Sheffield Hallam University

Faculty of Arts, Computing, Engineering & Sciences

DEPARTMENT OF COMPUTING

55-604708 Project (Technical Computing)

2019/20

Author: Gergo Kekesi

Student ID:: b6017384

Year Submitted: 2020

Supervisor: Dr. Michael Meredith

Second Marker: Mr. Hsi-Ming Ho

Degree Course: Computer Science

Title of Project: ShuWays – An indoor navigation solution for campus to aid visitors and students in finding their way around the university.

Confidentiality Required?: No

Acknowledgements

I would like to thank my wife for putting up with me during this project, for tirelessly giving me space and time to work and always encouraging me to do my best. Without her sacrifice, this project would not be what it is today. Secondly, I would like to thank my supervisor Michael Meredith for his continued support and his challenging questions that pushed me to achieve the best I can.

Abstract

Sheffield Hallam University spreads across multiple buildings located near the city centre as well as around the Ecclesall area. For anyone new to the University, finding where to go can be a real challenge. On a typical open day, visitors and prospective students rely on printed maps (that are often not detailed enough) or student ambassadors to help them locate where they need to go to. New students may also find it difficult for the first couple of weeks to navigate the vast area of the campuses. This project aimed to develop a product that could help address the issue of way-finding by providing a user-friendly, easily accessible way to locate and navigate to various Points-Of-Interest (POIs) around the University as well as enable open day visitors to browse and find the events taking place.

This report discusses the research, design, development, and testing of ShuWays, a cross-platform mobile accessible application that allows users to search for POIs such as rooms, help desks, toilets, water fountains, and other useful locations, combining outdoor and indoor route planning. The application is deployed as a progressive web app (PWA) that enables access to the application without the need for it to be installed. As indoors is usually more difficult to navigate for limited mobility users than outdoors, the route planning functionality allows the planning of routes that will be easier for them to take (such as taking them to elevators instead of stairs). Furthermore, the application lists the events of an open day or other events which the user can browse and locate on the map in a user-friendly manner.

Contents

List of Figures

List of Abbreviations

API - Application Program Interface

GPS - Global Positioning System

IDE - Integrated Development Environment

POI - Point Of Interest

PWA - Progressive Web App

QR - Quick Response

SDK - Software Development Kit

SUS - System Usability Scale

UCD - User-Centred Design

UI - User Interface

1 Introduction

With advances in building architecture and the development of complex structures, the need for way-finding indoors has grown exponentially. A 2016 Global Research Report by Vanson Bourne research company (Vanson Bourne Ltd., 2016) demonstrated the "widespread and growing demand for scalable, accurate and low-cost indoor positioning systems. The survey polled business and marketing decision-makers, and other executives in charge of location-based apps in North America, UK, and Asia-Pacific markets". Furthermore, the indoor location market was valued at USD 9.58 billion in 2018 and is projected to reach USD 145.68 billion by 2026 (Verified Market Research, 2018).

The results of these two surveys provide a good basis for the development of ShuWays. People are using mobile phones more than ever (Figure 1.1) and companies are actively looking into developing indoor navigation solutions with an ever-increasing market share.

A further motivation for the development of ShuWays is personal experience. Even after studying four years at Sheffield Hallam, there were still times where help was required to locate facilities in buildings not frequently visited. The project specification and aims can be seen in Appendix C.

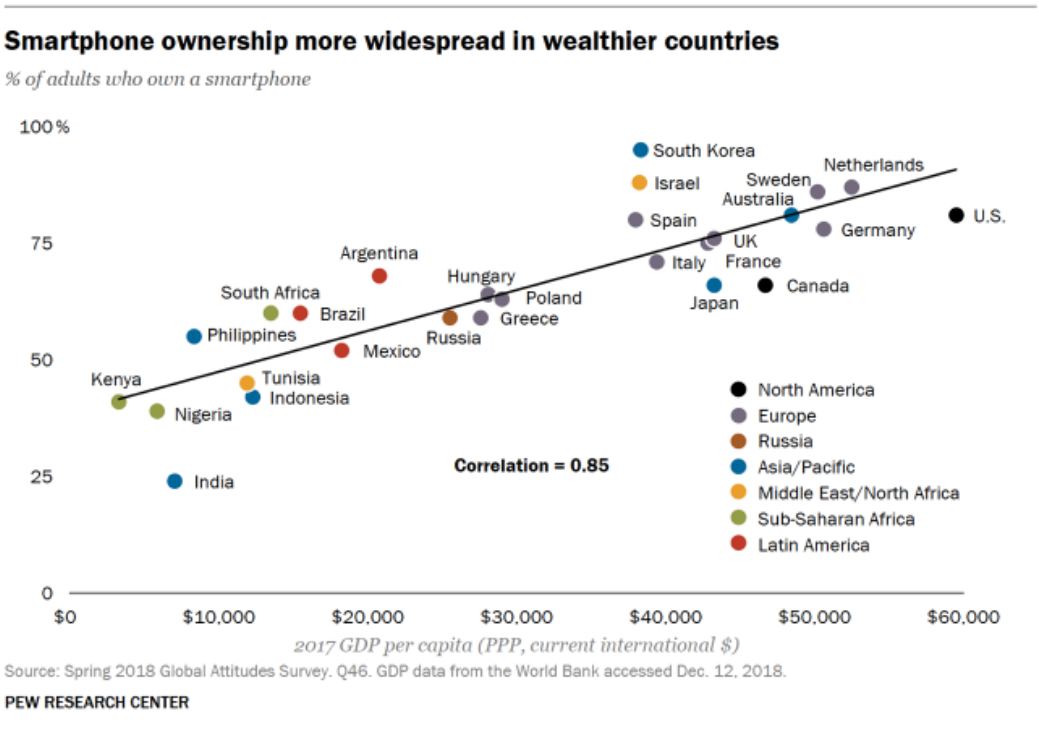


Figure 1.1: Smartphone ownership per country (Pew Research Center, 2018)

2 Literature Review

2.1 Indoor Navigation and Positioning

2.1.1 Navigating Complex Buildings

A persons ability to navigate indoors is affected by many different variables that must be considered to achieve success in the development of ShuWays. The project aims to deliver a mobile accessible map and the experiments conducted by Sarot and Delazari (2018) confirm that the "provision of information about the inside of buildings through maps helps the user who is navigating in the area". When navigating, people use landmarks and structural points of reference to orient themselves (Dalton, Hölscher, & Spiers, 2015). It is also important to display different levels of detail to not overwhelm users (Chen & Clarke, 2020).

2.1.2 Colour-blind users

Colour-blind users of applications are often underrepresented, especially when designing maps (Skupien, 2013). Skupien also mentions that signage can often be very confusing to colour-blind people if not well designed. To ensure that colour-blind people are well equipped to use ShuWays care must be taken when choosing colours, especially the colour of the displayed indoor map. Coblis (Wickline, 2018) is a tool that can display how people with different colour blindness would see an image.

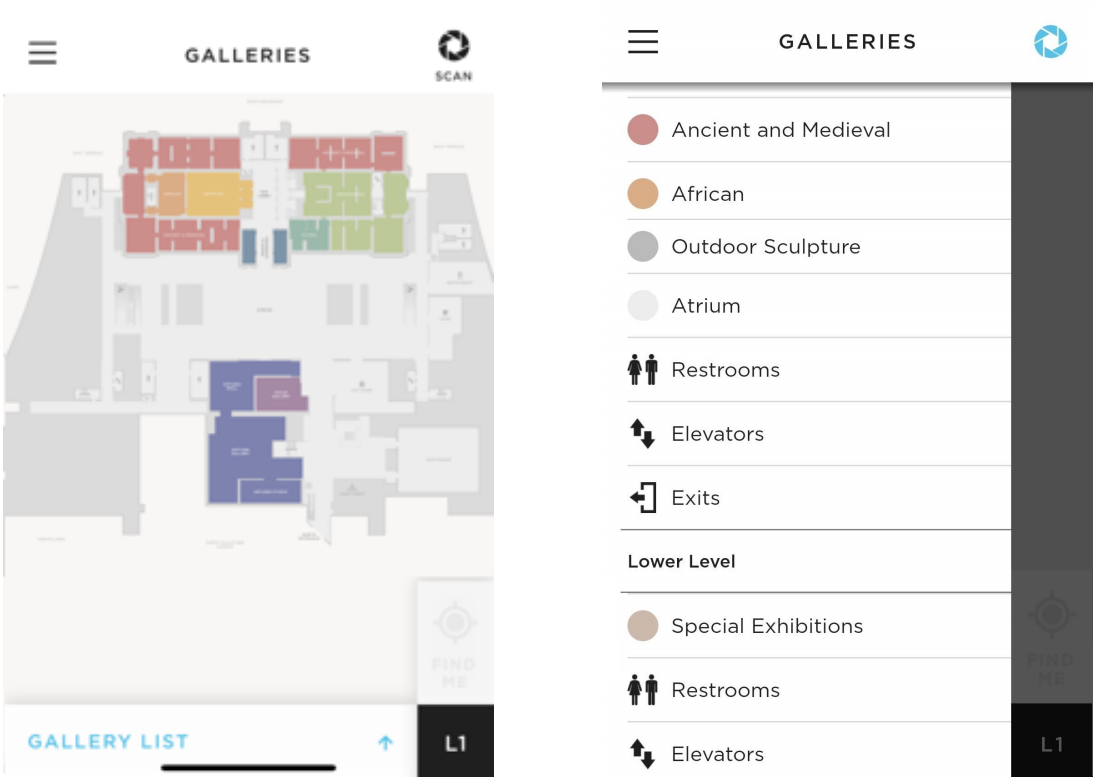
2.1.3 Inaccessible areas

Not all areas of a building are accessible. To show inaccessible areas, darker colours are often used (Midtbø, 2014). Gray, for example, is a common colour to use for disabled user interface (UI) elements (Google, n.d.-c) and would be suitable to indicate these areas.

2.2 Existing Solutions

2.2.1 ArtLens

ArtLens (*ArtLens App*, 2019) is an application developed for the Cleveland Museum of Art. It includes various features related to viewing art, but crucially it uses iBeacons (Apple, 2014) and Bluetooth technology to provide navigation to a specific artwork in the museum. The app also uses its indoor positioning capabilities to suggest nearby artwork to the user. It is available on both Android and iOS.



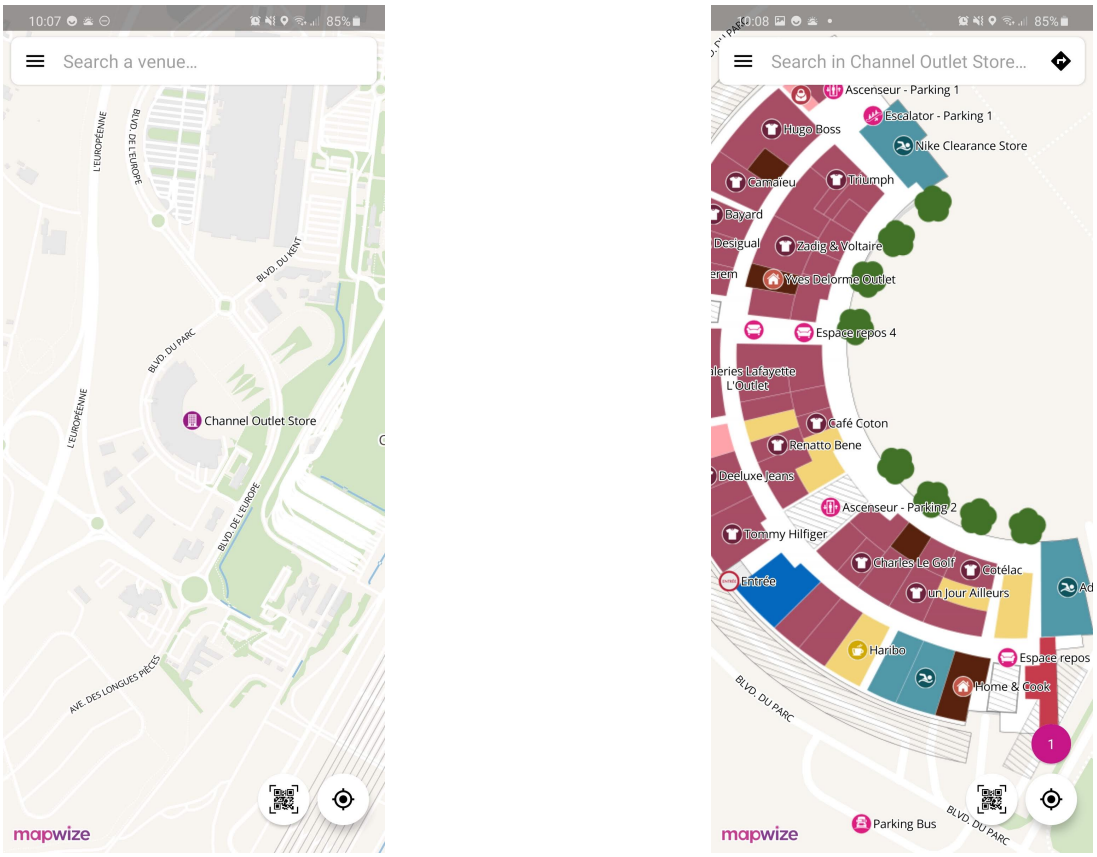
(a) Gallery view (b) Gallery list expanded

Figure 2.1: Artlens Application on Android

The downside of ArtLens is the inability to perform searches. To locate restrooms, for example, the user must first select Gallery from the main menu which loads the map, then select "Gallery List" at the bottom of the screen (Figure 2.1a) to bring up a list of galleries per floor as well as the facilities (Figure 2.1b). It is not immediately clear that toilets would be listed under "Galleries". The UI of the application is unfamiliar meaning users are required to learn it before being able to use it. There is a unique target audience, however, UI design patterns "provide consistency, familiarity and efficiency to users" (Nacheva, 2016) Creating a familiar UI will be vital for the project. ArtLens uses iBeacons to provide positioning. These must be distributed around a building first before they can be used. This is not going to be possible for this project due to large scale and cost.

2.2.2 Mapwize

Mapwize is a company providing indoor navigation services to purchase and a Software Development Kit (SDK) for those wanting to build on the platform. The UI design of the application is similar to Google Maps (Figure 3.3b). The application is available on Android and iOS, however, it must be installed to be used. It allows the user to get a route between POIs, to pan and zoom a map of a building, and to search for venues and POIs. Mapwize does not list any events in the application. Venues and buildings in the application are shown with a building marker and a label with the name, although they are rather small and hard to see (Figure 2.2a). Once the user zooms in close enough, the floor automatically loads and displays. This is nice but if a user accidentally pinches in then the whole screen will transition into the indoor view even though the user did not want that. While the indoor map is visible and displayed, it is possible to accidentally scroll the building out of view which results in the application thinking the user exit the building, transitioning UIs from Figure 2.2b to Figure 2.2a. This should be considered when developing ShuWays. Mapwize shows the location of the user on the map but does not have any functionality to navigate from where the user is. This results in the user having to first determine where they are and then locate a POI close to them on the map and use that as the starting point for the journey. To carry out this task a user will spend a considerable amount of time and the route won't be exactly from their location, increasing the chances of the user making an error. ShuWays will allow users to use their location when routing to increase the usability of the application, as per the requirements (Appendix C).



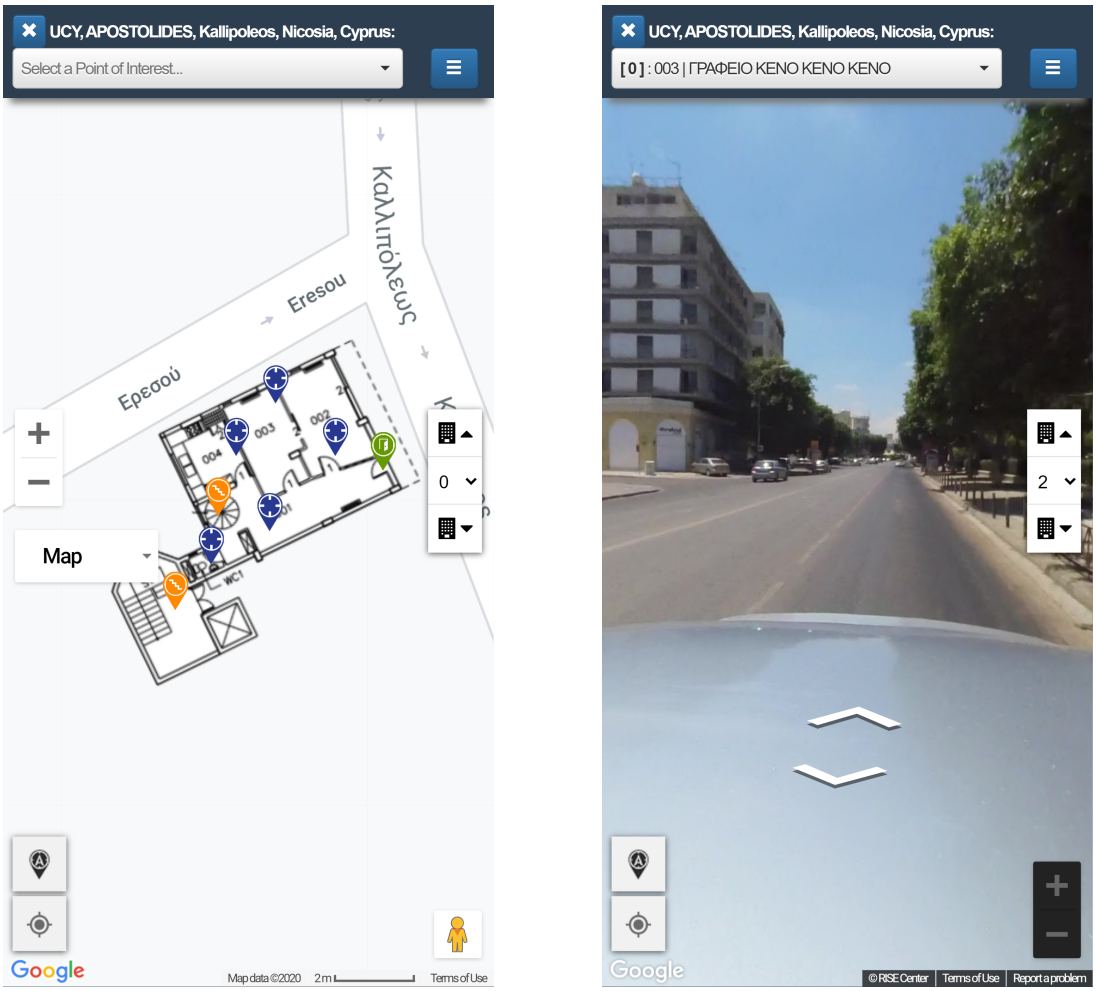
(a) Building marker in Mapwize

(b) Indoor view of a building

Figure 2.2: Mapwize application

2.2.3 Anyplace

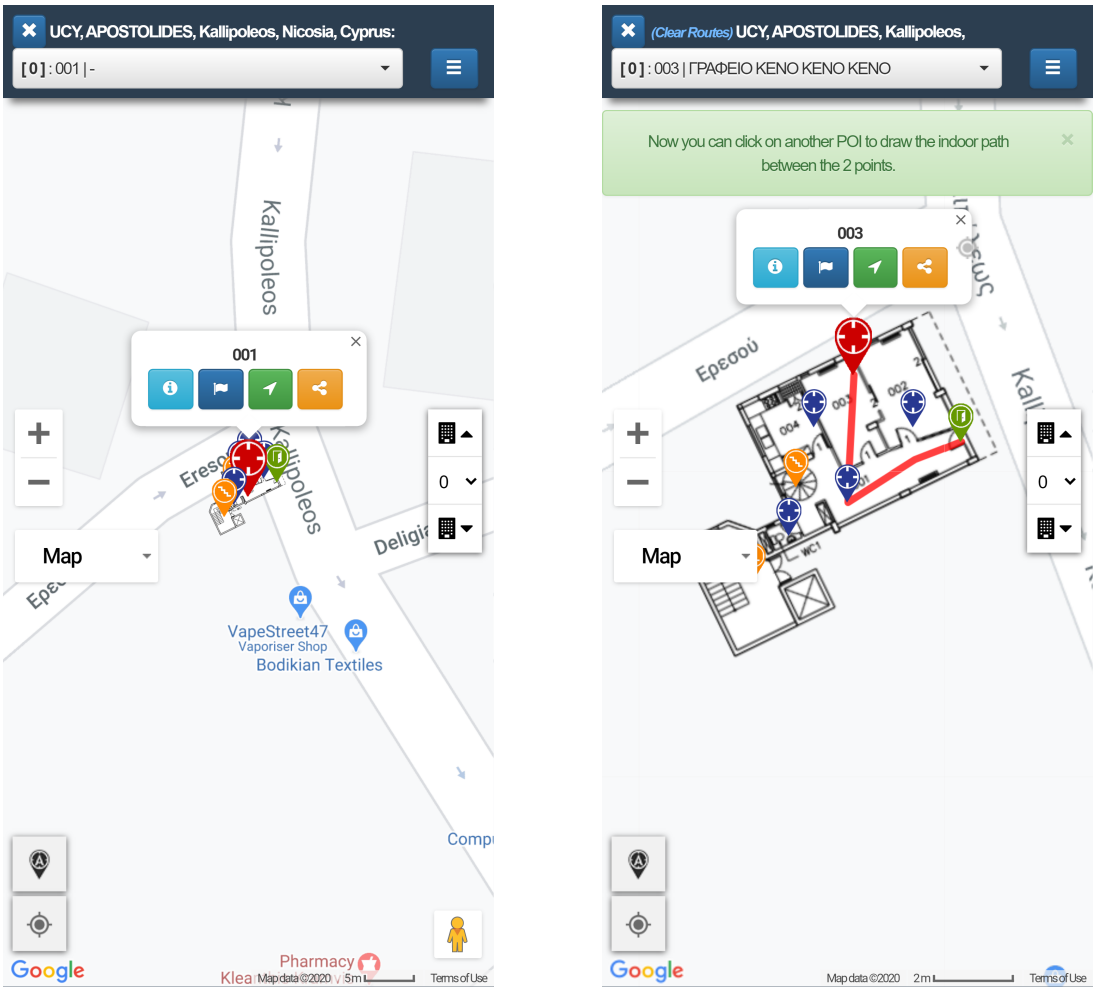
Anyplace is an indoor information service offering WiFi-based indoor localization, navigation, and search inside buildings using smartphones (Zeinalipour-Yazti & Laoudias, 2017). The stack includes multiple applications including the Anyplace Viewer (Figure 2.3a), which is a web-based front-end and an Android application. The web-based solution uses the Global Positioning System (GPS), while the Android-based one utilises WiFi positioning as well to position the user. The development team has focused a lot on providing an accurate indoor positioning solution. This is clear from the UI of the application. The controls on the screen are different sizes with some remaining visible even when they are not relevant. For example, the map type selection is visible when viewing an indoor map (Figure 3.2b) even though it is not relevant. Also, the street view pin from Google Maps is visible at all times, and dropping the pin loads the street view even though, the user should be browsing the indoors of a building according to the UI (Figure 2.3b). When selecting a POI, the map zooms too far out from the zoom level the user is on (Figure 2.4a), resulting in the user needing to zoom back in to see the map. To request a route, the user must manually select the origin POI then click on the destination one (Figure 2.4b). When displaying a route on the map, the navigation cannot be finished, and there is no change in the UI of the application to indicate the user is viewing a route.



