Application for navigation inside Buckingham building

Nikoleta Koleva
UP899244

School of Computing
Final Year Engineering Project
M24739

May 5, 2022

Abstract

Navigation inside university buildings has become a common issue nowadays as both new and experienced students struggle to find their way inside large buildings. Indoor navigation is a topic that has not been covered thoroughly in popular resources, while mostly navigating outdoors has been considered. This project shows the process of creating an application which will help students navigate inside Buckingham building as it is one of the most commonly used buildings in which students have a difficulty with finding a staff office. The process involves conducting research, writing a literature review, elicitating requirements, designing and implementing an artefact. After the final artefact is completed, it will be compared to the initial plan and the requirements. An evaluation will be made to how well they were satisfied and a conclusion will be deduced.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank all the people who helped me and supported me while writing this engineering project - my supervisor Soraya Harding and my moderator Alex Bennett for their time and guidance, the Estates Department construction manager (Jonathan Craner) and the security in Buckingham building for giving me more information on Buckingham building's architecture, the School of Computing staff who assisted me with the development of the artefact and my family members and friends for their encouragement to continue working on this project until the end.

Chapter 1

1.1 Problem overview

Buckingham building is a large building with three entrances (two for staff and one for students), multiple exits, stairwells and elevator.

Upon requesting more information from the University of Portsmouth estates staff by email, a map was obtained which shows the building's room numbers. Additionally, more information was collected about the accessibility features inside the building (AccessAble, 2022) and a website was found showing the location of each staff member inside Buckingham building (University of Portsmouth, 2022). By analysing the gathered information on the building, a conclusion has been reached that the map provided by the Estates department combined with the accessibility information and the staff location would be essential to the development of the application.

Reaching a staff office inside Buckingham building has proven to be a common issue for both new and experienced students. There is currently no information desk and the only staff located in the building are the caretakers, the security and the staff members. Since they may not always be available and able to help with navigation inside the building, an application which will give instructions to reach their target destination would assist students greatly.

This project will show the process of designing and implementing an artefact for indoor navigation inside Buckingham building as currently there is no application available related to this topic.

1.2 Aims and objectives

The goal of this project is to create an application for indoor navigation that will allow people to find a staff office easily within Buckingham building.

Considering time constraints, this project will be aimed towards having the key features available. As a result, students' data gathered from questionnaires will be analyzed and a few destinations will be taken into consideration to prove the application's functionality - for example, a staff office on the ground and one on the first floor.

The project objectives are:

- Research the corresponding development tools and programming languages for the development of the application
- Test and review similar applications. List key features, evaluate their design and performance
- Compare alternative development models
- Choose development model and give proof of the project's time management
- Use questionnaires to gather information related to the requirements elicitation methods
- Produce design documentation (System architecture, Database design, Graphical User Interface Style, UML diagram, UI mockups)
- Create prototype
  - Develop server side
  - Develop client application
- Conduct appropriate testing for the application
- Evaluate the prototype against the Aims and Objectives, the Requirements and the project methodology
- Summarise the project's outcome and give recommendations prior to future work

1.3 Target audience

This project will seek a diverse audience of University of Portsmouth students who visit Buckingham building and are interested in using a navigation

application which will assist them in finding a staff office. Since students with impaired mobility would require specific routes to navigate inside the building and time is a serious constraint, this project will focus on the wider audience and will recommend implementing accessibility features to the application in the future.

1.4 Ethical, Social and Professional issues

Currently the prototype will avoid requiring personal data. Aside from this, there are no ethical, social or professional issues related to the development of the artefact.

For future work, the application will ask users to sign up to Google in order to access their timetable. Ethical, social and professional issues will be managed by Google (Google, 2022).

1.5 Structure overview

- Chapter 1 - Outlines the problem the project solves, its aims and objectives, target audience, ethical, social and professional issues and gives a brief structure overview
- Chapter 2 - Literature Review - Literature is researched, similar applications are tested and reviewed along with the suitable development tools and programming languages
- Chapter 3 - Methodology and project management - Compares alternative development models, describes and justifies the chosen model with its project plan and shows evidence of time management kept throughout the project. The development tools used are described here.
- Chapter 4 - Requirements and specification - Shows the requirements gathering methods, sets out the functional and non-functional requirements and groups them into featureSets.
- Chapter 5 - Design - Presents the design documentation, including Use case diagrams, System Architecture, Database design, GUI Style and UI mockups.
- Chapter 6 - First Prototype and Testing - Shows the development process of the prototype from beginning to end, with appropriate system tests.

- Chapter 7 - Second Prototype - The development process of the second prototype is described with the difference between the second and the first prototype. The final system is implemented.
- Chapter 8 - Testing - Shows the testing strategies conducted on the first prototype and the second prototype.
- Chapter 9 - Evaluation - Evaluates the prototype against the Aims and objectives of the project, the requirements set out in Chapter 4, the project methodology and a personal evaluation is made.
- Chapter 10 - Conclusion - Gives a brief summary and overview of the project and offers recommendations prior to Future work.

Chapter 2

Literature Review

2.1 Introduction

This literature review will investigate the key topics related to the development of an indoor navigation application for Buckingham building and the relevant academic literature. Existent solutions will be considered along with their features in common and their advantages and disadvantages. Additionally, the development tools used for the creation of the artefact will be reviewed and the requirements will be determined.

2.2 Cognitive factors

According to Golledge (1999), users depend on cognitive maps, which are not the accurate representation of a space, but rather the abstraction. Ishikawa and Montello (2006) deduct that the difficulty of navigating depends on a person's image between the physical space and their cognitive map. Based on their research, people have different spatial capabilities which affects their ability to create cognitive maps. Downs (1977) describes navigation as a set of subtasks: self-localization, route planning, traversing the path and discovering the location. Dasler et al. (2021) indicates that an application which covers these successfully would be overly simple, will provide orientation and route planning, reduce the need for precise cognitive maps and will guide the users to their desired destination in order to reduce the demand of maintaining spatial information.

2.3 Navigation techniques

Fallah et al.'s work (2013) brings out the problem that indoor systems can't rely on global positioning system (GPS), typically used for outdoor navigation, as GPS signals cannot be received indoors. However, a solution was found to this problem using an indoor positioning system with GPS signals in the 433 MHz Industrial Scientific Medical (ISM) band (Uzun, 2021). Montague's paper shows a different view as it states that GPS would not be sufficient as a pin on a map will suffice when we are looking at a road or a street map but would not help with indoor navigation as the building may have different floors (2010) which shows that a different solution is required.

Two main approaches can be considered to indoor localization - infrastructure-based and infrastructureless (Subedi et al., 2020). The infrastructureless method utilizes environmental features like sound, light, magnetic field or uses smart phone sensors like accelerometer, gyroscope, etc. Infrastructure-based approaches typically makes use of pre-installed visual sensors or wireless technologies such as ZigBee, WiFi, Ultra-Wideband (UWB) (Orjales et al., 2021, Vazquez-Lopez, 2021), Radio Frequency Identification (RFID) (Fu et al., 2020) and Bluetooth Low Energy (BLE) (Subedi et al., 2020). The downside of infrastructure-based indoor positioning system is that it can be expensive, the cause of which are the methods needed or the hardware components. Additionally, installing the necessary infrastructure often takes a lot of time and effort, which increases the cost (Simon Tomažič, 2021). However, UWB and Wi-Fi approaches (network-based navigation systems) deliver more accuracy on positioning than Bluetooth and RFID (Ozdenizci B, 2011).

2.4 Application development

In this section the key app development concepts will be discussed.

Device

One of the challenges of the development of the application is choosing the type of device that the application will be developed on. Indoor localization and navigation have undergone rapid development due to many studies related to advanced mobile and communication technologies (Subedi et al., 2020). Smart phones are widely used for navigation purposes nowadays and provide kinematic navigation techniques (accelerometers, rotation sensors (gyroscopes) and magnetometers) (M. Jain et al., 2013). As navigation in large and

complex environments can be difficult and requires a lot of computational power, with the evolution of today's smart phones these issues have been addressed adequately (Barberis et al., 2014), resulting in the smart phone being one of the most used devices for navigation.

Platform

Another important aspect of application development is choosing a platform. The main platforms used in the UK for mobile-based applications are Android and iOS with only a 0.32 percent difference in market share (Jkielty, 2019). Both platforms have advantages and disadvantages which need to be considered carefully as both platforms have their relevant limits (White, 2013).

Regarding application availability, Nield and Aitken (2015) consider that iOS wins over Android due to the simpler interface, higher buying of expensive games which leads to more profitability and more recommendations for new applications. Considering software upgrades, Ananiadis's report shows that both platforms are regularly updated but Apple offers a safer and more up-to-date update of its versions (2015). On the contrary, Android is more serious and stable system in terms of the iOS system (Walter, Nield).

An example of an indoor navigation application based on Android is L. C. de Oliveira et al.'s navigation application (2017), which is a mobile augmented reality application using beacon technology. It uses the software library AndAR (Android Augmented Reality) developed with ARToolKit (P. Chen, 2016). Navigation applications with iOS are Cankaya et al.'s (2015) mobile indoor navigation system with augmented reality and Nikander J. et al.'s indoor and outdoor mobile navigation system using map data (2013).

As both platforms have application in indoor navigation and the artefact needs to reach as many users as possible, the issue of whether the mobile application should be a native application, web application or a hybrid web application will be discussed (White, 2013).

Native vs Web vs Hybrid applications

Native applications are apps developed for a specific operating system with a platform-specific programming language - Java for Android and Objective C for iOS (Jobe, 2013). They offer the fastest, most reliable and most responsive experience to users but are restricted to a specific platform (Saccomani, 2019). Native apps are built using an operating system's SDKs and can also interact with all the device features, such as the microphone, camera,

GPS, device storage, etc. Their main disadvantage is that a bigger budget is required to build the app on multiple platforms (Marcak, 2021).

A web app is a website that is designed fluidly, with the option of being viewed on a smart phone in mind. It is accessed via the web browser and will adapt to whichever device the user is viewing them on. As compared to a mobile app, it is not native to a particular system, and doesn't need to be downloaded or installed (Stevens, 2021). Web apps use JavaScript, CSS, and HTML5. Unlike an iOS or Android app, there is no software development kit (SDK) for a developer to work with and the most popular development frameworks are Angular, React, and Vue.js (Saccomani, 2019).

Weeks (2012) states that native applications are preferred over web applications as they have a simple, intuitive, and clear navigation and layouts. Saccomani supports this view by stating that developing a mobile application is more suitable for an application which will be used frequently and has a better overall user experience through a less distracting, more enclosed environment and a direct route through a single tap on users' home screens (2019). Since Android and iOS users have a similar market share (Jkielty, 2019), hybrid web applications will be taken into consideration as they provide cross-platform support.

Hybrid web applications are applications that are neither truly a mobile web app nor a native app. They are written with HTML5, CSS and use techniques such as JavaScript APIs, but run inside a 3rd party native app container (Jobe, 2013). They are built using cross-platform frameworks like React, Ionic, Sencha and Xamarin (Marcak, 2021). However, after being developed, they need to be packaged for different platforms (Sagara Technology Idea Lab, 2019) and the speed of the application depends entirely on the speed of the user's browser (Marcak, 2021). By using plugins, cross platform applications can have full access to the mobile device's features (Griffith, 2021).

Goetz and Zipf (2013) developed a mobile web-based platform for indoor routing with volunteered geographic information (VGI) and OpenStreetMap (OSM). The system has two parts, on server side it uses PostgreSQL (EnterpriseDB, 2022) database with PostGIS (PostGIS, 2020) extension and on the client side the WMS (Open Geospatial Constortium, 2022) data can be rendered and visualized by any GIS application. The application has floor plans and the user can select start (level, staff room) and target (level, staff room) location from drop down box and compute the route.

On the other hand, there are numerous cross platform application developed.

Hansen et al.'s study shows an implementation of an open source positioning and navigation application for indoor navigation which is developed for the major mobile platforms - Android, iPhone and Windows Phone (2013). Another application developed for multiple platforms is Micello (HERE, 2022), which can be used on Android and iOS platforms. MazeMap is a cross platform application which can work in a browser on PC or mobile platforms (MazeMap, 2022).

A major concern when it comes to developing the artefact is that the developer would need experience with the relevant programming languages as it would take time and would be more costly to hire more developers (White, 2013). Additionally, the budget for the development of the application must be appraised and the most used platform of the artefact's audience must be taken into consideration upon deciding on the type of application. As a result, web applications will be considered for the development of the artefact as they offer both cross platform support and can be developed using tools known to the developer.

Local databases

Since there is a variety of databases that can be used for web application development, the low-cost and most relevant ones to the application will be considered.

Realm is an object-oriented database built for offline-first and real-time applications that handles large amounts of data and requires high-performance (Pandya, 2021), making it the best choice for large-sized apps. Although Realm is an open-source platform, its advanced version is paid (Solanki, 2020). In terms of security, it uses different security standards to ensure the data is well protected on the application's platform (Patel, 2021).

Firebase is a popular service provided by Google that supports a real-time NoSQL database - Firestore (QA InfoTech, 2021). It is mainly selected for its offline data change and data synchronization (Solanki, 2020). The advantage of using Firestore is that the data is stored in encrypted format on the server side and decrypted automatically when read by an authorized user, which makes it well protected against security threats and theft (Google Cloud, 2022).

SQLite is an embedded SQL database engine which does not have a separate server process like most other SQL databases. It is a popular database engine choice on memory constrained gadgets like cellphones, PDAs, and MP3 players (Bhosale et al., 2015). SQLite is a free-to-use database connection

library, which supports data encryption with an extension named SQLite Encryption Extension (SEE) (Pandya, 2021).

Realm databases have been used mostly for mobile and web development and have been proven to run queries and sync objects significantly faster than SQLite and others (Shah, 2017). SQLite has proven to be compact, but not widely used for web applications. Firestore is a popular choice for web applications, with a large community providing support to developers worldwide. Additionally, it offers various services including Google service authentication, making it suitable for the artefact.

2.5 Similar applications

In this section already existing applications will be reviewed with their common and distinct features. For this research, Google Maps (Google LLC, 2021), CampusGo (Mapsted, 2021) and Path Guide (Microsoft Corporation, 2017) were chosen as they use various indoor navigation techniques.

Google Maps

Google Maps is an app on Google Play Store with a focus on navigation outdoors (Google LLC, 2021). Additionally, it contains indoor maps with buildings that are a part of the Google Maps directory (Figure 2.1), allowing you to navigate within them effectively (Smith, 2021). Google Maps offers two views for walking navigation: the 2D map and Live View (Google LLC, 2021). However, Live View (Figure 2.5.1) can only be used for outdoor navigation, on devices that are compatible with ARKit (Apple) or ARCore (Android) and the walking area must have good street view coverage (Google, 2021).

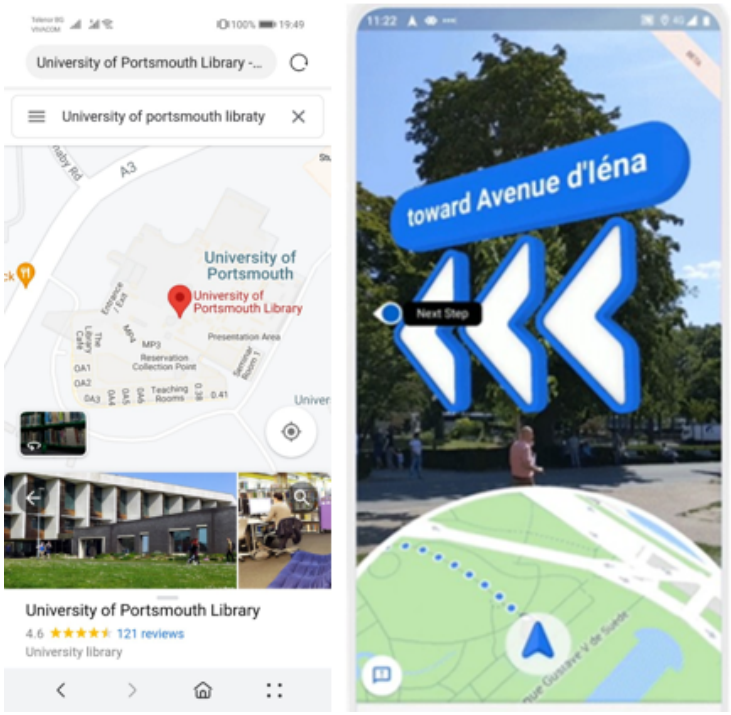

Figure 2.5.1: Google Maps Screenshots (Google LLC, 2021)

A recent new feature announced by Google is the updated Live View AR feature (Figure 2.5.2), which overlays digital guides on top of the real world to provide directions. For example, if the user is in an airport, it will show arrows or markers on the phone's display to the user's gate or ATM (Haselton, 2021).

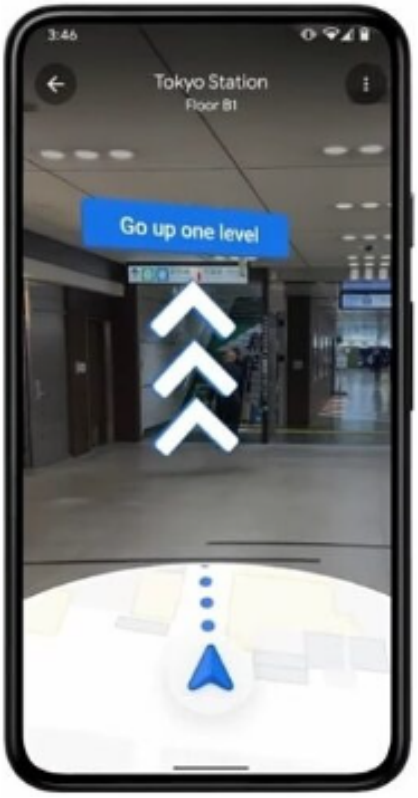

Figure 2.5.2: Google Maps Screenshots (Google LLC, 2021)

CampusGo

CampusGo is an application launched by the University of Windsor in Canada that offers accurate and convenient navigation inside the university campus (Figure 2.5.3). CampusGo was created by Mapsted, the world's leading indoor positioning system developers. The navigation technique implemented is precise to within three metres and requires no WiFi or GPS positioning beacons. Campus accessibility is also promoted through the app - accessible entrances and parking are listed, along with general safety and convenience items such as the location of campus emergency call stations and visitor parking (University of Windsor, 2021). Other advanced accessibility options feature speech-to-text search, nearby environmental alerts and voice navigation (Mapsted, 2021).

The app can be downloaded via Google Play (Android) or App Store (iOS) (Mapsted, 2021). The navigation technology and interactive campus maps are also available through any mobile or desktop web browser.

