

Specification Document

Project Name: Urban Sciences Building Tour App

Team Number: 6

|  |  |  |
| --- | --- | --- |
|  |  |  |

Document Information

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| Project Name: | Urban Sciences Building Tour App | | |
| Prepared By: | Will Comber | Preparation Date: | 22/11/18 |
| Email / Phone: | w.comber@newcastle.ac.uk |  |  |
| Document Version No: | 2.11 | Document Version Date: | 06/03/19 |
|  |  |  |  |

Version History

| Ver. No. | Ver. Date | Revised By | Description |
| --- | --- | --- | --- |
| 1.0 | 8/12/2018 | Will Comber | Added in executive summary and created contents table |
| 1.1 | 8/12/2018 | Imran Mohammed | Added project plan |
| 1.2 | 10/12/2018 | James O’Connor | Added background and analysis parts 1 & 2 |
| 1.3 | 10/12/2018 | Joe Lowthian | Added project purpose |
| 1.4 | 10/12/2018 | Joe Lawrence | Added software design overview |
| 1.5 | 11/12/2018 | Oscar Gilley | Added roles and deliverables |
| 1.6 | 11/12/2018 | Ethan Goodchild | Added hardware and software |
| 1.7 | 12/12/2018 | Will Comber | Added contribution matrix |
| 2.1 | 04/02/2019 | Edward Phillips | Added other considerations |
| 2.2 | 04/02/2019 | Will Comber | Updated numbering of requirements |
| 2.3 | 04/02/2019 | Imran Mohammed | Added semester 2 project plan |
| 2.4 | 05/02/2019 | Ethan Goodchild | Updated hardware and software |
| 2.5 | 06/02/2019 | Will Comber | Updated ordering of requirements based on priority |
| 2.6 | 06/02/2019 | Joe Lowthian | Updated the project purpose |
| 2.7 | 06/02/2019 | Titas Alvikas | Added UML package and behaviour diagrams |
| 2.8 | 07/02/2019 | James O’Connor | Updated Background Analysis |
| 2.9 | 26/02/2019 | Titas Alvikas | Updated UML package and behaviour diagrams |
| 2.10 | 01/03/2019 | James O’Connor | Updated References and added UML diagrams. |
| 2.11 | 03/03/2019 | James O’Connor | Added GUI designs. |
| 2.12 | 06/03/2019 | Will Comber | Added in GUI descriptions and updated formatting |
| 2.13 | 07/03/2019 | James O’Connor | Added UML descriptions and architecture diagram |

Table of Contents

Contents

[1. Purpose 4](#_Toc2776755)

[2. Background & Analysis 5](#_Toc2776756)

[2.1. Analysis Process 5](#_Toc2776757)

[2.2. Analysis 7](#_Toc2776758)

[2.3. Project purpose 8](#_Toc2776759)

[3. Roles and Deliverables 9](#_Toc2776760)

[3.1. Roles 9](#_Toc2776761)

[3.2. Deliverables 10](#_Toc2776762)

[4. Project Plan 11](#_Toc2776763)

[5. Hardware and software platforms to be used for developing and running your solution 12](#_Toc2776764)

[6. Solution requirements 12](#_Toc2776765)

[6.1. Functional Requirements 12](#_Toc2776766)

[6.2. Non-Functional Requirements 14](#_Toc2776767)

[7. Other considerations 14](#_Toc2776768)

[7.1. Assumptions 14](#_Toc2776769)

[7.2. Constraints and Dependencies 15](#_Toc2776770)

[8. Software Design 15](#_Toc2776771)

[8.1. Overview of what the software will do and not do. 15](#_Toc2776772)

[8.2. Modifications in light of comments made or changes deemed necessary 16](#_Toc2776773)

[8.3. System Architecture – describe design decisions (Architecture diagram) 17](#_Toc2776774)

[8.4. High level overview of how the functionality and responsibilities of the system are partitioned and assigned to components (deployment diagram, component diagram) 17](#_Toc2776775)