(a) Anyplace Viewer showing an indoor floor

(b) Street-view UI

Figure 2.3: Anyplace Viewer Application - 1



(a) POI Info Window

(b) A route being displayed

Figure 2.4: Anyplace Viewer Application - 2

2.3 Software Development Methodologies

2.3.1 Waterfall

The waterfall model is a basic, linear software development model commonly consisting of five phases done one after another: Requirement Analysis and Specification, Design, Implementation and Unit Testing, Integration and System Testing, and the Operation and Maintenance phase (Kannan, Jhajharia, & Verma, 2014). The waterfall model provides a structure. The work that is being done is easier followed linearly. If the procedures are correctly followed, time and work estimation can also be done easily. Furthermore, the documentation written as part of the process helps customers know what to expect as well as enable better maintenance. On the other hand, changes aren't accommodated easily. The model also does not cope with risk and uncertainty. If there isn't a feedback loop, initial mistakes can't be improved. Finally. the requirements must be clearly defined from the start which in most projects is not possible.

2.3.2 Agile

Agile is an umbrella term covering many methodologies, including Scrum and Extreme Programming (Kannan et al., 2014). In agile, the work is split into iterations (or sprints). Each iteration has a defined duration with a running list of deliverables, planned at the start. When the work is completed, if the software is ready, it is released and feedback from the customers is taken into account for consecutive iterations. Agile is built around the idea that requirements evolve and change, empowering a company to constantly enhance a product. Agile delivers working software in the early phases and responds to change as opposed to following a strict plan (Langr, 2011). However, completing work on time and to a good quality becomes more difficult with agile due to the iterative and evolving nature. It is also less predictable than waterfall. (Kannan et al., 2014)

2.3.3 User-Centred Design

User-Centred Design (UCD) is an iterative design process where the focus is on the end-users and the tasks they want to perform (Abras, Maloney-Krichmar, Preece, et al., 2004). In UCD, research is conducted upfront as opposed to the agile way of designing and developing at the same time. It makes use of various design tools including personas, scenarios, wireframes, and others (Garrett, 2011). Prototypes are created and user testing is conducted. The feedback from the testing is then used to refine the designs and iterate. UCD is often combined with an agile development strategy to achieve a product that meets the needs of the end-users with opportunities to expand and enhance features (Da Silva, Martin, Maurer, & Silveira, 2011). ShuWays will be developed in such a manner as the end-user is at the centre of a mapping application and the project involves prototyping by nature.

2.4 Design Tools

2.4.1 Wireframing

Wireframe creation can happen in many different ways. The traditional "quick and dirty" way of creating wireframes is by hand drawing them (Hamm, 2014). These sketches are then often transferred into higher fidelity wireframes draw using software. One of these softwares is Moqups (S.C Evercoder Software S.R.L., 2020). It allows the creation of wireframes with various built-in UI elements which is an advantage as it speeds up the workflow. On the downside, the free plan only allows up to 200 stencils and a 5MB limit for the file which is not enough to cover the designs of ShuWays. Mockflow (MockFlow, 2020) on the other hand has a much more generous free tier and also contains many free UI elements. Storage is not limited and therefore it will be sufficient to use for this project.

2.4.2 Floor Plan Creation

As the architectural plan supplied by the Estates Department at Sheffield Hallam University (Appendix G) is not designed for public wayfinding, a trace of it will need to be created, omitting unnecessary information for users and colouring the map with a clear colour scheme that helps distinguish between locations. Gimp (The GIMP team, 2020) is an image manipulation software that is free and comprehensive. It enables layered editing of images which would allow tracing of the floor plan, however, it is complex to use without experience. Inkscape (Inkscape, 2020) on the other hand is an image manipulation software with a focus on creating vector images. It includes simple tools such as the path tool that allows drawing of lines and filling them with colour. Just like Gimp, it also allows multiple layers. Due to Inkscapes specific feature set and its ease of use, it would be a good choice to draw the floor plans.

2.5 Development Technologies

2.5.1 Deployment platform

Native development refers to developing code for one specific platform (such as iOS or Android). Hybrid development, on the other hand, is a development method that allows developers to write code once, but run on multiple platforms (Pinto & Coutinho, 2018). This ensures a larger coverage for the software, reaching more users. To enable university visitors with all kinds of devices to use the application, a hybrid approach is suited for ShuWays. A limitation of the hybrid approach, however, is the lack of access to hardware. (Pinto & Coutinho, 2018). If compiled for a native platform, plugins could be used to access some hardware, but when running in the browser access to these is limited. Using a hybrid development platform, ShuWays could be accessible in the browser and when installed, native plugins could provide access to the required WiFi hardware. PWAs are a flavour of hybrid apps that run in the browser and can be added to the home screen to behave and look like a native application (Sheppard, 2017). Furthermore, Fransson and Driaguine (2017) concluded that there aren't any significant limitations of PWAs (other than those mentioned above).

2.5.2 Application and UI Frameworks

There are various technologies available to develop hybrid applications.

- **React** (Facebook, 2020a) is a UI library that allows the development of reusable components in JavaScript. The components contain all the logic and functionality for them to work (Mardan, 2017). An advantage of using React is the ability to develop UI faster and with less code. It also uses something called the Virtual DOM which is used to "find the difference between what's already in the browser and what's new" - Mardan. This results in only a partial refresh of the page as opposed to it all refreshing. However, React is not a framework that solves it all. Third-party libraries are needed to have a complete application.

- **Angular** is a TypeScript based web application framework that, similarly to React, also uses components to build web applications (Seshadri, 2018). Unlike React, Angular does not use the Virtual DOM but instead refreshes the whole page when rendering changes (Wohlgethan, 2018). This results in a slightly slower performance compared to React. Angular is a very comprehensive framework with a steep learning curve while React is easier to pick up. However, Angular uses the MVC architecture which creates a clearer separation of code logic as opposed to storing all the logic in the component code like in React.

- Finally, **Ionic** (Ionic, 2020b) is an open-source hybrid app development SDK. Ionic adds many UI elements and features on top of existing frameworks. It is available with Angular and it is also available with React. As mentioned above, React requires the use of third party libraries as it is only a UI framework. Ionic adds these libraries by default and implements their own to allow complete app development out-of-the-box. The Ionic CLI (Ionic, 2020a) allows developers to create templates and set up an Ionic application with a skeleton structure, speeding up development. There are no major downsides to using Ionic apart from needing to learn the components it provides. When used with React, it is effectively a bundle of necessary libraries to make PWAs work.

In conclusion, using Ionic with React will be a good choice for development due to the available UI elements and the quicker performance over Angular. React is also a lot more popular than Angular (Potter, 2020), resulting in a lot more support available should it be required.

2.5.3 Development Environment

The development environment choice is very often personal preference but there are some points to consider when choosing an Integrated Development Environment (IDE). Atom (Calkins, n.d.) is an IDE developed by Github. The functionality of it can be extended by many packages available to download, including support for beautifying text, rendering web pages and running React scripts. However, it does not come complete out-of-the-box. Packages must be installed to configure it for web app development. On the contrary, Visual Studio Code (Sole, 2019) comes ready for development with trademarked Intellisens code completion and debugging tools. To enable the project to get up and running quickly, Visual Studio Code will be used as the choice of IDE.

2.6 Mapping

There are various SDKs out there that can be leveraged to add indoor mapping functionality to the project. These are discussed below.

2.6.1 Indoor Atlas

Indoor Atlas (IndoorAtlas, 2020) provides a framework for doing indoor positioning and mapping. Their solution is limited to iOS and Android and is not accessible on the web. Their solution provides a hybrid indoor positioning method by merging multiple sources of information such as geomagnetic and WiFi fingerprints. They provide a mapping component but also allow the use of 3rd party maps such as Google Maps. Documentation, however, is very lacking which would slow down the development of the project. Furthermore, the SDK does not support limited mobility routing which is a requirement of this project (Appendix C).

2.6.2 Mapwize SDK

Mapwize offers a whole suite of tools for indoor mapping, including their map creator software and SDK for Android and iOS but not for JavaScript (Mapwize, 2018). Their free tier is also very restrictive with only 30 POIs allowed. This makes their solution unsuitable for ShuWays.

2.6.3 Leaflet

Leaflet (Crickard, 2014) is an open-source JavaScript library for mobile-friendly interactive maps. The issue is that Leaflet does not have built-in routing capabilities, therefore to render routes, one would have to manually create the symbols to draw on the map. The library also leaves all of the calculations to be done on the client-side as it is not a service or an Application Programming Interface (API). This can affect the performance of the application. Due to its limited feature set, this library would not be able to achieve all the requirements of the project.

2.6.4 Google Maps API

Google provides indoor mapping but only as a service, not as a development tool (Google, 2020a). However, they do provide a mapping SDK for Android, iOS, JavaScript and more. They have very good documentation and extensive support on all platforms. The JavaScript API has been built upon and extended to support many frameworks including wrappers for React and Angular. The API includes outdoor directions and street-view capabilities. The downside of the SDK is that on the web, it does not allow rotation of the map but on other platforms it does. The Google Maps API has much better support and documentation compared to Leaflet and it also allows developers to calculate directions outdoors and integrate street-view in the app. Leaflet does not have these capabilities and as these are requirements of the project (Appendix C), a React wrapper around the Google Maps API will be used (Klar & Lyakhov, n.d.).

2.6.5 Anyplace

Anyplace (Department of Computer Science, 2018) provides a RESTful API (Anyplace, n.d.) which can be used to enable indoor mapping and positioning functionality in any application consuming that API. The service is completely free to use with no limitations. Using the Anyplace Architect, it is possible to add an indoor floor and populate it with POIs. These POIs can then be connected using connectors and edges to allow routing. Using their API all of this data can then be queried and used in an application. On top of this, Anyplace is open-source with the source code available on Github (Anyplace, 2020). This is a major advantage as the Anyplace service cannot plan routes for limited mobility users but due to the open-source code, this can be developed and added. Using Anyplace will allow ShuWays to have less reliance on public services and give the freedom to add required functionality where needed. The Google Maps API can be used on top of that to handle the map rendering and other functionalities described above.

2.7 Backend

As the project will use the Anyplace service, the only consideration for the backend needs to be the storage of events. Even though the Anyplace service is open source and it could be modified to store the events, their solution was not designed for this. AWS is a collection of web services offered by Amazon. They offer a NoSQL service called DynamoDB (Amazon Web Services, Inc, 2014). It can be accessed via a RESTful API. On the other hand, Firebase offers many useful tools in the form of an SDK for JavaScript which means tighter integration for ShuWays, reducing the need for constructing REST calls and instead allowing developers to make queries in code (Google, n.d.-a). Both are NoSQL databases with mostly the same functionality but Firebase will be used to store the events due to its ease of integration with JavaScript.

2.8 Project Management

2.8.1 Work Management

Pivotal Tracker is a project-planning tool designed to help manage and estimate work to be done. Trello is another tool designed for this purpose. Both list user stories (or tasks) in a side-by-side view allowing users to see the progression of items through various stages of the development lifecycle. However, unlike Trello, Pivotal Tracker allows the use of "Epics" which organise related stories, reports on the performance of the team and allows the estimation of work (Pivotal Tracker, n.d.). Based on this, Pivotal Tracker is chosen to keep track of the work.

2.8.2 Version Control

Two major players exist in this area: Github and Bitbucket. Both support the use of Git, however, Bitbucket is better integrated with Mercurial (UpGuard, 2020). They both offer similar features, including unlimited private repositories. Bitbucket integrates better with JIRA while Pivotal Tracker offers Github integration. Due to having previous experience with Github and having chosen to use Pivotal Tracker for project management, Github will be used for version control.

3 Application Design and Architecture

The following section describes the design of the application with consideration of the potential users and their use cases described in Appendix A and Appendix B.

3.1 Process

Before proceeding with development, personas, and scenarios (Appendix A and B) were created that describe potential users of the application. Wireframes of the UI were designed with the personas in mind. Design patterns and reference designs were also used. Usability testing was performed to gather feedback from users on the application (Section 5), but this was only done once due to poor time management (discussed in Section 6).

3.2 Platform

One of the requirements of the project is to be mobile accessible, as well as to research the suitability of a PWA design. Based on the research conducted in Section 2.5, ShuWays is deployed as a PWA. This results in more potential users as the application lives in the cloud as opposed to needing to be natively installed.

3.3 Main menu (Landing page)

When the application is loaded the user is presented with the main menu (Figure 3.1). The application is designed to be a Single Page App (SPA). These take a bit longer to load initially, but once they are loaded they are much faster as all the resources are available (Hakim, 2016). The app will also be visually complete faster. The design of the menu is inspired by the well known Garmin systems layout (Griffin, 2011). Many of Garmin's systems show two options upon turning on the device, "Where to?" and "View Map". To help users be more familiar with the application, especially people like Matthew (Appendix B), the menu is designed with a similar layout, except with the "Where to?" replaced by POIs and Events.

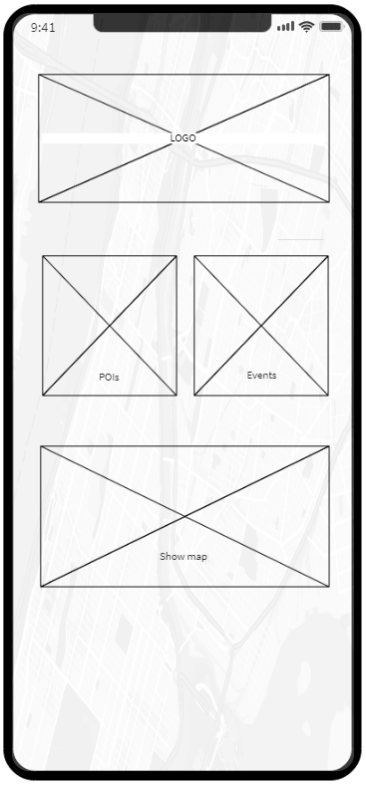


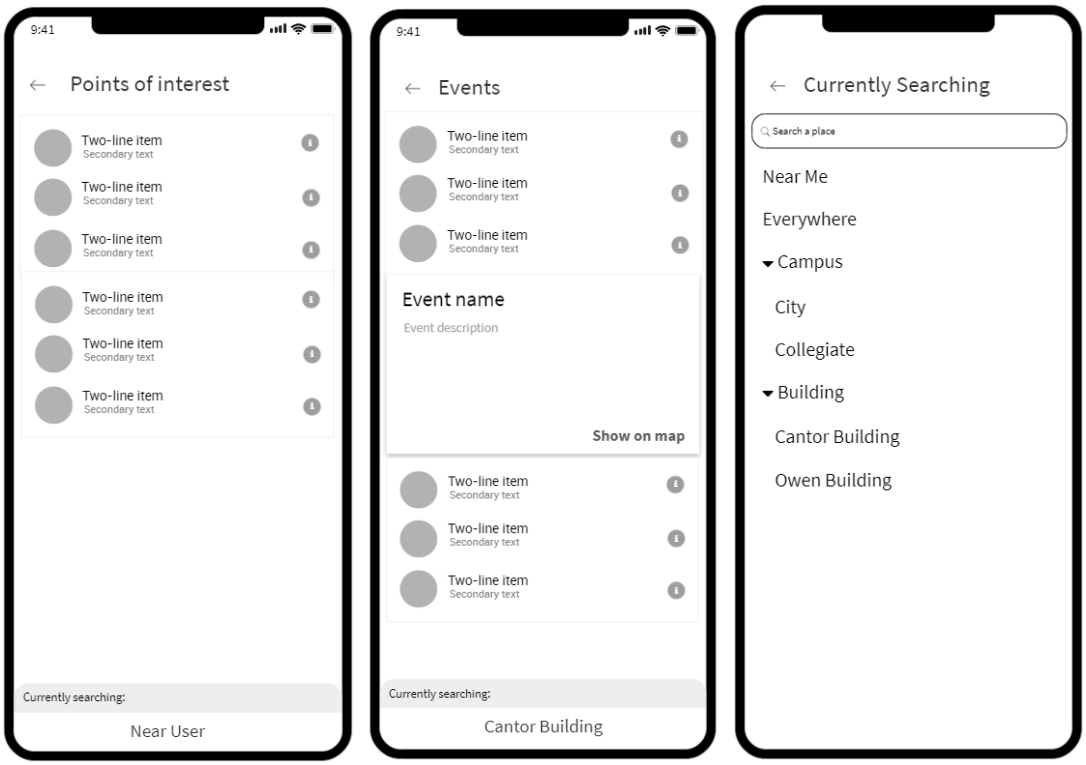
Figure 3.1: Wireframe of the landing page

3.4 Colour theme

The colour theme of the application's interface is important as different colours convey different messages (Stone, Adams, & Morioka, 2008). For example, red is associated with passion, energy, and power. Stone et al. also write that blue is associated with knowledge, coolness, and intelligence. As the project aims to deliver indoor knowledge to the user, the interface of the application is a few different shades of blue.

3.5 POIs and Events page

The POIs page (Figure 3.2a) is designed to be similar to Garmin systems. The Garmin Nuvi 2360 for example, lists categories of POIs and shows where the user is searching (Griffin, 2011). This allows the user to narrow down their results and find what they want. The Events page (Figure 3.2b) is also designed to allow the user to select where they want to find an event, by using the "Currently searching" section. When clicked, this brings up a list of options to select where the user would like to limit the searching to (Figure 3.2c). There is also an option to view an event on the map which loads the map centered on where the event is located.

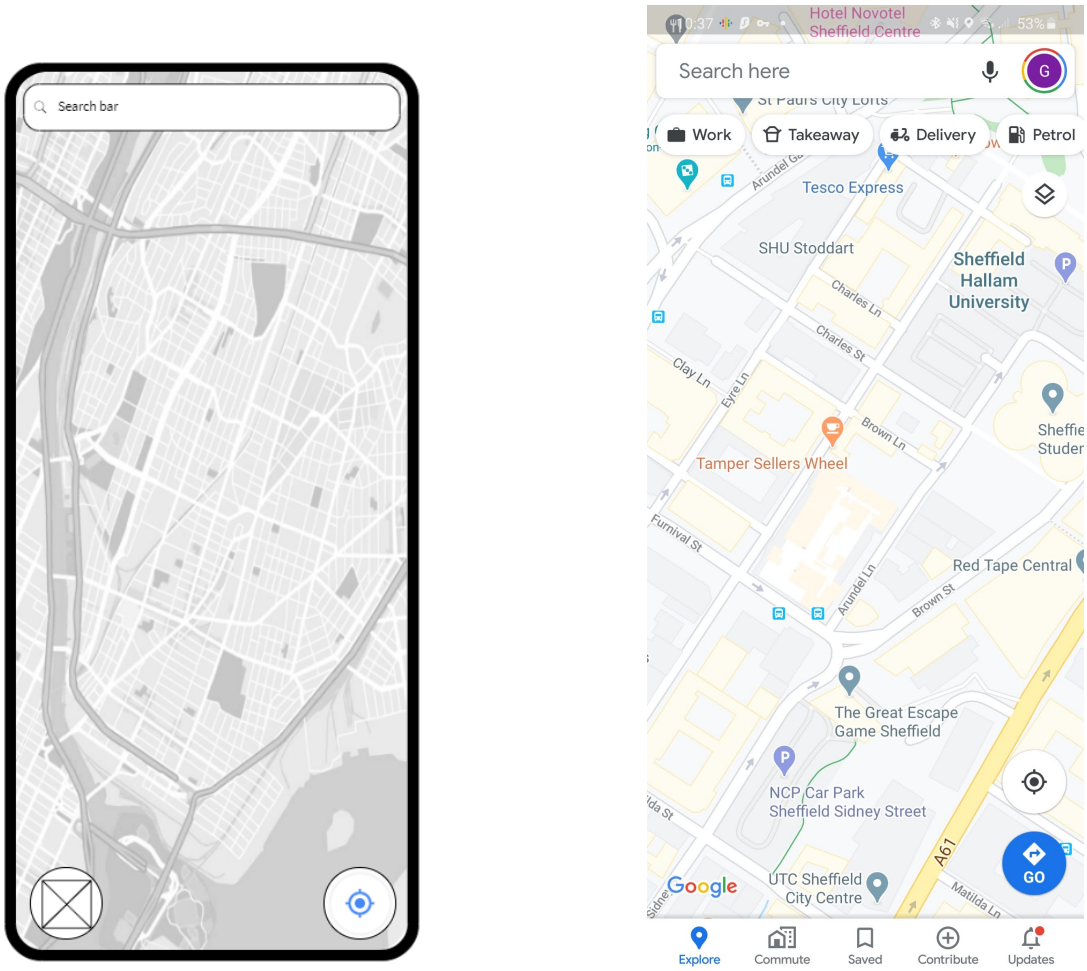


(a) Pois page (b) Events page (c) Currently searching

Figure 3.2: POIs and Events page including the Currently Searching option

3.6 Map view

ShuWays follows the design of Google Maps closely. The map view displays a search bar at the top of the screen, just like in Google Maps as well as styles and positions the GPS location button in the bottom right corner of the screen. (Figure 3.3a and Figure 3.3b). When the users select the "Show Map" item in the menu (Figure 3.1) they are taken to the map view. To allow the user to see the menu again, a menu button is displayed in the bottom left of the screen (Figure 3.3a). The map's boundaries have been restricted to Sheffield City Center to discourage the use of the application for routing from too far as that is not what the application was designed for.