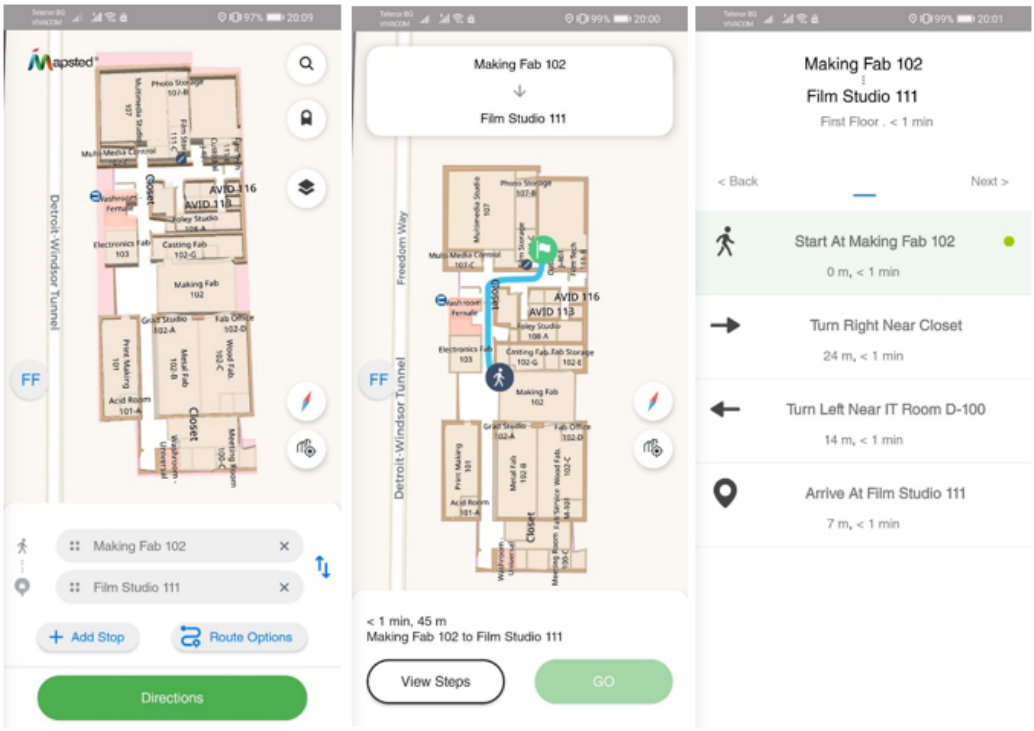

Figure 2.5.3: CampusGo Screenshots (Mapsted, 2021)

Path Guide

Path Guide is an easy-to-use indoor navigation app that relies on user generated data to create its indoor maps by recording the path of earlier travellers (Figure 2.5.4). Compared to Google Maps, the app does not rely on GPS signals and uses the on-board tech such as the barometer and magnetometer of smart phones (Smith, 2020). The app is based on Android and can be installed directly onto a user's smart phone, without the need for indoor maps or for the building to have any special pre-installed hardware (Microsoft, 2017).

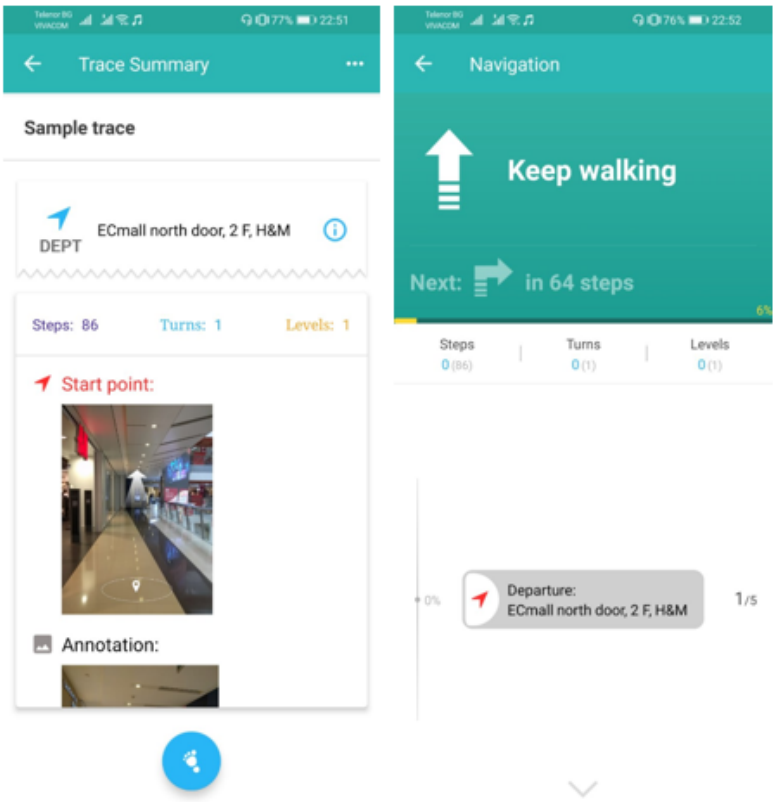

Figure 2.5.4: Path Guide Screenshots (Microsoft Corporation, 2017)

The following table compares the features in common and the distinct features found in the apps (Table 2.5.1).

Table 2.5.1: Key Features of tested applications

Feature	Purpose	Found in
Map	Interactive map which displays the indoor environment	CampusGo
Start/End point	The user provides the start and end point (room/location)	All apps
Instructions	Provides instructions on how to reach the destination	All apps
Shortest route	Shows the shortest route from the user's location towards the destination	CampusGo
Steps/Turns/Levels Counter	Gives instructions based on previous user's path	PathGuide
Walking time	Time taken to reach the destination	CampusGo
Distance	Distance towards the target destination	All apps
Arrows/markers on phone's camera using AR	Helps the user to navigate by using augmented reality	Google Maps
Information about other rooms on the map	Provides information on the indoor environment	CampusGo
Accessible entrances and parking	Helps users with accessibility needs	CampusGo
Nearby event notifications	Push notifications providing information on nearby events	CampusGo
Push notifications	Push notifications showing that the app is still running	CampusGo
Offline Mode	App is available offline	Path Guide
Favourite route	Keeps information on the user's routes saved as favourite	CampusGo
Account feature	The user is able to login with their account and access previous data	Path Guide, Google Maps

After reviewing the apps, it was concluded that using infrastructure-based hardware is not a requirement and different approaches can be taken depending on the artefact's requirements. However, from the research made related to navigation techniques, it was concluded that infrastructureless techniques would not be applicable as environmental features may not always be available inside the building and not all phone models have the smart phone sensors required for navigating indoors. UWB and Wi-Fi approaches were found unsuitable for the application since access to the university Wi-Fi would not be granted due to security measures. Additionally, network equipment can't be installed on site - the service provider in charge of the building would not support installing student-owned software or using hotspots due to interference with the university Wi-Fi.

As a result, low cost infrastructure-based techniques will be implemented for the development of the artefact. Radio Frequency Identification will not be considered as they are expensive and require the installation of bulky equipment in the building. Consequently, Bluetooth Low Energy beacons

will be used as they will not interfere with the network and would have a relatively low cost to achieve the project's objectives.

Design

The design of the applications is comparable to Google's Material Design which is known for the primary, secondary colour and a third colour which is a shade of the first two (Developers, 2020). As the application's main purpose is to help with navigation, more colours would be distracting and would have an impact on its usability.

Performance

The overall performance of the applications was good based on their responsiveness. All applications were responsive to user input and returned to their correct state when interrupted. In terms of loading time, Path Guide took the least, consequently it didn't require any loading information to be displayed on screen. Google Maps and CampusGo provided the relevant loading information to display that background processes were happening while loading the application's features. Upon developing the artefact, the artefact's responsiveness and load time will be taken into consideration to ensure that the overall user experience is good.

2.6 Chapter Summary

In conclusion, indoor navigation can be approached in numerous ways, with the main issues to consider being the cost and complexity of the indoor navigation system to be developed. Since the literature research showed that a navigation system should not be overly complicated in order for the user to get to their target point fast, the system designed will aim to have a simple interface. After discussing the development methods and the navigation techniques, it was concluded that the chosen type of application for the artefact will be a cross platform web application which will use the Bluetooth Low Energy beacon technology and Firestore as a database.

Chapter 3

Methodology and Project management

There are many possible lifecycle models for the development of the application, each with different strengths and weaknesses. In order to develop this project efficiently, the most efficient software development methodologies will be discussed for its development and one suitable for an individual piece of work will be chosen.

3.1 Alternative development models

Waterfall

The waterfall methodology uses a sequential or linear approach to software development. A project following this methodology is broken down into a sequence of tasks and works based on fixed dates, requirements and outcomes. One of the advantages of this model are that the developers implementing this model are not expected to provide status reports as often as with the Agile approach since the project requirements are gathered and understood upfront. Additionally, the processes for the development of the application are predefined with specific cost, design, and time requirements. Consequently, once coding has already begun, design changes can be difficult and expensive to make.

Since the artefact's requirements might change due to working with new technology and the system might need to be redesigned at any point during its development, this approach will be more time consuming and expensive.

Agile

The agile software development model is a flexible, iterative methodology that tackles the planning, design, implementation and testing of tasks in shorter, repeating cycles, called iterations. Unlike the waterfall model, the iterative process adds features one-by-one, providing a working product at the end of each iteration which is its main advantage. However, the system following this model is executed in short time and requires close interaction with its clientele, which makes it unsuitable for the project.

Prototype development model

The prototype development model is a software development model in which a prototype is built, tested, and reworked until an acceptable prototype is achieved. An advantage is that it works best in scenarios where the project's requirements are not known in detail. It is an iterative, trial and error method which takes place between the developer and client. This model was not chosen as it requires continuous customer evaluation and the requirements may change which will increase the project's cost.

3.2 Chosen development model

Iterative and incremental development model

Iterative and incremental development is an approach that separates the product into fully working slices called increments. All new increments build on top of the existing released functionality and each of the project's modules goes through the requirements, design, implementation and testing phases until the complete system is implemented.

Since this model is flexible and allows for developing features based on their priority, it was chosen as a development model. Additionally, its suitable for a not very skilled or trained software team and a project which will require quick release. However, the artefact's development will need to be considered carefully as there will be a need of well defined module interfaces and the time might be a constraint to creating fully working features without bugs.

Compared to the Waterfall model, if a part of the system needs to be redesigned on any of the increments, returning to a previous stage is always possible. Since the module interfaces will need to be well defined, tested and implemented on each increment, it is unlikely that an increment will need to be repeated, in this way reducing the cost.

The project will start with the Initial planning stage, followed by the first and the second prototype implementation which will be conducted in the Iterative element of Figure 3.2.1 (containing the Planning, Requirements, Analysis and Design, Implementation, Testing and user feedback, and Evaluation stages). In the Requirements stage the featureSets from Chapter 4 will be assigned to the Prototype. After the chosen featureSet is implemented, the code is uploaded on Git Hub to keep a backup of the artefact until the final system is deployed.

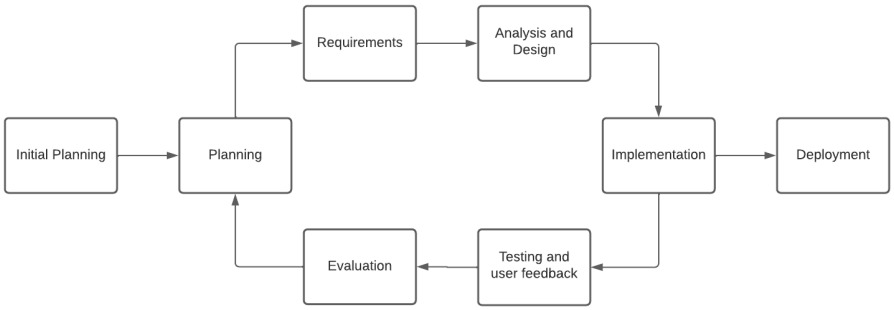

Figure 3.2.1: Iterative and Incremental Development Model

3.3 Project plan

In this section the implementation of the Iterative and Incremental development model (Figure 3.2.1) will be described in more detail and its project plan.

Initial Planning

The initial plan for the development of the whole artefact will be described in this section.

Iteration plan

Each iteration will be split into a Planning, Requirements, Analysis and Design, Implementation, Testing and User Feedback and Evaluation stage as seen below:

Planning

Shows research done beforehand in relation to the implementation of the Prototype's features.

Requirements

The functional and non-functional requirements will be defined from the literature review and by gathering requirements from a questionnaire aimed at the target audience.

The questionnaire was designed using Google Forms as its easily accessible and readable by the participants (Google Forms, 2022). The questions were grouped into subsections based on similar issues, which will help enhance the requirements elicitation gathering. A total of 30 responses were collected. The questionnaire can be found in Appendix C and the analysis of the responses in Chapter 4.

The questions included didn't ask for any personal information or information that would identify individual participants. Each student was handed a questionnaire and filled it for approximately 5 minutes. Their email address was not collected and their response was collected as a number to ensure anonymity.

After the requirements have been defined and determined based on priority, they will be grouped into featureSets based on their priority and will be assigned to each increment.

Analysis, Design and Implementation

In this stage analysis is included with the design of the artefact. The implementation of the Prototype is described and a backup of the system is kept on Git Hub. The next increment is started and the whole process is repeated until the system is completed.

Testing and user feedback

Each feature will be tested by performing user testing to identify bugs and user feedback will be gathered. After successful testing of the prototype the next step will be considered. If the testing stage fails, the debugging tool will be used. Consequently if the test fails again other methods to implement the feature will be considered and the design will be redone.

Evaluation

Each working feature will be evaluated with its purpose, advantages and disadvantages for the user.

Deployment

The final system is deployed on Git Hub and a backup is kept.

3.4 Development Tools

The Firestore Database and IoT devices - Low-Energy Bluetooth beacons will be used to build the artefact, as evaluated from the Literature review. The programming languages which are planned to be used are JavaScript, HTML and CSS as they are familiar to the developer and suitable for a web application.

3.5 Time management

As time management, the original Gantt chart from the PID will be used (Appendix A), which will provide information on the development process of the project. Each of the project's milestones with their corresponding time will be taken into consideration and the implementation of the artefact will be shown.

3.6 Chapter Summary

This section has defined the alternative development models with their advantages and disadvantages for this project. The chosen methodology has been justified and the project plan for implementing it has been described, along with the development tools which are planned to be used for the project. The technique used for time management has been outlined to be the same as in the PID.

Chapter 4

Requirements and specification

4.1 Requirements elicitation methods

Human participation is mandatory for the successful development of the application and to ensure that it meets industry standards. The main methods of gathering requirements were researching literature on the topic, reviewing similar applications and asking the target audience questions with a questionnaire to receive feedback from a large number of participants and avoid face to face meetings which pose a risk to the project development due to the covid pandemic.

4.2 Requirements from research

In the literature review, similar navigation applications were compared in order to see the features they have in common and their overall performance and design.

The following features were taken note of upon reviewing similar applications:

- Map - Interactive map which displays the indoor environment
- Start/End point - Start location and end location
- Instructions - Instructions on how to reach the destination

  CampusGo example instructions include:

  - Start at Location
  - Turn right at Location
  - Turn left at Location
  - Arrive at Location

  Path Guide example instructions include:

  - Turn right
  - Turn left
  - Keep walking

- Shortest route - Shows the shortest route from the user location towards the destination
- Steps/Turns/Levels counter - Gives instructions based on previous user's path
- Walking time - Time taken to reach the destination, measured in minutes
- Distance - Distance towards the target destination, measured in ft or metres
- Arrows/markers on phone's camera using AR - Helps the user navigate by using augmented reality
- Information about other rooms on the map - Provides information on the indoor environment
- Accessible entrances and parking - Helps users with accessibility needs
- Nearby event notifications - Push notifications providing information on nearby events
- Push notifications - Push notifications showing the app is still running in the background
- Offline feature - The application is available offline
- Favourite route - Keeps information on the user's routes saved as favourite
- Account feature - The user is able to login with their account and access previous data
- Overly simple interface - The user should be able to reach their target destination quickly without being obstructed by the interface

- Design - The design of the application should conform to the Material Design guidelines
- Responsiveness - The app should be responsive at any time and load fast

Analysis and results

After the analysis of these features, it was determined that event notifications won't be applicable for the navigation in Buckingham building scenario as there are currently no events held there. The steps, turns and levels feature was not chosen as it lacks real time indoor environment information and would not be applicable for users with accessibility issues. The accessible entrances and parking are not suitable as Buckingham building has no parking spaces and the main entrance is step free. Instead, stairlifts on the first floor and lifts will be considered. These accessibility features were taken note of from Buckingham building's accessibility web page (AccessAble, 2022).

4.3 Requirements from questionnaire

Questions

The questionnaire targets the audience of students of all age groups who are interested in the development of an indoor navigation application for Buckingham building. Its purpose was to determine the platform which the application will use by asking a question on the student's most used device, phone operating system and its version. The question whether students prefer using an application or a website was also considered, in order to determine the preferred type of application. Students were also asked questions on the indoor applications and other navigation methods they have used in order to consider additional implementation tools for the artefact and the features they would find most useful. To conclude the questionnaire, students were queried about issues or concerns they might have about the development of the application. A copy of the questionnaire can be found in appendix C.

Analysis and results

A total of 30 responses were collected for the study, with the same number of females and males. The questionnaire took responses from students in each

Table 4.3.1: Percentages of participants with the year they are in

Students Year-in	Percentage (%)
First Year	23.3
Second Year	16.7
Third Year (Placement)	0
Final Year	23.3
Masters	80

year and from all age groups, with the majority between 21 - 25 age range (Figure 4.3.1) and doing a Masters degree (Table 4.3.1).

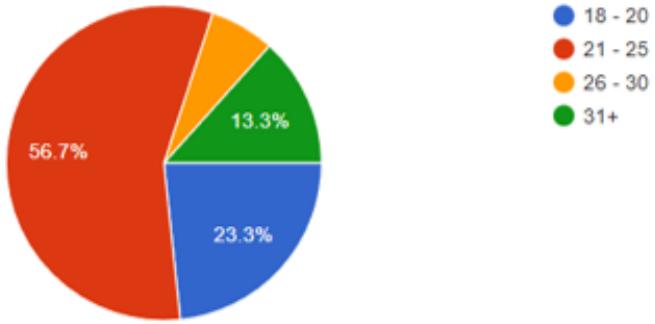

Figure 4.3.1: Age Range Results

The responses from the questionnaire showed that smart phones are the most used device for university students which will result in the smart phone to be prioritized as a device for the application (Figure 4.3.2).

