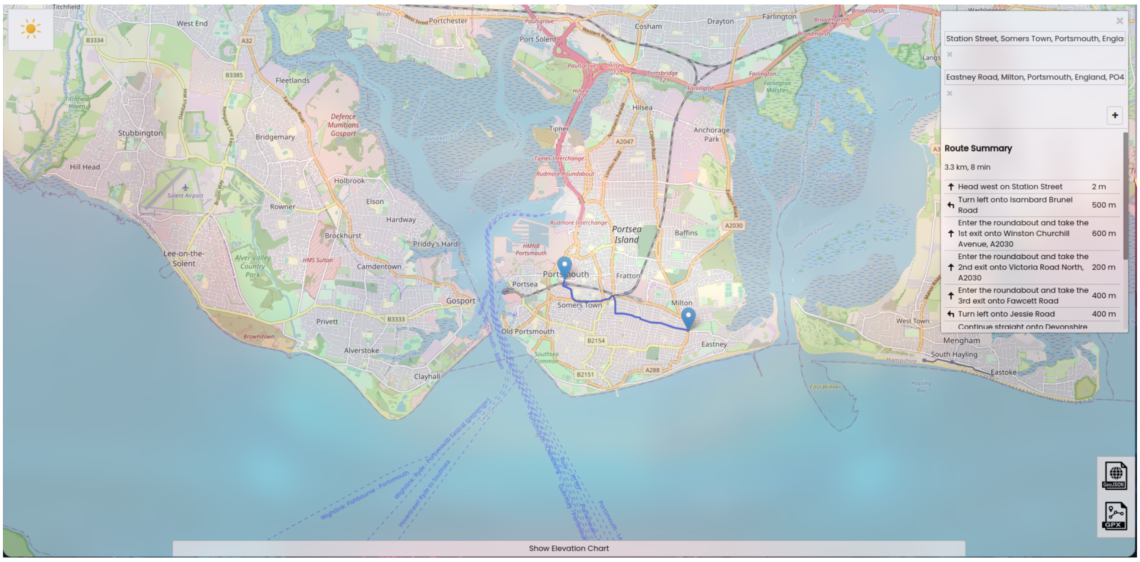


Figure D.3: GPX Icon Added

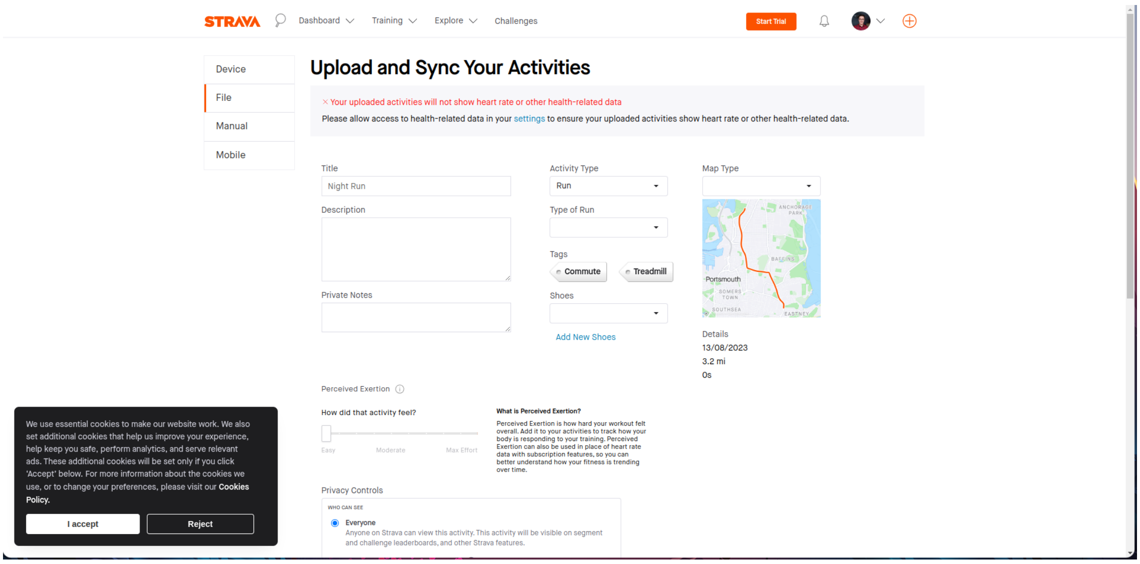


Figure D.4: Manual Strava GPX Upload Test

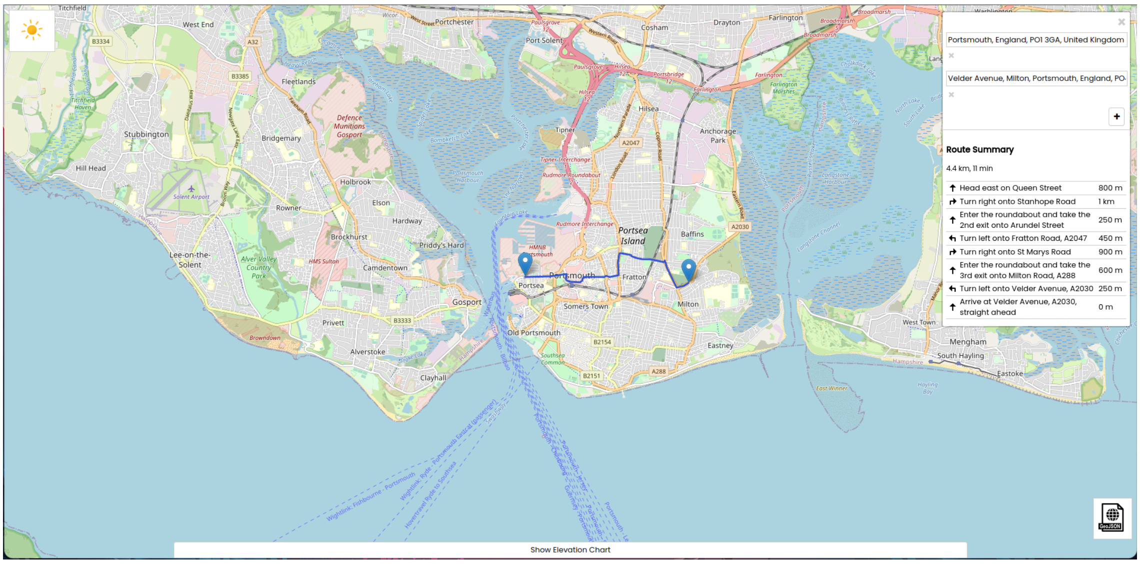


Figure D.5: GeoJSON Export

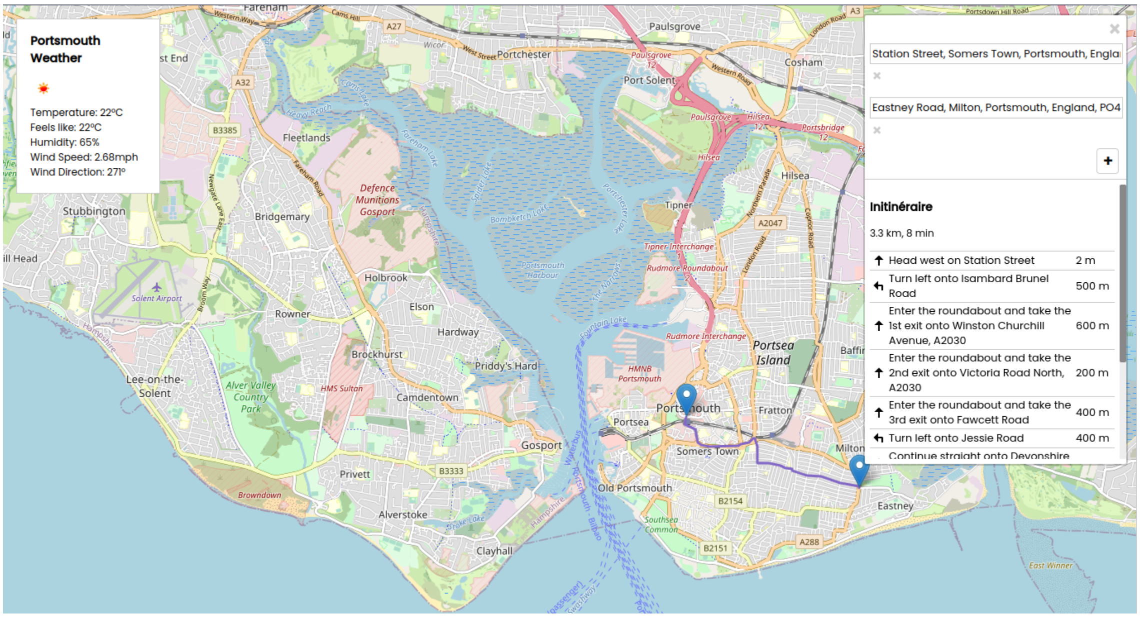


Figure D.6: Old Weather Panel

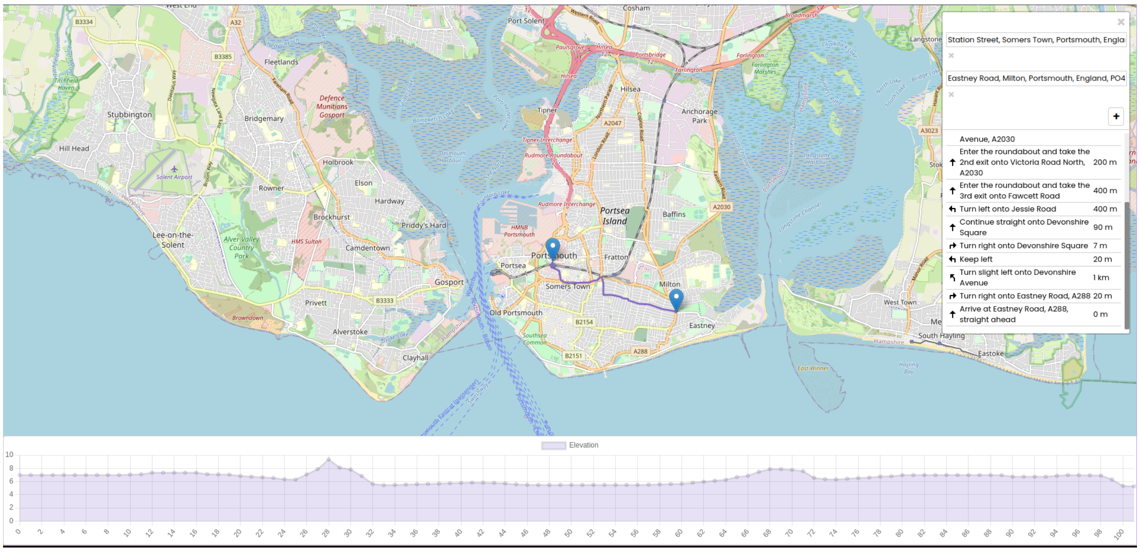


Figure D.7: Basic Elevation Plot

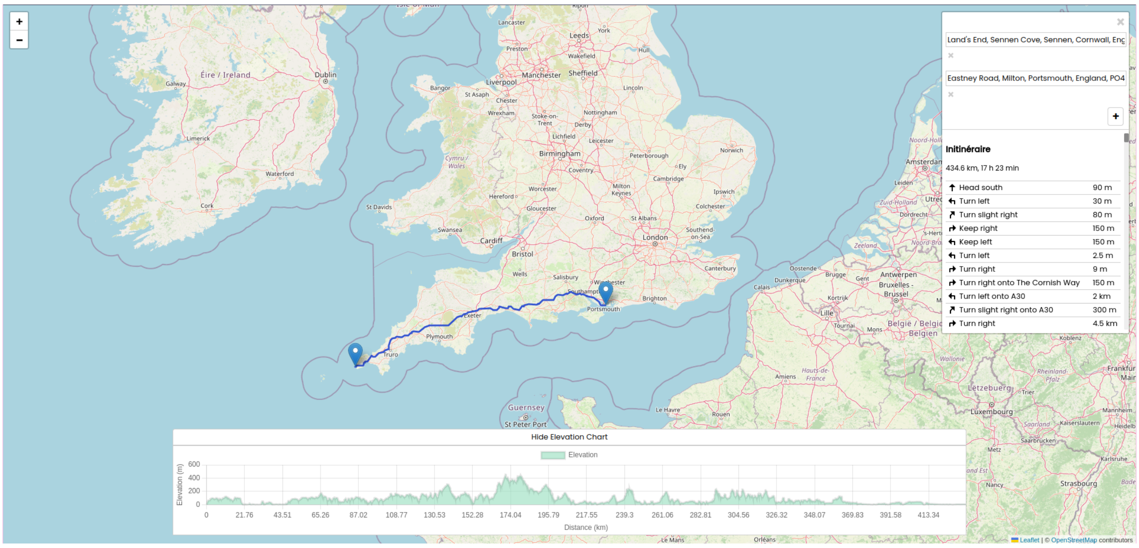


Figure D.8: Basic Elevation Plot Show

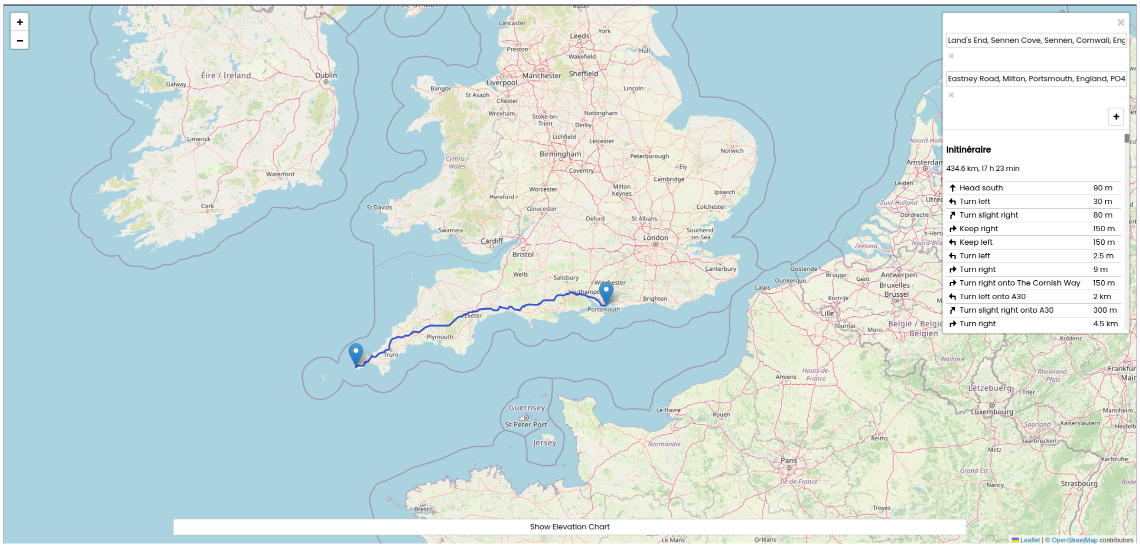


Figure D.9: Basic Elevation Plot Hide

Appendix E

Increment 3 Screenshots

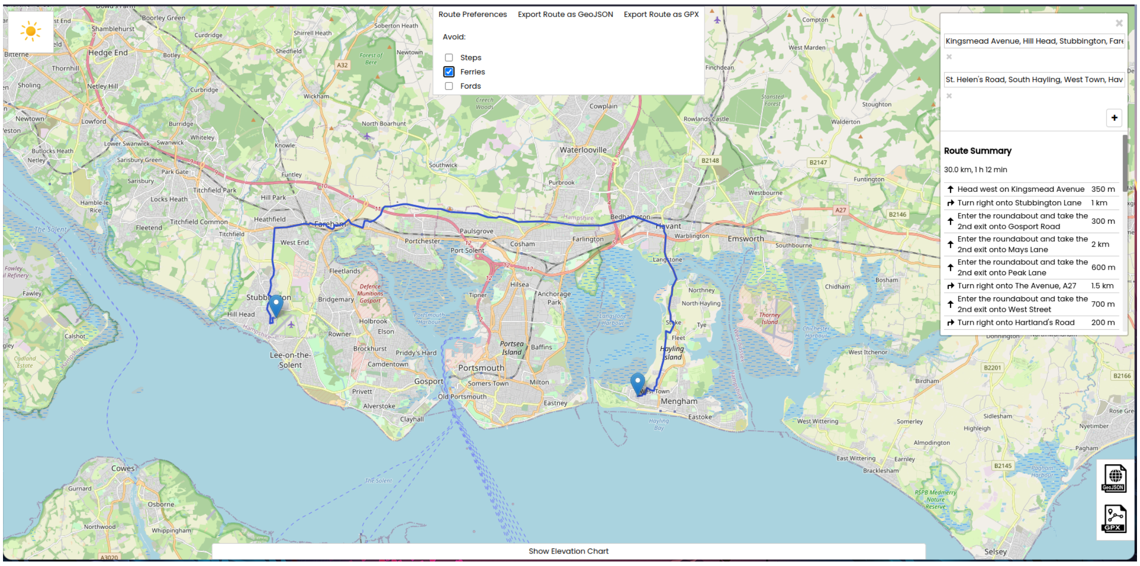


Figure E.1: Top Panel Added

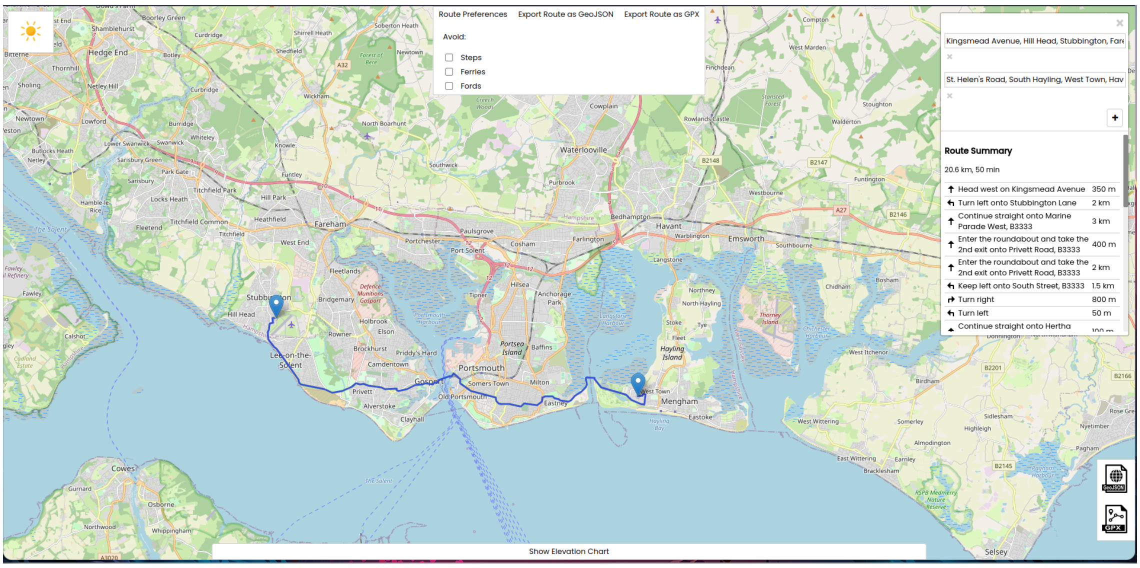


Figure E.2: Top Panel Added 2

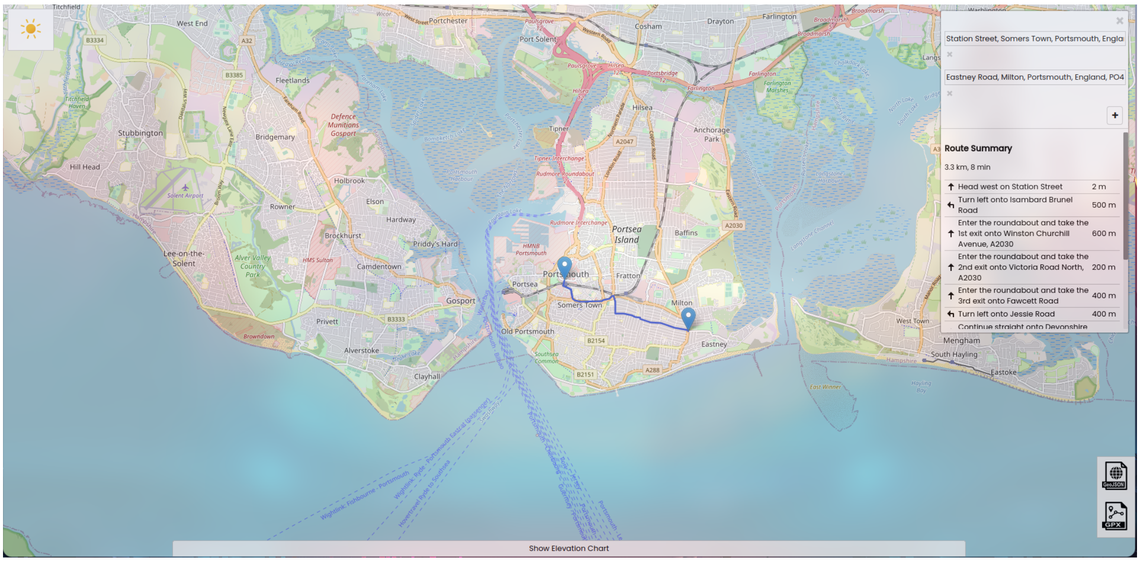


Figure E.3: GPX Icon Added

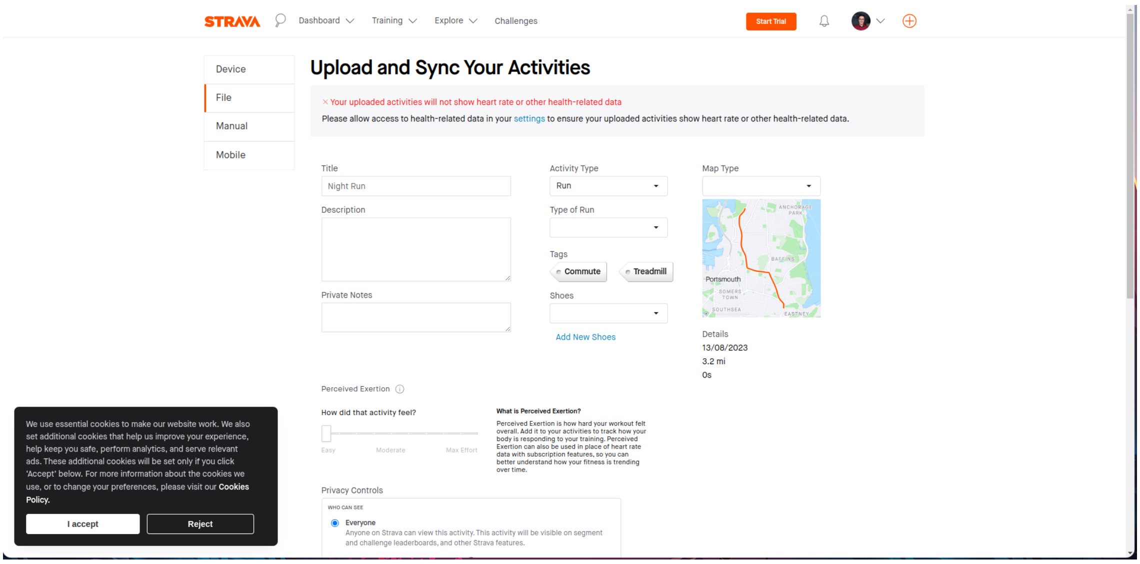


Figure E.4: Manual Strava GPX Upload Test

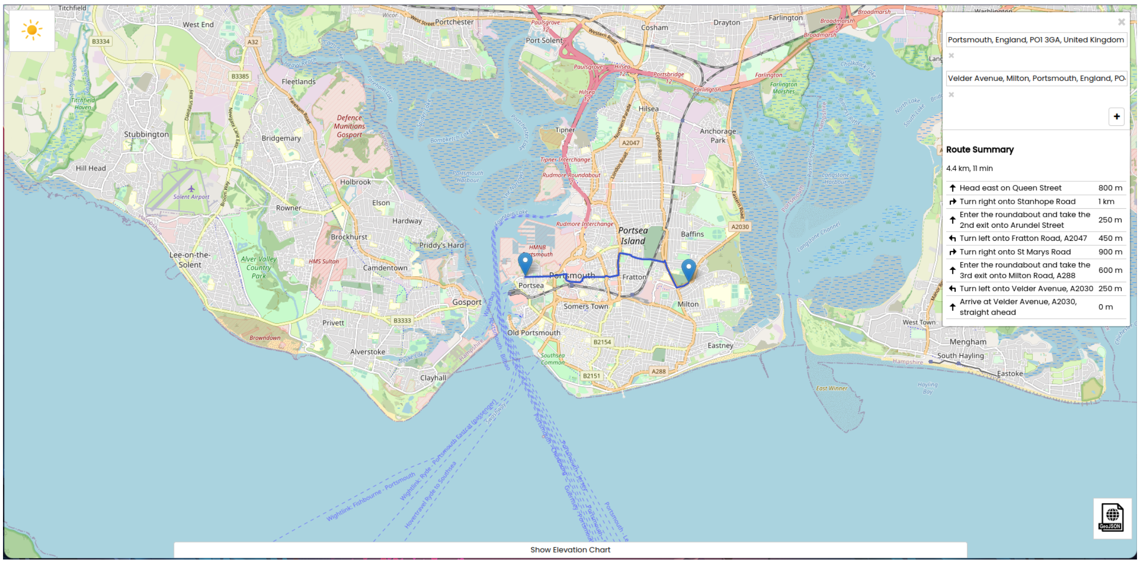


Figure E.5: GeoJSON Export

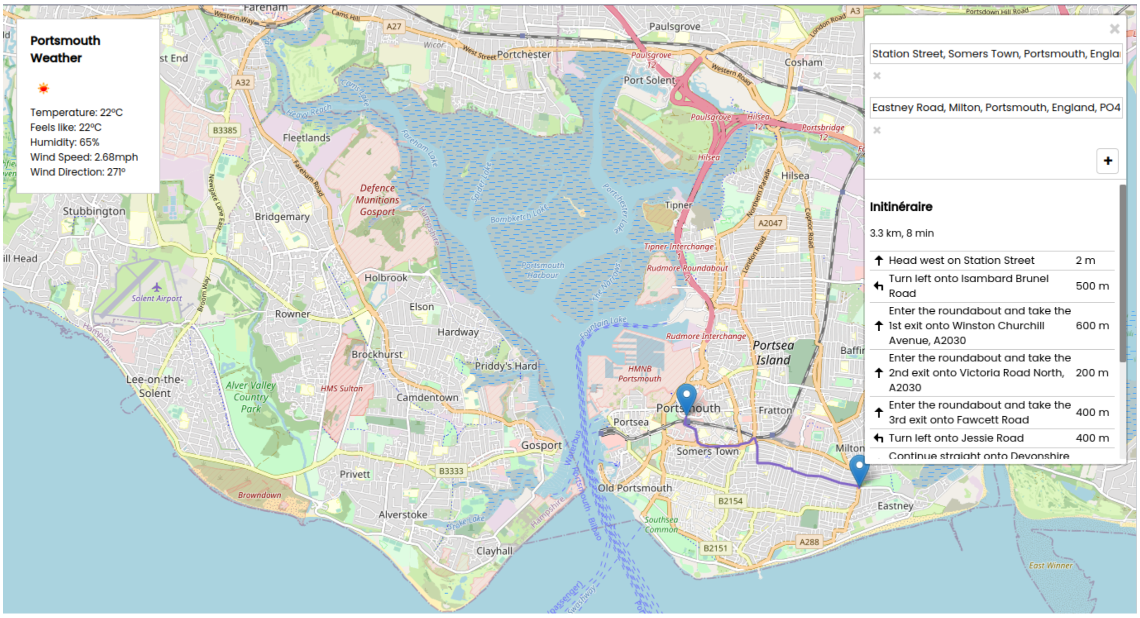


Figure E.6: Old Weather Panel

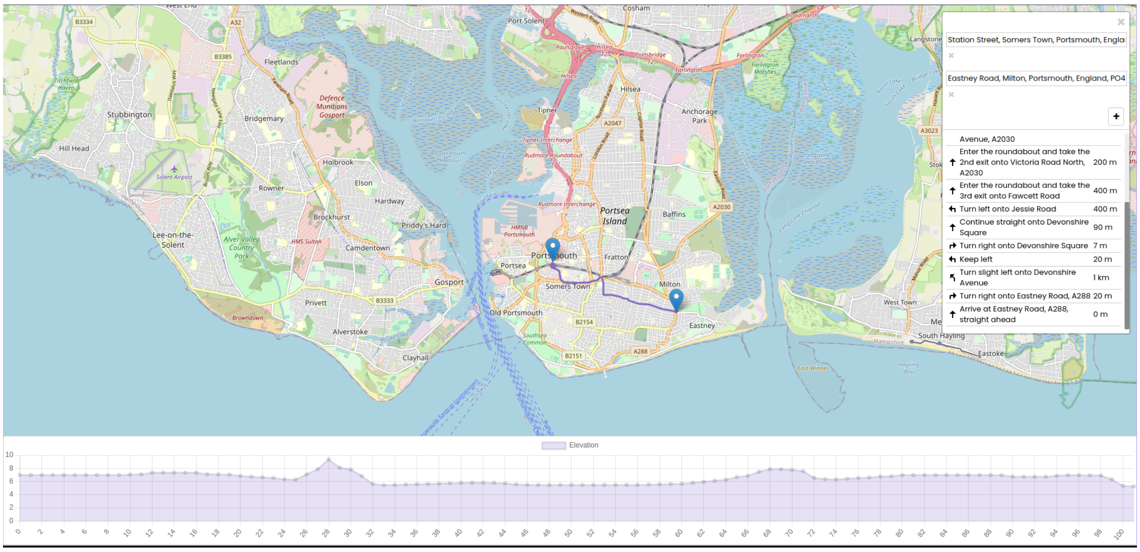


Figure E.7: Basic Elevation Plot

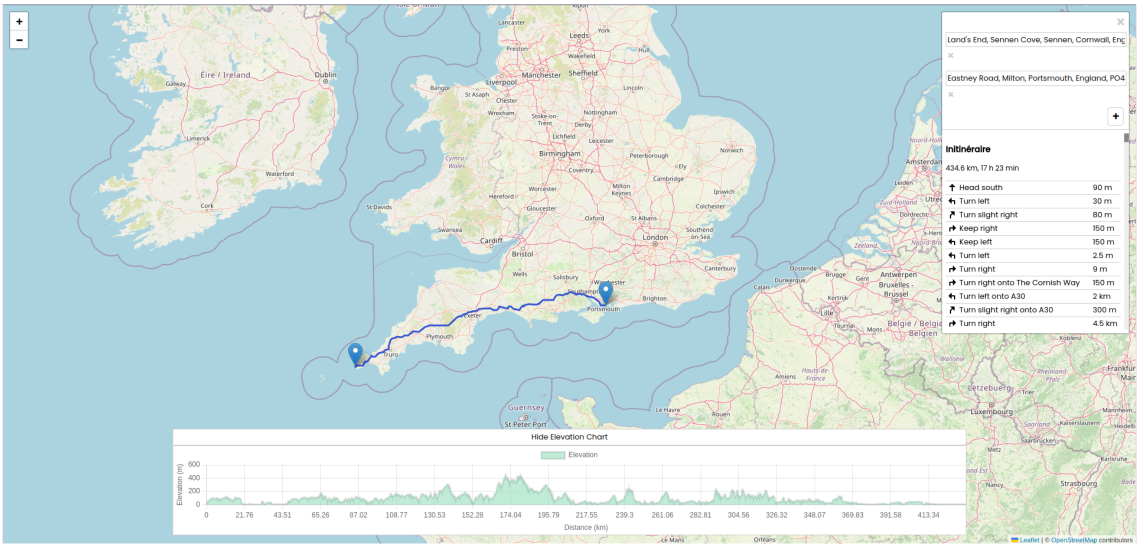


Figure E.8: Basic Elevation Plot Show

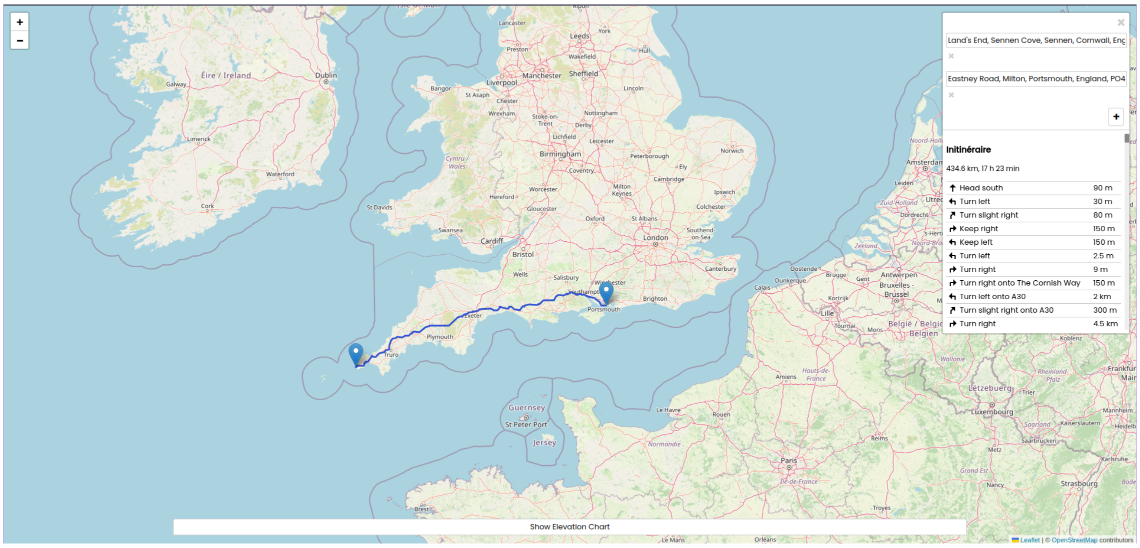


Figure E.9: Basic Elevation Plot Hide

Appendix F

Ethics Certificate

Certificate of Ethics Review

Project title: Bike Route App

Name:	Jake Bailey	**User ID**:	2002753	**Application date**:	11/10/2023 15:39:18	**ER Number**:	TETHIC-2023-106421