(a) ShuWays

(b) Google Maps

Figure 3.3: Map view of ShuWays and Google Maps

3.7 Search

According to Gaigg (n.d.-b), a "user expects to receive the correct results regardless of what they type and search for." The search functionality in ShuWays lists both events and POIs and the location of the search bar is again influenced by Google Maps as seen in Figure 3.3b. As the user types the results list is expanded and relevant results for POIs and events are shown (Figure 3.4), meeting the aim of the project (Appendix C).

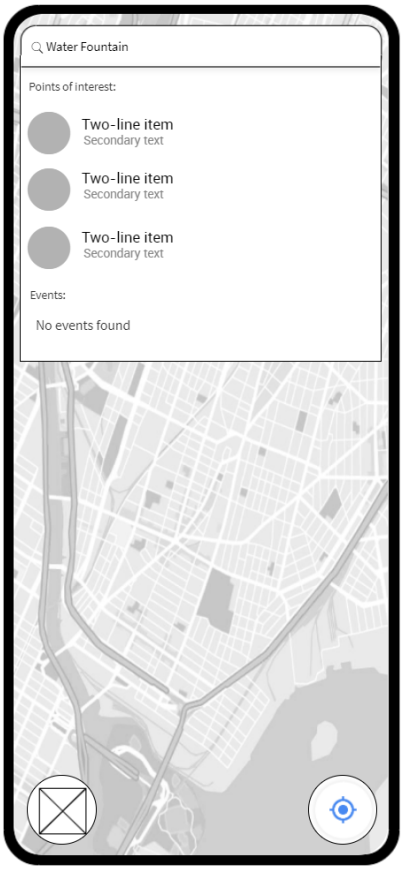


Figure 3.4: Wireframe of the search bar showing results

3.8 Entering a building

Mapwize (Section 2.2.2) loads the indoor map as soon as a user reaches a given zoom level, whereas Anyplace (Section 2.2.3) requires the user to click the building marker but then keeps the indoor map shown at every zoom level (it only hides POIs). ShuWays has been designed so that a user cannot zoom in beyond zoom level 18 (Google, n.d.-b) without entering a building first. This is done by selecting a building marker and clicking enter, as seen in Figure 3.5. If the user zooms out to zoom level 18 again, the application automatically closes the building and loads the building markers. This design allows a clear separation between the user looking at the surroundings of a building or looking indoors. The user has to choose to look indoors, unlike with Mapwize. Furthermore, while the user is viewing the indoor map panning is restricted to the bounds of the floor, preventing the user from accidentally scrolling the map out of view.

Figure 3.5: Wireframe of the info window pop-up when a building marker is selected

3.9 Indoor map

A major component of the project is the display of the indoor map. Different locations are designed to be colour coded, with different floors following the same colour scheme. "The purpose of color on a map is to help a map viewer decipher the symbols to make meaningful inferences" - (Peterson, 2009). The architectural floor plan (Appendix G) acquired was traced over using Inkscape (Section 2.4.2) to create the indoor maps for Cantor Floor 0, 1, and 2 (Figure 3.6). Grey is commonly used to show disabled UI elements (Google, n.d.-c); therefore the areas of the map inaccessible to the public have been coloured grey (Figure 3.6). During early prototyping the indoor map used a pink shade for the room colours (Appendix E.d), but to ensure good visibility for colour blind users such as Matthew (Appendix B). This was changed to the yellow seen in Figure 3.6. Appendix E shows how people with different colour blindness would see the indoor map (Wickline, 2018). It is evident that the finalised colour scheme is clearer to see for colour blind users, especially those with green colour blindness like Matthew (Appendix E.b compared to Appendix E.f)

Figure 3.6: Indoor map of Cantor Building Level 1 (91XX)

3.10 POI markers

The POIs displayed in ShuWays are represented using map markers. The map markers have been styled to be round, which is how Mapwize displays markers (Figure 2.2b), as opposed to having the upside-down water-drop style Google Maps uses (Figure 3.3b). For this project, the POIs (Figure 3.7) have been limited to the following:

- Stairs (Up, Down, and Bi-directional)
- Toilets (Female, Male, and Disabled)
- Food outlets
- Help desks
- Rooms
- Water fountains
- Showers
- Entrances
- Lifts

Figure 3.7: List of POI icons used in ShuWays

3.11 Floor selector

The floor selector component (Gaigg, n.d.-a) is displayed in the bottom right corner of the screen when the user is inside a building. It allows the user to go up and down floors one by one (Figure 3.8a) or select from the list of available floors (Figure 3.8b) as recommended by Gaigg. It also displays the current floor.

(a) Floor selector collapsed

(b) Floor selector expanded

Figure 3.8: Wireframes of the UI when displaying the indoor map

3.12 Zoom to reveal

Essential POIs (entrances, lifts, stairs, and help desks) are always shown when the indoor map is loaded, with all other POIs loaded as the user zooms beyond zoom level 19 as per the zoom-level design discussed by (Peterson, 2014).

3.13 Requesting directions

Both Google Maps and Mapwize give the user a list of options when they select a POI, so users are familiar with tapping something on the screen. To request directions in ShuWays a popup window is opened, the same way as when entering a building (Section 3.8). When the user taps on a POI the map centers on it and displays the window showing the name and location, as well as giving the user the option to get "Directions" to that POI. (Figure 3.9)

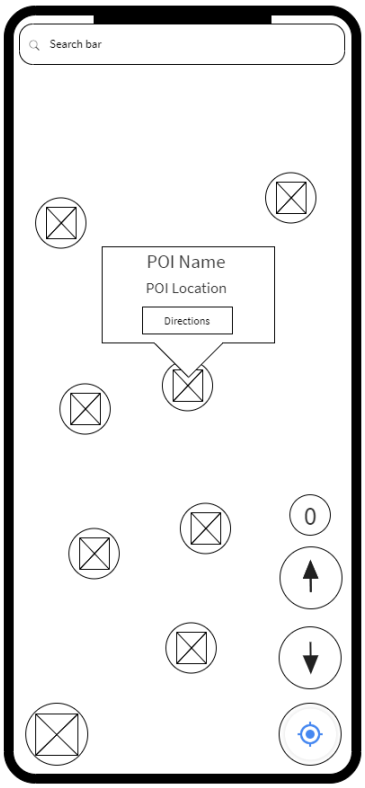
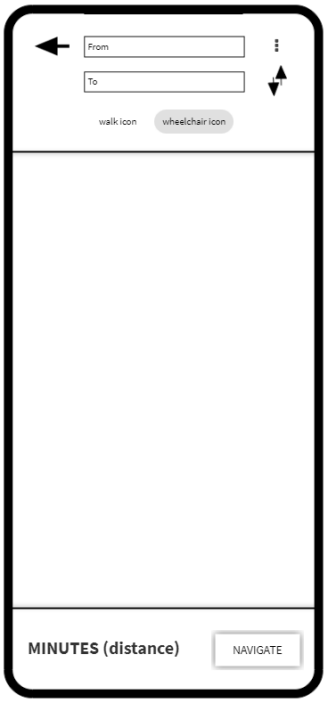


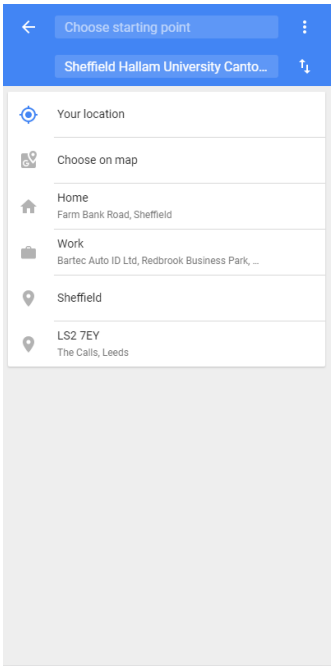
Figure 3.9: POI popup showing the name and the directions button

3.14 Route details view

When the user asks for directions to a POI the Route Details view (Figure 3.10a) is shown with the "To" section pre-populated, so they do not have to manually type it in. One of the requirements of the project is to allow for limited mobility users to find routes they can take (Appendix C). This requirement is met by giving the user route options. They can select the wheelchair or walk icon seen in Figure 3.10a similar to Mapwize (Section 2.2.2). Alternatively, they can select the three dots beside the "From" input field to set a routing preference like in Google Maps (Figure 3.10b). The user is then displayed the time it takes to arrive as well as the length of the route. The navigate button then initiates the drawing of the route on the map and transitions the view to the Navigation UI.



(a) ShuWays Route Details wireframe

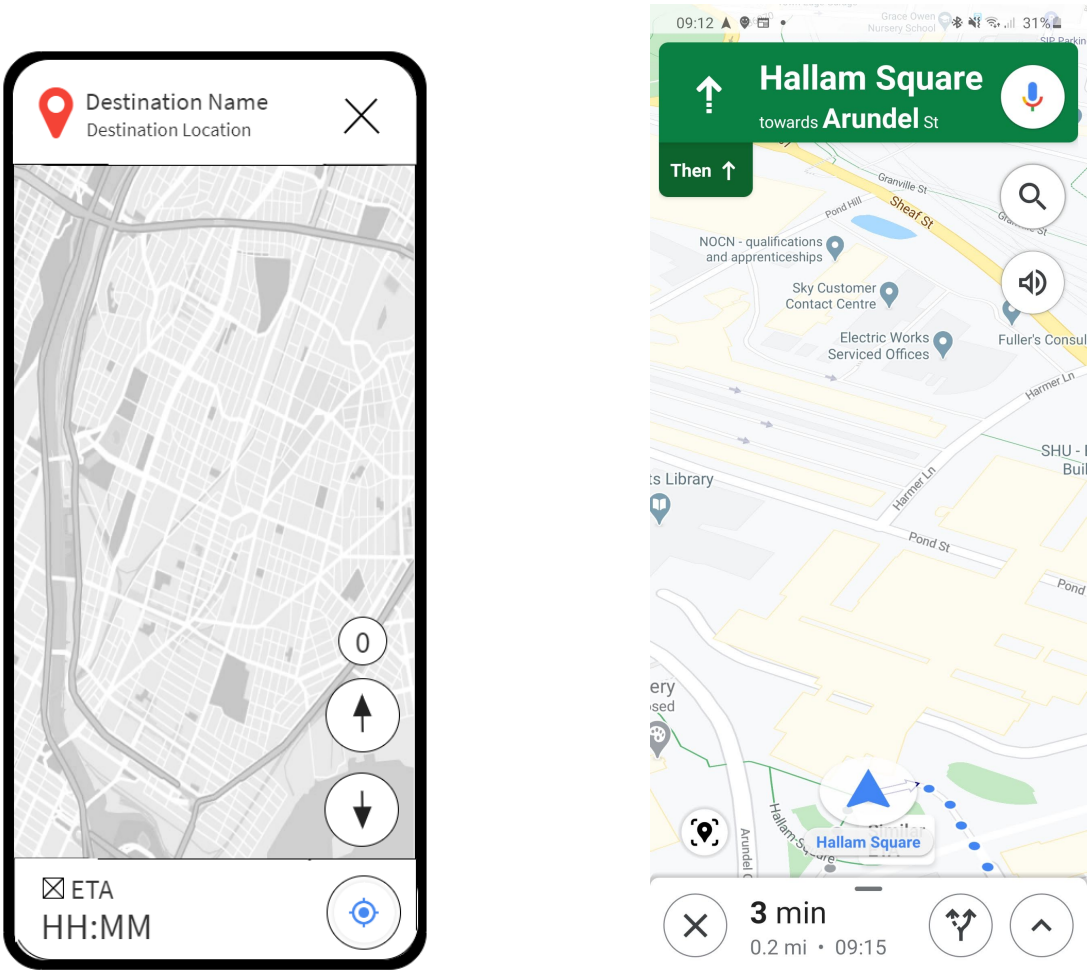


(b) Google Maps (Web) Route Details

Figure 3.10: Route details view of ShuWays and Google Maps (Web)

3.15 Navigation UI

If a user is happy with the origin and the destination on the Route Details view (Section 3.14) they can choose to get a route and navigate to the destination. This action will bring up the UI seen in Figure 3.11a, the design of which is based on Google Maps (Figure 3.11b). The floor selector component (discussed in Section 3.11) is visible only if the user has entered a building and there are floor changes in the route. When navigating outside it is hidden.



(a) ShuWays Navigation UI wireframe

(b) Google Maps (Android) Navigation UI

Figure 3.11: Navigation UI of ShuWays and Google Maps (Web)

3.16 Deep-linking to POIs

ShuWays has been designed so that external URLs can open the map centered on a specific POI. This allows the encoding of a URL in a Quick Response (QR) code, which in turn means that when a user scans the QR code, they can be taken to a specific location in the map. These QR codes could then be placed into open day guides or on posters.

3.17 Backend

The project uses the Anyplace (Section 2.2.3) backend to store the buildings, indoor maps, POIs, and calculate the routes between POIs. To achieve the requirement of finding routes for limited mobility users (Appendix C) the API call to get a route between two POIs has been extended to include a "preference" option (Figure 3.12), which can be set to lift, stairs or none.

Figure 3.12: The "/navigation/route" API call with the preference parameter added

The events are stored on Firestore (Section 2.7). The NoSQL structure used can be seen in Figure 3.13. This is scalable as to add a new event one simply needs to add a new entry. The structure also allows for easy query filtering such as asking for events in a given building using the built-in functions to query and order data (Google, 2020f).

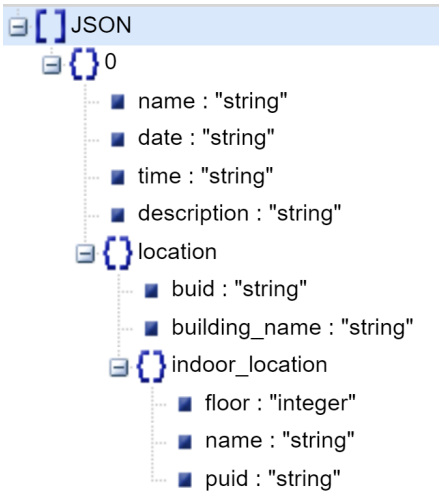


Figure 3.13: The structure of the events database

The Anyplace backend also has the functionality to do real-time positioning using WiFi, which can be leveraged by ShuWays when compiled for Android (Ionic, n.d.).

3.18 Hosting

To allow testing of the application by users, the application was hosted publicly using Firebase Hosting as Firebase integration was already decided to be added (Section 2.7). However, to access mapping functionality provided by the Anyplace backend over the web NGROK (Inconshreveable, n.d.) was used. NGROK allows developers to expose a local development webserver to the internet securely. The only drawback of the free package is that each time NGROK is restarted the web address changes, resulting in the need to re-deploy ShuWays.

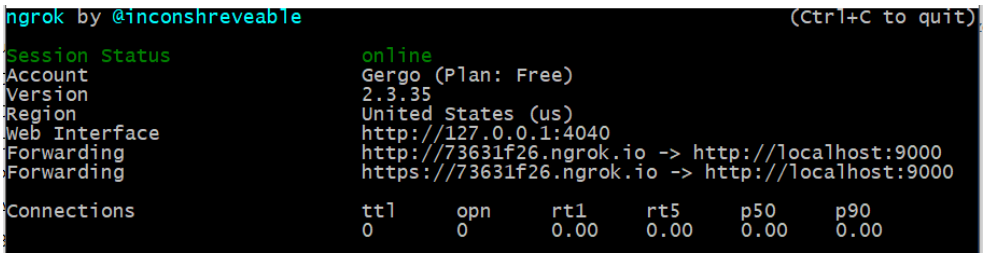


Figure 3.14: NGROK being used to expose the Anyplace backend to the internet

4 Development

4.1 Project Setup

Pivotal Tracker was set up and the work during development was managed there (Figure 4.1). A GitHub repository was used to store the application files and manage the versions of the software (Figure 4.2). Work would be regularly committed to the repository to ensure it was backed up and the progress of the work could be tracked.

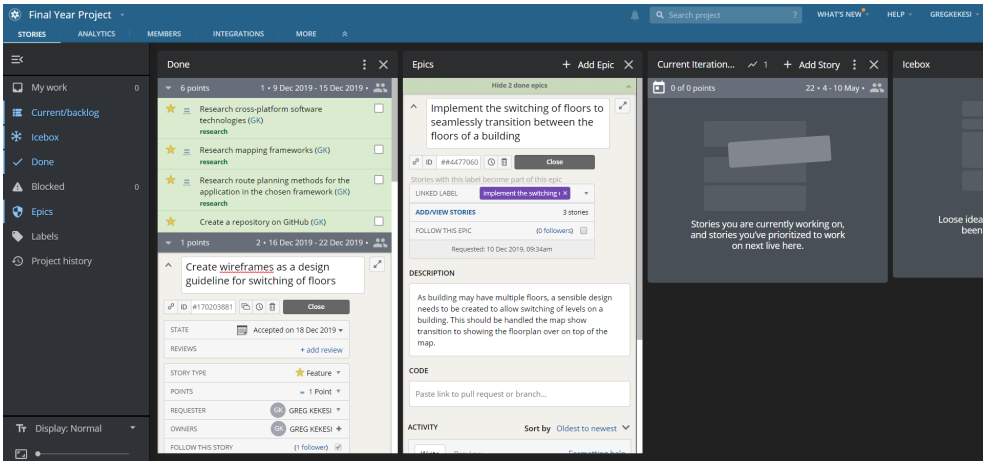


Figure 4.1: Pivotal Tracker page showing work items for the project

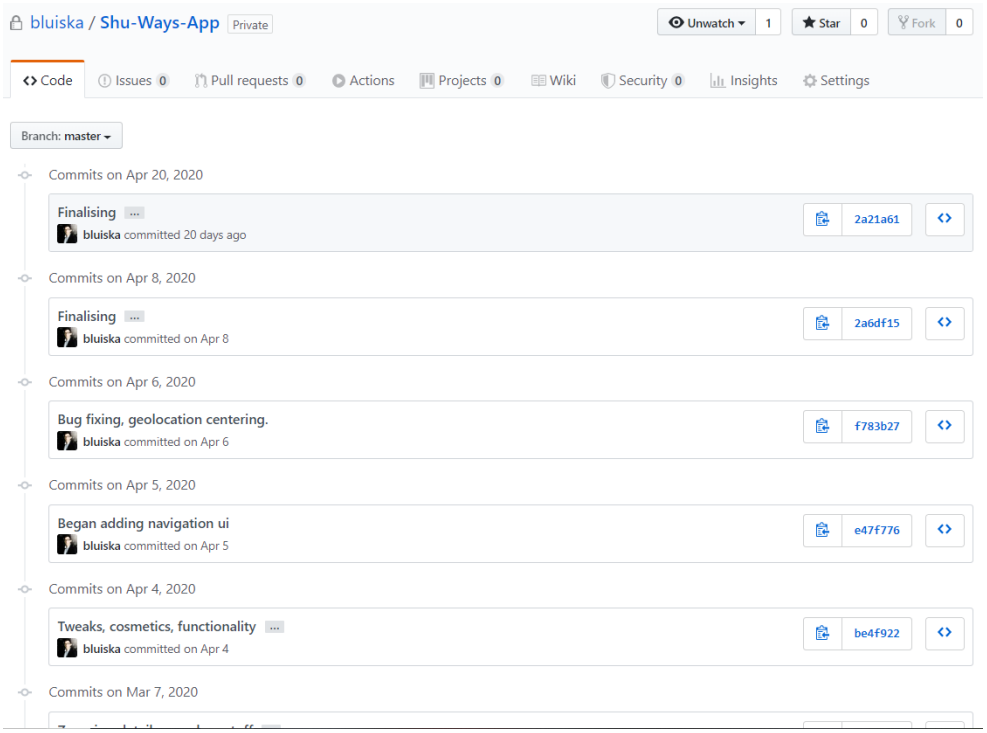


Figure 4.2: GitHub repository showing the latest commits

4.2 Prototyping and Learning

It was necessary to learn the basics of React to begin development. This was done by going through the tutorial on the official react website (Facebook, 2020b) and creating a TicTacToe game (Figure 4.3). It was also necessary to ensure the Anyplace servers could be accessed and queried. Therefore, some prototyping was done using the AXIOS library (Uraltsev & Zabriskie, n.d.) and some basic JavaScript. As the required knowledge of React was acquired and the anatomy of the Anyplace service was understood, it was possible to begin the development of the deliverable.