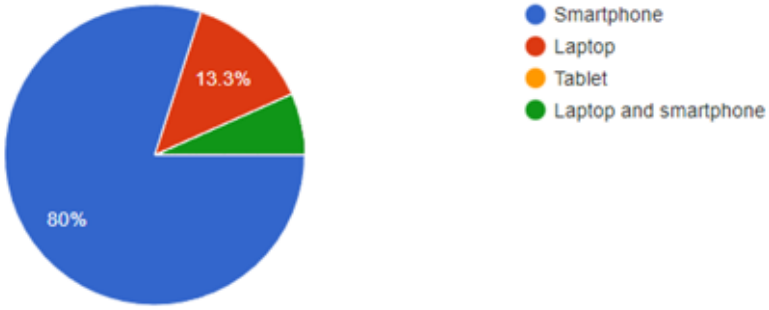

Figure 4.3.2: Students Most Used Device Results

From the literature review and the questionnaire, it is concluded that the users' percentage using these platforms is close and a cross platform application would be most suitable (Figure 4.3.3).

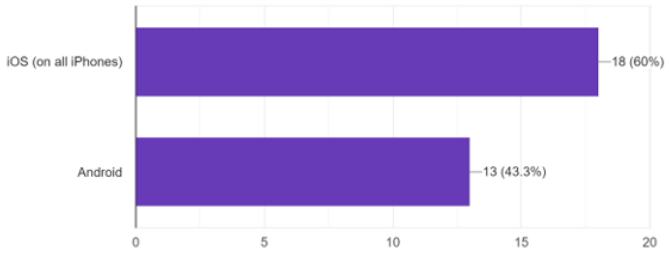

Figure 4.3.3: Most used Operating System

The questionnaire showed that the most used Android version is 10, while the Apple version is 15. This will result in the application being tested on the before mentioned OS versions.

The majority of students preferred using applications over websites (Figure 4.3.4).

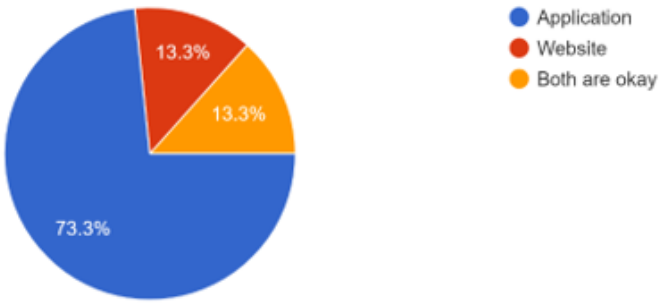

Figure 4.3.4: Application vs Website Results

Since the literature review stated that knowledge of the programming languages is mandatory and websites are built with languages that the developer is familiar with, a cross-platform website with application-like features will be considered for the development of the artefact.

To the question asking about the indoor navigation tools students have used, males responded that they have used Google Maps and mainly asked for directions, while females also used kiosks, for example the University of Portsmouth Library ones to navigate. Most of the students from both genders had not used an indoor navigation application. From this question it was

estimated that a kiosk can't be implemented as an artefact as the equipment and installation has a high cost.

Application features were discussed and results showed that the majority of students were in favour of the following features - a map, the ability to see where you are on the map, a start and end point with instructions. As a result, these features were found essential for the development of the application. Additional information can be displayed about rooms on the floor (Figure 4.3.5).

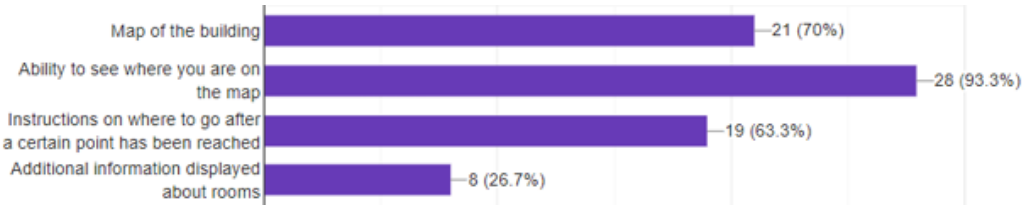

Figure 4.3.5: Application Features

The overall feedback on an indoor navigation application showed that it should be with a simple interface, without too many features and text as students would like to reach their goal fast.

The results from the questionnaire and the literature review revealed that a navigation application would also need to be secure and accessible.

4.4 Application Accessibility

Accessibility features considered for the application include:

- Appropriate level of contrast between colour combinations in order for the text to be visible
- Text alternatives for photos, videos or maps
- Change between light and dark mode

4.5 Security

As security issues can't be fully considered due to the constantly emerging new threats, any personally identifiable information will be avoided to be stored.

- A possible feature is to login the user with their Google account in order to connect to their University of Portsmouth timetable. The security will be managed by Google.
- As not all students may have access to their Google Account, a feature which should be considered is the option to login as a Guest.

4.6 Ethical and Legal Requirements

The system will be storing user information which will require the storage of the data to be handled with care and respect to the legal guidelines such as GDPR (Irwin, 2018). Requirements related to this include:

- Any personally identifiable information collected will have purpose for the application
- Personally identifiable information will be avoided to be stored
- Data should be as accurate as possible

Data will be encrypted by implementing the HTTPS protocol on the artefact. This will be done by making the website hosted to ensure data integrity and confidentiality between the user's computer and the website (Google Developers, 2022).

4.7 Requirements

This section describes the functional and non functional requirements gathered from the requirements elicitation methods.

Organisation and prioritisation of requirements - MoSCoW prioritization technique

The MoSCoW prioritization technique will be used in order to group each requirement into a featureSet (Table 4.7.1). It states that each requirement is assigned one of the following priority levels (ProjectSmart, 2022):

Table 4.7.1: MoSCoW prioritization technique

Requirement	Description
M: Must have	Necessary requirement, without it the product cannot be considered a success
S: Should have	High priority requirement that should be included if possible
C: Could have	Desirable requirement, but not necessary
W: Would like to have	Requirement that won't be implemented in the current release, but could be considered in future releases

The following functional and non-functional requirements were elicitated from the literature review and the questionnaire (Table 4.7.2, Table 4.7.3).

Table 4.7.2: Functional requirements

Requirement Number	Requirement	Purpose	Priority
F-1	Database	The system must store application data	Must have
F-2	Guest Login Feature	The user is able to login as a Guest	Must have
F-3	Map	Map which displays the indoor environment and accessibility features such as lifts. Provides information on the rooms	Must have
F-4	Locating the user on the map/Start point	The user must be able to start from his location or select a location	Must have
F-5	End point/destination	The user provides their destination (room/location)	Must have
F-6	Instructions	Provides instructions on how to reach the destination	Must have
F-7	Shortest route	The map should show the shortest route towards the room	Should have
F-8	Accessibility information on the map	Information about the accessibility features on the current floor	Should have
F-9	Push notifications	Push notifications notifying that the application is still running in the background	Should have
F-10	App is available offline	The application will be responsive offline	Should have
F-11	App is installable on screen	The user can install the app on the device's screen	Should have
F-12	Google Account Login feature	The user is able to login with their account, access previous data, their UOP timetable and save favourite routes	Could have
F-13	Favourite Route	The user saves the current route as favourite	Could have
F-14	Walking time	Time taken to reach the destination	Would like to have
F-15	Distance	Distance towards the target destination	Would like to have
F-16	Interactive Map	The map could be used to select a start point or a destination	Would like to have
F-17	Additional information displayed about the rooms	Information describing each room and staff member	Would like to have

The walking time, distance, interactive map and the additional information displayed about rooms features were decided not to be implemented in the

current version of the project as they are add-ons to the navigation features and the main functional features will be considered first.

Table 4.7.3: Non-Functional Requirements

Requirement Number	Requirement	Purpose	Priority
NF-1	IOS and Android Compatibility	The application can be used on the most recent version of iOS (15) and Android (10)	Must have
NF-2	Smartphone compatibility	The application can be used on smartphones	Must have
NF-3	Responsiveness	The application can return to its correct state if interrupted	Must have
NF-4	The system should not store any personal data	Personal data is not stored to avoid data exposure	Must have
NF-5	Data encryption	Dummy data is encrypted to enhance security	Must have
NF-6	Interface simplicity	Application will have an overall improved user experience	Must have
NF-7	Material design with appropriate level of contrast between colour combinations	User interface will conform to the best practices of user interface design	Should have
NF-8	Text alternatives	Pictures and the map have a text when they are not loaded on the page	Should have
NF-9	Change between light and dark mode	The light emitted by the device's screen is reduced while maintaining the minimum colour contrast ratios required for readability	Should have

FeatureSets

The functional system requirements are grouped based on priority level and functionality into featureSets (Table 4.7.4). Only functional requirements are considered as the non-functional requirements should be considered throughout all stages of development.

Table 4.7.4: FeatureSets

Featureset	Requirement Number	Requirement
Core Featureset	F-1, F-9	Database, Push notifications
Navigation	F-3, F-4, F-5, F-6, F-7, F-8	Map, Locating the user on the map/Start point, End/destination, Instructions, Shortest route, Accessibility information on the map
Application welcome page	F-2, F-10, F-11, F-12, F-13	Guest Login Feature, App is available offline, App is installable on screen, Google Account Login feature, Favourite Route
Navigation Add-ons	F-14, F-15, F-16, F-17	Walking time, Distance, Interactive Map, Additional information displayed about the rooms

4.8 Chapter Summary

The research from the literature review and the data collected from the questionnaire allowed for analysis of the requirements for the application. Requirements were elicitated and considered along with accessibility, security, ethical, legal requirements and grouped into featureSets. The featureSets defined above will be analyzed throughout the Design chapter and will provide order for the development of the application in the Prototype and Testing chapters.

Chapter 5

Design

5.1 Use Case

The use case diagram on Figure 5.1.1 shows the expected system to be available to students and to receive the properties needed for navigation from the IoT device.

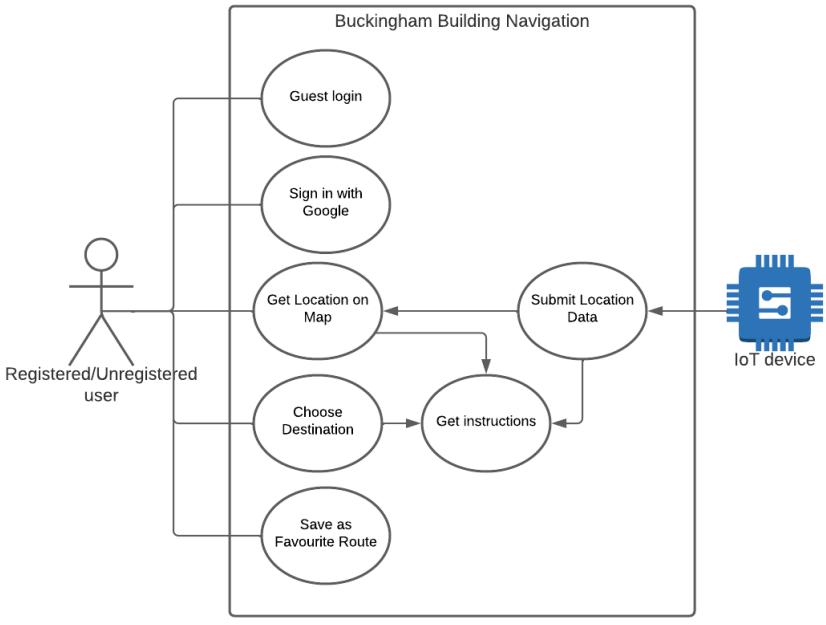

Figure 5.1.1: Use Case Diagram for the proposed system

5.2 System Architecture

To choose the architectural model of the artefact, the project methodology and the requirements were taken into consideration.

Chosen Architectural Model

The chosen architectural model for the artefact is the Client-Server model. It was chosen for its suitability for a small system and functionality which is available to all users. The artefact is expected to be comprised of one server, a database, an IoT device (Bluetooth Low Energy beacon) which will communicate with the server through the Web Bluetooth API to locate the user and a Google API for the Google Sign in if chosen to be implemented (Figure 5.2.1).

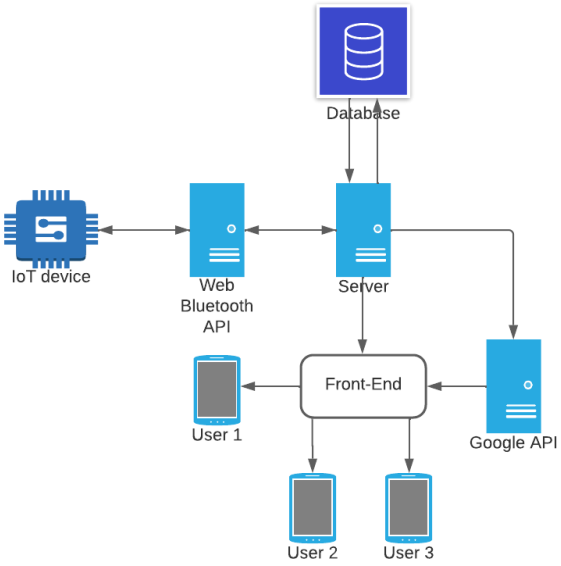

Figure 5.2.1: Client-Server Architecture

In the future, if the artefact becomes popular and requires more than one server, the Client-Server architectural pattern can be replaced by a Object-broker architecture.

Alternative Architectural Models

Other architectural models which were considered are the Object-broker and the Pipe and filter.

- Object-broker - The object-broker architectural pattern was regarded to be inefficient for the artefact as it can have multiple brokers and multiple servers. Consequently, it is appropriate for a large system with many users.
- Pipe and filter - This architectural pattern was not considered as it is one directional and suitable for applications that require a defined series of independent computations to be performed on data.

5.3 Database Design

The project will use a relational database as the entities will require links to ensure data consistency (Figure 5.3.1). As outlined in the literature review in Chapter 2, Firestore database was chosen for the artefact implementation as it provides a wide range of services supported by Google, including offline data persistence and authentication. However, Firestore is a NoSQL document database which differs in the way it describes relationships between data objects from SQL databases by using documents and collections to store data. As a result, the implementation of the links between the tables for the artefact would need to be considered carefully.

The Room and the StartLocation table would be connected to the Route table in order to store the Route information for each student in the Student table. The routeId in the Student table will represent the Favourite route for the student. In future work, as more functionality is expected to be needed, the student's email and name can be added from their Google Account. As directions are expected to be added to the artefact, they will be stored inside the Directions table.

Since each room can have multiple staff members, the relationship between the Room table and the StaffMember table was noted as one to many. The InOfficeTime was added to the StaffMember table in order to notify the Student/Guest user if the staff member is not in their office. By checking the inOfficeTime startTime and endTime against the guestLoginTime and the studentLoginTime, this feature is expected to provide more accurate information about the searched staff member.

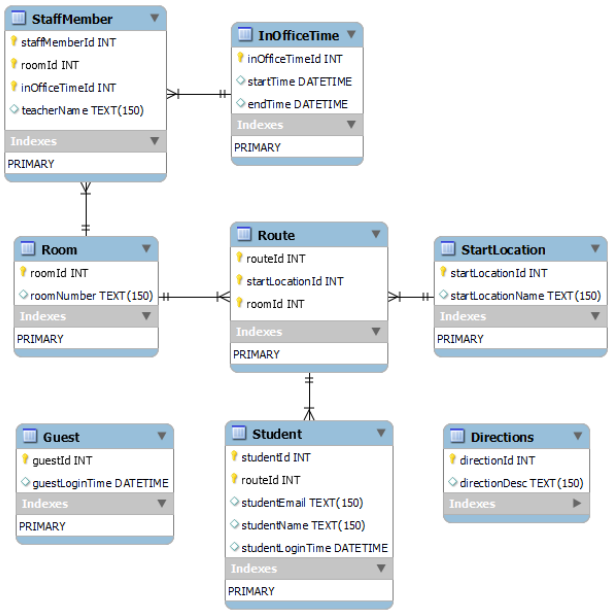

Figure 5.3.1: Database Design

5.4 Graphical User Interface (GUI) Style

The style of the UI was chosen upon reviewing the applications from the Literature review (Chapter 2). The main design concepts were determined in this section - the font and colour schemes.

Roboto Bold was decided on as a primary font as it allows letters to take up as much space as they need, improving the readability of the user (Figure 5.4.1). It is widely used in both Android and iOS mobile apps.

Roboto Bold AaBbCc012345

Figure 5.4.1: Font Scheme

The colours chosen provide an appropriate level of contrast between the colour combinations which will make both the elements on the UI and the text easily visible. Depending on whether the Dark mode colour scheme is chosen to be implemented, it will be similarly styled to the Light one, with the only difference being the background colour and the buttons (Figure 5.4.2).

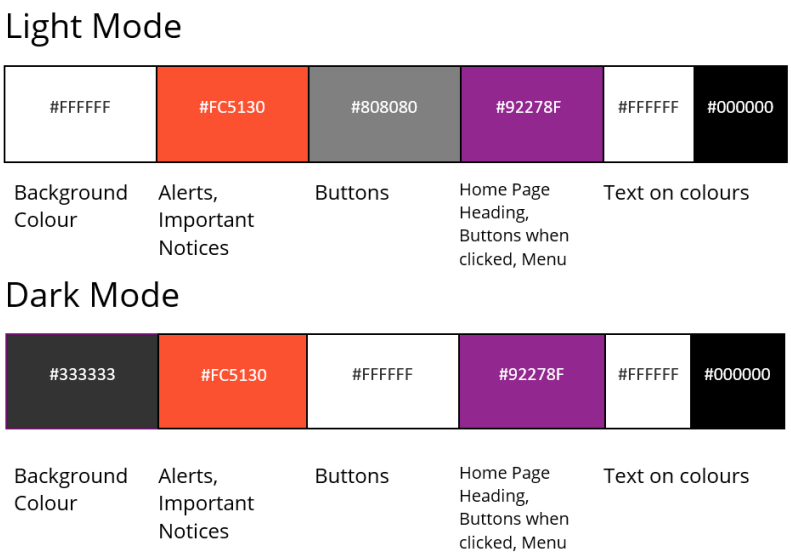

Figure 5.4.2: Colour Scheme

5.5 Graphical User Interface UI Mockups

To ensure simplicity of the user interface, the structure of the application uses two pages and a menu. As a result less information is shown at once and the user can access more information when desired without being overwhelmed.