[8.5. Package and Class diagrams which show dependencies between components 19](#_Toc2776776)

[8.6. Show the dynamic behaviour of the system (sequence diagrams that shows communication between objects. Activity diagrams that show the state of objects). 21](#_Toc2776777)

[8.7. GUI, Human Interface Views – Describe how the user accesses functionality 24](#_Toc2776778)

[9. References 27](#_Toc2776779)

[10. Definition of terms 27](#_Toc2776780)

## Purpose

The purpose of this document is to formally describe in detail the application we plan to create and how we will produce it, giving Newcastle University a clear idea of our interpretation of their needs and how we can deliver it. Upon reading this document, one should be able to understand the objectives and goals of the project.

This application we are tasked with creating is an Urban Sciences Building (or USB) Tour Guide Application. The primary feature required from this is enabling new and prospective students to navigate around the USB, to find their tutor or a specific room, or to simply see the main features of the building. More detail on this will be given in the **background and analysis** section of this document.

The **roles and deliverables** section describes what is required from each role and the responsibilities that are involved. This includes tasks they will have to complete, and the deliverables associated with this.

Following this there is the **project plan**, which is used to give a breakdown of all tasks over the course of the project. This information is displayed in a Gantt chart broken down into the main sections: research, design, implementation. Each of these sections is then broken down further to indicate deliverables associated with these sections.

A **hardware and software platforms** section is included. This contains what will be used while developing our solution and what will be required for end users to run our final solution.

The details of what our final solution will contain is broken down under **solution requirements** into functional and non-functional requirements. Each requirement is given priority of inclusion and supplier compliance, indicating if this is planned to be included in the final system. A comment is provided if a feature will be only partially included or not at all.

Any **other considerations** that we have made are broken down into any assumptions made about requirements, accounting for any ambiguity and constraints and dependencies that will affect the delivery of our final solution.

The **software design** section details all elements of how the final application will look, how the underlying system will work and its behaviour.

Finally, all **references** and **definitions of terms** are given to identify any sources we have used and also clarify any terms that need defining.

Terms in bold are separate elements of the document that will follow.

**--------------------------------------------------------------------------------------------------------------------------**

## Background & Analysis

### Analysis Process

As part of our research we looked into different tour apps for other buildings and places to see what they did well and didn’t do well so we could use this to see what we should do for our tour app.

Other systems looked at:

ActionTourGuide [Show app, Action 2016]

* Pros: Showed a clear image of every room which is good for navigation, Easy to understand where the doors are located on the floor plans, faster to find a room if you know the name while searching and Gives a bit of information on the home page.
* Cons: To look for a specific room you needed to swipe through other rooms first, doesn’t give any additional information about the room other than the name before selecting it when searching and the home page Takes up a lot of space.

Natural History Museum [Vusiem Limited. 2015]

* Pros: Had clear images of every section of the museum **(NH1)**, the floor plan was Very clear what area is what on the map also had colour-coded system makes it easy to tell what floor you’re looking on and has additional information on map **(NH1)**, such as toilet locations and entrance location.
* Cons: but it provided no information about each part and how to get to it **(NH1)**, on the floor pan some images might be a bit unclear what they represent (each image is a silhouette, so not always clear what an image is) **(NH2)** and Other than the shape of the room, there is not much information about the contents of the room. The search bar was not integrated well and was hard to use making it not easy to find a specific room. **(NH3)**

Newcastle Castle [Activ, Hello 2018]

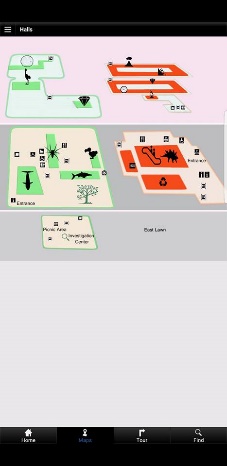
* Pros: Avoids having too much information in one place, has a clear image of the certain area and a lot of information for the user to read, the transport links **(NC1)** were useful to anyone looking to plan trips and it has a thumbnail to give a bit more information/understanding of the event. **(NC2)**
* Cons:Design is very basic - not very visually appealing, it had a very basic layout with a lot of information in the same place which could be a bit overwhelming on a first glance and it needs to be continually updated over time and doesn’t give an exact location before clicking the event.