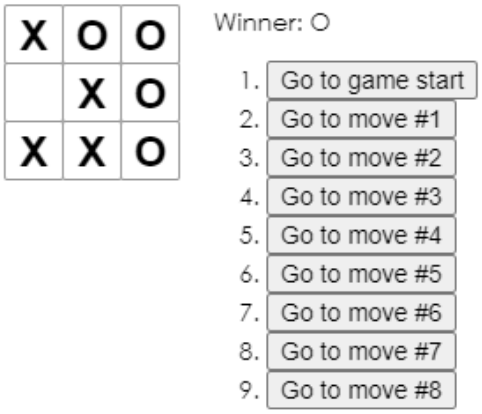
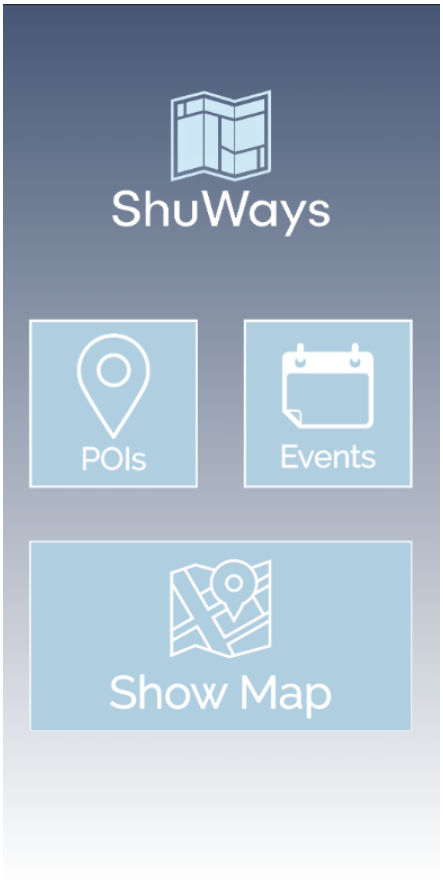


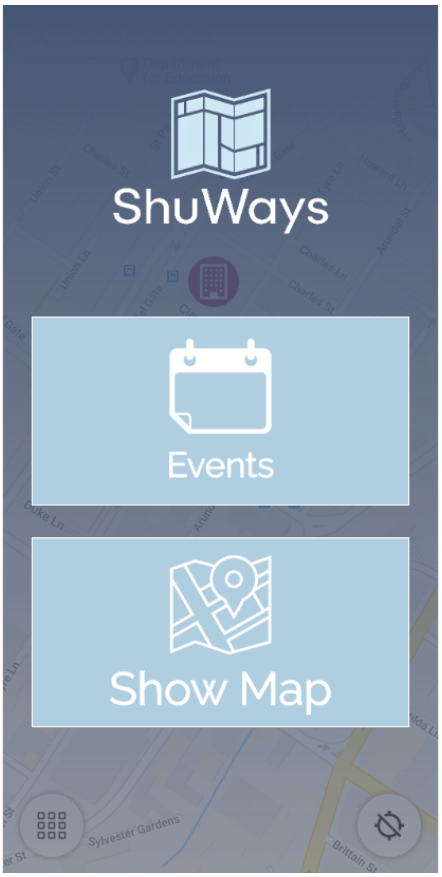
Figure 4.3: TicTacToe game developed during the React tutorial

4.3 Base application and main menu

Using the Ionic CLI (Ionic, 2020a), a React-based application template was created to serve as the skeleton for the project. The basic project created a blank page with all the necessary basic configuration done to run as a PWA. The menu was then styled and implemented as discussed in Section 3.3. The initial design seen in Figure 4.4a had to be modified to not include the POIs menu due to time constraints, resulting in the final implementation seen in Figure 4.4b. The loading of the map in the background was also added in the final implementation.



(a) Initial implementation



(b) Final implementation

Figure 4.4: Implementation of buildings

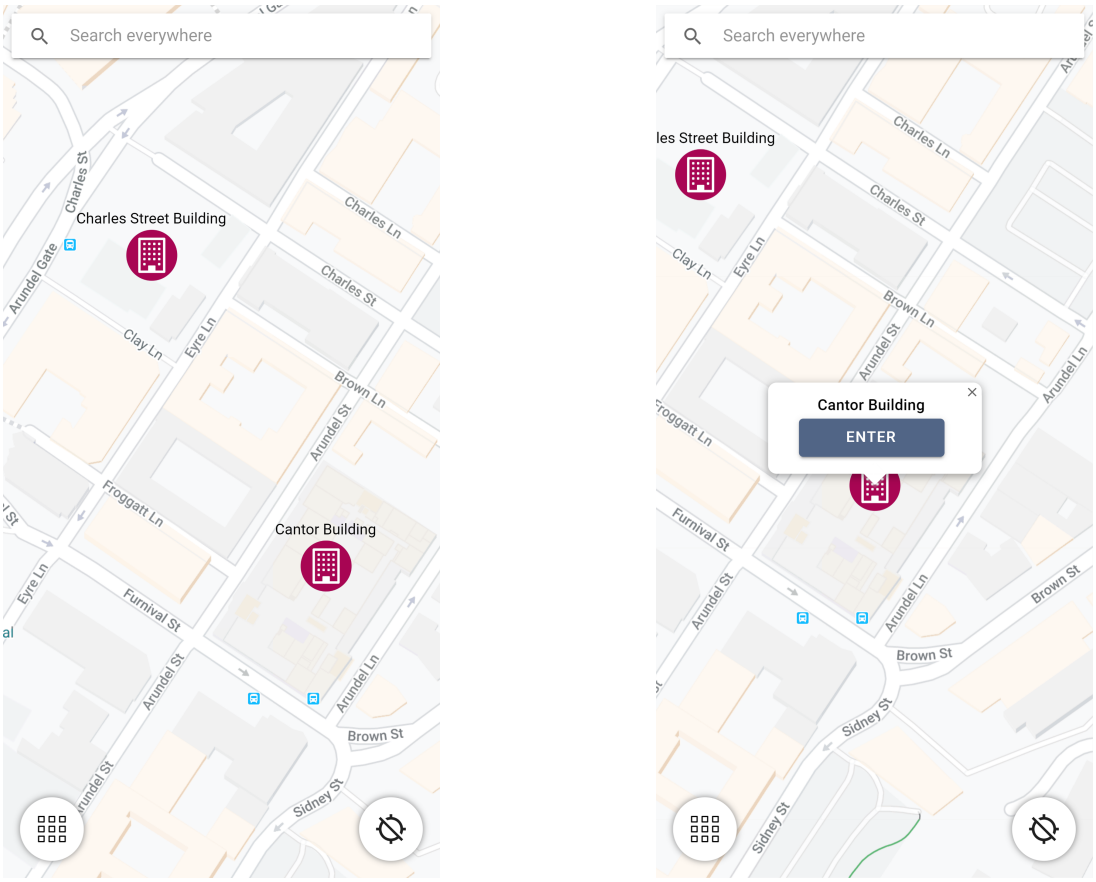
4.4 Populating Anyplace with data

Indoor maps of the first 3 floors of the Cantor Building were uploaded initially to the public Anyplace servers using the Anyplace Architect (Figure 4.5), with floor 3 and 4, as well as the first floor of Charles Street Building using a placeholder image (Appendix H). Over a couple of weeks, POIs of the ground floor and first floor of the Cantor Building were added including the route between POIs. Using the public Anyplace servers allowed for quicker prototyping in the initial stages of the project. All data was later moved to the locally hosted server.

Figure 4.5: Anyplace Architect

4.5 Buildings

The building markers were added to the map for the Charles Street Building and Cantor Building using the Marker component available in the react-google-maps (Klar & Lyakhov, n.d.). Figure 4.6a and Figure 4.6b show the implementation of building markers and entering of buildings respectively, as discussed in Section 3.8. When a building is entered using the "Enter" button, the indoor map is loaded and the map is zoomed in closer to show it.



(a) Building markers

(b) Entering a building

Figure 4.6: Implementation of buildings

4.6 Displaying the indoor map

The floor plan images are returned in a base 64 Portable Network Graphics (PNG) image format by the Anyplace server. This can be directly displayed in an "img" component in HTML. To anchor the image to the bounds of the building and display it, the OverlayView component was used from react-google-maps (Klar & Lyakhov, n.d.). Figure 4.7 shows the loaded indoor map on top of Google Maps, as discussed in Section 3.9. As the building and map data can take time to download when entering a building, to increase application's responsiveness the POIs and floor plan images are downloaded in the background as soon as the user loads the application.

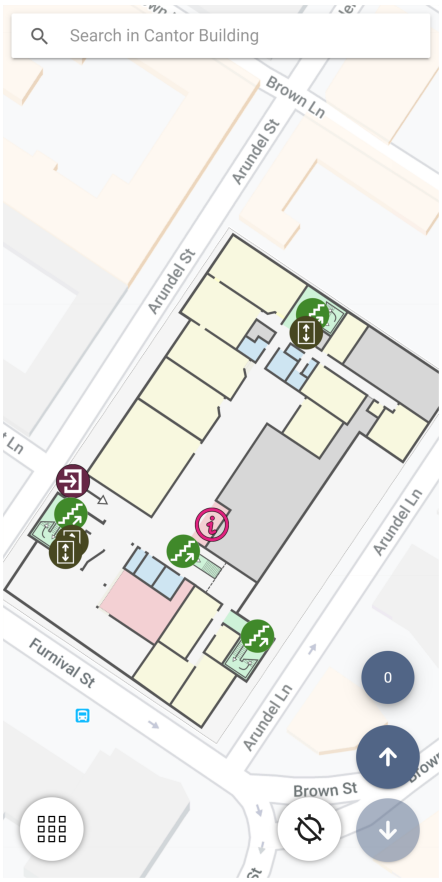


Figure 4.7: ShuWays displaying the Cantor Building map at zoom level 19.

4.7 Changing floors

Changing the floors required the development of the Floor Selector component discussed in Section 3.11. Changing floors was slow initially due to having to download the floor plans when the user requested to change the floor, but once caching of was implemented the change was much smoother. Figure 3.6 shows the floor selector in its initial state, while Figure 4.8 shows the floor selector component expanded to list the available floors as discussed in Section 3.11.

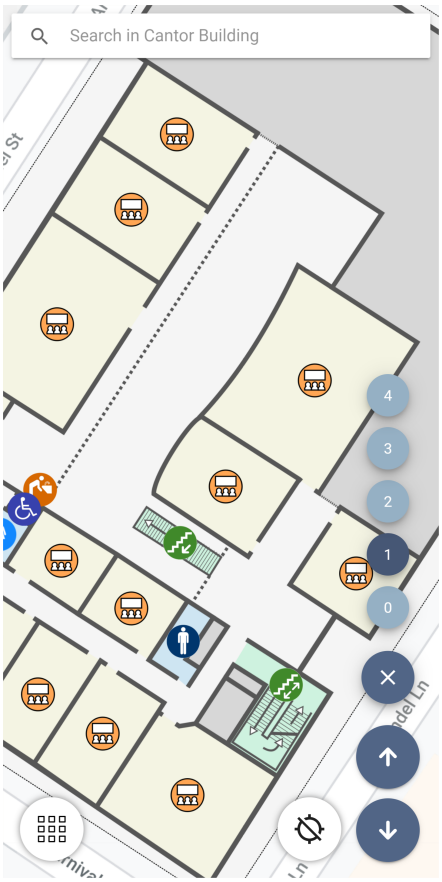


Figure 4.8: ShuWays displaying the Cantor Building map at zoom level 19.

4.8 POIs

The POIs are retrieved from AnyPlace as soon as the application loads to enable the search functionality to search them. For display purposes, a custom POI component was developed that extends the Marker component available in react-google-maps (Klar & Lyakhov, n.d.) and implements the selection of the correct POI image based on the type assigned. This ensures that to change the icons only one file needs to be modified. The images for POIs have been acquired and modified using IconFinder (ApS, 2020). Figure 4.9a shows various POIs being displayed in the Cantor Building as discussed in Section 3.10. Figure 4.9b shows the info box displayed when tapping a POI to get directions as discussed in Section 3.13.

(a) Various POIs displayed

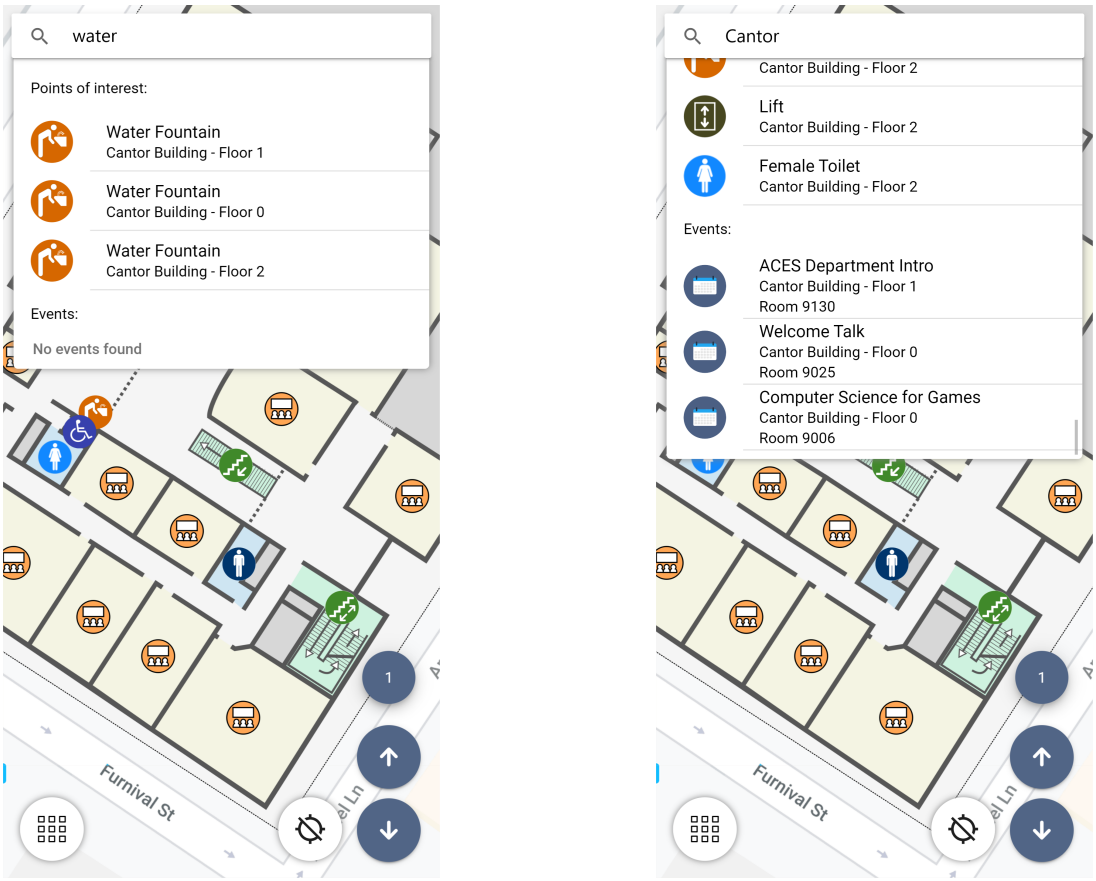
(b) POI info window

Figure 4.9: POIs implementation

4.9 Search

Ionic includes a search bar component that comes with various event handlers already implemented, as well as a cross-platform styling. This search bar was leveraged and customised to work for ShuWays. An expanding results list was created (see Figure 4.10a) as a reusable component so that the search results could be shown on the Navigation UI, discussed in Section 3.14. FuseJS (Risk, 2020) was the library used to perform a fuzzy search on the list of POIs and events. To ensure that relevant results would be listed the hyper-parameters were tweaked to what is believed to be an optimal setting based on experimentation (Figure 4.11). Furthermore, to help find what is relevant, results from the floor the user is viewing are prioritised and listed first with the rest listed in ascending order. For example, in Figure 4.10a the user is viewing Floor 1, therefore when searching "water", the results show the water fountain on Floor 1 first. Selecting the result will focus the map on the POI.

Figure 4.10a shows the implementation of the search functionality when searching for a POI and Figure 4.10b shows it when searching for events, as discussed in Section 3.7.



(a) Searching for POIs

(b) Searching for events

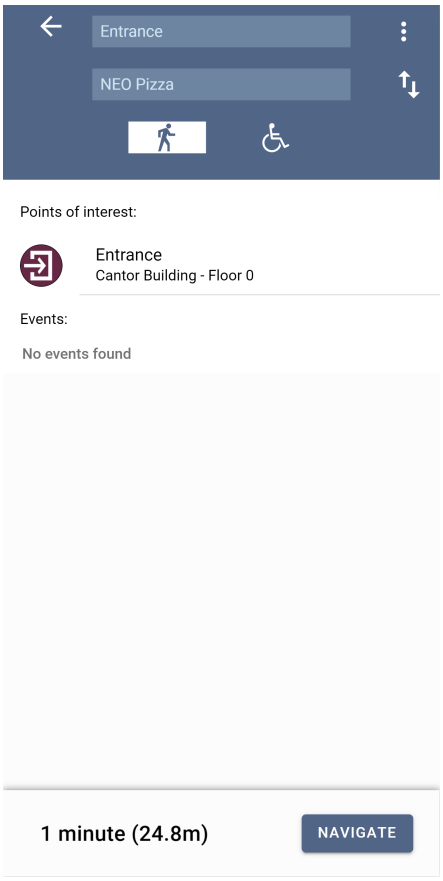
Figure 4.10: Implementation of the search functionality

```
const fuseOptions = {
    isCaseSensitive: false,
    shouldSort: true,
    threshold: 0.2,
    location: 0,
    distance: 100,
    maxPatternLength: 32,
    minMatchCharLength: 1,
    tokenize: false,
    keys: [
        "name",
        "description",
        "floor_number",
        "pois_type",
        "building_name",
        "location.building_name",
        "location.indoor_location.name",
        "time",
        "type",
    ],
};
```

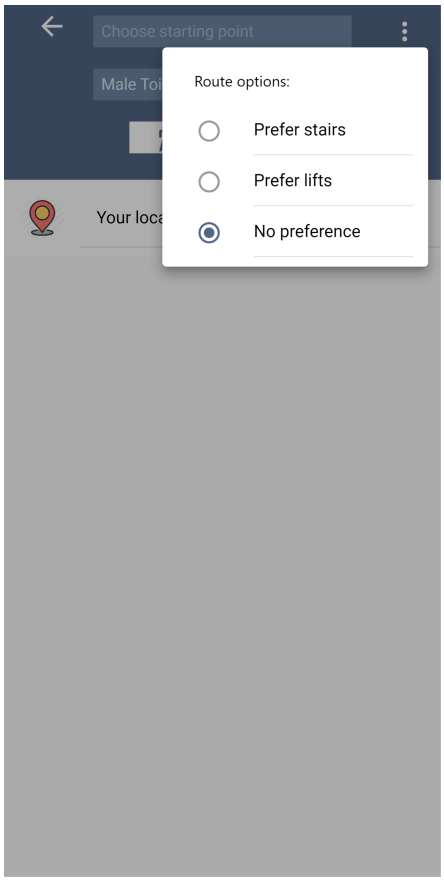
Figure 4.11: FuseJS parameters in ShuWays

4.10 Route Details View

The route details view was implemented as a separate component that is rendered when the "Directions" button is pressed in a POI popup window. The "To" and "From" sections implement the same search functionality as discussed in Section 4.9 and allow the user to search for their destination (or even origin) without having to leave the view. Figure 4.12b shows the three dots menu, which allows the setting of a routing preference as discussed in Section 3.14. If a valid route has been found, a bottom bar is displayed, which shows in minutes the length of the journey as well as the distance in metric units. No metric/imperial preference has been implemented as part of the prototype, but this would not be a difficult extension. Metric was used due to the distance calculations being performed in the metric system too. As ShuWays can be expected to be used by all age groups, the average walking speed of 1.25 meters per second was used for the calculation of journey time (Montufar, Arango, Porter, & Nakagawa, 2007). Figure 4.12a shows the implementation of the Route Details View as discussed in Section 3.14.



(a) Getting a route from the Cantor entrance to NEO Pizza

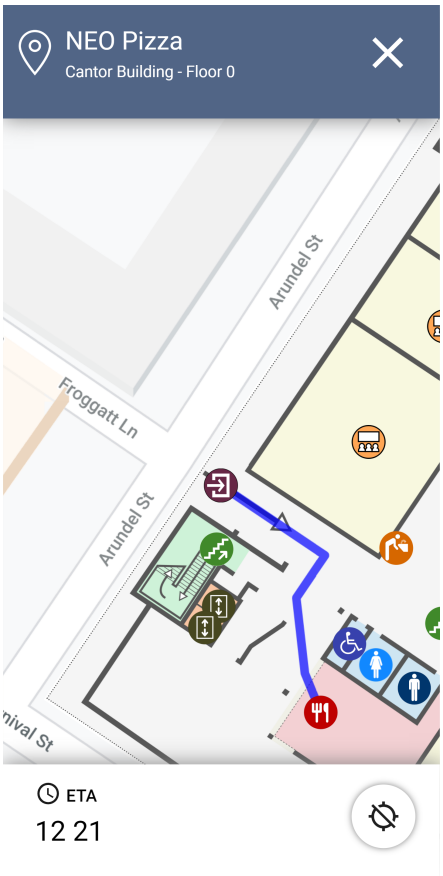


(b) Route preference

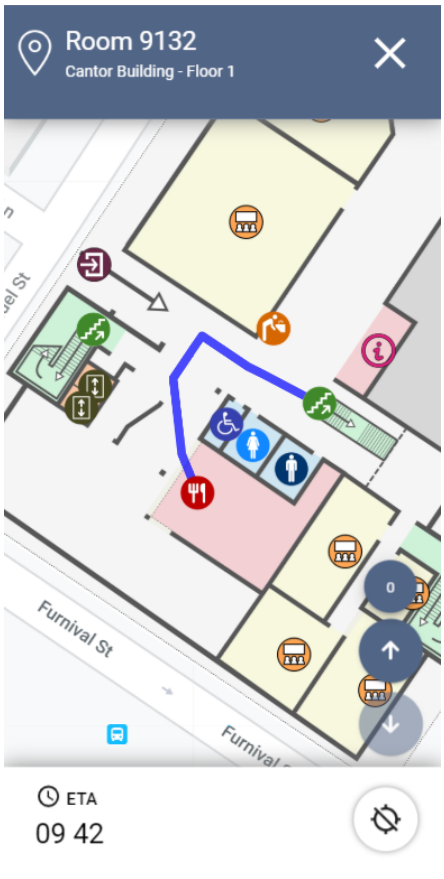
Figure 4.12: Implementation of the Route Details view

4.11 Indoor routes

Routes are a collection of LatLng objects (Google, 2020e). An array of LatLng objects can be passed to a Polyline component from the react-google-maps (Klar & Lyakhov, n.d.) to form a path, which is then styled blue and slightly opaque to form the path between POIs. The route is only visible when the indoor map is also visible. When getting directions to a POI located on a different floor to the starting point the shortest path will be taken (disregarding upward distance). However, it is possible to select a preference for stairs or lifts (discussed in Section 3.14). Figure 4.13a shows the implementation of the indoor routing between two POIs on the same floor and Figure 4.13b shows it when routing to a POI on a different floor.