Home Page

For each page, a UI Mockup has been created for both an Android and an IOS smart phone device. The Home page (Figure 5.5.1) will include:

- The University of Portsmouth logo on the top left
- Title "Buckingham building navigation" on the top right
- Google Sign in option which will grant the application access to the user's timetable. Currently, a prototype version of the Google Sign in option will be considered during the implementation.
- Information Icon next to the Google Sign in option which will explain its functionality with a short description when the user clicks on it.
- Guest Login since students may not have access to their Google Account

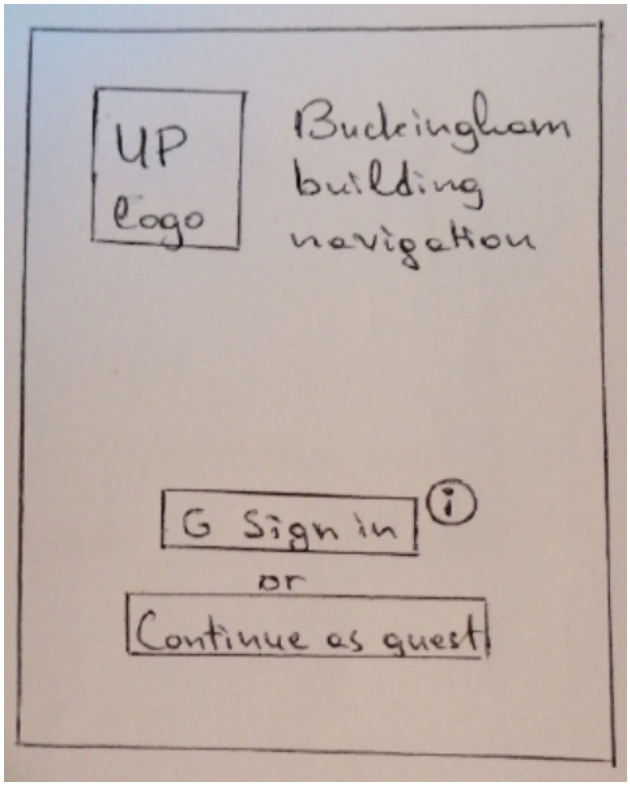

Figure 5.5.1: Home page

Navigation Page

On signing in with Google or continuing as a Guest the navigation page will be displayed (Figure 5.5.2). This screen consists of:

- Main menu which is located at the top left part of the map.
- Favourite Routes feature depending on whether it is chosen to be implemented. It will be located at the top right, if the user is not signed in, the Favourite Routes option will not be visible as it will not be possible to save user information.
- Floor Map which will be displayed in the background with the features implemented in Chapter 6. The user must be able to see his location on the map or select a start point.
- "Choose Destination" dropdown list with the staff offices.
- Back button which will take the user to the Home page.

- Start button which will be initially greyed out until the user has selected a destination and the start point is clear.

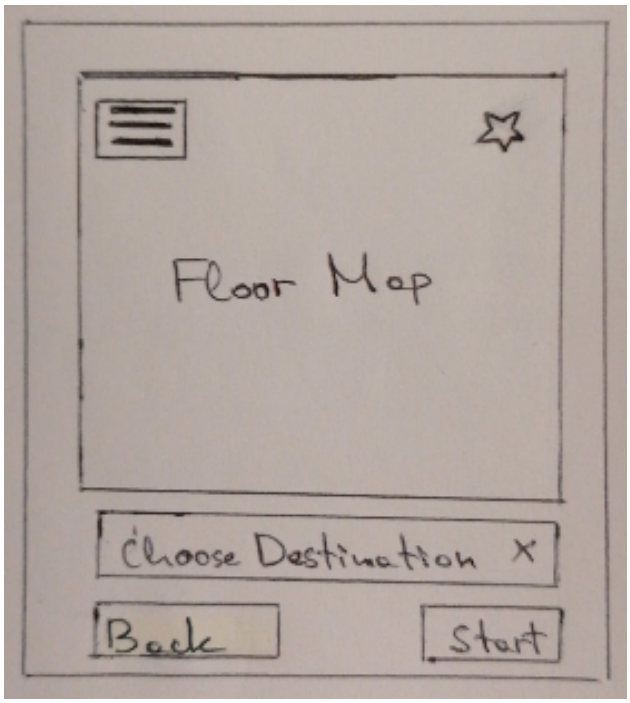

Figure 5.5.2: User Signed in Navigation Page (Choosing a destination)

If both the start and end point are valid, the shortest route towards the destination will become visible on the map if decided to be implemented in Chapter 6. Instructions on how to reach the destination will be given as seen on Figure 5.5.3. Example instructions chosen for the application include:

- Go to the left towards the stairs
- Go up one level
- Go past Staff Room A
- Go past Room B
- Destination reached!

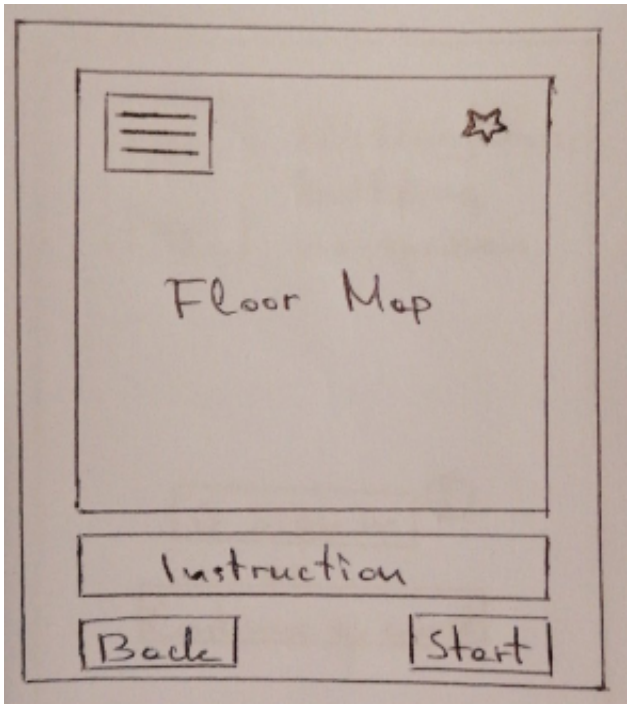

Figure 5.5.3: User Signed in Navigation Page (Receiving instructions)

Main menu

When the user is signed in with his Google account, he will be able to see his Google account name and email in the application's main menu (Figure 5.5.4). The following buttons were chosen to be added to this screen:

- Settings button - Depending on whether the light and dark mode is implemented, the Settings button will be added to the menu. On clicking the button, a window will open which will give the user the option to switch in between the two modes (Figure 5.5.5). In the future, if more functionality is added, it will be added to the Settings window.
- About button will provide basic information on the application.
- Sign out button which will log out the user and take him back to the Home page.

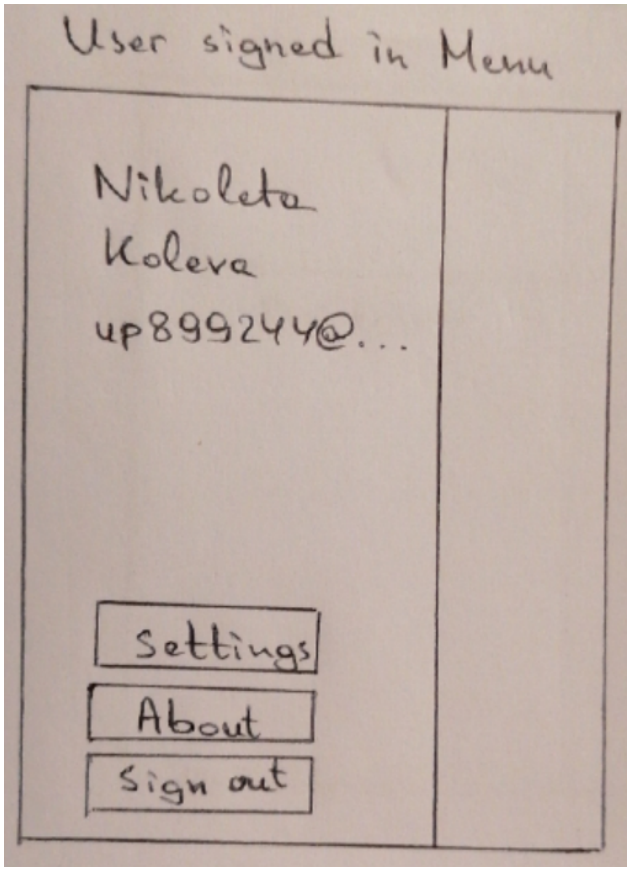

Figure 5.5.4: Main Menu User Signed In

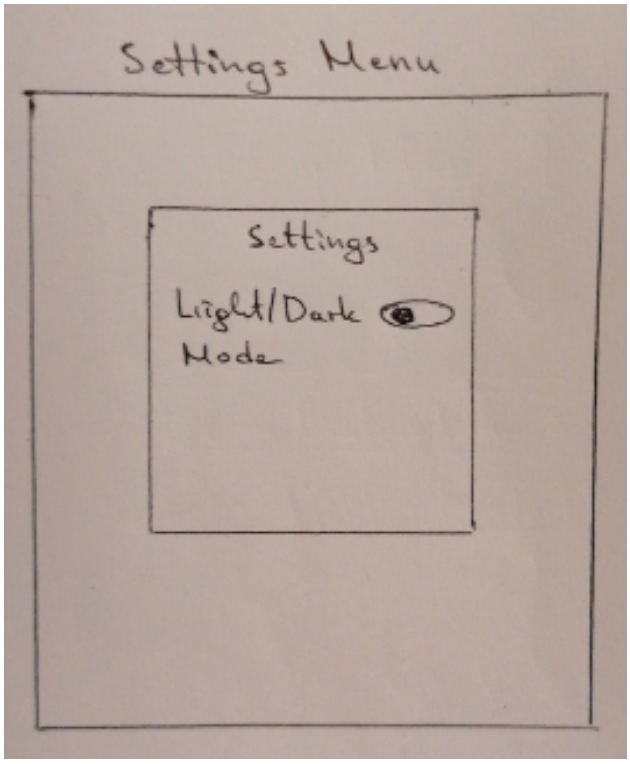

Figure 5.5.5: Settings Menu

If the user is logged in as Guest, he will see "Guest" on the menu's top, which will indicate his current state (Figure 5.5.6). The "Sign out" button will be replaced by a "Go back" button which will take the user to the Home page when clicked on.

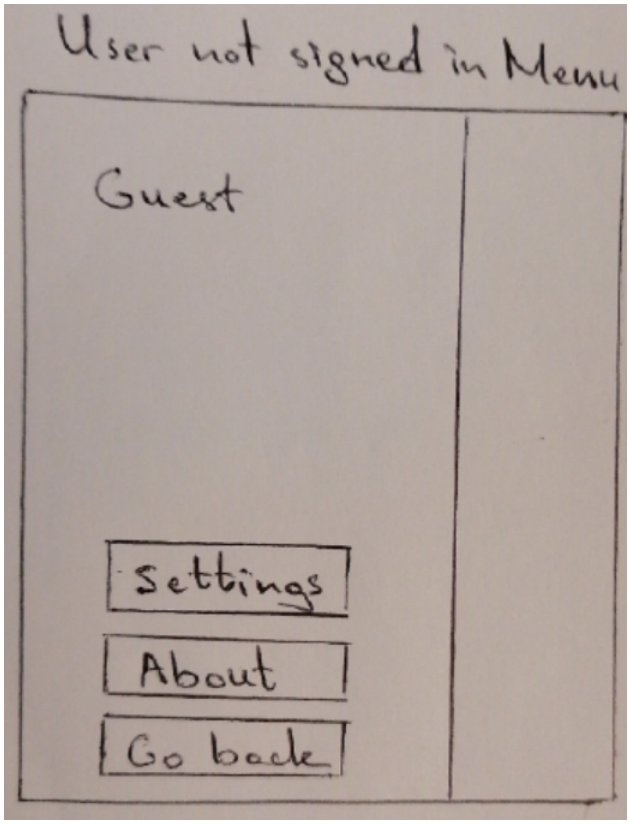

Figure 5.5.6: Main Menu User Not Signed In

5.6 Chapter Summary

The design choices in this section provide insight into the application's architecture and the UI design which will determine how the initial implementation of the artefact will be conducted.

Chapter 6

First Prototype

6.1 Initial Planning

The artefact will be developed in two prototypes, each consisting of the stages described in Chapter 3. The final system will be deployed after the implementation of the second prototype.

6.2 Planning

As evidenced from the Literature review, its possible to build a web application for indoor navigation. The results from the requirements gathering technique showed that a native application is the preferred device for the user.

The chosen type of application for the artefact was a progressive web app as it combines the features of a native application and a web app, and can be built with the languages from Chapter 3. Node.js will be used in the project since it can handle a large amount of simultaneous connections in a non-blocking manner, which would be suitable for the app if the project gains popularity in the future.

6.3 Requirements

The first prototype will focus on the non functional requirements of the system and creating the Application Welcome Page FeatureSet with requirement numbers F-2, F-10, F-11, F-12 and F-13.

6.4 Analysis, Design and Implementation

Back-end Structure

File hierarchy

The file hierarchy was implemented first in order to lay out the backend structure. Firebase Hosting was set up which provides fast, secure, and reliable way for the artefact to host its static assets (HTML, CSS, JavaScript, media files, etc.). Additionally, by hosting the artefact, HTTPS was implemented to increase security of data transfer between the user and the application. The project's structure of the First Prototype is described in Appendix D.

Front-end Structure

Home Page and App Installation

The Home Page was modified in order for the logo and the title of the application to have a suitable text size and visibility for the user. The logo was provided an alternative text in case its not displayed on the user's device. It was chosen for the Information tip from the Design chapter not to be included as it had a fixed position and was not displayed properly on devices of different screen sizes (Figure 6.4.1):

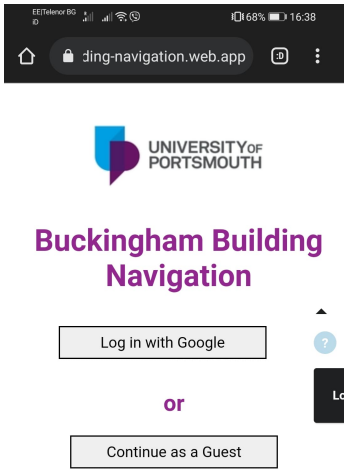

Figure 6.4.1: Tooltip Design issues

The Important Alert colour scheme was not implemented in this increment as there were no important alerts to be added in the application, but the Light and Dark Mode were added. For both themes the primary goal was to have an appropriate level of contrast between the colour schemes and the text, by following Google's Material Design guidelines mentioned in the Design chapter.

The Guest login feature was implemented, with the user being able to login as a Guest. It was chosen for the app to store its static files in the browser's cache in order to improve its responsiveness (Figure 6.4.2). This resulted in the web page loading the files faster.

```
self.addEventListener( type: 'install', listener: function (event : Event ) {
  console.log('SW Installed');
  event.waitUntil(
    caches.open( cacheName: 'static')
      .then(function (cache : Cache ) {
        cache.add('/');
         cache.addAll( requests: [
             '/',
             '/index.html',
             '/navigationPage.html',
             '/src/css/homePage.css',
             '/src/css/main.css',
             '/src/css/navigationPage.css',
             '/src/css/themes.css',
             '/src/js/app.js',
             '/src/js/homePage.js',
             '/src/js/navigationPage.js'
         ]);
      })
  );
});
```

Figure 6.4.2: Static files stored in browser's cache

A Service (Web) Worker was added to make the app installable on the user's screen and available offline (Figure 6.4.3).

```
/* Service Worker */
if ('serviceWorker' in navigator) {
  navigator.serviceWorker.register( scriptURL: '/server.js')
    .then(function() {
      console.log('SW registered');
    });
}
```

Figure 6.4.3: Service Worker

Since the application required the service worker to be installed to store the static files in the cache, it was decided to add a cache response to ensure the files are successfully added (Figure 6.4.4).

```
self.addEventListener( type: 'activate', listener: function () {
  console.log('SW Activated');
});

self.addEventListener( type: 'fetch', listener: function(event : Event ) {
  event.respondWith(
    caches.match(event.request)
      .then(function(res : Response | undefined ) {
        if (res) {
          return res;
        } else {
          return fetch(event.request);
        }
      })
  );
});
```