After researching other apps that would be similar to ours it allowed us to gain an understanding of what features would be good to add to our building app, such as clear navigation and a well-integrated search feature and also what things to try and avoid such as not having too much information in one place or making the design very basic. This allowed us to move on to create features that we would want to add.

**Appendix for App research:**

**Natural History Museum (NH):**

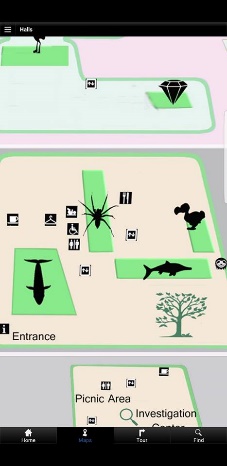
**1)**



**Pro:** Clear images of every sections of the museum, makes it very easy to find what the person is looking for and making it a lot easier to navigate the museum.

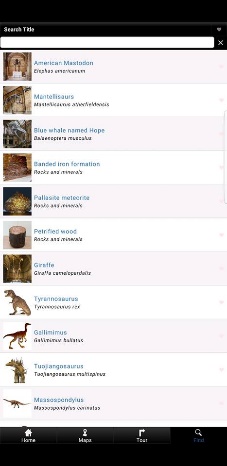
**Con:** No clear route planner, so you may be able to see what in each room but not how to get to each room.

**2)**



**Con:** Shadows don’t show what is in each room specifically and so it may take multiple attempts for a person to find the room with the exhibit they want.

**3)**



**Con:** Search bar not very well implemented as only certain things can be searched for and that makes it hard to find something if it is not in the search bar function.

**Newcastle Castle (NC):**

**1)**

**A picture containing building

Description automatically generatedA screenshot of a social media post

Description automatically generatedPro:** Lots of transport links, ‘By car’, ‘By Train’, ‘Public Transport’, ‘On Foot’. Also shows location and other attractions in the local area, giving landmarks for tourists to find in order to get to the castle.

**2)**

**A screenshot of a cell phone

Description automatically generatedPro:** Thumbnails give brief description of upcoming events, can be clicked in order to get to event page, good for people wanting to find basic information on the event but not too intrusive for those who do not.

### Analysis

The environment that our App would fit into would be that of the university. The app will be specifically designed to give information about the USB and be a tour guide for the USB, which includes giving people directions within the building to get from room to room and provide a search feature for users to search for a specific room or person within the USB to find the way to that room.

Our main stakeholder is Nicola Dolman, she is the one who has tasked us with creating this system and so she will be the client who we show the system to during development to gain feedback from. During our first interview she stated that the main features she was looking for where that the app allowed students to find their tutor rooms, be able to navigate around the building and to provide floor plans that are colour coded to what the room has in them e.g. tutor room, teaching space etc.

The other stake holders or the main people that will be using the app will be students who may be new and want to know how to get to places or students who have never been to a room before and need to know how to get to it, prospective students, those on open days and admission days where the app will give them information about the building and allow them to flow through the day in an organised way which may influence their decision to come to Newcastle. and those from other places who wish to tour the USB such as those from other university computer science departments who want to see how Newcastle works or those from other departments within the university who may have some need to come to the USB.

There is no such tour app which is specifically for the USB, but Newcastle university do have their own app for students and staff only, as a login is required. This however only shows locations of certain buildings and what they are called, using an inbuilt google maps. This doesn’t show what subjects are in each building or what tutors or how to get around the buildings, which will be the main aim for our app.

### Project purpose

The Urban Sciences Building (USB) is the home of The School of Computing at Newcastle University. The USB is in need of a mobile application that can act as a tour guide around the building for anyone visiting or working there. This project aims to build a mobile application to satisfy that need. The proposed project will run on the Android mobile operating system and will be available to download from the Google Play Store.