You must download your referral certificate, print a copy and keep it as a record of this review.

The FEC representative(s) for the **School of Computing** is/are **Elisavet Andrikopoulou, Kirsten Smith**

It is your responsibility to follow the University Code of Practice on Ethical Standards and any Department/School or professional guidelines in the conduct of your study including relevant guidelines regarding health and safety of researchers including the following:

- University Policy
- Safety on Geological Fieldwork

It is also your responsibility to follow University guidance on Data Protection Policy:

- General guidance for all data protection issues
- University Data Protection Policy

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Dr Rich Boakes**
Will you gather data about people (e.g. socio-economic, clinical, psychological, biological)?: No
Will you gather data from people about some artefact or research question (e.g. opinions, feedback)?: Yes
Confirm whether and explain how you will use participant information sheets and apply informed consent.: No personal information will be captured, only personal opinions. A participant information sheet will be provided alongside the research application where the participant MUST agree to the conditions before they are able to submit their opinion on the topic.
Confirm whether and explain how you will maintain participant anonymity and confidentiality of data collected: All that will be stored is an array of numbers that is the result of a card sort relating to features to be implemented into the application.
Will the study involve National Health Service patients or staff?: No
Do human participants/subjects take part in studies without their knowledge/consent at the time, or will deception of any sort be involved? (e.g. covert observation of people, especially if in a non-public place): No
Will you collect or analyse personally identifiable information about anyone or monitor their communications or on-line activities without their explicit consent?: No
Does the study involve participants who are unable to give informed consent or are in a dependent position (e.g. children, people with learning disabilities, unconscious patients, Portsmouth University students)?: No
Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants?: No
Will blood or tissue samples be obtained from participants?: No
Is pain or more than mild discomfort likely to result from the study?: No
Could the study induce psychological stress or anxiety in participants or third parties?: No
Will the study involve prolonged or repetitive testing?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No

Will your project or project deliverables be relevant to defence, the military, police or other security organisations and/or in addition, could it be used by others to threaten UK security?: No
Please read and confirm that you agree with the following statements: I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc.), I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs, I confirm that I will act ethically and honestly throughout this project

Supervisor Review

As supervisor, I will ensure that this work will be conducted in an ethical manner in line with the University Ethics Policy.

Supervisor comments:

Supervisor's Digital Signature**: rich.boakes@port.ac.uk** Date**: 18/10/2023**

Faculty Ethics Committee Review

Ethics Rep comments: **simple app, no concerns**

Faculty Ethics Committee Member's Digital Signature(s)**: elisavet.andrikopoulou@port.ac.uk** Date**: 19/10/2023**

Appendix G

Project Initiation Document

School of Computing
Final Year Engineering Project

Project Initiation Document

Jake Bailey

Long Distance Cycling Route Planner

1. Basic details

Student name:	Jake Bailey
Draft project title:	Long Distance Cycling Route Planner
Course and year:	BSc (Hons) Software Engineering Year 3
Project supervisor:	Dr Rich Boakes
Client organisation:	
Client contact name:	

2. Degree suitability

The artefact relates to my Software Engineering degree because it will utilise various technologies learned in this course and during my placement year thus far. The challenge of developing this application is greater as I will demonstrate my skills in building a web application to calculate and plot routes on OpenStreetMap based on a set of conditions determined by user input and publicly available databases of hazard data and weather predictions in real-time.

3. Outline of the project environment and problem to be solved

I'm going to build a prototype web application targeted at cyclists who wish to plan their route before their ride. The route planner will focus on customisability of the routing algorithm and safety of the cyclist. The prototype may not only consider user input into the routing algorithm, but other conditions such as, weather, road, elevation, traffic and many more.

Most of this functionality in pre-existing route planners either doesn't exist and/or is locked behind a paid service. The aim is to provide cyclists with as much flexibility as possible when planning a route whilst considering the safety of the cyclist, whether on a short or long ride.

The application will be developed as a modular system to ensure this planned flexibility. This means users can tailor the functionality according to their needs, allowing different components of the system to seamlessly interact when planning different types of routes. Furthermore, the system should ensure efficient route calculation even when utilising these distinct modules.

4. Project aim and objectives

Overall aim:
To build a prototype route planning application to be further expanded beyond the scope of this project. The prototype will allow cyclists to customise a range of datasets to plot a route custom-fit to the user's needs.

Objectives:

- Research open-source routing algorithms to understand how different options function.
- Research into mapping options, for example:
    - OSM (Open Street Map).
    - Google Maps.
- Discover what data is available and how I can access it.
- Understand how to utilise the available data in the different routing algorithms.
- Build an intuitive and accessible UI (User Interface).

5. Project constraints

- Fixed deadline for project completion.
- Only one team member.
- Limited funds to pay for API access.
- Limited funds to test routing algorithms on high-end machines.

6. Facilities and resources

I will use my home PC, Laptop, and university devices to conduct secondary research and to develop and test the proposed artefact. Mobile phones and activity trackers may also test certain functionality, such as importing the generated GPX file into a mapping application.

A potential constraint could arise where I cannot test using activity trackers like Garmin devices due to the cost of such devices. As I do not own an activity tracker other than a mobile phone, I could request permission from students who own such devices to conduct primary research and test the exported file generated by the artefact. Using other students' devices, however, would develop a time constraint for the project, whereby I am required to return the device after some time.

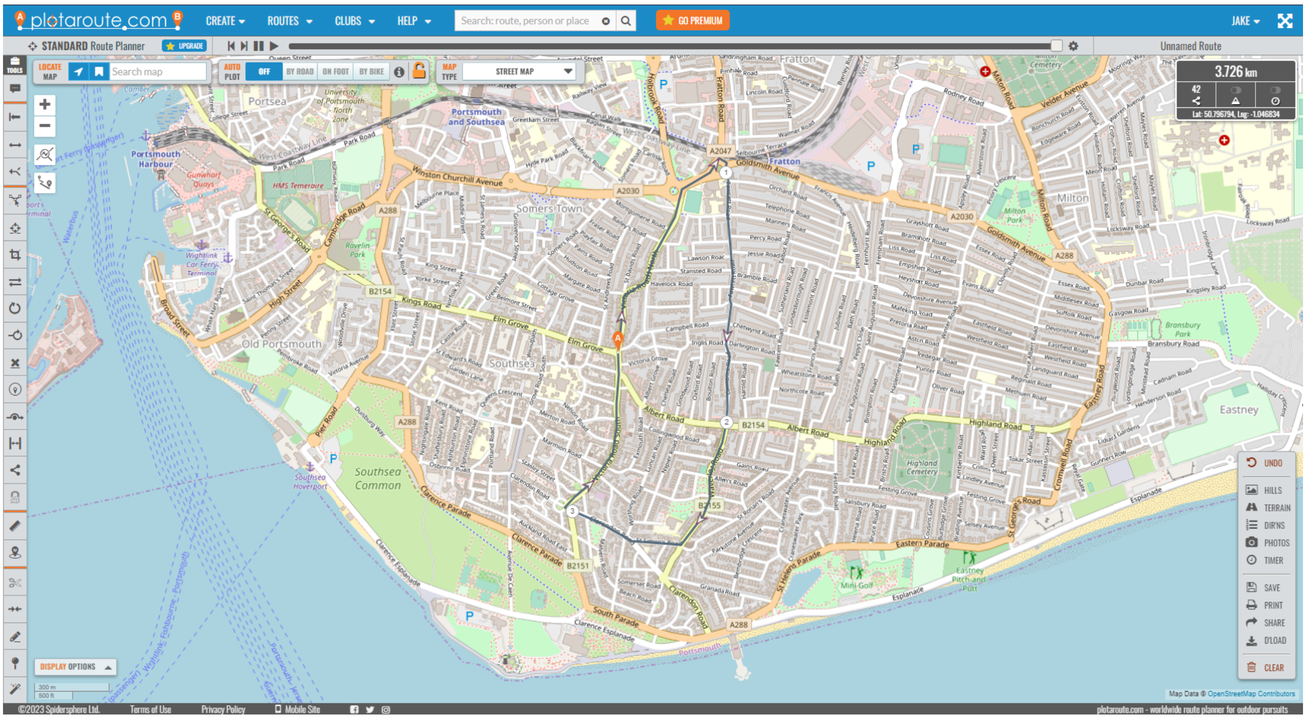
7. Log of risks

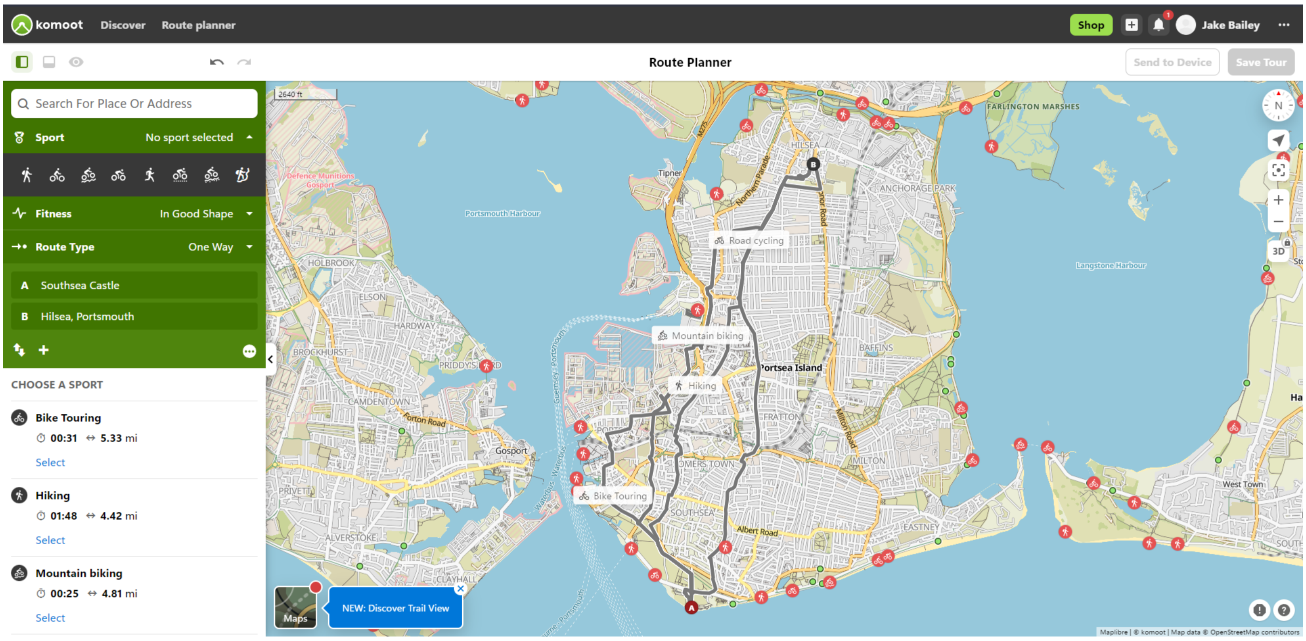
Description	**Impact**	**Mitigation/Avoidance**
Supervisor becomes unwell	Cause delays in the project due to lack of insight and guidance.	If possible, conduct online meetings and plan time in preparation for the potential delay.
I become unwell	Impact on the scope of the project, delaying the timeline.	Visit a doctor if unwell and allocate time for rest around university work.
Change in requirements.	Impact on the scope of the project, delaying the timeline.	Regular communication with the supervisor to discuss the feasibility of the current requirements to prevent any future changes.
Time availability	Decreases the feasibility of	Plan tasks ahead of time to consider potential future setbacks.

	completing the project by the submission date.	
Data loss	The application will no longer work. The development would have to start from scratch.	Back up all project files regularly. Use git and GitHub to back up and manage the codebase.
Hardware failure	Delays the project timeline; new devices would need setting up.	Use multiple devices to ensure progress continues while the other device is in repair.