Figure 6.4.4: Check Cache response

The installation of the app was handled by the app manifest which gave the web app the properties of a progressive web application (Figure 6.4.5). The icons for the app on the Figure have different sizes in order to be displayed in consistency with the device settings.

```
{
  "short_name": "Buckingham building navigation",
  "start_url": ".",
  "display": "standalone",
  "orientation": "portrait",
  "background_color": "#ccc",
  "theme_color": "#5FAAE5",
  "icons": [
    {
      "src": "/src/images/app-icon-96x96.png",
      "sizes": "96x96",
      "type": "image/png"
    },
    {
      "src": "/src/images/app-icon-144x144.png",
      "sizes": "144x144",
      "type": "image/png"
    },
    {
      "src": "/src/images/app-icon-256x256.png",
      "sizes": "256x256",
      "type": "image/png"
    },
    {
      "src": "/src/images/app-icon-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Figure 6.4.5: App Manifest

The app's icon design, which can be found in the First Prototype screenshots was chosen based on popular app icons such as Goodreads, Facebook and Pandora (Upland, 2022). As they use the first letter of the company and avoid the usage of many colours and images in the app icon, it was decided to make the artefact's icon similar. This was done by keeping the icon focused and simple, including only one letter from the application's name and the purple colour (92278F) from the GUI Style in Chapter 5.

Navigation Page

The Google Sign in feature was implemented unsuccessfully as it required a database.

The main front-end features of the navigation page were added - Menu, the star icon to add a favourite route, the outline for the map to be added and the buttons required for the user to navigate easily in the application (Back/Sign out). They will be changed to match the Design in the second increment.

Navigation Page Menu

The application's menu was modified to include the Favourite route and the About button which gives more information on the application. The Favourite routes button is currently unresponsive as its functionality is required by the Google Sign in feature. The Settings button was not included as the Dark and Light theme were implemented on start-up of the application, depending on the user settings. The Terms of service and the Privacy policy were not included as the application will not be published online at this stage.

6.5 First Prototype screenshots

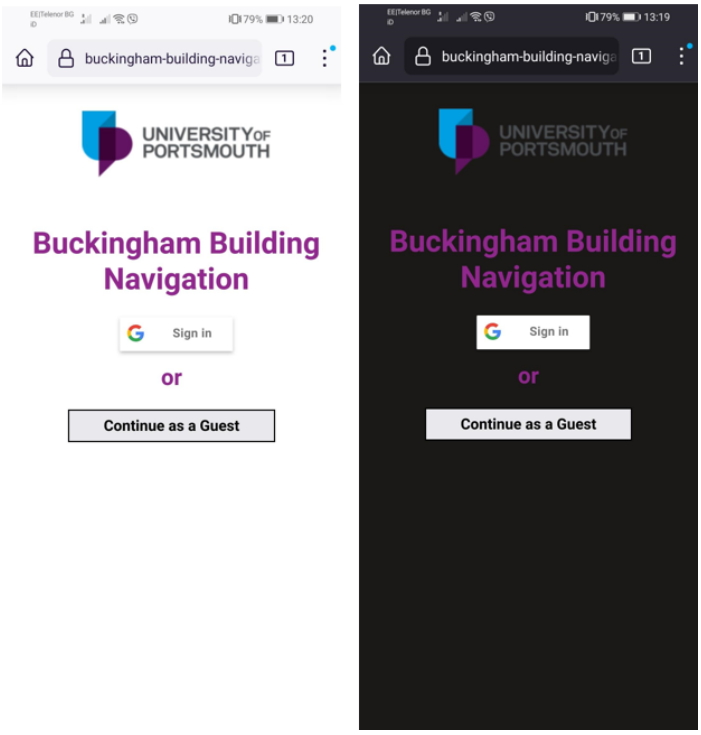

Figure 6.5.1: Home Page

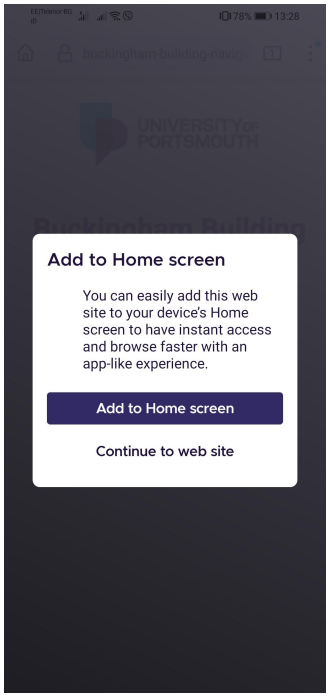

Figure 6.5.2: Add to Home screen message

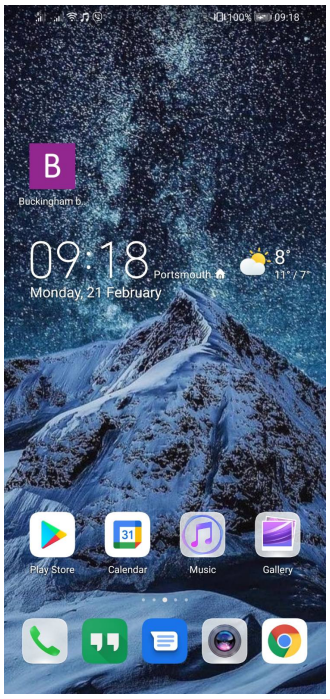

Figure 6.5.3: App installed on user screen

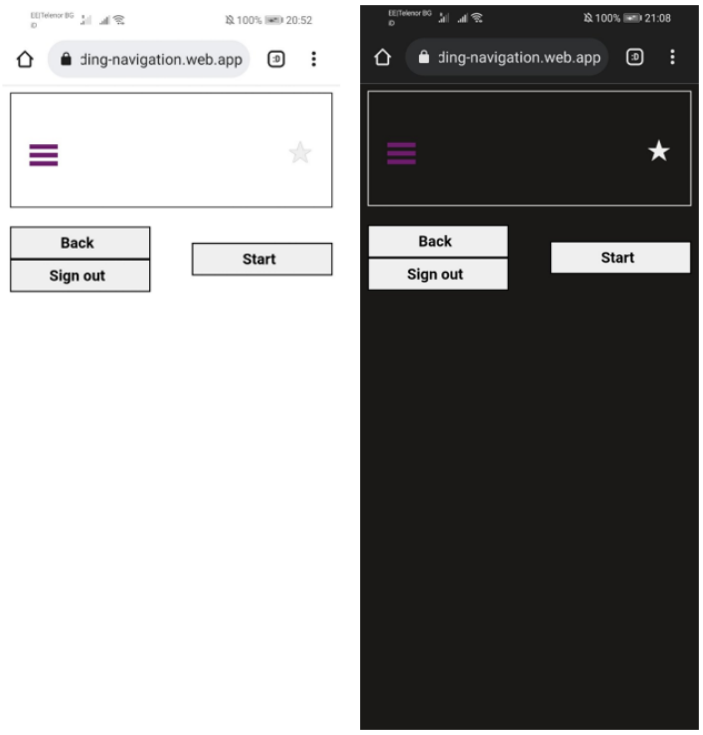

Figure 6.5.4: Navigation Page

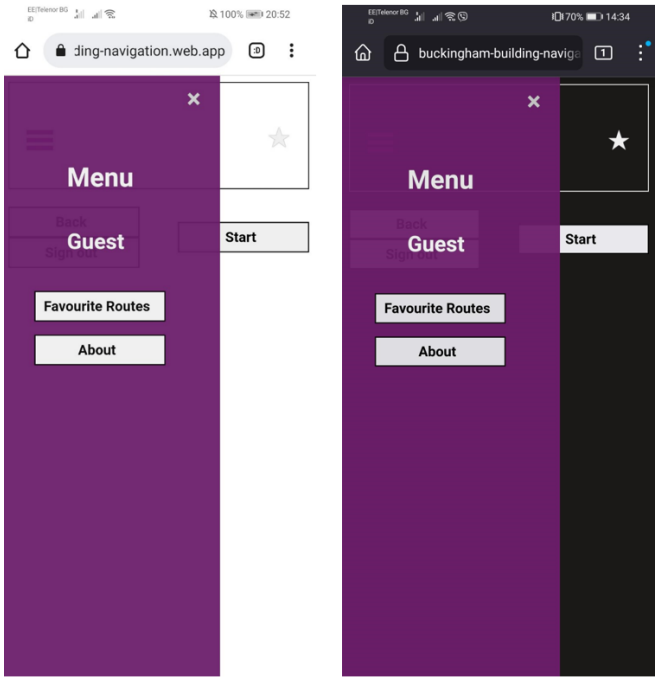

Figure 6.5.5: Menu

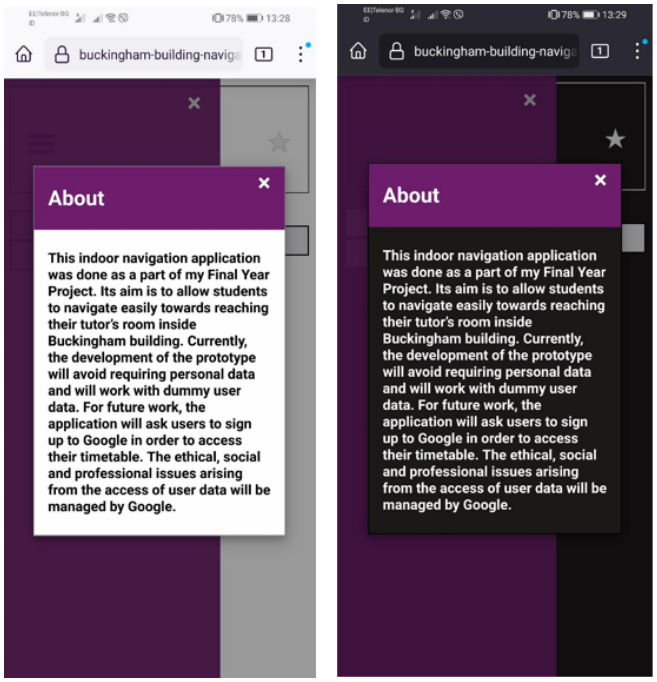

Figure 6.5.6: About Menu

6.6 Chapter Summary

In the first prototype the front end was prioritised to satisfy the non functional requirements and user requirement numbers F-2, F-10, F-11, F-12 and F-13. The Guest Login feature (F-2), the application's offline availability (F-10) and the ability to add the app to the user's home screen (F-11) were implemented. The Google Sign in (F-12) and the Favourite routes feature (F-13) will not be implemented in the final prototype of the project as they have a lower priority in the table of the project's functional requirements.

Chapter 7

Second Prototype

7.1 Planning

The maps required to lead the user to their chosen staff office are the maps of the ground floor and the first floor (Figure 7.1.1, Figure 7.1.2) since the student is expected to start navigating from the entrance on the ground floor. Although Buckingham building has five floors and a roof floor, the application will be considering the ground and the first one due to the high device cost and the limited number of equipment.

Map navigation

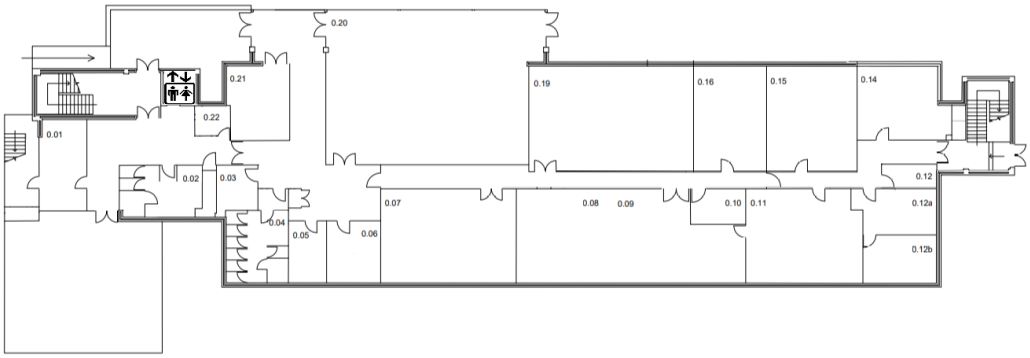

Figure 7.1.1: Ground Floor Map

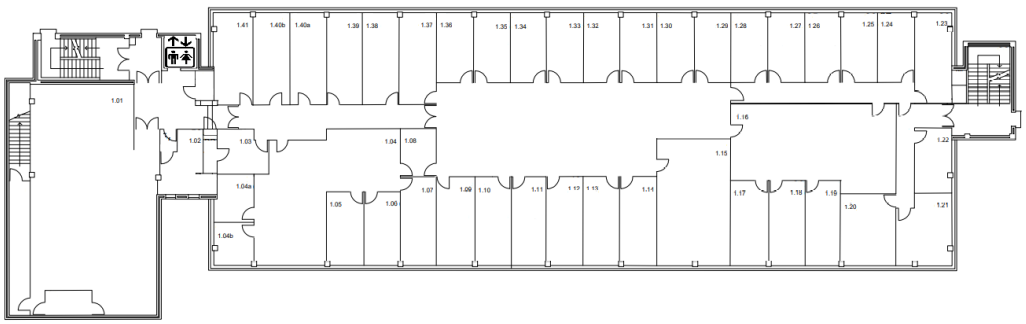

Figure 7.1.2: First Floor Map

Localization Method

A method that was considered is trilateration - determining a position by knowing the distance of the user from at least 3 known points. This localization method was found unsuitable as Buckingham building has narrow corridors and the beacons would need to be placed inside rooms to get an accurate position resulting in the signal being blocked by the walls.

The technique which was decided to be implemented was proximity based localization based on the Received Signal Strength Indicator (RSSI) broadcasted by the beacons. Each beacon will be mounted on the ceiling of the corridors in order to avoid getting the signal blocked by walls or objects, ensuring the functionality of the application is fulfilled.

Internet of things device

The Bluetooth Low Energy device beacon model chosen for use is the JINOU Bluetooth BLE 5.0 Programmable Beacon (Figure 7.1.3) provided by BeaconZone Ltd. (BeaconZone, 2022). It was considered fitting for this project as it was usable for both Android and iOS, has good connecting range (30m) and an average battery life of 12 months. Additionally, both the iBeacon and the Eddystone protocol were transmitted by the beacon which gave more options for programming the application.

Figure 7.1.3: Beacon Model

As both protocols were researched, it was decided that the Eddystone protocol will not be used for the artefact since it is no longer supported by Google due to the low number of users and the poor user experience (Android Developers, 2018). The iBeacon protocol was chosen for implementation for its compatibility with progressive web applications and wide range of services offered.

Beacon Locations

In regard to the equipment installation, the planned installation of the beacons can be seen on Figure 7.1.4 and Figure 7.1.5. Each beacon is marked in green, and expected to be installed on an intersection area, where the user can receive clear instructions to his target destination. The beacons were placed as expected from the Testing scenario, in which Staff Office 0.10 and Staff Office 1.41 will be used to prove the application's functionality.

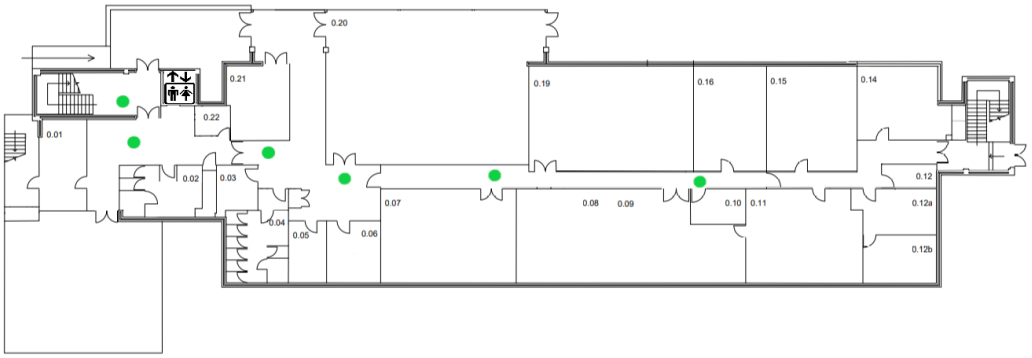

Figure 7.1.4: Ground Floor Beacon Location

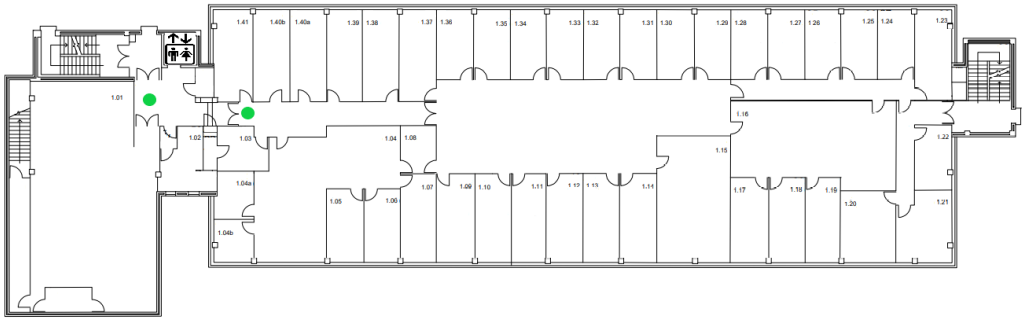

Figure 7.1.5: First Floor Beacon Location

In the future, if using this technique, it is recommended to install a beacon in front of each door in order to get more accurate positioning. Another option would be to upgrade the equipment with a more powerful one, place it inside rooms or open areas and use trilateration to pinpoint the exact location of the user.

7.2 Requirements

The second prototype will implement the Core FeatureSet (F-1, F-9) and the Navigation FeatureSet (F-3, F-4, F-5, F-6, F-7 and F-8).

7.3 Analysis, Design and Implementation

Back-end Structure

The API which was decided to be used to connect the beacons to the progressive web application is the Web Bluetooth API. The code which attempted to find the Bluetooth beacons can be seen on Figure 7.3.1.

```
async function startNavigation() {
  let filters = [];

  let filterName = "GroundFloorBeacon";
  if (filterName) {
    filters.push({name: filterName});
  }

  let filterNamePrefix = "e2c56db5";
  if (filterNamePrefix) {
    filters.push({namePrefix: filterNamePrefix});
  }

  let options = {};
  if (document.querySelector( selectors: '#allAdvertisements').checked) {
    options.acceptAllAdvertisements = true;
  } else {
    options.filters = filters;
  }

  try {
    log('Requesting Bluetooth Scan with options: ' + JSON.stringify(options));
    const scan = await navigator.bluetooth.requestLEScan(options);

    log('Scan started with:');
    log(' acceptAllAdvertisements: ' + scan.acceptAllAdvertisements);
    log(' with filters: ' + JSON.stringify(scan.filters));

    navigator.bluetooth.addEventListener('advertisementreceived', event => {
      log('Advertisement received.');
      log('  RSSI: ' + event.rssi);
      event.manufacturerData.forEach((valueDataView, key) => {
        logDataView('Manufacturer', key, valueDataView);
      });
      event.serviceData.forEach((valueDataView, key) => {
        logDataView('Service', key, valueDataView);
      });
    });

    setTimeout(stopScan,  timeout: 10000);
    function stopScan() {
      log('Stopping scan...');
      scan.stop();
      log('Stopped.  scan.active = ' + scan.active);
    }
  } catch(error)  {
    log('Argh! ' + error);
  }
}
```

Figure 7.3.1: Code connecting the beacons to the application

As seen from the screenshot, one of the beacons was named GroundFloorBeacon and its name prefix was e2c56db5. However, the Web Bluetooth API was unable to find the beacon and receive any information from the device. Due

to the time constraint and the little experience the developer had with the API, it was decided for the application to use a start point instead. This resulted in the application not being able to locate the user and to give a predefined path based on the start and end point chosen.

Due to the scarce information found on the algorithms implementing paths on maps and little experience with working with JavaScript on images, it was established that it is not mandatory to use algorithms to implement the shortest path to the end destination. Instead, the path was drawn on the image for each scenario (Room 0.10/1.41) and the maps displayed were handled by JavaScript functions. This solution successfully implemented the navigation part of the application as seen from the Second Prototype screenshots.

Since the initial plan for the implementation of the prototype was changed fundamentally, the instructions provided to the user for navigation from Chapter 5 were also changed accordingly (Second Prototype Screenshots). The Favourite Routes feature (star sign) and the Sign out button were removed as the Google Account feature was not implemented in the First Prototype. The Start button was modified to appear only after the user had chosen a start and end point.

The new files which were added in this increment were added to the cached files from the First Prototype (Figure 7.3.2).

Figure 7.3.2: Adding the new static files to the browser's cache

Changes were made to the server file as seen on Figure 7.3.3. The server was configured to run on port 8080 for testing purposes and an event listener was added to check the server events. Express.js was used as it is designed for applications requiring the use of APIs, making it suitable for the artefact.