The goal is that this application will provide an easy and efficient way to gather information about the USB, without having to go online to a website, thus improving portability of information. Having information readily available to a user anywhere they go is crucial as it is easy to access and allows for peace of mind that any information they need will always be available. Having the application available to download from the Google Play Store will also allow multiple users to have access to the same information, removing the need to have one person per group being the information hub. The application will be free for anyone to download and will not be age-restricted.

The application will run on all devices running Android 5.0 Lollipop and above. By using Android 5.0, over 85% [Google 2019] of Android users will be able to run the application. This decision was made to target a large audience, without compromising on speed and functionality. In addition to this, the application size should not be too large, meaning users will not have to worry if it is taking up too much storage space on their device.

Using Android Studio, a big objective is to create an intuitive system by designing a clean graphical user-interface (GUI). By providing a simple GUI, the aim is to make users comfortable using the application and not struggle to understand how anything works. By using familiar and recognisable icons throughout the application, it should be easier for users to transfer their knowledge of other computer software to the USB app. For example, by using a recognisable search icon on the search bar, the functionality of that button should be clear without the user having to press it to find out.

Furthermore, a focused objective is to provide clear, easy-to-follow directions and navigation guides to users by using descriptive and understanding language. By using clear, short and simple navigation directions to guide a user around the USB, the goal is that anyone will be able to understand the instructions and be able to find their way around the building.

The application should also be designed to have a colour scheme without any colour clashes and with an option to change text size to improve readability. This will allow colour blind users to use the app as well as those with partial blindness who need larger text on their screen [Collinge, R. 2017].

Making the application run on different screen sizes is another key objective. By achieving this, the application will be available on a wide range of devices, with both small and large screen sizes.

Another objective is to provide users with additional information about the USB, such as architectural and contact information.

## Roles and Deliverables

### Roles

Within the team, each of us have specified roles to better delegate our contributions to the project. While we have these assigned roles, people are free to help in other areas as they see fit, and some aspects such as the programming are a whole group effort that everyone will contribute on and discuss in meetings.

#### Roles:

T. Alvikas – Lead Tester, Junior Web Developer

W. G. Comber - Team Leader, Junior Tester, Junior Documenter, Junior Programmer

O. J. E Gilley – Lead Web Developer, Junior Programmer

E. J. Goodchild – Junior Programmer, Junior Documenter

J. P. Lawrence – Lead Programmer, Junior Web Developer

J. E. P. Lowthian – Lead Designer, Junior Programmer, Junior Documenter, Junior Researcher

I. J. Mohammed – Lead Administrator, Junior Programmer

J. E. O’Connor – Lead Researcher, Junior Designer

E. R. G. R Phillips – Lead Documenter, Junior Tester

#### Responsibilities:

* Lead Programmer– Lead force behind developing the application detailed in our specification. They are in charge of designating tasks to all the junior programmers and overseeing the project, making sure everything is coming along at a good pace. They may also help the junior programmers with some tasks while contributing to the program themselves.
* Lead Documenter – Lead force behind the documentation of the application/website. They will work closely with the programming team to document the app’s functions and methods in detail and document the use of the app itself. They will also help the Junior Documenters with their work when needed.
* Lead Designer – Lead force behind the design of the app itself, creating mock-ups for the app and also helping the program with various aspects related to the user interface of the app. Works closely with assistant designers to discuss ideas for different aspects of design.
* Lead Tester – Lead force behind the testing of the app, overseeing planning for the testing and working with implementation team to successfully test the app. Works closely with assistant testers to create new test cases and carry out the tests with various phone devices.
* Lead Web Designer – Lead force behind the design of the app’s promotional website, doing the main design of the website in conjunction with the design team and deciding on the layout of the webpage itself. Works closely with the junior web designers to create the website’s content and styling.
* Junior/Assistant Roles – