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, p14).

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 (see Section 2.3, p17)(see Section 2.4, p20). 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 2.5, p22).

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 2.6.1, p23) and weather conditions (see Section 2.6.2, p25).

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 JavaScript and JavaScript Frameworks

JavaScript has existed for many years and is now one of the most widely known and used front-end programming languages in web development (Mariano, n.d.). Due to the language's wide popularity within web development, various frameworks have been developed to extend the pre-existing functionality of vanilla JavaScript. JavaScript frameworks enable web development by utilising a component-based approach, allowing for these components to be designed and developed to be reused. Creating reusable components aids in reducing the software development time and increasing the reliability of the product (Crnkovic and Larsson, 2002).

There are a range of different JavaScript frameworks that both increase and decrease in popularity over time. React, Vue and Angular are at the current forefront of JavaScript frameworks, whereas other frameworks, such as JQuery, used to be popular but have recently decreased in popularity. The Stack Overflow insights tool demonstrates the decline in user questions surrounding different JavaScript frameworks, paralleling the decline of JQuery and the adoption of React, Vue, and Angular (see Figure 2.1, p18).

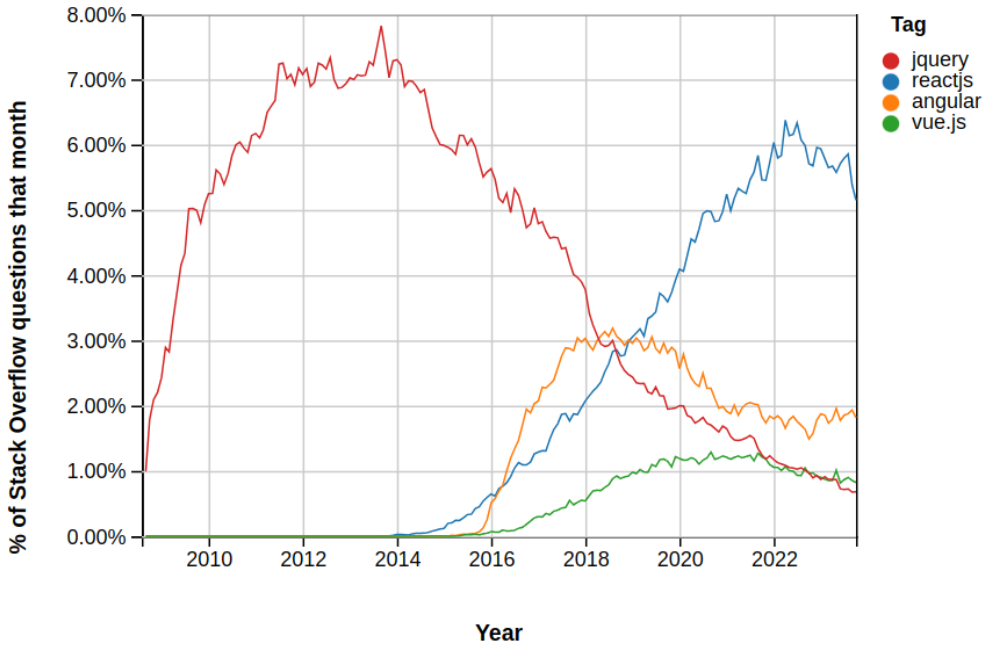

Figure 2.1: Stack Overflow Trends JavaScript Frameworks (14/11/2023) ("Stack Overflow Trends", n.d.-a)

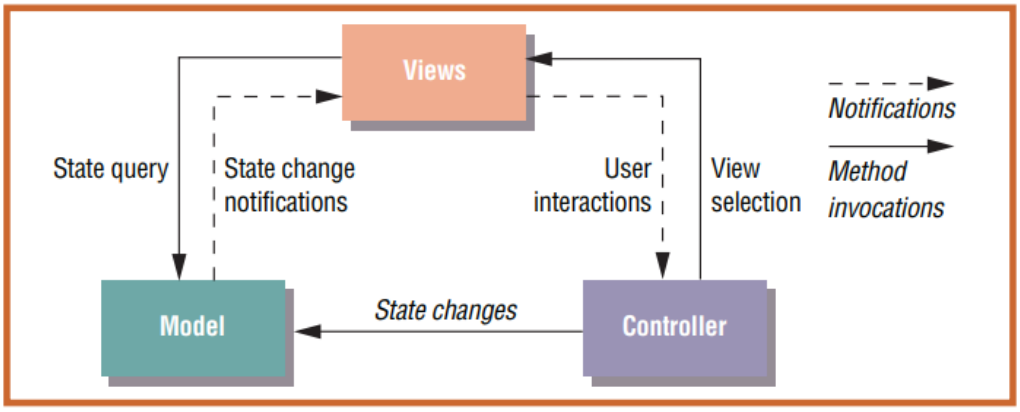

Figure 2.2: Model View Controller Model (Curry and Grace, 2008)

2.3.1 React

The React.js framework was originally built by software engineers at Facebook with the intention of developing a JavaScript library for building user interfaces. React's main focus is to enable developers to "combine a number of short independent code fragments into a complex UI interface" (Xu, n.d.). React works similarly to the MVC (Model View Controller) model (see Figure 2.2, p18); it simply acts as the 'View' part of the model where it interacts with and utilises the VDOM (Virtual Document Object Model). When a state or props change occurs for a component, React compares the newly returned VDOM component with the previously rendered DOM component. If these components are not equal, the entire DOM is re-rendered (Mariano, n.d.).

2.3.2 Next.js

Next.js is a framework built on the original React.js framework and has recently become the new default for new React projects. It natively incorporates multiple production-ready such as server-side rendering. This feature, in particular, is one key improvement Next.js introduces whereby JavaScript modules are dynamically loaded on the server, drastically decreasing the client-side loading times (Dinku, 2022). Despite these improvements to React, introducing features such as server-side rendering can reduce the framework's compatibility with libraries such as Leaflet for OpenStreetMap which is pivotal to this project.

2.3.3 Vue

In contrast to React, Vue was released in 2014, and instead of being built from the MVC model, it uniquely provides MVVM (Model View View-Model) data binding and a composable component system (Xu, n.d.). The MVVM data binding is two-way, where the component is divided into a model and a view; once the binding is created, the binding will be synchronised with the data (Li and Zhang, 2021). This enables Vue to have a similar state management system to React whereby the DOM will be re-rendered when the state of a component changes.

2.3.4 Angular

Angular has two separate versions, AngularJS (2010) and Angular2+ (2016); these are similarly both referred to as AngularJS; however, Angular2+ is the latest, maintained version based on Typescript, whereas AngularJS is built with JavaScript and is no longer actively maintained. Angular2+ is discussed within this review and will be referred to as Angular from now onwards.

Angular interacts with the DOM differently to React and Vue as Angular does not utilise the VDOM, instead handling only direct DOM manipulations (Levlin, 2020), but is still based on the MVC model (see Figure 2.2, p18). Angular interacts most similarly to JQuery out of all the modern frameworks. It implements create functions, such as

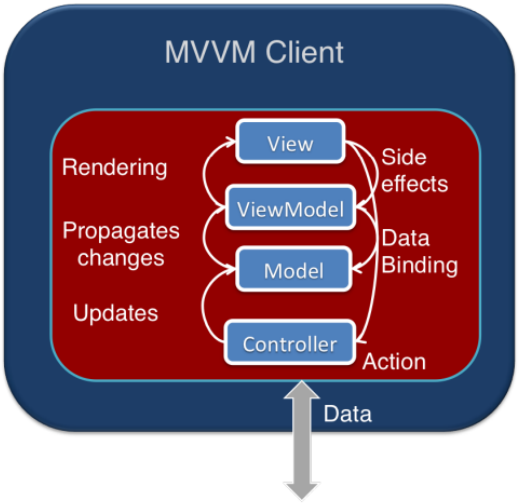

Figure 2.3: Model View View-Model Model (Zarifis, n.d.)

`createCustomElement()`, which can be used to covert existing components into a class to be displayed on the DOM (Levlin, 2020). These classes contain decorators, which not only provide metadata but are also responsible for updating the component. Updates made to components via the decorator will update the existing DOM without the need to update a VDOM and compare it with the DOM (as done in React and Vue).

2.4 Go (Programming Language)

Go, sometimes referred to as Golang, was developed by Google in 2007 and "arose through experience building large-scale distributed systems, working in a large codebase shared by thousands" (Cox et al., 2022). Golang was built during a time when multi-core systems were becoming more and more common. Software Engineers at Google at the time found that the whole industry struggled to write efficient systems for multicore systems, and Go was their solution to the problem. Go has become increasingly popular over recent years (see Figure 2.4, p21), primarily because the entire industry now faces those original issues highlighted by Google in 2007 when they were developing Go (Cox et al., 2022).

Concurrency is one of Go's headline features. The language splits its concurrency model into two fundamental elements: Goroutines (lightweight threads) and Channels, which enable multiple Goroutines to communicate and synchronise execution (Meyerson, 2014). Go follows the fork-join concurrency model; fork refers to how, at any point, it can split a child branch of execution which runs concurrently alongside the parent branch (Cox-Buday, 2017). Once the child process has finished execution, it will rejoin the parent at the join (see Figure 2.5, p21).

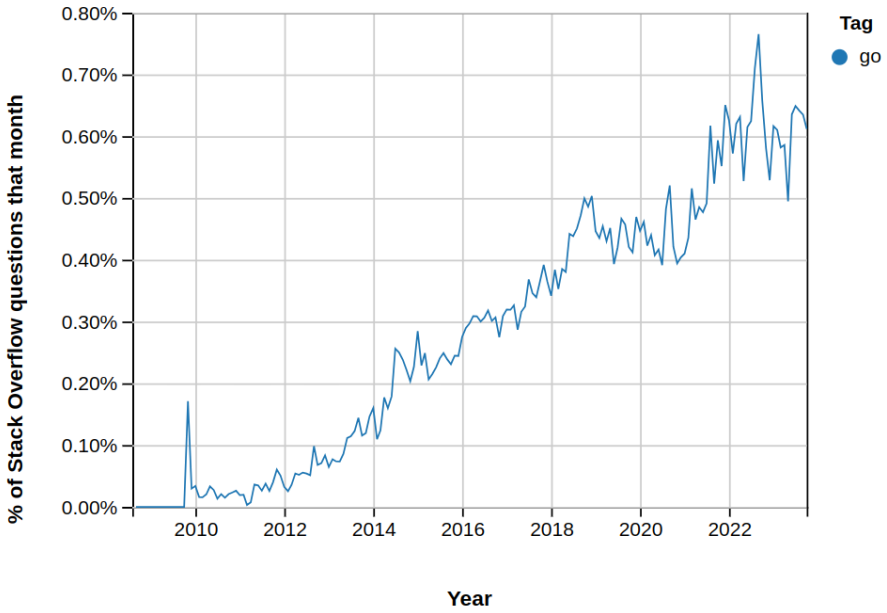

Figure 2.4: Stack Overflow Trends Go (14/11/2023) (“Stack Overflow Trends”, n.d.-b)

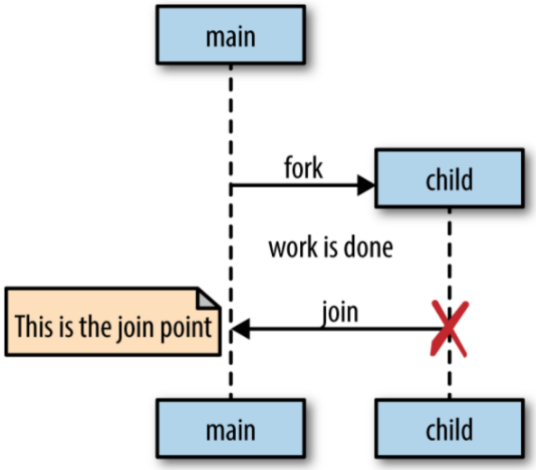

Figure 2.5: Fork-Join Model (Cox-Buday, 2017)

2.5 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 Table 2.1, p22).

		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	

Table 2.1: Table of current solutions and their included features

2.5.1 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 Table 2.1, p22). 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, p41). 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.

2.5.2 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, p42). 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.

2.5.3 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, p42). 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.

Figure 2.6: Features in priority order based on user feedback

2.6 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.

2.6.1 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. Furthermore, Hong states how investing in effective cycling infrastructure "mitigates the

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

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 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 2.7, p24) (see Figure 2.8, p25) (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.

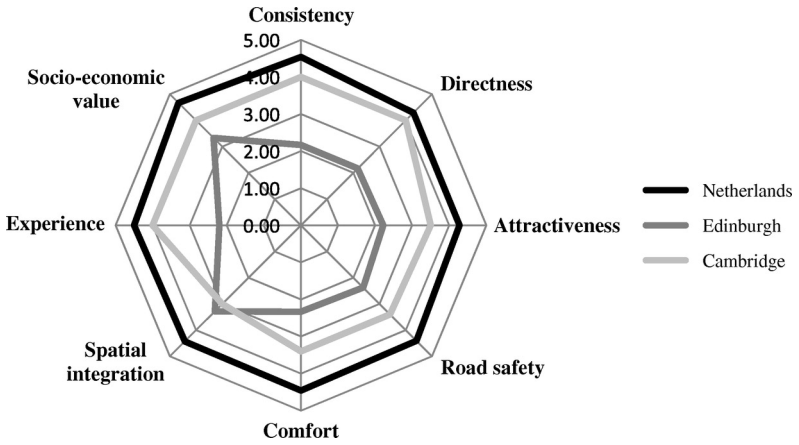

Figure 2.8: Spider web diagram comparing the Bicycle Infrastructure Scores (Hull and O'Holleran, 2014).

2.6.2 Weather Conditions

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 2.6.1, p23), 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).

2.7 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 (“urrent weather and forecast - OpenWeatherMap”, n.d.) 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, as mentioned in Section 4.

3.1 Chosen Methodology - Incremental

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

Requirements

4.1 Introduction

4.2 Requirements Gathering

4.3 Prioritising Requirements

4.4 Identifying Users

4.5 User Stories

Table 4.1: 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 4.2: 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 4.3: 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 4.4: 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 4.5: 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 4.6: 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 4.7: 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 dropdown 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

Chapter 5

Design

5.1 Architecture Design

5.1.1 Client-Server Architecture?

5.1.2 Alternative Architecture Patterns

5.1.3 Architecture Diagram

5.2 System Design

5.2.1 REST API

5.2.2 Open Weather Map API

5.2.3 OSRM (Open Source Routing Machine)

5.2.4 Google Distance Matrix API

5.2.5 Use Case Diagrams

5.2.6 Hierarchical Task Analysis (HTA)

5.3 User Interface Design

5.3.1 Low Fidelity Prototype

5.3.2 High Fidelity Prototype

5.4 Database Design

5.4.1 Users

Chapter 6

Implementation

6.1 Development Environment

6.2 Programming Languages

6.3 Database

6.4 Increments

6.4.1 Increment 1

6.4.2 Increment 2

6.4.3 Increment 3

6.5 Documentation and Linting

6.6 Challenges and limitations

Chapter 7

Testing

7.1 Unit Testing

7.1.1 Testing here

7.2 Postman

7.3 Chrome Developer Tools

Chapter 8

Evaluation

Chapter 9

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

Cox, R., Griesemer, R., Pike, R., Taylor, I. L., & Thompson, K. (2022). The go programming language and environment [Publisher: Association for Computing Machinery]. *Communications of the ACM*, *65*(5), 70–78. https://doi.org/10.1145/3488716

Cox-Buday, K. (2017, July 19). *Concurrency in go: Tools and techniques for developers* [Google-Books-ID: R3MtDwAAQBAJ]. "O'Reilly Media, Inc."

Crnkovic, I., & Larsson, M. (2002). Challenges of component-based development. *Journal of Systems and Software*, *61*(3), 201–212. https://doi.org/10.1016/S0164-1212(01)00148-0

Curry, E., & Grace, P. (2008). Flexible self-management using the model-view-controller pattern. *IEEE Software*, *25*(3), 84–90. https://doi.org/10.1109/MS.2008.60

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

Dinku, Z. (2022). React.js vs. next.js.

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

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

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

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

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/

Levlin, M. (2020). *DOM benchmark comparison of the front-end JavaScript frameworks react, angular, vue, and svelte* [Accepted: 2020-05-28T06:40:59Z Publisher: Åbo Akademi]. Retrieved November 15, 2023, from https://www.doria.fi/handle/10024/177433

Li, N., & Zhang, B. (2021). The research on single page application front-end development based on vue [Publisher: IOP Publishing]. *Journal of Physics: Conference Series*, *1883*(1), 012030. https://doi.org/10.1088/1742-6596/1883/1/012030

Mariano, C. L. (n.d.). Benchmarking JavaScript frameworks.

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

Meyerson, J. (2014). The go programming language [Publisher: IEEE]. *IEEE Software, Software, IEEE, IEEE Softw.*, *31*(5), 104–104. https://doi.org/10.1109/MS.2014.127

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/

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

Stack overflow trends. (n.d.-a). Retrieved November 14, 2023, from https://insights.stackoverflow.com/trends?tags=reactjs%2Cangular%2Cvue.js%2Cjquery

Stack overflow trends. (n.d.-b). Retrieved November 15, 2023, from https://insights.stackoverflow.com/trends?tags=go

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

Xu, W. (n.d.). Benchmark comparison of JavaScript frameworks react, vue, angular and svelte.

Zarifis, K. (n.d.). In-depth survey of MVVM web application frameworks.

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

urrent weather and forecast - OpenWeatherMap. (n.d.). Retrieved November 16, 2023, from https://openweathermap.org/

Appendix A

Appendix

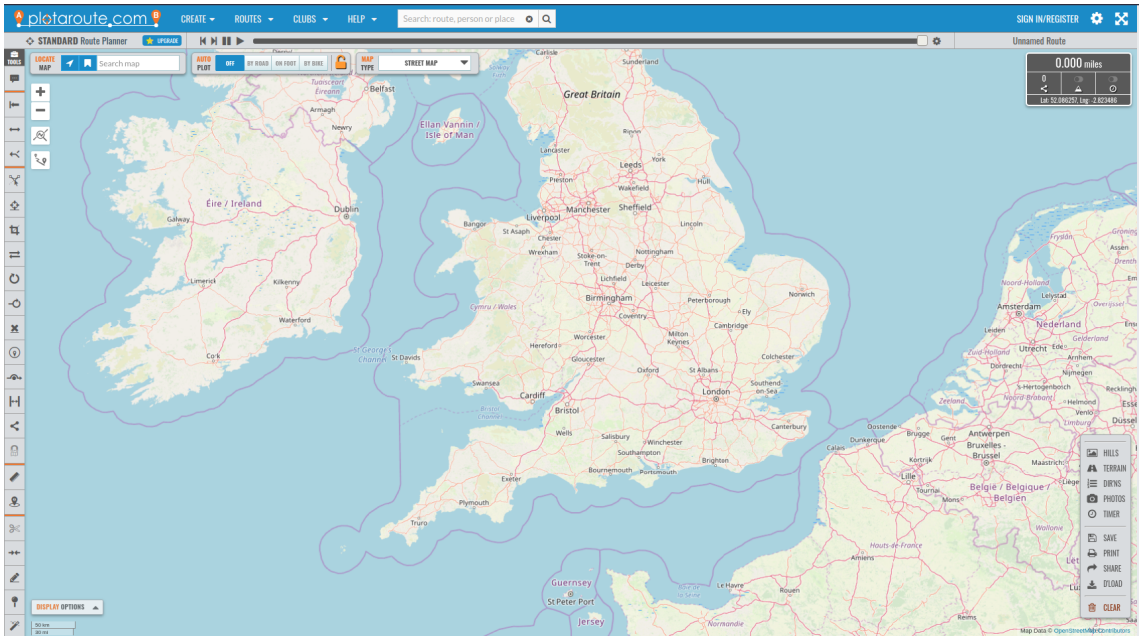

Figure A.1: Plotaroute.com UI

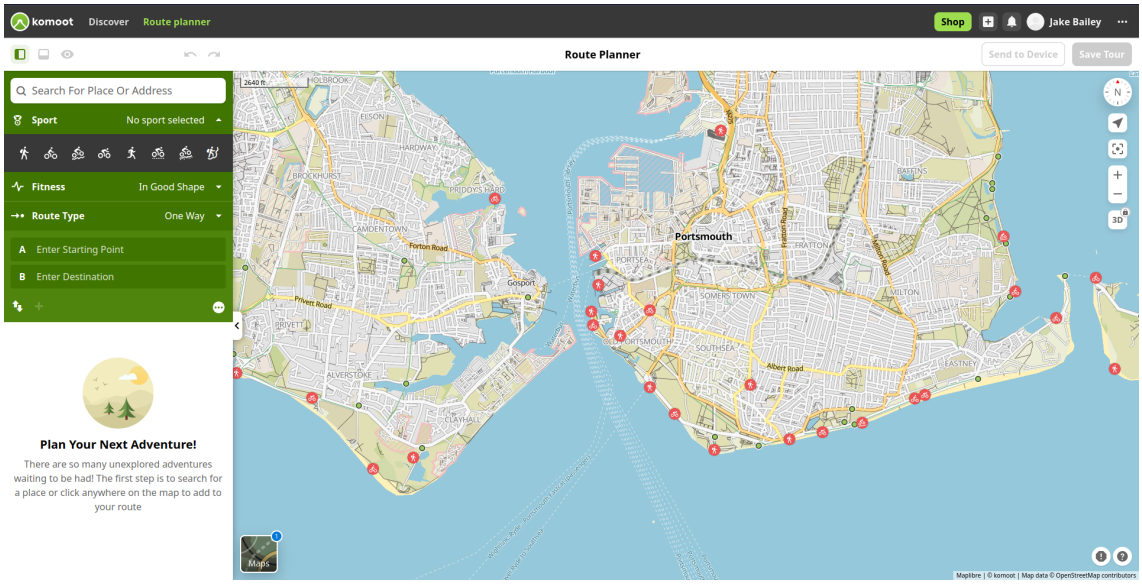

Figure A.2: Komoot UI

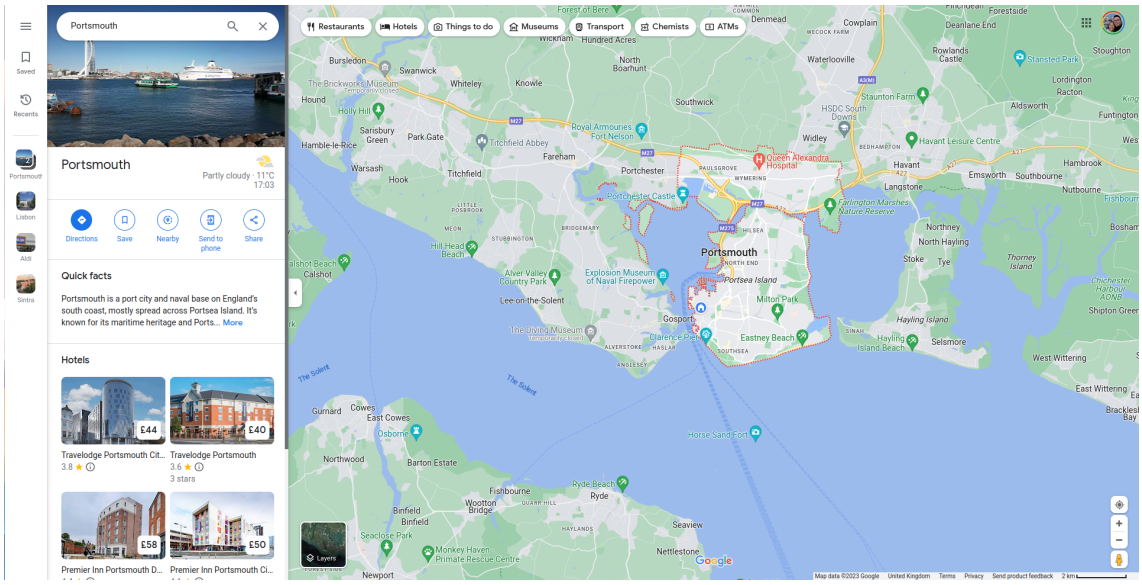

Figure A.3: Google Maps UI