(a) A route between two POIs on the same floor



(b) A route to a POI on a different floor

Figure 4.13: Implementation of the indoor routing

4.12 Outdoor routes

To navigate the user from outdoors to indoors, outdoor routing capability was implemented using the Directions API (Google, 2020c) and Distance Matrix API (Google, 2020d). The directions service was used to get the route and render it while the Distance Matrix API is used to calculate the distance on the Route Details View (Section 4.10), as it can take multiple destinations in a single call and return the overall shortest route. This is necessary when getting a route to a POI in a building with multiple entrances. The algorithm developed compares the outdoor route to each entrance and the indoor route from each entrance to the destination, returning the shortest route overall. Outdoor navigation is only possible from the location of the user (if known) in the current prototype. The outdoor path uses the default styling with two markers (A and B) showing the origin and the destination (which will be an entrance of a building). This styling could be changed in future work.

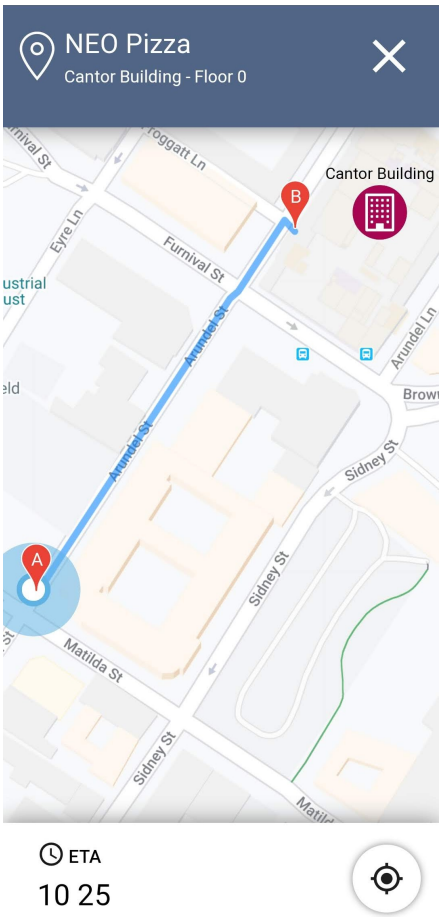
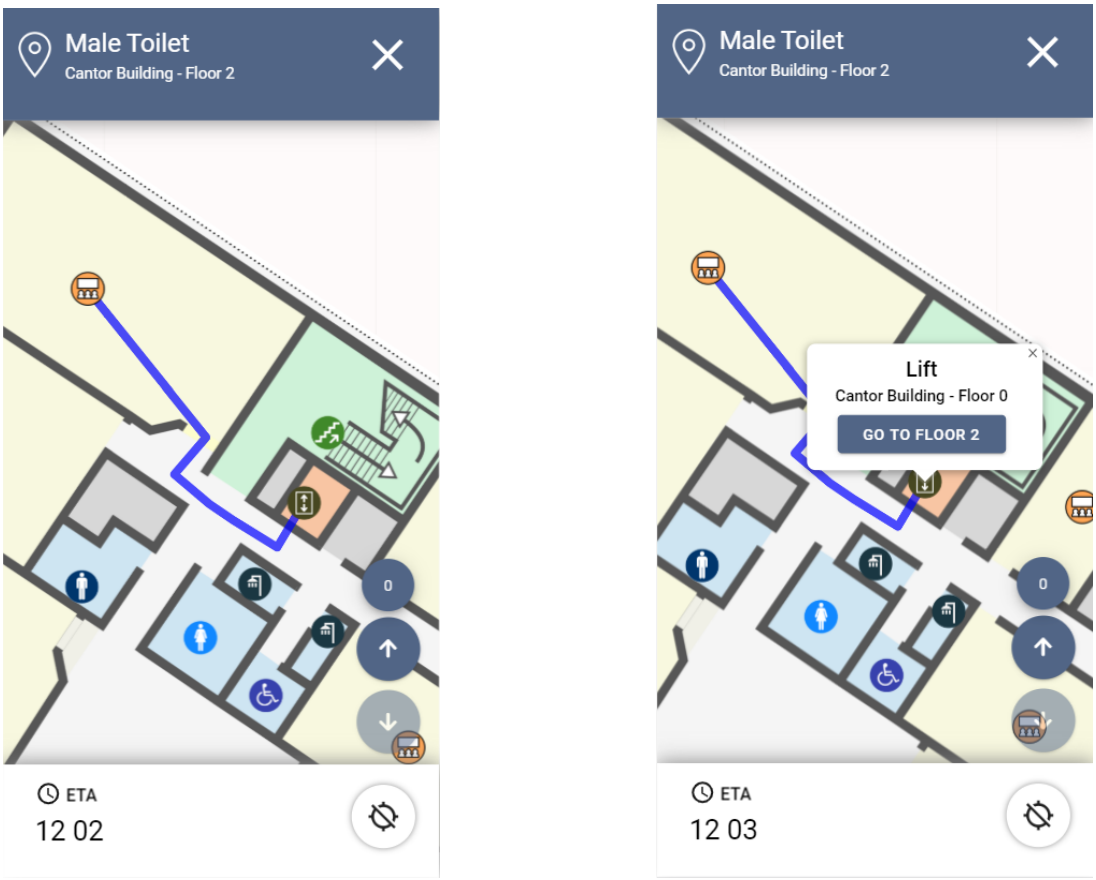


Figure 4.14: An outdoor route to inside the Cantor Building

4.13 Navigation UI

When the navigate button is pressed in the Route Details view, the Navigation UI is displayed over the map. Figure 4.15a shows the implementation as discussed in Section 3.15. The Navigation UI gives the user the estimated time of arrival (ETA) and focuses the user's attention on following the route displayed. The X button closes the navigation and clears the route. If the route includes a floor change, clicking on stairs or lifts that are part of the route will bring up a popup. The button in the popup (Figure 4.15b) will take the user directly to the next floor in the route, skipping floors where the user is just going up or down in the lift of stairs. Floors can also be changed using the floor selector discussed in Section 4.7. To ensure that the user does not request new directions without ending the navigation first the "Directions" button is hidden on POI popups.



(a) The Navigation UI

(b) Floor change popup

Figure 4.15: Implementation of the Navigation UI

4.14 Routing for limited mobility users

The routing algorithm works by creating a graph of the POIs using the connectors and edges specified in the Anyplace Architect. For the project, routing for limited mobility users means preferring lifts over stairs. This could be extended in the future to include more, such as taking the ramp as opposed to the route with a couple of steps in it. To enable the front-end to get routes with a preference, as discussed in Section 3.17, the unsuitable nodes (lift or stair POIs) and edges connecting to those nodes are removed from the graph. In the current prototype there is always a route for limited mobility users, so it is assumed that a valid route will be found. Figure 4.16a shows the route calculated when no preference was selected, whereas Figure 4.16b shows the same origin and destination but with limited mobility routing.

(a) Route from 9002, no route preference

(b) Route from 9002, limited mobility

Figure 4.16: Difference between route based on preference

4.15 Events

The project required to also list events that are happening at the university. The choice was made to connect the application to Firebase Firestore and store the events there (Figure 4.17), as designed in Section 3.17. Unfortunately the Events page could not be completed to look exactly as designed in Figure 3.2b. For the deliverable, 3 events have been added to the database and the "Show on map" functionality was developed, but the "currently searching" filtering has not been implemented. Figure 4.18 shows the implemented events page.

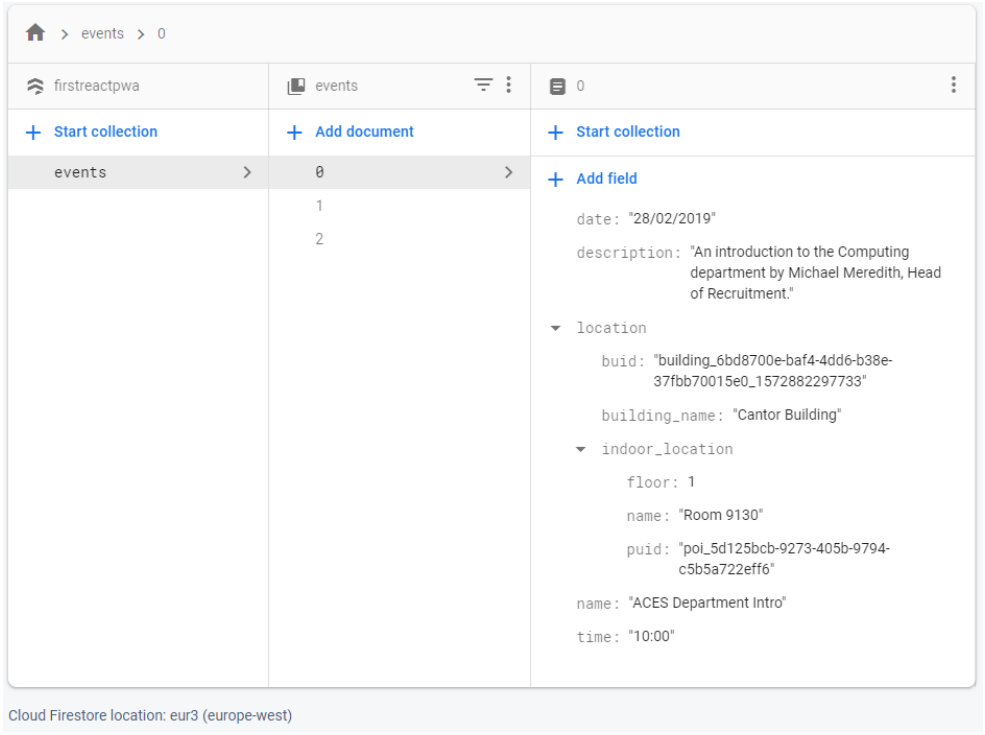


Figure 4.17: The Firebase Firestore showing the stored events

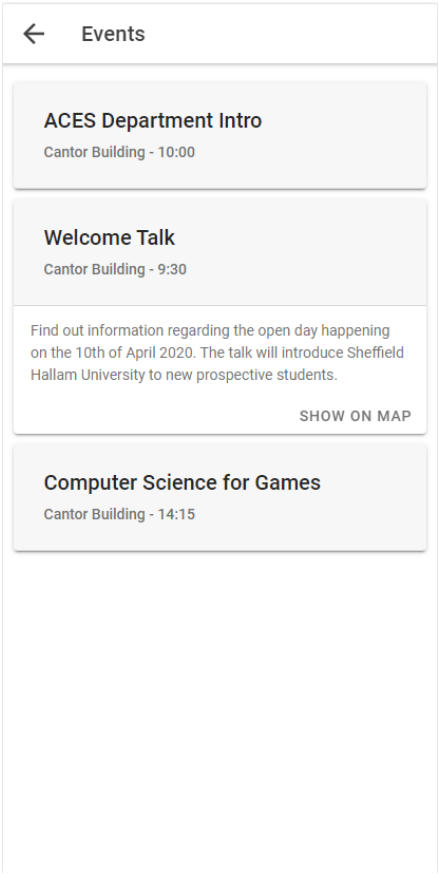


Figure 4.18: The implemented events page

4.16 Geo-location and "Your Location"

The application was originally intended to support real-time positioning using the WiFi system. This feature was not implemented (discussed in Section 6.2) so the project uses the HTML Geo-location API (Holdener, 2011) only. This is not as precise as WiFi positioning could have been (Woo et al., 2011), but it does try it's best to locate the user (depending on the phone's location settings). To show the location of the user a pulsating blue dot is displayed on the map (Figure 4.19). The location button was implemented as discussed in Section 3.6. When pressed, if location permission is allowed and the user is located in the Sheffield City Centre area the map will pan and focus on the user. If no permission is allowed, an error message is displayed (Figure 4.20). If the user is outside of the Sheffield City Centre area, a prompt is displayed informing the user to use other navigation apps to come closer to the campus (Figure 4.21). When a building is open and the button is pressed, if the user is outside the building bounds, the building will be closed and the map will focus on the user. If the user's location is inside the building bounds, the map will focus on the location of the user. However, if the user is navigating and the user is detected to be outside the building, a prompt will confirm if the user wants to leave navigation and exit the building (Figure 4.22).

When requesting directions from the user's location in the Route Details View (Section 3.14), if the user is inside the building they must confirm which floor they are

on as elevation cannot be determined using the HTML Geo-location API. The location of the user is then compared to the location of all known POIs and connectors (Section 2.6.5) and using the Euclidean distance (Ivis, 2006) the closest one is determined. A dotted line is then drawn from the user to the found POI with a route retrieved between it and the destination.

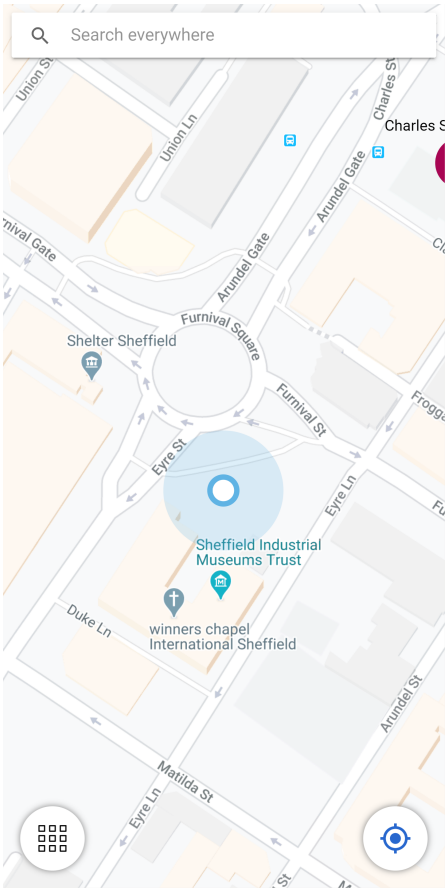


Figure 4.19: The map centred on the user's location

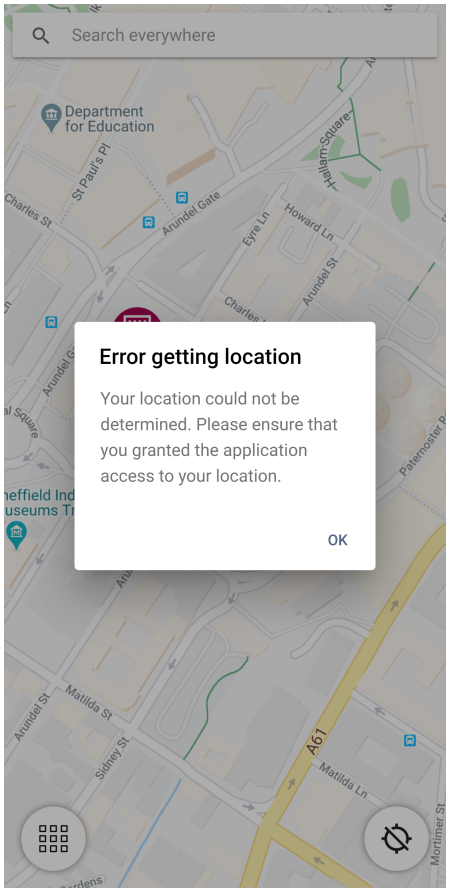


Figure 4.20: Difference between route based on preference

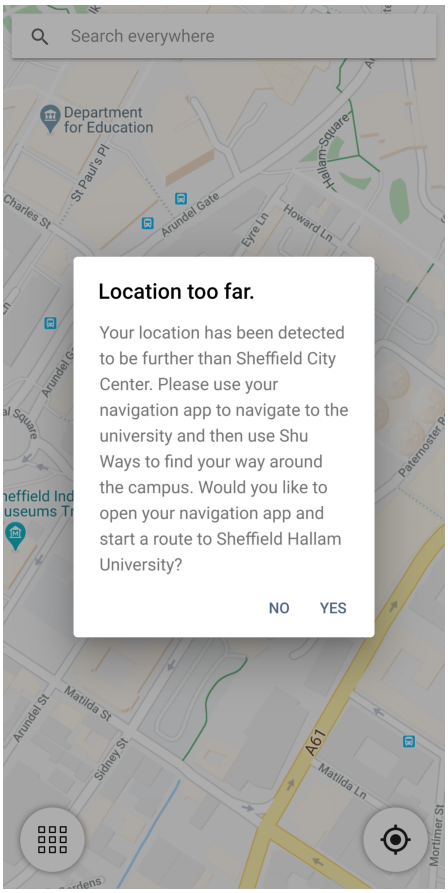


Figure 4.21: Location outside the bounds of Sheffield City Centre

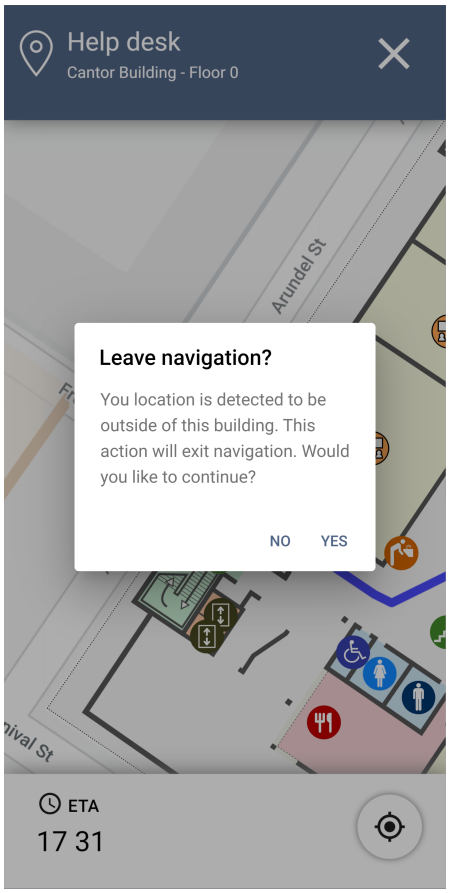


Figure 4.22: Location outside the bounds of the opened building while navigating

4.17 External URLs

Using the React Router it is possible to pass URL parameters to a page allowing external URLs to set up the application state when opened (Freeman, 2019). When the application loads the URL is parsed and checked for any parameters. If a POI ID is found, the map opens the building and focuses on the POI, displaying the same window that opens when a user clicks a POI icon.

5 Testing and Evaluation

5.1 Software Testing

During development, regression testing would often be performed to ensure changes have not affected previously working functions. Care was taken limit refactoring, although this proved to be difficult due to the prototyping nature of development, resulting in frequently breaking functionality. When the development was stopped, exploratory testing (Itkonen & Rautiainen, 2005) was performed to ensure the stability of the system before being handed to users.

5.2 User Evaluation

Due to restrictions imposed by the situation regarding Covid-19, the Cantor Building could not be accessed to perform user evaluation in the context the application was developed for. Had it been possible, at least 10 volunteers would have been asked to perform a set of tasks, using the application to locate POIs and navigate inside the building. The participants chosen would have been unfamiliar with the building to ensure they truly needed aid to find locations. Their interaction with the application would have been observed with both qualitative and quantitative data gathered.

Despite the circumstances, user evaluation of ShuWays was still performed albeit remotely. Skype was used to remotely observe the participant's phone screen as they interacted with the application, performing 4 given tasks (Appendix I). The tasks were designed to cover a wide range of use cases. Participants were aged between the ages of 20-42, only one of whom have previously been to the Cantor Building. Notes were taken during the test, which can be seen in Appendix F. When the tasks were completed, the two surveys seen in Appendix D and Appendix K were filled out by the participants.

Summarised results of the surveys:

- Which feature did you like the most in the application?
  - Navigation x 3
  - Clarity of the map x 2
  - Events list x 1
  - Search bar x 1
  - Going up floors using stairs and lifts when navigating x 1
- If you could change one thing about the application, what would it be?
  - Point on map functionality x 1
  - Better back button handling x 2
  - Better representation of multiple POIs that are the same type in the search x 1
  - Allow map rotation x 1
  - "Show on map" button stand out more x 1
  - To and From boxes highlighted better x 1

  - Nothing x 1

- What is the most important feature you think is missing? (if any)
  - Nothing x 4
  - Bookmark POIs x 1
  - List of POIs page x 1
  - Show which POIs have events x 1
  - Navigate back where you came from x 1

- Were the controls on-screen clear?
  - The average rating for the 8 participants was 8.375/9 (or 93%), which is very positive.

- Was the amount of information you saw display on-screen adequate to complete the tasks given?
  - Every participant has responded with "Yes", suggesting the UI is not too complicated and that it is clear.

- How easy did you find the product to use?
  - Participants found the product easy to use with an average rating of 8.375/9 (or 93%).

- How likely would you be to use this product if you had to visit the university and you were not familiar with the building you were visiting?
  - Participants would likely use this application with two participants rating 7/9 and six rating 9/9, resulting in a 94.4% likelihood of being used.

- Other observed remarks during the evaluation
  - A drawback of doing the evaluation remotely was that, for the scope of the project, the application was not fully optimised for all mobile device screen, which resulted in the UI not being perfectly aligned in all cases due to the different phone screen sizes and resolutions.
  - Some users phones were slower in performance, which due to the lack of progress indicators in parts of the application resulted in them thinking that the application has stopped working.
  - No user has zoomed in close enough to see the POI labels
  - Many participants found the map so clear they didn't need to use the search or the routing feature as they could see the POIs needed clearly and route they would need to take.
  - Many participants had trouble with the back button functionality, with most making the mistake of exiting the application due to incorrect handling of the navigation history
  - Participants could only assume they could get a hot drink at "NEO Pizza" as it was not displayed anywhere in the application what was available there.

- Many participants remarked that they found the application easy to use as it was similar to Google Maps. This further strengthens the persona of Jordan (Appendix A) and the relevance of using it.

The System Usability Scale (SUS) score was also calculated by asking the participants to complete the standard SUS form (Lewis, 2018). ShuWays has achieved a very good **90.3%** score. In conclusion, the application has performed well and was liked by the participants with all participants being likely to use it if they had to visit the university.

6 Critical Reflection

6.1 Deliverable successes

The deliverable of the project has met most of the aims set out in the project specification (Appendix C) and it has been praised during user testing, with all participants having an enjoyable experience using it. Research was successfully undertaken to identify existing solutions in the indoor mapping scene and learn from their design. Personas, scenarios, and good design patterns have also been effectively utilised to create wireframes and designs according to UCD principles. The final implementation would be familiar to use for Jordan (Appendix A) as well as Matthew (Appendix B) and their use case scenarios would be achieved. ShuWays was effectively implemented in a cross-platform manner by deploying it as a PWA, allowing it to be mobile accessible on all platforms (given an internet connection). The map of the Cantor Building was successfully designed with colour blind users in mind (which was not an original requirement but a consideration realised later in development), although for the prototype, only the ground floor and floor one are fully complete. The search functionality requirement was also achieved, allowing both POIs and events to be searched and located on the map. Route planning, including routing for limited mobility users, was also accomplished, with this feature being highlighted by participants during user testing. On top of the initial requirement of using the Cantor Building, the Charles Street building was added with a placeholder map and a few POIs to demonstrate more complex routing solutions. ShuWays successfully combined outdoor and indoor routing too, as per the original requirements. Project management tools were successfully utilised to track the workflow and the GitHub source control tool was used to keep versions of the software.

6.2 Deliverable limitations

Though the project has been largely successful it has shortcomings. The main menu, as well as the POIs and Events page was not implemented as designed (Section 3.5) due to issues with routing and the search functionality taking precedence during development. This is not a major limitation as POIs can still be searched using the universal search bar on the Map view, with the Events view looking similar to the design. The current implementation of the Events view would fail if there were many events, as there is no way to filter them. The biggest limitation of the project is the lack of accurate indoor positioning. Due to Covid-19 buildings were inaccessible and the development of this feature was also left late in the project. The architecture of the project makes it possible for this technology to be implemented, especially by building on top of Anyplace's implementation, as they expose an API, which takes a list of visible access points around the user and estimates the position based on the known radio map of the building. This method would never have worked in the browser due to no access to the WiFi hardware (Section 2.5.1). A requirement of the project was to develop a proof-of-concept Android application. Using Ionic and Capacitor, ShuWays can be compiled for Android (Ionic, n.d.) with access to the WiFi hardware provided by a Cordova plugin (tripflex, 2019), but access to WiFi scanning is only available on Android, not iOS (Apple, 2017), and even on Android, newer versions have imposed restrictions on the number of scans an application can perform (Google, 2020g), which greatly reduces the suitability of this method.

As noted by a participant in user testing, when more then one of the same POI exists on a floor the search results become unclear as there is no way to identify where each one is located. This issue was considered during development but it could not be addressed due to lack of time. A wire-frame drawn during development (Appendix M) shows the intended implementation, which is to list all POIs that are the same and highlight them as Google Maps does. This would allow the user to select the most appropriate POI. Thankfully, users also found the indoor map and the icons displayed very intuitive, with many not even using the search to locate POIs (Section 5.2).

Unfortunately, the street-view feature was not implemented due to the lack of access to the buildings. The Google Maps API used by ShuWays would have allowed the addition of a custom street-view path with custom images (Google, 2020b).

The requirement to "drop a pin" to use for navigation was also missing from the final deliverable. This feature was suggested by a participant as something they would have liked. It is possible to achieve the touch and hold functionality using the Touch-Start event and a timer, however, the current implementation only brings up the Route Details view indoors. Given more time, this feature would not be too difficult to implement.

Finally, a limitation of the Google Maps API for JavaScript is the inability to rotate the map. As discussed in the literature review (Section 2.6.4), this choice was made early on but the importance of the choice was not realised till later on. Some users pointed out the inability to rotate the map and said they would prefer to be able to rotate it but on the positive side they felt that this did not impact them too much.

6.3 Future work

Below are some considerations for future work to improve and add new functionality to ShuWays on top of addressing the limitations highlighted above:

- Less fragmentation at the backend
  - The Anyplace project was a great starting point for the application as the mapping functionality didn't need to be re-invented. However, to add events it was better to do this in Firebase. Such fragmentation can cause headaches as multiple sources need to be maintained. A better solution would be to have one central backend that is either built on top of the Anyplace implementation or potentially other sources.
- UI on different devices
  - The current implementation in Ionic is a PWA; therefore, it works in the browser. This means a wide range of devices can access the application very easily. As future work, the UI of ShuWays could be optimised to look good on larger screen devices like laptops, desktop monitors, and smaller screens than the mobile phone used for prototyping (Galaxy S20+).
- Advanced limited mobility routes
  - Currently, routing for limited mobility users means that the algorithm routes the user to lifts instead of stairs when changing floors. There are buildings at Hallam, however, where a certain route includes steps but no floor changes (Appendix J). In this case the university provides a ramp to

allow access. The routing algorithm could be extended to consider these ramps by adding a property to the edges connecting POIs. This property could identify if a particular path is suitable for limited mobility users.

- Advanced deep-linking
  - At the moment, ShuWays can be opened up to focus on a POI by scanning a QR code. This functionality could be extended to setup advanced application states such as a pre-configured navigation route from a parking lot to an event or to load a specific list of events using filtering. This would increase ease of use with the application on open days as less configuration would be needed. Scan a QR code and go.
- Turn based navigation
  - Turn based navigation would hugely boost the usability of the application as it could provide clear directions at every step of the routing process. Users could follow exactly what next step they need to take such as "Take the lift to floor 3" or "Turn left".

6.4 Project management

The work undertaken during the project could have been managed better. Often time and focus was placed on other assignments taking place during the semester. Pivotal Tracker was also neglected at times resulting in needing to go back and create a backlog of work to ensure progress was tracked. A diary of development was taken only towards the end of the development phase resulting in needing to go back on the Git commits and Pivotal to reflect on what work was done. The user-centred design process also usually involves multiple runs of user testing, which was only done once for this project at the end. If the project was better managed and development would have started earlier, user testing could have been performed more often, aiding the design process. On the other hand, the project was successfully adapted when the circumstances due to Covid-19 were introduced. Some additional work was carried out to make up for the lack of access to the Cantor Building, with user testing adapted to be performed remotely.

6.5 Ethical issues and considerations

As the application does not store any personal information, GDPR was not a major concern during the development of the application. In future work, if user logins were to be created, this would have to be considered in more detail. The only consideration was location data. This data is never stored to ensure compliance with GDPR. It is only used during the runtime to show the current location of the user and allow navigation from where the user is.

The project included the use of user testing, which involved the collection of data. To ensure the ethical use of the data, participant consent was obtained (Appendix L) following GDPR regulations.

Lastly, the project addresses disability by allowing routing for limited mobility users, so it was important to ensure that no discrimination was present. When routing, the user can select from a "walking person" icon or a "wheelchair" icon. However, since

not all limited mobility users might be in a wheelchair, when the "walking person" icon is selected the user has the option to set a routing preference to prefer lifts, stairs or no preference (Figure 4.12b). This ensures the whatever needs or preferences a user has, they have an option.

6.6 Personal development

I have developed immensely during the course of the project both personally and professionally. I started the project with some vague knowledge of JavaScript and no knowledge of React and Ionic web app development at all. Throughout the whole project I was constantly learning best practices and new concepts, all of which allowed me to deliver the successful application ShuWays is. I have also developed a deeper understanding of the user-centred design process, the importance of focusing on the end-user and prototyping, and learned the significance of getting user feedback, which unfortunately I only carried out once. It is now clear that it would have been very beneficial to seek out potential users of the application to gather feedback during the design phase of development. During the whole development process I have faced many challenges especially with getting the Anyplace system working due to the lack of documentation on their part, but all of these challenges allowed me to gain new knowledge. I have learnt more about writing scientifically (using LaTeX) and about justifying the decisions behind my actions. In retrospect, I would have allocated more time to the project as I have enjoyed my time working on it, and given more time, some of the limitations discussed in Section 6.2 could have been addressed. Before working on future project I would ensure that a good development cycle and plan was in place, to ensure all the requirements can be realised and done so on time.

7 Appendices

7.1 Appendix A - Jordan Young

Persona of Jordan Young
Jordan Young is a 17 year old male from Nottingham. He is currently finishing his A-Levels and is looking to study Computer Science at Sheffield Hallam University. He is familiar with technology having played with gaming consoles and computers since he was 3 years old. He is also a regular smartphone user. He uses his phone, specifically Google Maps, often to find directions to new places he visits (Speake, 2015). Jordan is much more likely to google to find out information as opposed to finding it out from brochures, guides or ask someone. (Antunovic, Parsons, & Cooke, 2018)

Jordan Young - Scenario 1
It is a warm and sunny Saturday morning. The time is 10:27. Jordan just arrived at Hallam Square for the open day happening today. He was travelling using the buses in Sheffield and knew how to get to Hallam Square from the Arundel Gate Bus stop of the 52 bus. He is looking to attend a talk that is described on the website as being in the Cantor Building. He has seen that there is a app available to help find where to go which he is eager to give a try. In the university guide he sees that the events contain QR codes besides them and that there is a note on the top of the page informing him that he can scan this to get directions. He pulls out his phone and scans the QR code of the "Welcome Talk" event. His phone loads the web page of the ShuWays app and focuses on the room where the welcome talk is being held at. Jordan finds this pretty cool. He is happy that it was this easy to load the application and the application did not have to be installed. In the application, Jordan presses the button to get directions and he is presented with a view where he can select where he wants to start from. On this screen, he selects to start from his location. As he selects this options, the application prompts him to grant access to his location. As he wants to get directions from where he is, Jordan sees no problem with that and presses allow. The application now calculates and displays the details of the route. He appreciates being given a time of arrival and selects the option to navigate. The application now displays the route from where Jordan is at Hallam Square to the entrance of the Cantor Building (where the Welcome Talk is). This view of the map makes it clear for Jordan that the application is first navigating him to the building entrance as he cannot see any indoor directions straight away. As he is walking towards the directions given, his location is being updated in real-time on the map with an indicator showing where he is. This allows him to know he is on the right path. While walking, he receives a text messages from his friend which he opens and reads. This takes him out of the application to his messaging app. When he is finished reading the messages he navigates back to the browser which is still on the page of the application and continues his journey towards the Cantor Building.

Jordan Young - Scenario 2
Jordan has been wanting to attend the Welcome Talk event happening in the Cantor Building at 11:00 and he has used the ShuWays application to get directions from Hallam Square. He has already followed the directions outside and has now entered the Cantor Building. In the application, Jordan selects the Cantor Building on the map as he can see he is near it and the application gives him the option to "Enter"

the building. He taps on this option and is presented with the route he will need to take indoors as well as the indoor floor plan of the Cantor Building. He is pleased with being able to see all the points-of-interests on the map clearly and identify what is what. He follows the route to the stairs next to the help desk. When he nears the stairs he knows he has to change floors as the application is still displaying the ground floor. He can see two arrows at the bottom right of the screen as well as an indicator that tells him he is on floor 0 and immediately knows that these buttons will allow him to move up and down the floors but out of curiosity he presses on the icon for the stairs. The application presents him with a window describing the stairs as going up to floor 1 and displays a button with the text "Go to floor 1". He is very pleased if not a little surprised by the intuitiveness of the application as he did not expect anything to happen when clicking on the stairs icon while navigating. Jordan walks up the stairs and follows the route to Room 9130 where the "Welcome Talk" is being held. When he arrives he closes the application tab in his browser and enters the room happy about the ease of navigation in a place he has never been to before.

7.2 Appendix B - Matthew West

Persona of Matthew West
Matthew West is a 57 year old English Teacher who lives in Sheffield and has 3 daughters ages 13, 15 and 17. His 17 year old, Holly, is looking to start university in September 2021 to study Architecture at Sheffield Hallam University. Matthew was late to adopt a smartphone having gotten his first one just 3 years ago when his oldest daughter had one since she was 11. However, since getting it he has been using it more and more often over his laptop which he was using before. He is increasingly doing his daily tasks on his phone such as online banking, communicating via instant messaging and searching for information. (Ofcom, 2016) While driving to places, Matthew often uses his cars in-built satnav to find locations which is a Garmin based system. It took him some time to familiarise himself with the system but since he has owned the car for 2 years now he is familiar with operating it. Matthew has also been diagnosed with deuteranopia (green colour blindness). This doesn't affect much of his daily life but he does struggle with some mobile applications that use colours which blend together, making it harder for Matthew to make out the interface.

Matthew West - Scenario 1
It is the 9th of April, 2020, 6'o clock in the evening. Sheffield Hallam University will be holding an open day event on the 12th of April, 2020. Matthew's daughter, Holly, would like to attend the event with his father. They have received an email just before dinner from the University which included details on the availability of the ShuWays application. As Matthew has not previously been to the university he is keen to take a look at where things are going to be. He opens the email on his Android smartphone (Samsung A70) and visits the link to the ShuWays application. The application loads and displays three menu items for him; one to view the points-of-interests, one to view events and one to show the map. As Matthew and his daughter are interested in attending an event, he selects events which takes him to a page listing the events upcoming. From the list of events, Matthew selects the "Welcome Talk for Architecture"

as he assumes that would be the first event they need to attend. The application now loads the map, displaying the indoor floor plan of the building the event is held at and focuses on Room 9130 of the Cantor Building as that is where the talk is happening. As the application has transitioned interfaces to the map interface, Matthew takes a couple of swipes on his screen to explore what he can do with this map view. He can see that the application is highlighting the room where the event is happening. He dismisses this highlight and browses the floor plan to orientate himself. He is impressed by the ability to see the floor plan and the immediate surroundings around it. He pinches to zoom out at which point the floor plan of the Cantor Building is hidden and the map of Sheffield is visible. He appreciates that he can see some major points of interest in Sheffield such as "The Moor" as this helps him quickly understand the area of Sheffield he is looking at. Out of curiosity, he wonders if there is somewhere where he could get some food. He sees the search bar present at the top of the screen, the same place he's seen it in other map applications. He clicks the search bar and types in "food". As he begins typing, the search results start listing results that match the closest to what he typed. As he completes typing food, the NEO Pizza at Cantor Building is listed. He selects this item from the search list which results in the application centering in on the POI, opening the Cantor Building floor plan and highlighting NEO Pizza. Matthew is pleased he was able to so easily search what is available and be presented with the results very quickly.

7.3 Appendix C - Project Specification

PROJECT SPECIFICATION - Project (Technical Computing) 2019/20

Student:	**Gergo Kekesi**
Date:	**24/10/2019**
Supervisor:	**Michael Meredith**
Degree Course:	**Computer Science BSc**
Title of Project:	**SHU Ways – An indoor navigation solution for campus to aid visitors and students in finding their way around the university.**

Elaboration

Sheffield Hallam University spreads across multiple buildings located near the city centre as well as around the Ecclesall area. For anyone new to the University, finding where to go can be a real challenge. On an open day, people need to speak to quite a few people to find the place they are looking for. New students may also find it difficult for the first couple of weeks (and often longer) to navigate the vast area of the campuses.

The aim of this project is to help address the issue of navigation by providing an accessible way to locate and navigate to various points of interests (POIs) around the University. A mobile accessible application will allow users to search for POIs such as rooms, help desks, toilets, water fountains and other useful locations. It will ideally combine outdoor GPS based navigation with indoor navigation functionality to help guide users from the outside all the way to a room inside the building. The SHU Ways application will be developed as a cross-platform application in order to ensure that most smartphone users are able to access the map. The map will be able to operate in both “offline” and “real-time” mode where “real-time” means the users are given real-time updates about where they are located inside the building instead of just being able to manually operate the map (“offline”). The native version of the app will implement a proof-of-concept, real-time indoor positioning and navigation system in the Cantor building.

Another consideration of the project will be the ability to plan a route for those with limited mobility. As indoors is usually more difficult to navigate for limited mobility users than outdoors, the route planner will consider the limited mobility of the user and plan a route that will be easier for them to take (such as taking them to elevators instead of stairs).

Furthermore, the application could be extended to list events of an open day or other event to help users find where they need to go without having to leave the application. For example, if there is a Computing Department Welcome Talk, the search function could allow users to search for this by name and allow users to navigate to the room the talk is held at.

Project Aims

The project aims to achieve the following:

- Research and investigate existing indoor mapping and navigation solutions as well as UI design patterns for map applications.
- Research the suitability and learn a cross-platform development framework such as React, Angular, Ionic, Xamarin and others.
- Research and learn a database solution to store the data used by the application (AWS, Google Cloud, Firebase...etc).
- Research the suitability of a Progressive Web App design for the application.
- Acquire and enhance a map of the Cantor building to be used as the basis for the indoor navigation concept.
- Apply a user centred design process for the development of the solution

- Use a project management tool to track the workflow as well as utilise a source control tool such as Git to manage source code.
- Create a mobile accessible application that allows users to browse the map manually as well as enable real-time positioning.
- Develop a search functionality allowing users to locate POIs (rooms, toilets, water fountains, elevators and others).
- Extend the search functionality with listing events happening on the day to allow users to easily locate and navigate to them.
- Develop a prototype real-time indoor positioning system using the existing WiFi infrastructure in place at the Cantor building to tell where the user is located inside the building.
- Implement route planning between two locations allowing users to plan a route either from a pinpointed location or from their location (if known) to a desired destination.
- Add the ability to plan routes for limited mobility users such as avoiding stairs in the route planning and instead taking them to elevators.
- Implement a navigation solution that can navigate users outdoors as well as indoors
- Extend the map with “street-view” like photos the user can look through to see the inside of the building.

Project deliverable(s)

The deliverable of the project will be a mobile accessible application. This application will connect to a database backend which stores the data used by the application (such as POIs, WiFi map…etc).

I will map out the Cantor building to be used for the prototype real-time positioning system. If time permits, I will map out other buildings at the City Campus too.

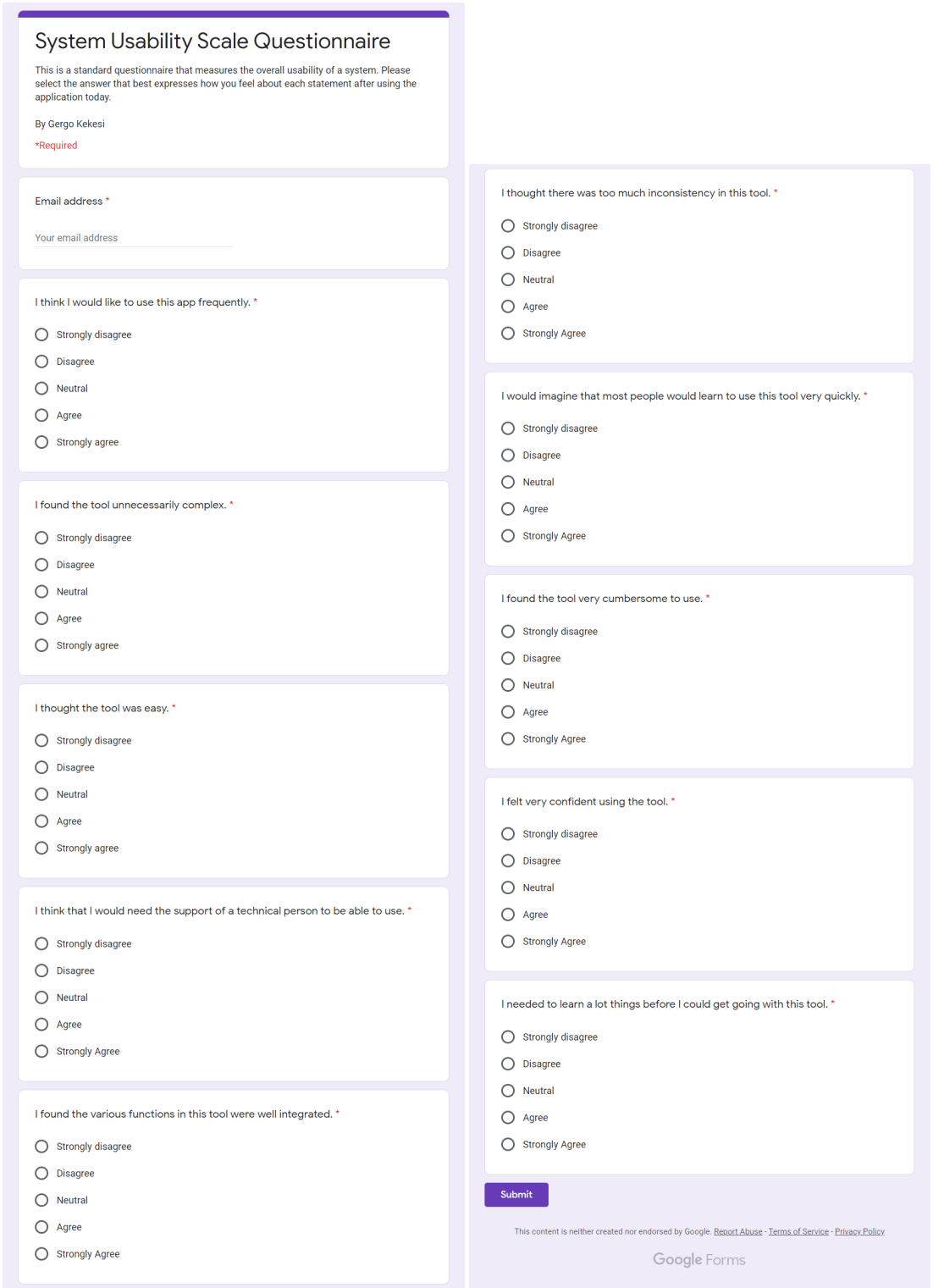
The real-time indoor positioning prototype will be delivered as an Android Application.