8. Project deliverables

The artefact that will be developed will be a prototype web application with an accessible and user-friendly UI design; it will display a map and a menu allowing the user to customise a route they want to plan. The artefact will also display information about the route with data to be useful to the user in deciding which route to pick. The UI will share similarities to plotaroute.com (*Route Planner for Walking, Running, Cycling - Plotaroute.Com*, n.d.) and Komoot (*The Best Route Planner for Cycling, Walking, Hiking and Running*, n.d.):





The primary documentation piece to be produced will be the final project report. All other documentation may include:

- Design Specification
    - User Interface Wireframes
    - User Interface Designs
- Unified Modelling Language
    - Use Case Diagrams
    - Sequence Diagrams
- System requirements
    - Functional requirements
    - Non-functional requirements
- User Guide
- Code documentation (inline)
- Installation and setup instructions
    - In a README.md file
    - In the User Guide
- Testing documentation – Jest will be used.
    - Test instructions
    - Test plan
    - Test cases
    - Test logs

9. Project approach

I will plan the project using elements of the Agile Software Development Lifecycle methodology, enabling development to adapt to an unplanned requirements change. I will use the GitHub Kanban board to manage and track the project's progress. Using GitHub's Kanban board will enable the code repository to be closely linked to the board and ensure tasks are completed within their expected timeframes. GitHub also allows pull requests to be linked to open issues and labels to be applied to those issues, which gives more context as to what the specific task is.

Secondary research will be completed before development begins; I will research the current systems in use to gain a greater understanding of how they function and the data they use. Different websites and online documentation for APIs, such as Open Street Map, will be useful in determining how I can utilise existing services and integrate them into the artefact.

Primary research will be necessary for the project. Utilising the preferences available for the routing algorithm, I will research how different datasets affect the quality of the final route when subjectively assessed by the users. This will determine which preferences are vital to the route planning algorithm and which will be optional.

10. Project plan

The key stages of the project will comprise:

- Secondary Research & Literature Review
- Create UI Wireframes for Artefact
- Create UI Designs for Artefact
- Establish requirements
    - Functional
    - Non-Functional
- Develop the artefact prototype
    - Conduct primary research (during development)
    - Develop tests (during development)
    - Write artefact documentation
- Conduct primary research (after development)
- Conduct tests (after development)
- Write the final report

I will conduct secondary research whilst writing the literature review during the initial stage of the project. The wireframe will be designed as a first UI design draft before the final, detailed UI design is developed. An initial set of requirements will be set before development begins, and these requirements will be converted to issues on GitHub's Kanban board; the requirements may likely change during the development of the artefact.

Accessing and manipulating the data from a range of APIs will be challenging to translate to a format accepted by the routing algorithm. I will be utilising my current knowledge of software engineering, web development and RESTful APIs whilst developing my knowledge further through research, other students, and my supervisor.

11. Supervisor Meetings

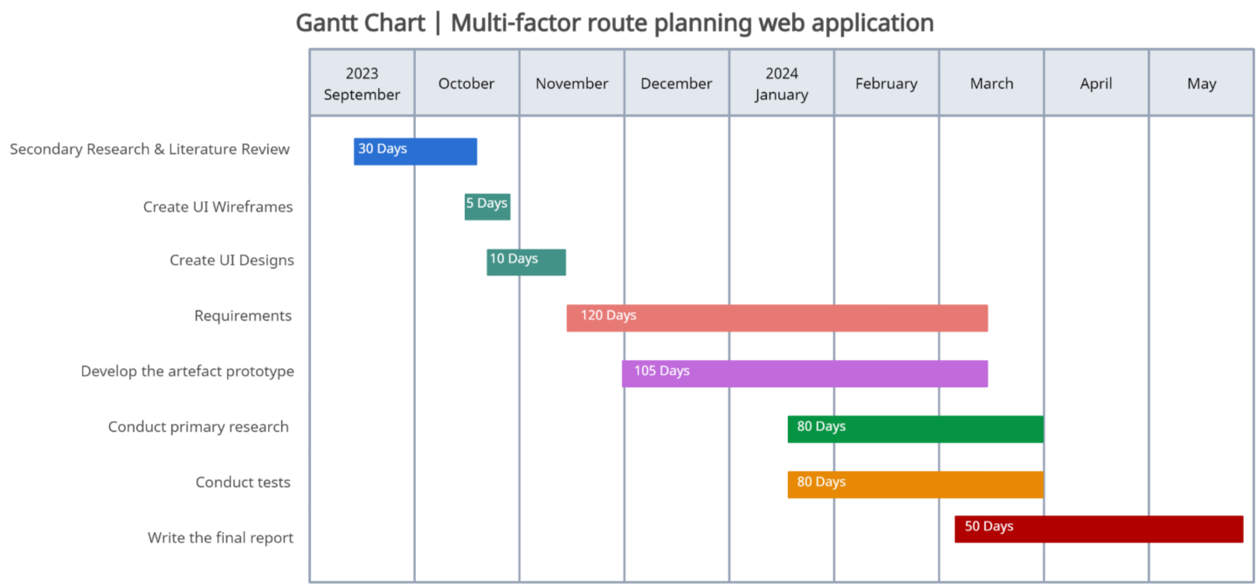
I will schedule meetings on a weekly or fortnightly basis; my supervisor allocates a range of slots each week where I can book a time which works for me. I aim to have all meetings face-to-face. However, if my supervisor or I become unwell, we will conduct a video conference where possible. If my supervisor has planned leave, or I will be away for some time, we will communicate ahead of time to devise a plan for the project while one or the other is away and unable to meet regularly. Doing so will further allow me to plan my workload ahead of time and effectively keep track of the progress throughout the project.

12. Legal, ethical, professional, social issues

The key legislation I must consider for this project is the Data Protection Act 2018 (DPA 2018). I am not planning on storing personal information. However, the user's current location will be requested upon the launch of the application; when the application no longer uses this data, it will be deleted and re-requested when the user enters the application again. Future iterations could implement accounts, storing a small amount of sensitive user information to include more features. However, the submitted artefact shouldn't contain this data; regardless of this fact, I will ensure the artefact abides by all principles of the DPA 2018 due to it handling the current location data of the user.

One social issue that could arise is that the artefact may entice more public members to start cycling more frequently; whilst this result will be a great incentive for protecting the environment, some road users are cautious, with many cyclists riding unsafely on the roads. The artefact will push users to ride safely and abide by all road safety laws, just as vehicles do; there will also be the option only to use cycle routes/lanes when plotting a route to ensure those cyclists who are less comfortable on roads still feel safe on the routes planned by the artefact.

Appendix B: Gantt chart



Bibliography

- *Route Planner for Walking, Running, Cycling—Plotaroute.com*. (n.d.). Retrieved 12 March 2023, from https://www.plotaroute.com/routeplanner
- *The Best Route Planner for Cycling, Walking, Hiking and Running*. (n.d.). Komoot. Retrieved 12 March 2023, from https://www.komoot.com/plan