```
app.use(express.static(path.join(__dirname, 'public')));

app.use(bodyParser.json());

/*Server*/

/*Get port from environment and store in Express.*/
const port = normalizePort( val process.env.PORT || '8080');
app.set('port', port);

/*Create HTTP server.*/
const server = http.createServer(app);

/*Listen on provided port, on all network interfaces.*/
server.listen(port);
server.on( event 'error', onError);
server.on( event 'listening', onListening);

/*Normalize a port into a number, string, or false.*/
function normalizePort(val) {
    const port = parseInt(val,  radix 10);

    if (isNaN(port)) {
        // named pipe
        return val;
    }

    if (port >= 0) {
        // port number
        return port;
    }

    return false;
}
```

Figure 7.3.3: Server Configuration

Push notifications were not implemented on the artefact as they required the platform specific language Kotlin to add them with Firebase Cloud Messaging and a higher priority was given to the database. The database was created on the Firestore website as seen from Figure 7.3.4 by adding each table as a collection and the relational data as a document storing a map of values.

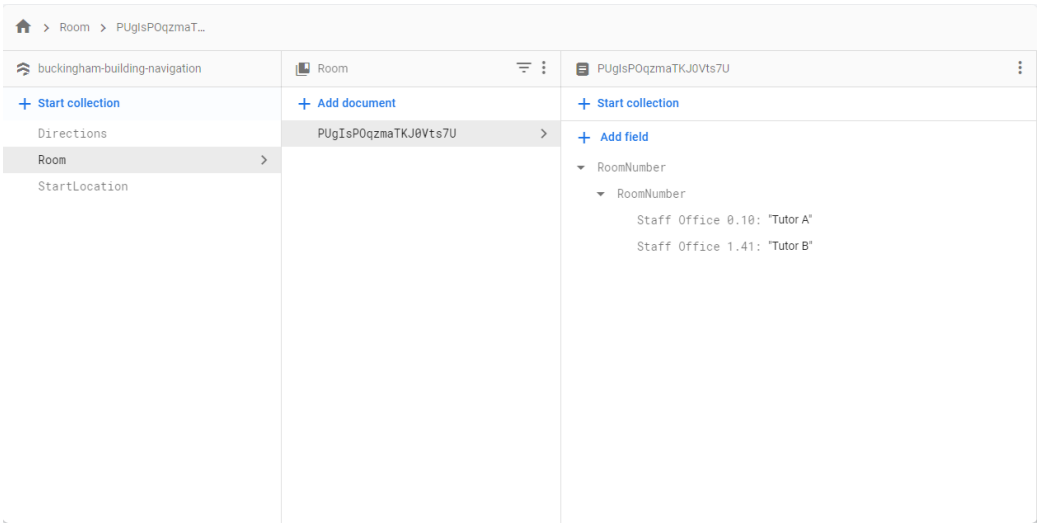

Figure 7.3.4: Firebase Design

The database was successfully added to the artefact on Figure 7.3.5. When the database was queried, no response was received and due to the little time left for the development of the artefact and the little experience the developer had with Firestore database, this feature is recommended to be implemented at an earlier stage in future work.

```
<script src="https://www.gstatic.com/firebasejs/5.0.3/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/5.0.3/firebase-firestore.js"></script>

<script>
    //Connects to Firebase backend and identifies the app
    const firebaseConfig = {
        apiKey: "AIzaSyAdfMrZSpB8_fsJNmT2oxDr2YeFyNKVdKc",
        authDomain: "buckingham-building-navigation.firebaseapp.com",
        projectId: "buckingham-building-navigation",
        storageBucket: "buckingham-building-navigation.appspot.com",
        messagingSenderId: "707024362581",
        appId: "1:707024362581:web:3dbbd4ee82df8f77f048dc",
        measurementId: "G-1FEC88T4ER"
    };

    firebase.initializeApp(firebaseConfig);
    const db = firebase.firestore();
    db.settings({ timestampsInSnapshots: true });

</script>
```

Figure 7.3.5: Connecting to the database

File hierarchy

The service worker and the cache response code were moved to new files named client.js and worker.js. This allowed for the project to have the server configuration added as a separate file and to replace them. As features were redesigned, the app.js and the homePage.js files were removed and the code required for the application's functionality was moved to the navigationPage.js file.

Overall, minor changes to the file hierarchy were made as the back-end structure did not cause major issues to the implementation of the features. The file hierarchy in the second increment is described in Appendix E.

Front-end Structure

Home Page

The Home Page was not changed in this increment.

Navigation Page

The map was added with the back-end changes made. It was made scrollable since the building is large and the map could not fit on the user's screen. Text alternative was added to it as it may not be displayed on all devices. Lifts were added on the map. Additional accessibility features were chosen to be added in future work as specific paths would be needed for users with mobility issues.

In terms of design choices, the overall colour combination from Chapter 5 was kept, with the only difference being the instructions. They were decided to be highlighted on to emphasize on the guidance of the app. The application was compatible on smart phones for both iOS and Android and responsive to user input. No personal data was required and the application's interface was kept simplified.

Navigation Page Menu

The application's menu was not modified in this increment.

7.4 Second Prototype screenshots

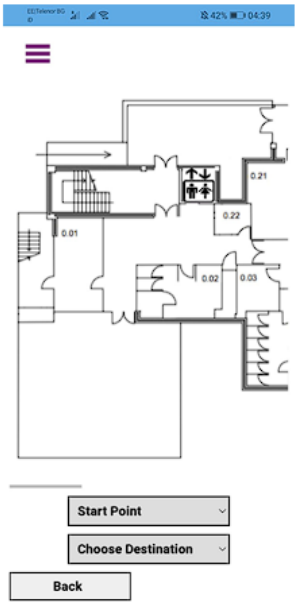

Figure 7.4.1: Navigation Page

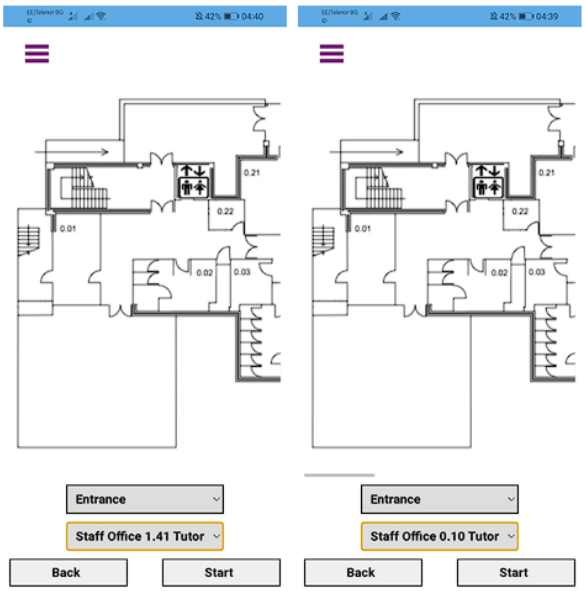

Figure 7.4.2: Choosing a Staff office

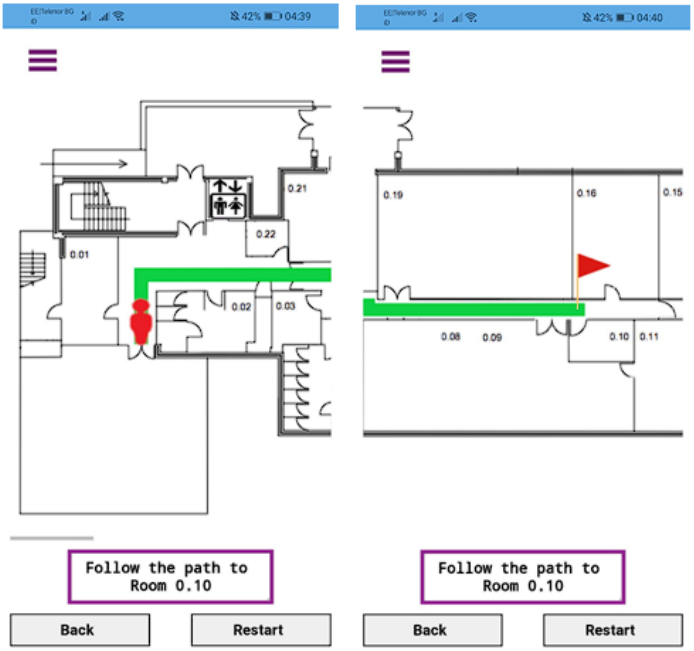

Figure 7.4.3: Staff office 0.10 Instructions and path

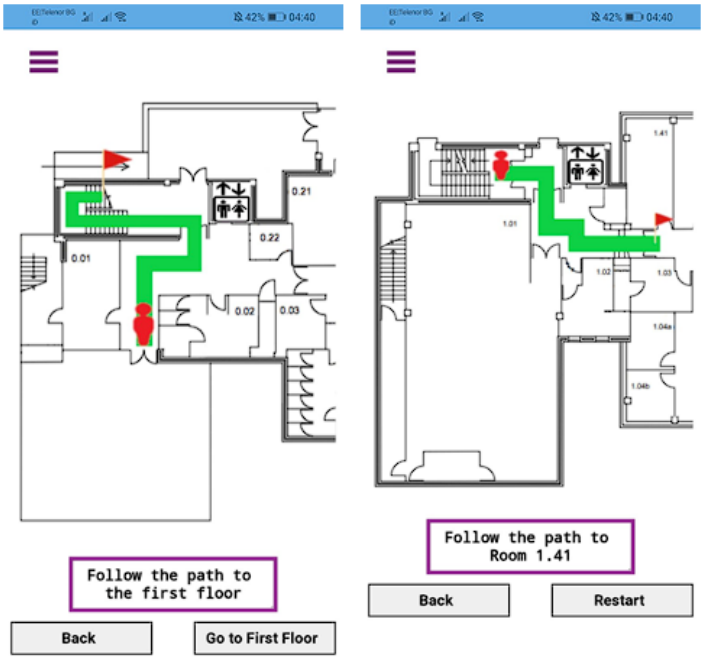

Figure 7.4.4: Staff office 1.41 Instructions and path

7.5 Deployment

After the prototype was designed and implemented, it was deployed on Git Hub to keep a backup copy for future work.

7.6 Chapter Summary

In the second prototype user requirement numbers F-3, F-4, F-5, F-6, F-7 were successfully implemented. The Core FeatureSet (F-1, F-9), requirement number F-8 from the Navigation featureSet and the Navigation Add-ons featureSet (F-14, F-15, F-16, F-17) were chosen to be developed in future work due to the time constraint. The system was delivered and a backup of the version is kept on Github.

Chapter 8

Testing

8.1 First Prototype Testing

Tables 8.1.1 and 8.1.2 show the implementation outcome of the functional and the non functional requirements by performing user testing. Features are considered to be "Satisfied", "Partly satisfied" and "Not implemented" depending on how fully the requirements have been met.

Table 8.1.1: Non Functional Requirements Testing

Requirement Number	Requirement	Satisfied?	Testing
NF-1	IOS and Android Compatibility	SATISFIED	The application is compatible with the most recent version of iOS (15) and Android (10).
NF-2	Smartphone compatibility	SATISFIED	The application is available on smartphones.
NF-3	Responsiveness	PARTLY SATISFIED	The application returns to its correct state when tested. It has still not been tested against user input.
NF-4	The system should not store any personal data	SATISFIED	No personal data is required by the user.
NF-5	Data encryption	NOT IMPLEMENTED	Dummy data is not encrypted as the application was not connected to the database.
NF-6	Interface simplicity	SATISFIED	Overall, less text was used on the interface to provide good user experience and unnecessary features were not implemented.
NF-7	Material design with appropriate level of contrast between colour combinations	SATISFIED	Material design guidelines were followed throughout the project.
NF-8	Text alternatives	SATISFIED	Images have an alternative description when not displayed.
NF-9	Change between light and dark mode	SATISFIED	Both modes were implemented, they change on start-up as the user switches his device's settings.

Table 8.1.2: Functional Requirements Testing

Requirement Number	Requirement	Satisfied?	Testing
F-1	Database	NOT IMPLEMENTED	The application does not store any data in the database in this increment. Will be considered in the Second Prototype.
F-2	Guest Login Feature	PARTLY SATISFIED	The user can access the app as a Guest. Staff member availability is not shown based on the guest login time.
F-3	Map	NOT IMPLEMENTED	Not implemented in this increment. Will be considered in the Second Prototype.
F-4	Locating the user on the map/Start point		
F-5	End point/destination		
F-6	Instructions		
F-7	Shortest route		
F-8	Accessibility information on the map		
F-9	Push notifications		
F-10	App is available offline	SATISFIED	The application is available offline.
F-11	App is installable on screen	SATISFIED	User can install the app on their screen and has the app icon with different sizes depending on the device's settings.
F-12	Google Account Login feature	PARTLY SATISFIED	The user can login with Google but doesn't receive information on his Account in the Menu. Buttons are not displayed according to the logins and the Sign out button does not sign the user out. This feature will be suggested for Future work due to data protection issues.
F-13	Favourite Route	PARTLY SATISFIED	The Favourite route feature was shown as an option (star) but was not added in this increment as the Google Sign in feature was not implemented fully. This feature will be suggested for Future work.
F-14	Walking time	NOT IMPLEMENTED	Not implemented in this increment due to time restrictions. Will be considered in the next iteration.
F-15	Distance		
F-16	Interactive Map		
F-17	Additional information displayed about the rooms		

The application was tested by the target audience, described the bugs found and provided feedback.

Found bugs

1. **Google Sign in feature signs in but is not receiving user information**

   Description: Google Sign in authorization is successful but user is

not receiving information on his timetable and staff availability from Google.

Cause: The library receiving these details was not implemented successfully.

2. **In the Menu the correct information is not displayed after Signing in**

   Description: The user should be able to see their name and university email in the Menu after signing in
   Cause: This bug was caused by the previous bug

3. **Sign out button is broken**

   Description: The Sign out button doesn't sign out the user
   Cause: This bug was caused by the Google Sign in bug

4. **Back button is displayed when the user is signed in**

   Description: The Back out button currently returns to the Home screen but is displayed when the user is Signed in.
   Cause: Google Sign in information not received by the application

5. **Adding a favourite route and displaying favourite routes from the "Favourite routes" button is not working**

   Description: Adding and displaying favourite routes is not working.
   Cause: This option would require a database in order to be implemented.

Feedback

The application pages closely matches the UI design with minor changes and the overall design was found to be satisfactory. Key functionality of the application is missing for indoor navigation which will be implemented in the second increment.

8.2 Second Prototype Testing

The following tables show the Non Functional (Table 8.2.1) and Functional Requirements Testing conducted on the Second Prototype (Table 8.2.2). The difference in the non functional requirements between the first and the second prototype was found to be mainly in the responsiveness as in the First prototype no user input was handled on the Navigation page.

Table 8.2.1: Non Functional Requirements Testing

Requirement Number	Requirement	Satisfied?	Testing
NF-1	IOS and Android Compatibility	SATISFIED	The application is compatible with the most recent version of iOS (15) and Android (10)
NF-2	Smartphone compatibility	SATISFIED	The application is available on smartphones
NF-3	Responsiveness	SATISFIED	The application returns to its correct state when tested and is responsive to user input as seen from the testing stage.
NF-4	The system should not store any personal data	SATISFIED	No personal data is required by the user
NF-5	Data encryption	NOT IMPLEMENTED	Dummy data is not encrypted as the application was not connected to the database.
NF-6	Interface simplicity	SATISFIED	Overall, less text was used on the interface to provide good user experience and features which don't work were removed.
NF-7	Material design with appropriate level of contrast between colour combinations	SATISFIED	Material design guidelines were followed throughout the project.
NF-8	Text alternatives	SATISFIED	The University of Portsmouth logo and the map have an alternative description when not displayed.
NF-9	Change between light and dark mode	SATISFIED	Both modes were implemented, they change on start-up as the user switches his device's settings.

Table 8.2.2: Functional Requirements Testing

Requirement Number	Requirement	Satisfied?	Testing
F-1	Database	NOT IMPLEMENTED	The application currently works with dummy data which is not stored in the database.
F-2	Guest Login Feature	PARLTY SATISFIED	The user can access the app as a Guest. Staff member availability is not shown based on the guest login time.
F-3	Map	SATISFIED	The app shows the map on the navigation screen. The user can scroll through the map to fully display it.
F-4	Locating the user on the map/Start point	SATISFIED	The user can choose a start point.
F-5	End point/destination	SATISFIED	The user can choose where he wants to navigate to.
F-6	Instructions	SATISFIED	Instructions are displayed based on the predefined route shown on the map.
F-7	Shortest route	SATISFIED	The user can see the shortest route on the map.
F-8	Accessibility information on the map	NOT IMPLEMENTED	The user can see elevators marked on the map, but not accessibility features such as stairlifts.
F-9	Push notifications	NOT IMPLEMENTED	Not implemented in this increment.
F-10	App is available offline	SATISFIED	The application is available offline.
F-11	App is installable on screen	SATISFIED	User can install the app on their screen and has the app icon with different sizes depending on the device's settings.
F-12	Google Account Login feature	PARTLY SATISFIED	The user can login with Google but doesn't receive information on his Account in the Menu. Buttons are not displayed according to the logins and the Sign out button does not sign the user out. This feature will be suggested for Future work due to data protection issues.
F-13	Favourite Route	NOT IMPLEMENTED	The Favourite route feature was removed from this Prototype as it should not be available in the Guest Login. This feature will be suggested for Future work.
F-14	Walking time	NOT IMPLEMENTED	Not implemented in this increment due to time restrictions. Recommended for Future Work.
F-15	Distance		
F-16	Interactive Map		
F-17	Additional information displayed about the rooms	NOT IMPLEMENTED	This feature will be implemented in future versions of the application as there is currently not enough detail provided on the information needed by the target audience.

After the final deployment of the artefact, the application was tested by the target audience.

Found bugs

1. **Google Sign in feature signs in but does not receive user information and no additional functionality is added**

   Description: Google Sign in authorization is successful but no additional features are shown. This does not affect the overall functionality of the application.

   Cause: Unsuccessful implementation in the First Prototype.

Feedback

The application fulfills its functionality and guides students to the locations implemented successfully. More staff offices can be added in the future, also teaching rooms can be considered as they are commonly used.

8.3 Chapter Summary

This chapter provides description of the outcome from the testing made by the target audience by showing bugs found in each Prototype and feedback provided by the target audience. Test cases were identified for both the functional and the non functional requirements and the testing outcome of each test scenario was represented.

Chapter 9

Evaluation

9.1 Evaluation against Requirements

The following tables critically evaluate the implementation outcome of the functional and the non functional requirements from the First and the Second prototype with each working feature's purpose, advantages and disadvantages.

Table 9.1.1: First Prototype Non Functional Requirements Evaluation

Requirement Number	Requirement	Purpose	Advantages	Disadvantages
NF-1	IOS and Android Compatibility	The application can be used on the most recent version of iOS (15) and Android (10)	The application is compatible with the most recent version of iOS and Android.	The application is not compatible on other versions of iOS and Android.
NF-2	Smartphone compatibility	The application can be used on smartphones	The application is available on smartphones.	The application is available on other devices but is not tested. It would likely need additional changes to be made to function properly.
NF-3	Responsiveness	The application can return to its correct state if interrupted	The application doesn't crash and is fast when changing states.	The application has still not been tested against user input.
NF-4	The system should not store any personal data	Personal data is not stored to avoid data exposure	No personal data is required by the user.	The application does not accept user data.
NF-6	Interface simplicity	Application will have an overall improved user experience	Overall, less text was used, and the user interface is clearer.	Less features and information on the user interface.
NF-7	Material design with appropriate level of contrast between colour combinations	User interface will conform to the best practices of user interface design	Material design applies mobile first sensibility. Material design can be applied for dark theme.	The design is only valid for Android devices.
NF-8	Text alternatives	Pictures and the map have a text when they are not loaded on the page	Visually impaired users using screen readers will be able to read the alt text.	Takes up visual display space.
NF-9	Change between light and dark mode	The light emitted by the device's screen is reduced while maintaining the minimum colour contrast ratios required for readability	Less light emitted from the device and less 'blue light' emitted from the user's phone.	Dark mode may increase eye strain in brightly lit conditions.

Table 9.1.2: First Prototype Functional Requirements Evaluation

Requirement Number	Requirement	Purpose	Advantages	Disadvantages
F-2	Guest Login Feature	The user can access the app as a Guest.	In the future when the Google Sign in option is added users who don't have a Google Account will be able to access the app as a Guest.	The user needs to go through the Home screen to reach the Navigation screen. Staff member availability is not shown based on the guest login time.
F-10	App is available offline	The application is available offline.	The application can be accessed without using Wi-Fi or mobile data.	These features require the use of Web Worker for progressive web apps which is currently in development.
F-11	App is installable on screen	User can install the app on their screen and has the app icon with different sizes depending on the device's settings.	The user can install the application on their home screen if they have selected the option. The icon has different sizes depending on the device's settings.	
F-12	Google Account Login feature	The user can login with Google but doesn't receive information on his Account in the Menu. Buttons are not displayed according to the logins and the Sign out button does not sign the user out. This feature will be suggested for Future work due to data protection issues.	The user can login with Google.	The user is not able to access their account's data currently.
F-13	Favourite Route	The user can add their route as a favourite to have access to them later.	The user can see the option (star) to add a favourite route.	This feature requires the Google Sign in feature to be fully working and store the route data.

The difference between the Non Functional requirements between the First and the Second Prototype is in the responsiveness as the First prototype did not accept user input (Table 9.1.3).

Table 9.1.3: Second Prototype Non Functional Requirements Evaluation

Requirement Number	Requirement	Purpose	Advantages	Disadvantages
NF-1	IOS and Android Compatibility	The application can be used on the most recent version of iOS (15) and Android (10)	The application is compatible with the most recent version of iOS and Android.	The application is not compatible on other versions of iOS and Android.
NF-2	Smartphone compatibility	The application can be used on smartphones	The application is available on smartphones.	The application is available on other devices but is not tested. It would likely need additional changes to be made to function properly.
NF-3	Responsiveness	The application can return to its correct state and accepts user input without being interrupted.	The application does not crash and is fast when changing states.	The application is not fully optimized as its not a native app.
NF-4	The system should not store any personal data	Personal data is not stored to avoid data exposure	No personal data is required by the user.	The application does not accept user data.
NF-6	Interface simplicity	Application will have an overall improved user experience	Overall, less text was used, and the user interface is clearer.	Less features and information on the user interface.
NF-7	Material design with appropriate level of contrast between colour combinations	User interface will conform to the best practices of user interface design	Mobile first sensibility. Material design can be applied for dark theme.	The design is only valid for Android devices.
NF-8	Text alternatives	Pictures and the map have a text when they are not loaded on the page	Visually impaired users using screen readers will be able to read the alt text.	Takes up visual display space.
NF-9	Change between light and dark mode	The light emitted by the device's screen is reduced while maintaining the minimum colour contrast ratios required for readability	Less light emitted from the device and less 'blue light' emitted from the user's phone.	Dark mode may increase eye strain in brightly lit conditions.

Table 9.1.4: Second Prototype Functional Requirements Evaluation

Requirement Number	Requirement	Purpose	Advantages	Disadvantages
F-2	Guest Login Feature	The user can access the app as a Guest.	In the future when the Google Sign in option is added users who don't have a Google Account will be able to access the app as a Guest.	The user needs to go through the Home screen to reach the Navigation screen. Staff member availability information is not shown.
F-3	Map	The user can see a map of the indoor environment.	The user can identify the indoor environment and locate himself based on the map if he is on the same floor.	The map does not always show the user's accurate location – if he is on the first floor it won't change.
F-4	Locating the user on the map/Start point	The user can select his start point.	The user can locate himself based on the start point.	The user is unable to see his location on other places on the map.
F-5	End point/destination	The user can select the end point (Staff Office) where they want to navigate to.	The staff offices the user wants to navigate to are easier to identify.	Currently not many staff offices are available. In the future as more are added, a scroll dropdown with a limited number of options will be likely needed.
F-6	Instructions	Provides instructions to the user on how to reach his target destination.	The user can find the way to their destination by receiving instructions.	Instructions are given based on the predefined path on the map.
F-7	Shortest route	Provides the shortest route from the starting point to the user's target destination.	The user can see the shortest route he needs to take to his target destination.	The user is not able to see his location on the map.
F-10	App is available offline	The application is available offline.	The application can be accessed without using Wi-Fi or mobile data.	These features require the use of Web Worker for progressive web apps.
F-11	App is installable on screen	User can install the app on their screen and has the app icon with different sizes depending on the device's settings.	The user can install the application on their home screen if they have selected the option. The icon has different sizes depending on the device's settings.	
F-12	Google Account Login feature	The user can login with Google but doesn't receive information on his Account in the Menu.	The user can login with Google.	The user is not able to access their account's data currently.

9.2 Evaluation against Aims and objectives

The aims and objectives of the project were met, with the necessary documentation provided. Higher priority requirements were prioritised and the project aimed

towards having the key features available due to the time constraint. As new changes in the floor plan of Buckingham building were made recently, this prototype won't be able to consider them as they were made too late to allow change of concept. The main goal of helping people navigate easily within Buckingham building was achieved.

9.3 Methodology and Project Plan Evaluation

The methodology chosen for this project allowed for the artefact to meet the base requirements during the period of development. The deadlines set out in the Project plan's Gantt chart, were not met due to unforeseen circumstances interfering with project progress. The Iterative and Incremental methodology allowed for deviations to the deadlines and the featureSets to be adjusted to enable the timely completion of the project. In the future, it is recommended to use a more realistic Gantt chart for the time management of the project which will take into consideration the possible circumstances which may delay the project development.

In terms of the navigation method which was initially selected, beacon technology was found to be complex and more time was required to study and implement it on the artefact which resulted in a different navigation method being chosen. As more staff offices are expected to be added in the future, using maps is currently not a good solution for most user devices since they take up space and this would slow down the application further. Consequently, in future work it is recommended to use an algorithm which will take the start point and end point and calculate the shortest route based on the map of Buckingham building. Additionally, more instructions can be provided on intersections of the building as currently they are set up on the predefined path shown on the map. By adding more instructions, the user will have a better image of the indoor environment and will navigate more easily.

9.4 Personal Evaluation

My time management throughout this project was poor due to illness which caused the project development to be slowed down. I also spent a significant amount of time on the Literature review as I found it hard to determine the development tools and programming languages which I would be using on the artefact. The literature and materials found with programming development methods described relating to the topic were found to be scarce and further guidance was sought during the development of the artefact by Networking,

Internet of things and Programming professionals in the University of Portsmouth Computing department. As getting in touch and booking a meeting at a suitable time delayed the progress of the artefact, if this project was to be redone it is advised to do the implementation of the artefact earlier on in the project development and to familiarize oneself with the tools earlier on in the implementation stage.

9.5 Chapter Summary

This chapter critically evaluates the outcome of the features implemented in the First and the Second Prototype against the Requirements, Aims and Objectives, Methodology and the Project Plan. The project implementation was evaluated from a personal point of view and deductions were made based on the evaluation findings.

Chapter 10

Conclusion

10.1 Recommendations prior to Future work

A more accurate Gantt chart can be used in the future to implement the artefact as the one used for the time management of this project represented the ideal development of artefact and did not allow for deviations due to unforeseen circumstances. The implementation of the artefact is recommended to be done earlier on in the development of the project to add more functionality to the artefact.

In future releases, any remaining bugs could be fixed and software maintenance could be conducted regularly to keep up with the target audience's needs. The web application can be updated each time it is deployed with Firebase when a stable version of the artefact is available. As more functionality is expected to be needed by the users, it could be added to the Settings button which was proposed to be added to the Menu in Chapter 5.

In regard to the requirements, the ones which were not fully implemented due to time restrictions could be satisfied. The core requirement which was not implemented due to difficulty and time constraint - the database - can be developed to ensure the application meets the security requirements. The Sign in with Google feature could be made available with the information icon which was drafted in the Design chapter. Additionally, locating the user on the map, showing the shortest route by using an algorithm, displaying the walking time and distance could be implemented as they were not fulfilled in this version due to the time restrictions and complexity. The Rooms additional information feature could be added in future releases by running a focus group of the users and gathering data on this feature. It is proposed

that the interface simplicity requirement (NF-6) should be kept at all times and should be considered while implementing the previous features. Additionally, the navigation can be improved further by providing voice instructions and a different route for users with accessibility instructions.

Since Buckingham building also has teaching rooms, it is likely that students would need to navigate to them. As the artefact currently uses maps to help the users navigate, it is recommended to use an algorithm which will calculate the shortest route between the start and the end point to make the application more efficient. Considering that the artefact currently provides navigation instructions to staff offices, it will be easier to implement it for a lecture theatre for example as it is larger than a staff office. As mentioned in Chapter 5, if the artefact becomes popular and requires more than one server, the Client-Server architectural pattern can be replaced by a Object-broker architecture to guarantee the application's responsiveness.

10.2 Chapter Summary and Project Overview

The overall aim of the project was satisfied and the report was completed. The project was developed throughout the academic year for a short period of time, with coursework and exam deadlines which affected its development. As a result of the difficulty met with the implementation of the database and the beacon technology, if this project was to be done again, it is advised for the developer to familiarize themselves with the tools for the implementation earlier on in the project development and to follow a more realistic Gantt chart which will be able to take into consideration possible issues with the development of the project. In the future, the features which were not added can be implemented to further improve the functionality of the application and the overall user experience. In conclusion, this project supports the University of Portsmouth community by fulfilling its main aim of providing navigation inside Buckingham building to a diverse audience.

Chapter 11

Bibliography and References

Fallah, N., Apostolopoulos, I., Bekris, K., Folmer, E. (2013). **Indoor human navigation systems: A survey.** Interacting with Computers, 25(1), 21-33. https://doi.org/10.1093/iwc/iws010

Subedi, S., Pyun, J.-Y. (2020). **A Survey of Smartphone-Based Indoor Positioning System Using RF-Based Wireless Technologies.** SENSORS, 20(24), 7230. https://doi.org/10.3390/s20247230

Tomazic, S. (2021). **Indoor Positioning and Navigation.** SENSORS, 21(14), 4793. https://doi.org/10.3390/s21144793

Uzun, A., Ghani, F. A., Najafabadi, A. M. A., Yenigun, H., Tekin, I. (2021). **Indoor Positioning System Based on Global Positioning System Signals with Down- and Up-Converters in 433 MHz ISM Band.** SENSORS, 21(13), 4338. https://doi.org/10.3390/s21134338

Montague, K. (2010). **Accessible indoor navigation.** In Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility. Association for Computing Machinery, 305–306. https://doi.org/10.1145/1878803.1878884

Ozdenizci, B., Ok, K., Coskun, V., Aydin, M. N. (2011, April). **Development of an indoor navigation system using NFC technology.** In 2011 Fourth International Conference on Information and Computing, 11-14. https://doi.org/10.1109/ICIC.2011.53

Jain, M., Rahul, R. C., Tolety, S. (2013). **A study on Indoor navigation techniques using smartphones.** In 2013 International Conference on Advances in Computing, Communications and Informatics, 1113-1118.

https://doi.org/10.1109/ICACCI.2013.6637333

Barberis, C., Andrea, B., Giovanni, M., Paolo, M. (2013). **Experiencing indoor navigation on mobile devices.** It Professional, 16(1), 50-57. https://doi.org/10.1109/MITP.2013.54.

Orjales, F., Losada-Pita, J., Paz-Lopez, A., Deibe, A. (2021). **Towards Precise Positioning and Movement of UAVs for Near-Wall Tasks in GNSS-Denied Environments.** Sensors 2021, 21(6), 2194. https://doi.org/10.3390/s21062194

Vazquez-Lopez, R., Herrera-Lozada, J.C., Sandoval-Gutierrez, J., von Bülow, P., Martinez-Vazquez, D.L. (2021). **Sensor Information Sharing Using a Producer-Consumer Algorithm on Small Vehicles.** Sensors 2021, 21(9), 3022. https://doi.org/10.3390/s21093022

Fu, W.; Liu, R.; Wang, H.; Ali, R.; He, Y.; Cao, Z.; Qin, Z. (2020). **A Method of Multiple Dynamic Objects Identification and Localization Based on Laser and RFID.** Sensors 2020, 20(14), 3948. https://doi.org/10.3390/s20143948

Walter, D. (2015). **Apple iOS 8 v Android 5.1 – which is best?**. http://thenextweb.com/dd/2015/04/06/apple-ios8-v-android-5-1-which-is-best/

Ishikawa, Montello (2006). **Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places.** Cognitive Psychology, 52(2), 93 – 129. https://doi.org/10.1016/j.cogpsych.2005.08.003

Downs, R. (1977). **Maps in Minds: Reflections on Cognitive Mapping.** New York; London Harper and Row, 1977

Golledge, R. G. (n.d.). **Wayfinding behavior: Cognitive mapping and other spatial processes.** Psycoloquy, 10.

Dasler, P., Malik, S., Mauriello, M. L. (2021). **"Just Follow the Lights": A Ubiquitous Framework for Low-Cost, Mixed Fidelity Navigation in Indoor Built Environments.** International journal of human-computer studies, 155, 102692. https://doi.org/10.1016/j.ijhcs.2021.102692

Jkielty. (2019). **Android v iOS market share 2019.** https://deviceatlas.com/blog/android-v-ios-market-share

White, J. (2013). **Going native (or not): Five questions to ask mobile**

application developers. The Australasian medical journal, 6(1), 7. https://doi.org/10.4066/AMJ.2013.1576

Weeks, K. M. (2012). **Going mobile: A guide to mobile website and application user preferences.** https://digitalcommons.calpoly.edu/grcsp/71

Nield, D. (2015) **iOS vs Android: The 2015 Edition.** http://fieldguide.gizmodo.com/ios-vs-android-the-2015-edition-1700461435

Aitken, N. (2015) **Android vs iOS vs Windows Phone 8 – Mobile OS Comparison.** http://whatphone.com.au/compare/android-vs-ios/

Ananiadis, V. (2015) **Good Mobility Index Report (Q1 2015).** https://www.slideshare.net/bassilis/good-mobility-index-report-q1-2015

Jobe, W. (2013). **Native Apps vs. Mobile Web Apps.** International Journal of Interactive Mobile Technologies, 7(4). https://doi.org/10.3991/ijim.v7i4.3226

Griffith, C. (2021). **What is Hybrid App Development?.** https://ionic.io/resources/articles/what-is-hybrid-app-development

Pandya, S. (2021) **Choosing the right database for your React Native app.** https://blog.codemagic.io/choosing-the-right-database-for-react-native-app/

Solanki, P. (2020) **Top Local Databases for React Native App Development.** https://www.mindinventory.com/blog/top-react-native-databases/

Patel, N. (2021) **Top 5 Databases For React Native App Development.** https://www.business2community.com/tech-gadgets/top-5-databases-for-react-native-app-development-02383205

Bhosale, S. T., Patil, M. T., Patil, M. P. (2015) **Sqlite: Light database system.** Int. J. Comput. Sci. Mob. Comput, 44(4), 882-885. https://doi.org/10.47760/ijcsmc

QA InfoTech (2021) **Top Local Databases For React Native Mobile App Development In 2021.** https://qainfotech.com/local-databases-for-react-native-mobile-app-development/

Shah (2017) **React Native Database – Choosing the right database for your React Native app.** https://www.simform.com/blog/react-native-database-selection-guide/2

Google Cloud (2022) **Server-side encryption.** https://cloud.google.com/firestore/docs/server-side-encryption

Saccomani, P. (2019) **Native Apps, Web Apps or Hybrid Apps? What's the Difference?.** https://www.mobiloud.com/blog/native-web-or-hybrid-apps

Marcak, M. (2021) **Cross-Platform Apps vs Native Apps vs Progressive Web Apps.** https://www.businessofapps.com/insights/cross-platform-apps-vs-native-apps-vs-progressive-web-apps/

Stevens, E. (2021) **What is the Difference Between a Mobile App and a Web App?.** https://careerfoundry.com/en/blog/web-development/what-is-the-difference-between-a-mobile-app-and-a-web-app/

Sagara Technology Idea Lab. (2019) **Mobile Apps vs Web Apps: Which is the Better Option?.** https://sagaratechnology.medium.com/mobile-apps-vs-web-apps-which-is-the-better-option-868106c88730

Smith, C. (2020) **4 of the best indoor navigation apps for Android.** https://knowtechie.com/4-of-the-best-indoor-navigation-apps-for-android/

University of Windsor (2021) **Smart campus technology and app support easier wayfinding at UWindsor.** https://educationnewscanada.com/article/education/level/university/1/927301/smart-campus-technology-and-app-support-easier-wayfinding-at-uwindsor.html

Microsoft (2017) **Path Guide: A New Approach to Indoor Navigation.** https://www.microsoft.com/en-us/research/blog/path-guide-new-approach-indoor-navigation/

Google LLC (2021) **Google Maps.** https://play.google.com/store/apps/details?id=com.google.android.apps.maps

Mapsted (2021) **CampusGo.** https://play.google.com/store/apps/details?id=com.mapsted.campusgo

Microsoft Corporation (2017) **Path Guide.** https://play.google.com/store/apps/details?id=com.microsoft.msra.followus.app

Google (2021) **Google Maps Help.** https://support.google.com/maps/answer/9332056?hl=enco=GENIE.Platform%3DAndroidzippy=%2Cnavigate-with-live-vie

Haselton, T. (2021) **Google Maps has a wild new feature that will guide you through indoor spaces like airports** https://www.cnbc.com/2021/03/30/google-maps-launches-augmented-reality-directions-for-indoor-spaces.html

Developers (2020) **Material Design for Android** https://developer.android.com/guide/topics/ui/look-and-feel

Google Forms (2022) **Get insights quickly, with Google Forms.** https://www.google.com/forms/about/

Google (2022) **Google Privacy Policy** https://policies.google.com/privacy

AccessAble (2022) **Buckingham Building.** https://www.accessable.co.uk /venues/ buckingham-building-hantsd73513bd-5a6c-fa44-8977-59cd30ff8c0a

L. C. de Oliveira, A. O. Andrade, E. C. de Oliveira, A. Soares, A. Cardoso and E. Lamounier (2017) **Indoor navigation with mobile augmented reality and beacon technology for wheelchair users.** 2017 IEEE EMBS International Conference on Biomedical Health Informatics (BHI), 37-40, https://doi.org/10.1109/BHI.2017.7897199

P. Chen, Z. Peng, D. Li and L. Yang (2016) **An improved augmented reality system based on AndAR.** Journal of Visual Communication and Image Representation, 63-9, https://doi.org/10.1016/j.jvcir.2015.06.016.

I. A. Cankaya, A. Koyun, T. Yigit and A. S. Yuksel (2015) **Mobile indoor navigation system in iOS platform using augmented reality.** 9th International Conference on Application of Information and Communication Technologies (AICT), 281-284, https://doi.org/10.1109/ICAICT.2015.7338563.

Nikander, J., Järvi, J., Usman, M., Virrantaus, K. (2013) **Indoor and outdoor mobile navigation by using a combination of floor plans and street maps.** In Progress in Location-Based Services (pp. 233-249). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34203-5_13

Gerstweiler G, Vonach E, Kaufmann H. HyMoTrack (2016) **A Mobile AR Navigation System for Complex Indoor Environments.** Sensors. 16(1):17. https://doi.org/10.3390/s16010017

EnterpriseDB, (2022) **PostgreSQL is an open source database support PostGIS spatial database.** www.enterprisedb.com

PostGIS, (2020) **About PostGIS.** http://postgis.net/

Open Geospatial Constortium, (2022) **Web Map Service.** http://www.opengeospatial.org/standards/wms

Goetz, M. and A. Zipf (2013) **Indoor Route Planning with Volunteered Geographic Information on a (Mobile) Web-Based Platform.** Progress in Location-Based Services, 211-231. https://doi.org/10.1007/978-3-642-34203-5_12

Hansen, R., Thomsen, B., Thomsen, L. L., Adamsen, F. S. (2013) **SmartCampusAAU–An Open Platform Enabling Indoor Positioning and Navigation.** In 2013 IEEE 14th International Conference on Mobile Data Management, 33-38. https://doi.org/10.1109/MDM.2013.62

HERE (2022) **Micello is now part of HERE Technologies. Together, we'll be mapping everything.** https://venues.here.com/

MazeMap (2022) **Indoor Maps and Wayfinding.** https://www.mazemap.com/

Irwin, L. (2018) **The GDPR: Understanding the 6 data protection principles.** https://www.itgovernance.eu/blog/en/the-gdpr-understanding- the-6 -data-protection-principles

Google Developers (2022) **Secure your site with HTTPS.** https://developers.google.com/search/docs/advanced/security/https

ProjectSmart (2022) **MoSCoW Method.** https://www.projectsmart.co.uk/tools/moscow-method.php

BeaconZone (2022). **JO-BEC09.** https://www.beaconzone.co.uk/Jinou/BEC09

Android Developers (2018). **Discontinuing support for Android Nearby Notifications.** https://android-developers.googleblog.com/2018/10/discontinuing-support-for-android.html

Upland (2022), **5 Tips for Designing the Perfect Mobile App Icon.** https://uplandsoftware.com/localytics/resources/blog/tips-for-designing-the-perfect-mobile-app-icon/

University of Portsmouth (2022). https://portsoc.github.io/photoboard/ https://portsoc.github.io/photoboard/

Appendix A

Project Initiation Document

School of Computing
Project Initiation Document

Nikoleta Koleva

Progressive web application for navigation inside School of computing buildings

Engineering Project

1. Basic details

Student name:	Nikoleta Koleva
Draft project title:	Progressive web application for navigation inside School of computing buildings
Course:	BSc Computer science
Project supervisor:	Mrs. Soraya Harding
Client organization:	N/A
Client contact name:	N/A

2. Degree suitability

This project is suitable for my degree as it requires me to develop a progressive web application that will test my knowledge and understanding of programming languages and software engineering acquired throughout my course. It will also challenge me to produce an artefact with efficient, easy-to-maintain code that will satisfy the user requirements and solve the problem represented in this dissertation.

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

Student navigation has proven to be a common issue for students in University of Portsmouth buildings. New students often struggle to find a room inside a building as they are not familiar with the building architecture and aren't aware of the resources available online related to the building they are in. This project aims to provide a progressive web application that will make finding a room faster and easier in the School of Computing buildings.

A progressive web application is suitable as it involves coding in programming languages already used during my course which will make the development faster. Additionally, it will be available on cross platforms, will reach a wider audience and will be able to load information fast which is essential for real-time navigation issues.

The application will be available on mobile devices. Mobile phones play a major part in our daily lives and users tend to carry them frequently which will make it easier to access the application and quickly check in which direction they need to go to reach the desired destination. This mobile application will help users memorize the building easily and navigate fast as it will use augmented reality to show real-time directions based on unique objects in the building such as rooms.

The project will seek an audience of different age groups who have classes in the School of Computing buildings and are interested in using a navigation application related to this topic.

4. Project aim and objectives

This project aims to provide a progressive web application that uses augmented reality to make finding a room faster and easier in the School of Computing buildings.

Since the project involves large buildings and time is a serious constraint, it will analyze the students' data gathered from the questionnaires. The destinations most often visited by students will be taken into consideration and the project will be aimed towards giving a solution for the rooms most commonly used. Considering the constraints, the project will be aimed towards having the key features available.

The project objectives are:

- Choosing a project supervisor and project topic
- Deliver PID
- Literature review write-up
  - Research other apps and tools which help with navigation inside a building
  - Research relevant literature
- Requirements (Software requirements specification)
  - Conduct interviews/use questionnaires to gather information
  - Elicitate requirements
- Design
  - Produce design documentation (UML diagrams, UI mockups)
  - Design database
- Create prototype
  - Develop server-side
  - Develop client application
  - Unit tests
- Application testing
  - Identify test plans
  - System tests
  - Evaluate test results
- Project evaluation
  - Conduct appropriate testing for the application
  - Perform user tests and gather user feedback
- Provide regular weekly reports to track progress
- Finalize the report
- Conduct the final demonstration

5. Project deliverables

The deliverables for this project are an artefact (a progressive web application) and a report which will give details of the development and research strategies. Documentation supporting the report will be the Project initiations document and ethics certificate, literature review, requirements specification, design specification, test strategies, questionnaires/interviews with students and feedback results.

6. Project constraints

A significant constraint is that the application will require experience with languages related to progressive web applications and augmented reality which wasn't taught in my course. Online courses and tutorials will help in gaining a better understanding and completing the project.

Another constraint is the accessibility of data such as the student's timetables needed to identify the destination room. It may be difficult to get a hold of it as it is stored in the university's system. It is possible to find a way around this, however this adds more work to the project as data integrity will need to be ensured.

Another constraint is time. Since the time is limited and the project is complex, the most commonly used university rooms will be made possible destinations. Respectively, the requirements will be prioritized so that the most essential features are developed within the time constraints.

A possible constraint is the risk of a lockdown due to the covid pandemic and the potential closure of university buildings that will result from it. This will have a significant impact on the number of interviews conducted as face-to-face meetings might not be possible and it might be more difficult to find participants for the research.

Lastly, the cost of the software used for the development of the product and its maintenance will be considered. Since it may be difficult to find software that satisfies the requirements and comes at a little price, possible solutions will be considered.

7. Project approach

The research I will do will be in the form of a literature review and information gathering from participants who are interested in the topic. I will prioritize the requirements using the Moscow method and will determine the highest priority ones. By keeping track of the development plan on the Gantt chart, I will implement the highest priority ones and will consider the rest based on their constraints. After the requirements have been decided, I

plan is to start with the development stages of a software engineering project and to decide on a software development methodology.

8. Literature review plan

The starting point of my research for the literature review will be looking into already existing navigation systems such as Google Maps, checking how they can be re-engineered to solve the problem and reading their relevant documentation to the topic.

I will also look into books or papers in journals that will be relevant to this project. The primary systems I have planned to use for my research are the Library website and Google Scholar. By looking up academic journals and books, I will identify key-words and narrow down my search to the specific information I would need.

9. Facilities and resources

The computers required for this project will be both university ones and my personal computer. I will use them as required for accessing apps such as Intellij, AppsAnywhere and GitHub. Additionally, the university-provided Linux server will be used to host the server-side content such as databases, user authentication and data processing. To test application functionality and device compatibility on mobile devices, I will use a personal android device and will use software that will allow testing on devices of different sizes and cross-platform mobile devices.

10. Log of risks

Description	Impact	Likelihood	Mitigation	First indicator
Covid lockdown	High	Likely	Decide on alternative ways to conduct interviews and find participants for the research since university buildings most likely won't be open. Entering the university buildings while I can gather the necessary information needed for the project in time. Make an alternate test plan	The UK government and the university give a warning on a possible lockdown.

			that does not need to enter the closed building.	
Software Failure/ Software becomes unusable	High	Likely	Keep a different version of the software on a different device or have an alternative of the software application	Software used for development crashes or is obsolete.
Hardware Failure	High	Likely	Keep a backup of the project files	Hardware is causing issues that could lead to loss of data
Project Deadline	High	Likely	Follow the project plan, provide weekly reports on finished work and allocate the necessary amount of time for each task	The project deadline nears and there are too many requirements/ features to fulfill for the remaining amount of time.
Becoming ill	High	Unlikely	N/A	N/A

11. Project plan

I have planned on creating 2 Gantt charts using Lucidchart, one representing my initial/planned progress and one representing my actual progress during the project development. I have planned for at least two weeks for each milestone. There is a risk of falling behind in the schedule. For example, since there might be an impact by the covid pandemic as mentioned in the risks section the requirements gathering may take longer. Consequently, I will attempt to work ahead of schedule to mitigate all possible risks occurring in the development of the project.

Below is the Gantt chart describing my initial project development plan.


Gantt chart
Nikoleta Koleva | October 29, 2021
2021
2022
Milestone
Week #
1 2 3 4 5 6 7 8 9 10 11 12 13 1 2 3 4 5 6 7 9 10 11 12 13 14 15 16 17 18 19 20 21 22
Choose topic and supervisor
4 - 15 Oct
Deliver PID
18 - 29th Oct
Literature review
1 - 14 Nov
Conduct interviews/give out questionnaires
8 - 15 Nov
Software requirements specification
8 - 28 Nov
Design
22 - 6 Dec
Prototype creation
6 Dec - 26 Feb
Application testing
10 - 23 Jan
21 - 6 Mar
Project Evaluation
28 - 3 Apr
User testing and feedback
21 - 10 Apr
Finalize report
21 - 7 May
Final demonstration
9 - 25 May

12. Legal, ethical, professional, social issues (mandatory)

The main legal, ethical, professional and social issues related to this project are the gathering, storing, processing and protection of data from the questionnaire/interview participants. To ensure these issues are taken into consideration, the data taken from the user will be anonymous and users won't be asked questions that will identify them in any way. The database will be secured to the highest level possible for the time given to complete the project. Since new risks are coming up and no method will ensure the complete security of the data, in case of a cyber-attack the data won't include any personal information so that it wouldn't pose a risk to the participants.

When using the application, the user will be asked to give consent for his university timetable data to be shared. This will be done through Google sign in and the security of the data shared will be managed by them. To guarantee that the requirements are met the data shared with the application won't be sent to any external server or a third-party and will be stored on the user's device.

As the artefact is a university-based application, the project will also aim not to damage the reputation of the university and have negative implications for it. To satisfy these conditions, the project will be developed with both university students and the university as an institution in mind.

In the future, if it's decided for the application to be made available out of the UK, the data-sharing regulations and the content in the application will be considered carefully to match the requirements and the restrictions of the countries it will be used in.

Since my project requires the project supervisor to agree with the ethics certificate and sign it, I have asked her through email correspondence and confirmed her agreement. She had also signed the ethics certificate.

Appendix B

Ethics Review Form

Certificate of Ethics Review

Project title: Progressive Web application for navigation inside School of computing buildings

Name:	Nikoleta Koleva	**User ID**:	899244	**Application date**:	27/10/2021 16:35:08	**ER Number**:	TETHIC-2021-101517

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

The FEC representative(s) for the **School of Computing** is/are **Philip Scott, Matthew Dennis**

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

- University Policy
- Safety on Geological Fieldwork

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

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

Which school/department do you belong to?: **School of Computing**
What is your primary role at the University?: **Undergraduate Student**
What is the name of the member of staff who is responsible for supervising your project?: **Mrs Soraya Harding**
Is the study likely to involve human subjects (observation) or participants?: Yes
Will peoples` involvement be limited to just responding to questionnaires or surveys, or providing structured feedback during software prototyping?: Yes
Will the study involve National Health Service patients or staff?: No
Do human participants/subjects take part in studies without their knowledge/consent at the time, or will deception of any sort be involved? (e.g. covert observation of people, especially if in a non-public place): No
Will you collect or analyse personally identifiable information about anyone or monitor their communications or on-line activities without their explicit consent?: No
Does the study involve participants who are unable to give informed consent or in are in a dependent position (e.g. children, people with learning disabilities, unconscious patients, Portsmouth University students)?: No
Are drugs, placebos or other substances (e.g. food substances, vitamins) to be administered to the study participants?: No
Will blood or tissue samples be obtained from participants?: No
Is pain or more than mild discomfort likely to result from the study?: No
Could the study induce psychological stress or anxiety in participants or third parties?: No
Will the study involve prolonged or repetitive testing?: No
Will financial inducements (other than reasonable expenses and compensation for time) be offered to participants?: No
Are there risks of significant damage to physical and/or ecological environmental features?: No
Are there risks of significant damage to features of historical or cultural heritage (e.g. impacts of study techniques, taking of samples)?: No
Does the project involve animals in any way?: No
Could the research outputs potentially be harmful to third parties?: No
Could your research/artefact be adapted and be misused?: No
Does your project or project deliverable have any security implications?: No

I confirm that I have considered the implications for data collection and use, taking into consideration legal requirements (UK GDPR, Data Protection Act 2018 etc)
I confirm that I have considered the impact of this work and and taken any reasonable action to mitigate potential misuse of the project outputs

I confirm that I will act ethically and honestly throughout this project

Supervisor Review

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

Supervisor's signature:  Date: 28/10/2021

Faculty Ethics Committee Review

Faculty Ethics Committee Member's signature: Date:

Appendix C

Requirements Questionnaire

Final Year Project - Application for navigation inside Buckingham building

Thank you for participating in my questionnaire. This questionnaire should take between 5-10 mins to complete.

All data which will be gathered from this questionnaire will be kept anonymous and will only be used for Primary Research.

If you have any questions please feel free to email me at:

up899244@myport.ac.uk

*Required

What is your gender? *

- Male
- Female
- Prefer not to say

What year are you in? *

- First year
- Second year
- Third year (Placement)
- Final year
- Masters
- Other:

What is your age? *

- ○ 18 - 20
- ○ 21 - 25
- ○ 26 - 30
- ○ 31+
- ○ Other:

Most used device *

- ○ Smartphone
- ○ Laptop
- ○ Tablet
- ○ Other:

Phone OS *

- ☐ iOS (on all iPhones)
- ☐ Android
- ☐ Other:

Please describe your OS version (skip if you are unsure)

Your answer

When you perform tasks like Messaging, Maps do you prefer using an Application or a website? *

- ◯ Application
- ◯ Website
- ◯ Both are okay

Have you used indoor navigation applications inside buildlings? *

- ◯ Yes
- ◯ No

If selected "Yes", please describe what applications you have used?

Your answer

If selected "No", please describe what other methods you have used for navigation?

Your answer

This questionnaire is about developing an indoor navigation application to help navigate inside Buckingham building and reach a room. What features do you consider such application should have? *

- [ ] Map of the building
- [ ] Ability to see where you are on the map
- [ ] Instructions on where to go after a certain point has been reached
- [ ] Additional information displayed about rooms
- [ ] Other:

What constraints or not useful features of the indoor navigation application do you find bad to have? *

Your answer

Further feedback - feel free to express any concerns here.

Your answer

Submit Clear form

Appendix D

Project Structure First Prototype

- .firebase folder
  - hosting - contains information on the files hosted with Firebase
- .github folder - holds the git hub settings for commits
- node modules folder - project packages are installed in this folder
- public folder - contains html, css, javascript, json files and image files
  - src folder - stores the files with code/images which are mainly read or edited
    - css folder - includes the style sheet documents used for describing the presentation of the HTML documents
      - homePage.css
      - main.css - includes the css of all the pages
      - navigationPage.css
      - themes.css - the dark and light theme are styled here
    - images folder - the application's images and icons for installing the app on the phone's screen are stored here
      - app-icon-96x96.png
      - app-icon-144x144.png
      - app-icon-256x256.png
      - app-icon-512x512.png
      - logo200x70.png - University of Portsmouth logo suitable for both dark and light theme
    - js folder - contains the javascript files which will be used for the application
      - app.js
      - homePage.js
      - navigationPage.js
  - index.html - home page
  - manifest.json - name of the application, configured as a progressive web app
  - navigationPage.html
  - server.js - server configuration
- firebaserc - stores the project's aliases
- gitignore - specifies intentionally untracked files that Git should ignore. Would be used upon pushing files to Git Hub

• firebase.json - lists the project’s configuration

• package.json - used for setup, where it tells node.js what libraries to install

• package-lock.json - keeps track of the exact version of every package that is installed

• README.MD file which shows the name of the application, its description, instructions on how to install, run the project and how to use the project.

Appendix E

Project Structure Second Prototype

- .firebase folder
  - hosting - contains information on the files hosted with Firebase
- .github folder - holds the git hub settings for commits
- node modules folder - project packages are installed in this folder
- public folder - contains html, css, javascript, json files and image files
  - src folder - stores the files with code/images which are mainly read or edited
    - css folder - includes the style sheet documents used for describing the presentation of the HTML documents
      - homePage.css
      - main.css - includes the css of all the pages
      - navigationPage.css
      - themes.css - the dark and light theme are styled here
    - images folder - the application's images and icons for installing the app on the phone's screen are stored here
      - app-icon-96x96.png
      - app-icon-144x144.png
      - app-icon-256x256.png
      - app-icon-512x512.png
      - app-icon.png – a copy of the app icon is kept
      - logo200x70.png - University of Portsmouth logo suitable for both dark and light theme
      - groundFloor.png – the rest of the images are included for navigation
      - groundFloorInstructionsToFirst.png
      - groundFloorNavigation.png
      - firstFloorNavigation.png
    - js folder - contains the javascript files which will be used for the application
      - navigationPage.js
  - 404.html – page included by Firebase hosting to display "Page not found" error
  - index.html - home page
  - manifest.json - name of the application, configured as a progressive web app
  - navigationPage.html
  - client.js – checks if service worker is installed
  - worker.js – service worker
- firebaserc - stores the project's aliases

• gitignore - specifies intentionally untracked files that Git should ignore. Would be used upon pushing files to Git Hub

• firebase.json - lists the project's configuration

• package.json - used for setup, where it tells node.js what libraries to install

• package-lock.json - keeps track of the exact version of every package that is installed

• README.MD file which shows the name of the application, its description, instructions on how to install, run the project and how to use the project.

• server.js - server configuration