Action plan

Task	Deadline
Project Specification completed	25/10/2019
Setup project tracking and source control tools for the project	29/10/2019
Organise weekly sprints in the selected project management tool and create personas and scenarios to be used for design purposes	05/11/2019
Research cross-platform software technologies to develop the app in	12/11/2019
Research mapping frameworks and investigate suitable options	19/11/2019
Research route planning methods for the application in the chosen mapping framework	19/11/2019
Research data storage solutions to store the data used by the app	26/11/2019
Investigate the option of adding “Street View” like functionality to the application	26/11/2019
Research and investigate indoor positioning techniques and evaluate their suitability for my use case	03/12/2019
Information Review	06/12/2019
Design a database schema to store and retrieve POIs and other data for the application	17/12/2019
Design the core components of the application including the interaction between the database and the app in the chosen development stack	17/12/2019
Create the base application, displaying a single floor of the Cantor building in a user-friendly manner	07/01/2020
Integrate the chosen database solution with the application using the designed schema	14/01/2020

Extend the base application with route planning indoors from one location to another.	28/01/202
Add “Street View” option to the map and take 360° images of the Cantor building	04/02/2020
Carry out a survey with the new starters at university to gather thoughts on navigation and on the application	11/02/2020
Populate the map with known POIs and test navigation from a set point to a POI	18/02/2020
Provisional contents page	21/02/2020
Develop a search functionality to search the POIs and display it in the applications search feature	25/02/2020
Implement the most suitable indoor positioning technique in the Android prototype	10/03/2020
Gather feedback from the target audience to analyse the suitability of the current solution	17/03/2020
Draft critical evaluation and draft report sections	27/03/2020
Final changes to the application based on the user feedback gathered	31/03/2020
Finalise report and application	14/03/2020
Final review and last-minute changes to the report	21/03/2020
Submit Project Report to TurnitIn	22/04/2020
Submit Project Report Physically and Electronically	23/04/2020
Demonstration of work	12/05/2020

7.4 Appendix D - System Usability Survey



System Usability Scale Questionnaire

This is a standard questionnaire that measures the overall usability of a system. Please select the answer that best expresses how you feel about each statement after using the application today.

By Gergo Kekesi

*Required

Email address *

Your email address

I think I would like to use this app frequently. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

I found the tool unnecessarily complex. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

I thought the tool was easy. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

I think that I would need the support of a technical person to be able to use. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I found the various functions in this tool were well integrated. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I thought there was too much inconsistency in this tool. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I would imagine that most people would learn to use this tool very quickly. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I found the tool very cumbersome to use. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I felt very confident using the tool. *

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

I needed to learn a lot things before I could get going with this tool. *

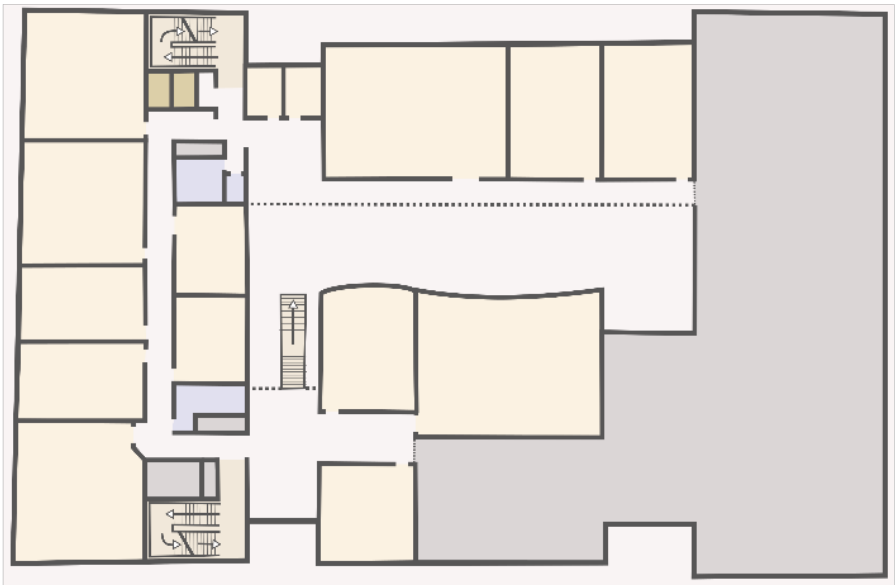
- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly Agree

Submit

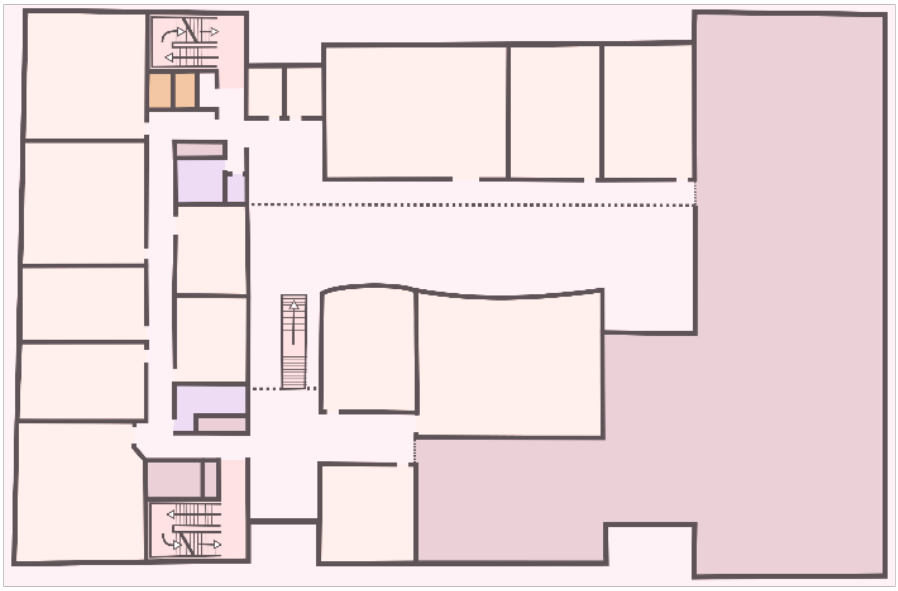
This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

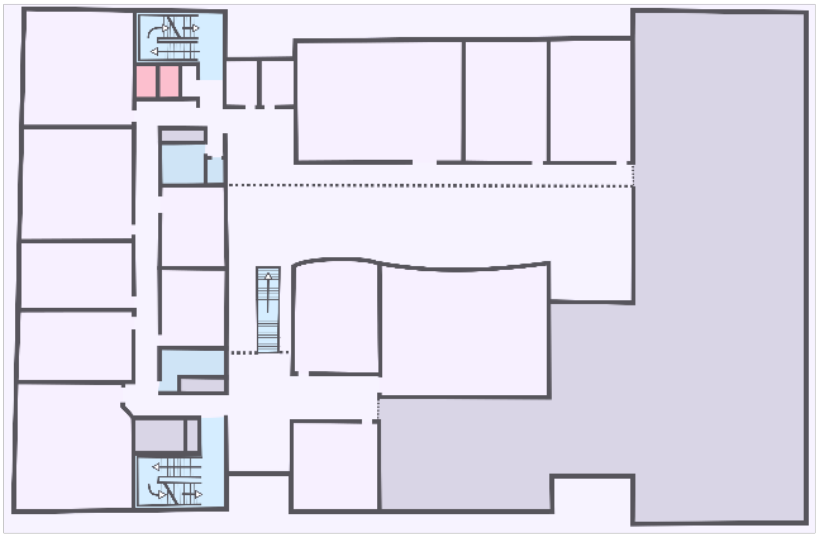
7.5 Appendix E - Colour blind view of the Cantor Building Level 1 map



(a) New colours - Red colour blind



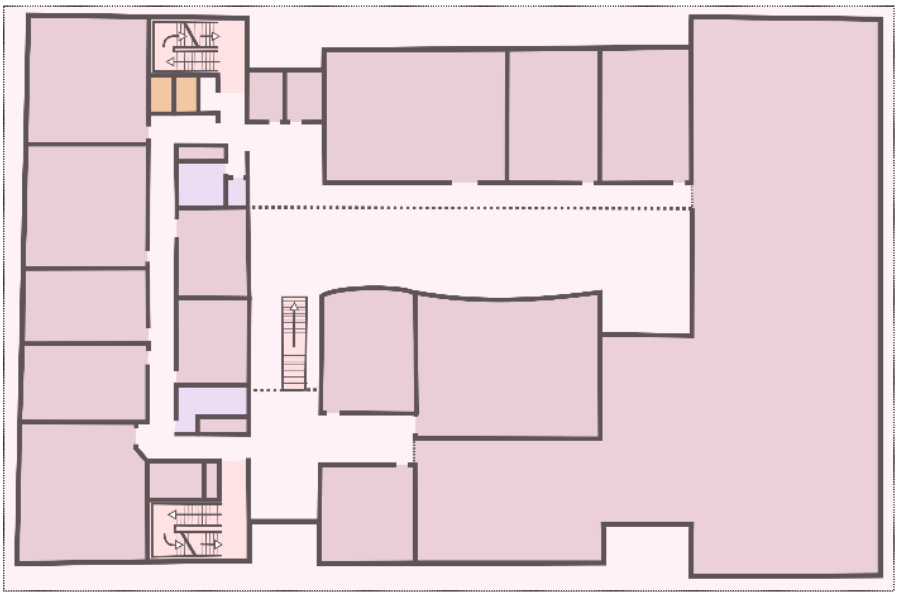
(b) New colours - Green colour blind



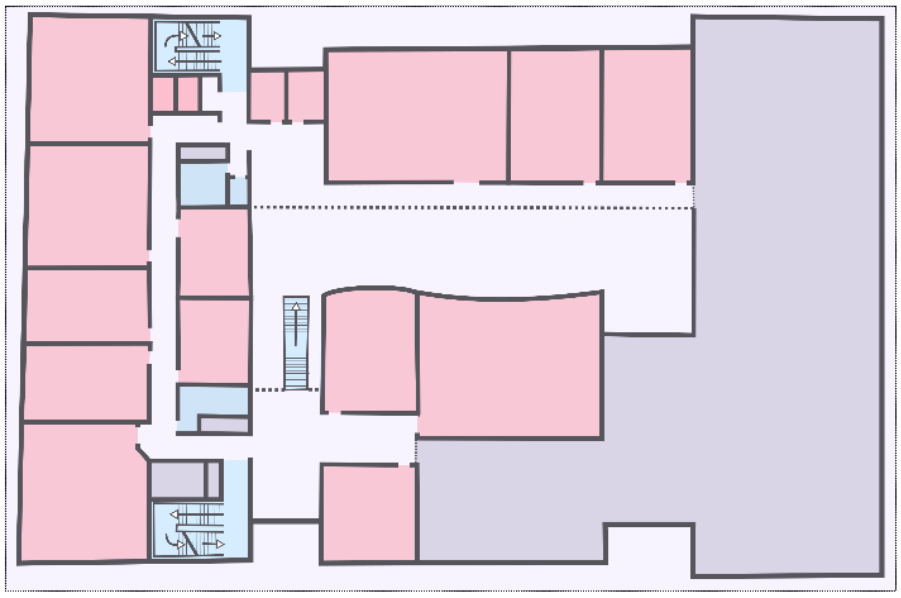
(c) New colours - Blue colour blind

(d) Old colours - Non colour blind view

(e) Old colours - Red colour blind



(f) Old colours - Green colour blind



(g) Old colours - Blue colour blind

7.6 Appendix F - User Evaluation Notes and Remarks

The following notes are a summary of what has been recorded during each participants use of ShuWays while conducting the user research.

Participant 1

The user found the event in Task 1 and navigated to the map using the "Show on map" button however they intended to return to the Events page by pressing back on their phone which resulted in them leaving the app. On Task 2, the user found the way by manually looking for the entrance of the building on the map and then flipping the direction in the Route Details view. According them this was because they were comparing the app to Google Maps and did not think Google Maps would find "Entrance" if they searched for it. For Task 3, the user simply observed the displayed indoor map and found the food symbol where they assume they can get a drink. During Task 4, the user repeatedly pressed the floor selector without any response from the app initially. Turns out the users phone was just slow (especially when compared to other participants). The user suggested the display of a loading bar or progress bar would have stopped them from clicking. The user also tapped on the bottom bar in the Navigation View expecting to see the full route plan which did not happen.

Participant 2

The user found the event in Task 1 and navigated to the map using the "Show on map" button however just like Participant 1, they intended to return to the Events page by pressing back on their phone which resulted in them leaving the app. Found the home button and navigated to events to progress with Task 2. Pressed "Show on map" and then selected "Directions" in the Info Box over the POI. The keyboard came up straight away when opening Route Details and the user typed "Entrance" to find the POI. Selected the item to finish Task 2. For Task 3, the user used the navigation feature to get to the rest room after seeing it displayed close to them. Then, scrolled the map manually to find somewhere they could get a drink. Saw "NEO Pizza" but unsure if they could get a drink there. No information available about hot drink. Decides that if they were in the building, they would give "NEO Pizza" a try. To complete Task 4, the user's immediate instinct was to just find the way manually by browsing the map quoting "I use the search functionality as a last resort when I'm lost". Found the route by browsing the map so did not use the routing feature.

Participant 3

Task 1 completed easily. Did not click back button but used the menu button to get back to Events. Clicks Directions and searches for "Entrance" to finish Task 2. Found it very straight forward. To complete Task 3 the user did not use the search but observed the map and found food icon. They assume hot drink is available there suggesting that some sort of display of available items would be good. To complete Task 4, they attempted to search for Lift but was confused with too many lifts showing (clear shortcoming of the app as I ran out of time to implement this) as they intended to get to Level 2 first then find the Water Fountain. Having browsed the map however, found the route manually to the water fountain easily.

Participant 4

Finds event easy for Task 1 but makes a remark about "Show on map" not looking a lot like a button. To complete Task 2, attempts to browse map and enter the Cantor Building but then realises that doesn't know where the event is so goes back to Events. Finds the event, clicks Directions and types Entrance. For Task 3, the user does not use the directions, the map is clear enough. Unsure if "NEO Pizza" has hot drinks. Found the way to the water fountain manually instead of using search.

Participant 5

Find the event easily and clicks "Show on map". Presses back accidentally and leaves the app. Upon returning the Info Box for the event is not displayed so they are not sure where it is. Attempts tapping a few POIs but makes a remark that the events are not listed under the POIs and that it'd be nice if they were. For Task 3, they find the food POI without searching. Taps it for more info, no more info given so assumes they could get a drink there. Manually searched directions as they thought the map was clear enough to follow and said aloud the route they would take, indicating they would take the stairs.

Participant 6

Finds events straight away in the Events menu. Clicks "Show on map" and locates where it is. For Task 2, the user found the events again using the menu button to go back and clicked "Directions" in the popup over the POI. Typed in entrance and completed Task 2. To complete Task 3, the user has observed the map and located the toilets, tapped on the toilets and remembering which room the event was in, typed in the room number. Then the user proceeded to swap the To and From to get the directions FROM the events location to the toilets. Then they located NEO Pizza by browsing the map but unsure if a hot drink would be served there. Task 4, the user attempted to figure out the route manually by looking at the map. They have successfully found the route to there without using the route planning. The user stated they like that it is similar to Google Maps.

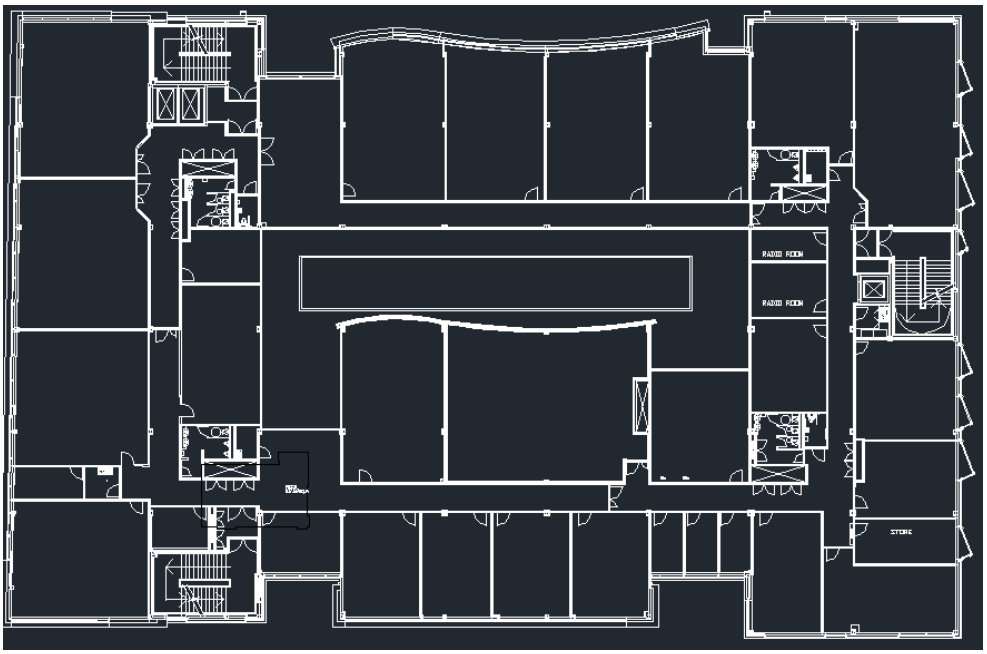
Participant 7

Finds events straight away in the Events menu. Clicks "Show on map" and locates where it is. For Task 2, the user found the events again using the menu button to go back and clicked "Directions" in the popup over the POI. They have however made the mistake of pressing back first. Typed in entrance and completed Task 2. The user found it clear from the map how to get to the toilets and did not need to use the navigation feature, however, found it hard to figure out where to get a hot drink as the name "NEO Pizza" confused them. Thought about going to help desk and asking as that was clar from the map. To complete Task 4, the user attempted initially to navigate manually but decided it was easier to just use the directions by finding the water fountain on level 2 and asking for directions to it. When following the route, the user used clicked the lift to go up to the second floor. Some remarks: The zoom-to-reveal pattern was confusing for the user as they thought that there was something wrong with the application.

Participant 8

Finds events straight away in the Events menu. Clicks "Show on map" and locates where it is. For Task 2, the user found the events again using the menu button to go back and clicked "Directions" in the popup over the POI. Typed in entrance and completed Task 2. Made a remark that it would have been nice if the TO and FROM inputs stood out more. Used the routing to get to the toilet by selecting the toilet and typing in the room they were in. Then manually searched for the food icon and found "NEO Pizza". They assume they could get a drink there.

7.7 Appendix G - AutoCAD floor plan of Cantor Level 1



7.8 Appendix H - Placeholder map for Charles Street Building

7.9 Appendix I - User Evaluation Tasks

ShuWays Application - User Tasks

Task 1

You are going to be visiting a Sheffield Hallam University open day and are interested to see where the "Welcome Talk" is going to be for the Computing Department. Please find out which building, floor and room number it will be happening in.

Task 2

Assume that you have arrived and the building. Please find out how to get the room where the Welcome Talk will be happening from the main entrance of the building, as if you just stepped into the building. How long will the journey take you and how far is the room from the entrance?

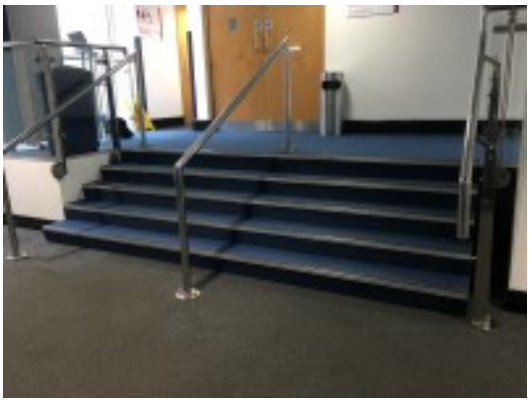
Task 3

Assume that the Welcome Talk has finished and you would like to get a hot drink. However, first you'd like to stop by the restroom. Please walk me through how you'd go about finding your way around doing the above task.

Task 4

You have just finished a talk in Room 9016 and are the lecture tells the room to meet upstairs at the Water Fountain on Level 2. Please walk me through how you'd go about finding your way around doing the above task.

7.10 Appendix J - Owen building ramp

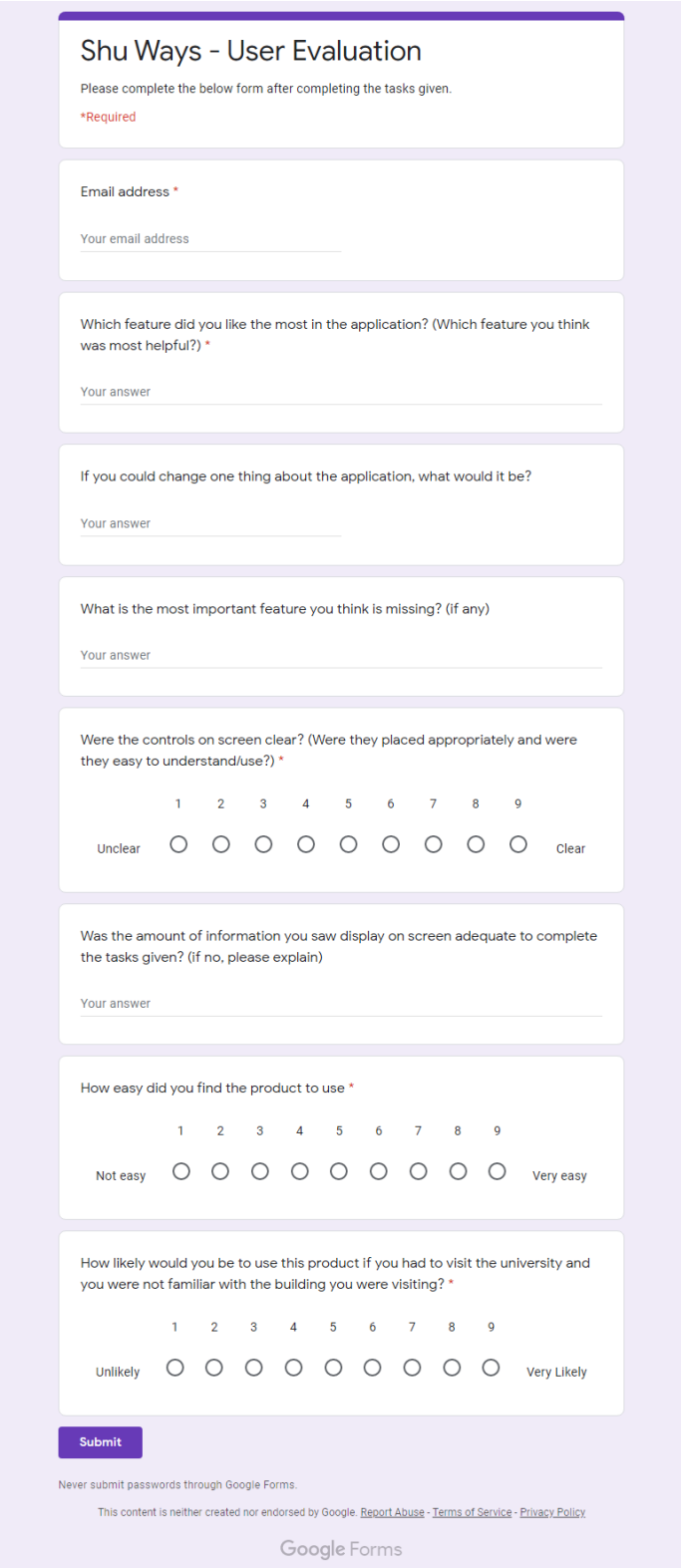


(a) Stairs route

(b) Limited mobility route

Access to the Peak Lecture theatre at the Owen building

7.11 Appendix K - User evaluation survey



Shu Ways - User Evaluation

Please complete the below form after completing the tasks given.

*Required

Email address *

Your email address

Which feature did you like the most in the application? (Which feature you think was most helpful?) *

Your answer

If you could change one thing about the application, what would it be?

Your answer

What is the most important feature you think is missing? (if any)

Your answer

Were the controls on screen clear? (Were they placed appropriately and were they easy to understand/use?) *

	1	2	3	4	5	6	7	8	9	
Unclear	○	○	○	○	○	○	○	○	○	Clear

Was the amount of information you saw display on screen adequate to complete the tasks given? (if no, please explain)

Your answer

How easy did you find the product to use *

	1	2	3	4	5	6	7	8	9	
Not easy	○	○	○	○	○	○	○	○	○	Very easy

How likely would you be to use this product if you had to visit the university and you were not familiar with the building you were visiting? *

	1	2	3	4	5	6	7	8	9	
Unlikely	○	○	○	○	○	○	○	○	○	Very Likely

Submit

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

Google Forms

7.12 Appendix L - GDPR Participant Information Sheet and Consent Form

Information Sheet

SHU WAYS Application – User Evaluation

You are invited to participate in a study to evaluate the usability of the SHU Ways application. You have been asked to participate as you fit the age group most likely to use this application. It is up to you to decide if you want to take part. A copy of the information provided here is yours to keep, along with the consent form if you do decide to take part. You can decide to withdraw at any time without giving a reason, or you can decide not to answer a particular question. During this evaluation you will be required to use the SHU Ways application on your mobile phone and allow observation of you using the application via Skype (screen-sharing). You will be asked to speak aloud your thoughts as you interact with the application (an example will be given during the evaluation on how this might look like) and complete a set of tasks given to you. The evaluation should take between 15-20 minutes. The screen of your mobile device will NOT be recorded, you are only asked to share your screen during the evaluation to allow me to follow along with your interaction. At the end of the evaluation you'll be required to fill out two short surveys via Google Forms. You will only be asked once to complete the study. Your results will be anonymized and no personal information will appear in any material using data from this study. The findings of the study will be summarized and presented as figures and quotes, part of the Bachelor's Thesis written by myself, Gergo Kekesi. No personal information or raw data will be published or be accessible to anyone but myself. I, Gergo Kekesi will be responsible for the raw data collected during the study and this data will be deleted within 3 months of the study. The University undertakes research as part of its function for the community under its legal status. Data protection allows us to use personal data for research with appropriate safeguards in place under the legal basis of public tasks that are in the public interest. A full statement of your rights can be found at https://www.shu.ac.uk/about-this-website/privacy-policy/privacy-notices/privacy-notice-for-research. However, all University research is reviewed to ensure that participants are treated appropriately and their rights respected. Further information at https://www.shu.ac.uk/research/ethics-integrity-and-practice.

Researcher contact details: Gergo Kekesi, Email: b6017384@my.shu.ac.uk, Mobile: 07578865686

You should contact the Data Protection Officer if:	**You should contact the Head of Research Ethics (Professor Ann Macaskill) if:**
• you have a query about how your data is used by the University • you would like to report a data security breach (e.g. if you think your personal data has been lost or disclosed inappropriately) • you would like to complain about how the University has used your personal data	• you have concerns with how the research was undertaken or how you were treated
DPO@shu.ac.uk	a.macaskill@shu.ac.uk
Postal address: Sheffield Hallam University, Howard Street, Sheffield S1 1WBT Telephone: 0114 225 5555	

Sheffield Hallam University

PARTICIPANT CONSENT FORM

TITLE OF RESEARCH STUDY: SHU Ways Application – User Evaluation

Please answer the following questions by ticking the response that applies

	YES	NO
1. I have read the Information Sheet for this study and have had details of the study explained to me.	☐	☐
2. My questions about the study have been answered to my satisfaction and I understand that I may ask further questions at any point.	☐	☐
3. I understand that I am free to withdraw from the study at any time, without giving a reason for my withdrawal or to decline to answer any particular questions in the study without any consequences to my future treatment by the researcher.	☐	☐
4. I agree to provide information to the researchers under the conditions of confidentiality set out in the Information Sheet.	☐	☐
5. I wish to participate in the study under the conditions set out in the Information Sheet.	☐	☐

Participant's Signature: _____ **Date:** _____

Participant's Name (Printed): _____

Contact details: _____

_____

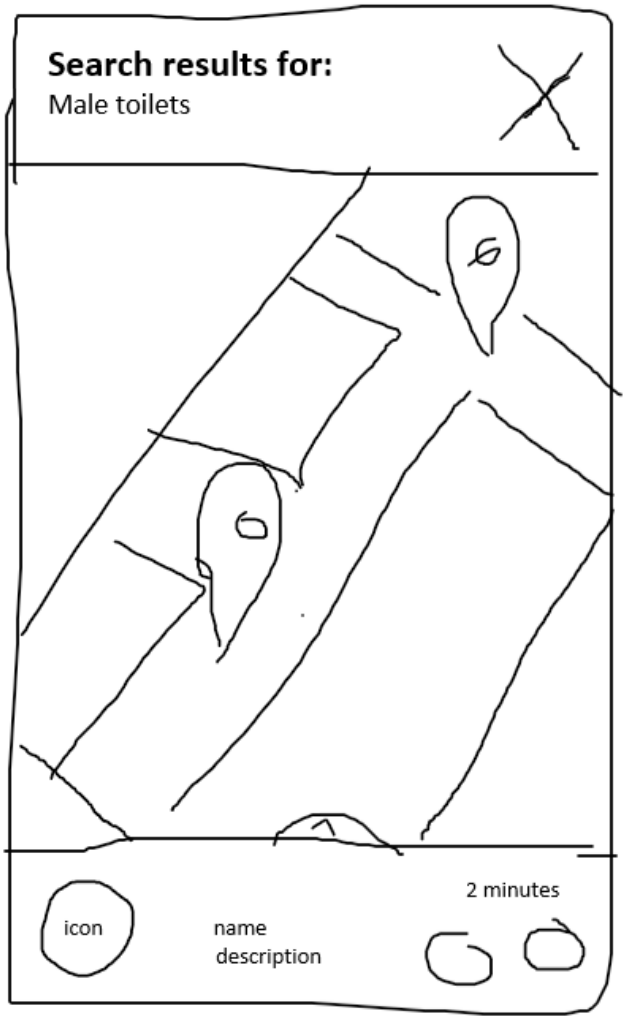
Researcher's Name (Printed): Gergo Kekesi

Researcher's Signature: 

Researcher's contact details:
Name: Gergo Kekesi
E-mail: b6017384@my.shu.ac.uk
Mobile: 07578865686

Please keep your copy of the consent form and the information sheet together.

7.13 Appendix M - Multiple POIs Search Wireframe



References

Abras, C., Maloney-Krichmar, D., Preece, J., et al. (2004). User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, *37*(4), 445–456.

Amazon Web Services, Inc. (2014). *Amazon dynamodb: Developer guide.* Retrieved from `https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Introduction.html` (Accessed on 17 May 2020)

Antunovic, D., Parsons, P., & Cooke, T. R. (2018, May). 'checking' and googling: Stages of news consumption among young adults. *Journalism*, *19*(5), 632-648. Retrieved from `https://journals.sagepub.com/doi/full/10.1177/1464884916663625` doi: 10.1177/1464884916663625

Anyplace. (n.d.). *Anyplace developers api.* Retrieved from `https://ap.cs.ucy.ac.cy/developers/` (Accessed on 12 May 2020)

Anyplace. (2020). *Anyplace repository.* Retrieved from `https://github.com/dmsl/anyplace` (Accessed on 12 May 2020)

Apple. (2014, Jun 2,). *Getting started with ibeacons.* Retrieved from `https://developer.apple.com/ibeacon/Getting-Started-with-iBeacon.pdf` (Accessed on 22 Apr 2020)

Apple. (2017, 08 14). *ios wi-fi management apis.* Retrieved from `https://developer.apple.com/library/archive/qa/qa1942/_index.html` (Accessed on 26 Apr 2020)

ApS, I. (2020). *Iconfinder.* Retrieved from `https://www.iconfinder.com/` (Accessed on 09 May 2020)

Artlens app. (2019). Retrieved from `https://www.clevelandart.org/artlens-gallery/artlens-app` (Accessed on 17 Apr 2020)

Calkins, M. J. (n.d.). *Best of atom: Features, plugins, acting like sublime text.* Retrieved from `http://scotch.io/bar-talk/best-of-atom-features-plugins-acting-like-sublime-text` (Accessed on 11 May 2020)

Chen, J., & Clarke, K. C. (2020). Indoor cartography. *Cartography and Geographic Information Science*, *47*(2), 95-109. Retrieved from `https://doi.org/10.1080/15230406.2019.1619482` doi: 10.1080/15230406.2019.1619482

Crickard, P. (2014). *Leaflet.js essentials.* Packt Publishing.

Dalton, R., Hölscher, C., & Spiers, H. (2015). Navigating complex buildings: Cognition, neuroscience and architectural design. In J. S. Gero (Ed.), *Studying visual and spatial reasoning for design creativity* (pp. 3–22). Dordrecht: Springer Netherlands.

Da Silva, T., Martin, A., Maurer, F., & Silveira, M. (2011, 09). User-centered design and agile methods: A systematic review. In (p. 77 - 86). doi: 10.1109/AGILE.2011.24

Department of Computer Science, U. o. C. (2018). *Anyplace — indoor information service* (Vol. 2019) (No. Dec 5,). Retrieved from `https://anyplace.cs.ucy.ac.cy/`

Facebook. (2020a). Retrieved from `https://reactjs.org/` (Accessed on 17 Apr 2020)

Facebook. (2020b). *Tutorial: Intro to react – react.* Retrieved from `https://reactjs.org/tutorial/tutorial.html` (Accessed on 12 Mar 2020)

Fransson, R., & Driaguine, A. (2017). *Comparing progressive web appli-*

cations with native android applications : An evaluation of performance when it comes to response time (Tech. Rep.). Retrieved from http://lnu.diva-portal.org/smash/get/diva2:1105475/FULLTEXT01.pdf (2018-01-13T18:02:21.793+01:00)

Gaigg, M. (n.d.-a). *Map ui patterns - floor selector.* Retrieved from https://www.mapuipatterns.com/floor-selector/ (Accessed on 02 May 2020)

Gaigg, M. (n.d.-b). *Map ui patterns - unified search.* Retrieved from https://www.mapuipatterns.com/unified-search/ (Accessed on 01 May 2020)

Garrett, J. J. (2011). *The elements of user experience: user-centered design for the web and beyond* (2nd ed ed.). Berkeley, CA: New Riders.

Google. (n.d.-a). *Adding firebase to a javascript project.* Retrieved from https://firebase.google.com/docs/web/setup (Accessed on 22 Apr 2020)

Google. (n.d.-b). *Google maps javascript api - zoom levels.* Retrieved from https://developers.google.com/maps/documentation/javascript/tutorial#zoom-levels (Accessed on 08 May 2020)

Google. (n.d.-c). *Material design - states.* Retrieved from https://material.io/design/interaction/states.html#usage (Accessed on 08 May 2020)

Google. (2020a). Retrieved from https://www.google.com/maps/about/partners/indoormaps/ (Accessed on 22 Apr 2020)

Google. (2020b, 04). *Custom street view panoramas.* Retrieved from https://developers.google.com/maps/documentation/javascript/examples/streetview-custom-simple (Accessed on 18 Apr 2020)

Google. (2020c, Apr). *Directions service — maps javascript api — google developers.* Retrieved from https://developers.google.com/maps/documentation/javascript/directions (Accessed on 10 May 2020)

Google. (2020d, Apr). *Distance matrix service — maps javascript api — google developers.* Retrieved from https://developers.google.com/maps/documentation/javascript/distancematrix (Accessed on 10 May 2020)

Google. (2020e, Apr 16). *Lat/lng object literal — maps javascript api.* Retrieved from https://developers.google.com/maps/documentation/javascript/examples/map-latlng-literal (Accessed on 12 May 2020)

Google. (2020f). *Order and limit data with cloud firestore.* Retrieved from https://firebase.google.com/docs/firestore/query-data/order-limit-data (Accessed on 06 May 2020)

Google. (2020g). *Wifi scanning overview.* Retrieved from https://developer.android.com/guide/topics/connectivity/wifi-scan0 (Accessed on 16 Apr 2020)

Griffin, D. (2011, Mar 04,). *Reviewed: Garmin nüvi 2360* (Vol. 2020) (No. Apr 11,). Retrieved from https://www.pocketgpsworld.com/Garmin-nuvi-2360-reviewed-37558.php (Accessed on 14 Apr 2020)

Hakim, M. (2016, Apr). Speed index and critical path rendering performance for isomorphic single page applications. In *The 16th winona computer science undergraduate research symposium* (p. 41-46).

Hamm, M. (2014). *Wireframing essentials : an introduction to user experience design : learn the fundamentals of designing the user experience for applications and websites.* Birmingham, England: Packt Publishing.

Holdener, A. (2011). *Html5 geolocation.* Sebastopol, CA: O'Reilly.

Inconshreveable. (n.d.). Retrieved from https://ngrok.com/ (Accessed on 08 May

2020)
IndoorAtlas. (2020, Mar). *Indooratlas.* Retrieved from https://www.indooratlas.com/ (Accessed on 13 Apr 2020)
Inkscape. (2020). Retrieved from https://inkscape.org/ (Accessed on 14 Apr 2020)
Ionic. (n.d.). Retrieved from https://ionicframework.com/docs/developing/android (Accessed on 09 May 2020)
Ionic. (2020a, Apr 02). *Ionic cli - ionic documentation.* Retrieved from https://ionicframework.com/docs/cli (Accessed on 11 May 2020)
Ionic. (2020b). *Ionic framework.* Retrieved from https://ionicframework.com/ (Accessed on 16 May 2020)
Itkonen, J., & Rautiainen, K. (2005). Exploratory testing: a multiple case study. In *2005 international symposium on empirical software engineering, 2005.* (p. 10 pp.-).
Ivis, F. J. (2006). Calculating geographic distance: Concepts and methods..
Kannan, V., Jhajharia, S., & Verma, S. (2014). Agile vs waterfall: A comparative analysis. *International Journal of Science, Engineering and Technology Research (IJSETR)*, *3*(10), 2680–2686.
Klar, U., & Lyakhov, A. (n.d.). *@react-google-maps/api.* Retrieved from https://www.npmjs.com/package/@react-google-maps/api (Accessed on 11 May 2020)
Langr, J. (2011). *Agile in a flash : speed-learning agile software development : Agile cards for agile teams.* Raleigh: Pragmatic Bookshelf.
Lewis, J. R. (2018). The system usability scale: Past, present, and future. *International Journal of Human–Computer Interaction*, *34*(7), 577-590. Retrieved from https://doi.org/10.1080/10447318.2018.1455307 doi: 10.1080/10447318.2018.1455307
Mapwize. (2018). *Mapwize.* Retrieved from https://www.mapwize.io (Accessed on 17 Mar 2020)
Mardan, A. (2017). *React quickly : painless web apps with react, jsx, redux, and graphql.* Shelter Island, NY: Manning Publications Co.
Midtbø, T. (2014). Indoor maps for orienteering sport events. *Scientific Journal of Orienteering*, *19*(1).
MockFlow. (2020). Retrieved from https://mockflow.com/ (Accessed on 14 Apr 2020)
Montufar, J., Arango, J., Porter, M., & Nakagawa, S. (2007, January 1,). Pedestrians' normal walking speed and speed when crossing a street. *Transportation Research Record*, *2002*(1), 90-97. Retrieved from https://doi.org/10.3141/2002-12 doi: 10.3141/2002-12
Nacheva, R. (2016, Oct). Business benefits of applying mobile user interface design patterns..
Ofcom. (2016). Adults' media use and attitudes 2016.
Peterson, G. (2009). *Gis cartography : a guide to effective map design.* Boca Raton: CRC Press.
Peterson, G. (2014). *Gis cartography : A guide to effective map design, 2nd edition.* Boca Raton, Florida: CRC Press.
Pew Research Center. (2018). *Smartphone ownership is growing rapidly around the world, but not always equally.* Retrieved from https://www.pewresearch.org/

global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/ (Accessed on 11 Apr 2020)

Pinto, C. M., & Coutinho, C. (2018). *From native to cross-platform hybrid development.* (ID: 1) doi: 10.1109/IS.2018.8710545

Pinto, C. M., & Coutinho, C. (2018). From native to cross-platform hybrid development. In *2018 international conference on intelligent systems (is)* (p. 669-676).

Pivotal Tracker. (n.d.). *Pivotal tracker vs. trello.* Retrieved from https://www.pivotaltracker.com/agile/tracker-vs-trello (Accessed on 07 May 2020)

Potter, J. (2020, May 02). Retrieved from https://www.npmtrends.com/angular-vs-react (Accessed on 02 May 2020)

Risk, K. (2020, Mar,). *Fusejs.io.* Retrieved from https://fusejs.io/ (Accessed on 26 Apr 2020)

Sarot, R. V., & Delazari, L. S. (2018, 12/). Evaluation of mobile device indoor maps for orientation tasks. *Boletim de Ciências Geodésicas*, *24*(4), 564-584. Retrieved from http://www.scielo.br/scielo.php?script=sci_abstract&pid=S1982-21702018000400564&lng=en&nrm=iso&tlng=en doi: 10.1590/s1982-21702018000400034

S.C Evercoder Software S.R.L. (2020). Retrieved from https://moqups.com/faq/ (Accessed on 14 Apr 2020)

Seshadri, S. (2018). *Angular : up and running : learning angular, step by step.* Sebastopol, CA: O'Reilly Media.

Sheppard, D. (2017). *Beginning progressive web app development.* Retrieved from https://doi.org/10.1007/978-1-4842-3090-9

Skupien, K. S. (2013). Colors and mapping: The right to receive information. Retrieved from https://scholarcommons.usf.edu/etd/4845

Sole, A. (2019). *Visual studio code distilled : evolved code editing for windows, macos, and linux.* California: Apress.

Speake, J. (2015). 'i've got my sat nav, it's alright': Users' attitudes towards, and engagements with, technologies of navigation. *The Cartographic Journal*, *52*(4), 345-355. Retrieved from https://doi.org/10.1080/00087041.2015.1108663 doi: 10.1080/00087041.2015.1108663

Stone, T. L., Adams, S., & Morioka, N. (2008). *Color design workbook: A real world guide to using color in graphic design.* Rockport Pub.

The GIMP team. (2020). Retrieved from https://www.gimp.org/about/ (Accessed on 14 Apr 2020)

tripflex. (2019). *Wifiwizard2.* Retrieved from https://github.com/tripflex/WifiWizard2

UpGuard. (2020, Mar). *Bitbucket vs github [updated for 2020].* Retrieved from https://www.upguard.com/articles/bitbucket-vs-github (Accessed on 17 Apr 2020)

Uraltsev, N., & Zabriskie, M. (n.d.). *axios - promise based http client for the browser and node.js.* Retrieved from https://www.npmjs.com/package/axios (Accessed on 14 Apr 2020)

Vanson Bourne Ltd. (2016, July). *The rise of indoor positioning.* Retrieved from https://www.indooratlas.com/wp-content/uploads/2016/09/A-2016-Global-Research-Report-On-The-Indoor-Positioning-Market.pdf

Verified Market Research. (2018). *Indoor location market.* Retrieved from https://www.verifiedmarketresearch.com/product/indoor-location-market/ (Ac-

cessed on 17 Apr 2020)

Wickline, M. (2018). *Coblis - colour blindness simulator.* Retrieved from https://www.color-blindness.com/coblis-color-blindness-simulator/ (Accessed on 03 Mar 2020)

Wohlgethan, E. (2018). *Supporting web development decisions by comparing three major javascript frameworks: Angular, react and vue. js* (Unpublished doctoral dissertation). Hochschule für Angewandte Wissenschaften Hamburg.

Woo, S., Jeong, S., Mok, E., Xia, L., Choi, C., Pyeon, M., & Heo, J. (2011). Application of wifi-based indoor positioning system for labor tracking at construction sites: A case study in guangzhou mtr. *Automation in Construction*, *20*(1), 3 - 13. Retrieved from http://www.sciencedirect.com/science/article/pii/S092658051000107X (Global convergence in construction) doi: https://doi.org/10.1016/j.autcon.2010.07.009

Zeinalipour-Yazti, D., & Laoudias, C. (2017, October). The anatomy of the anyplace indoor navigation service. *SIGSPATIAL Special*, *9*(2), 3–10. Retrieved from https://doi.org/10.1145/3151123.3151125 doi: 10.1145/3151123.3151125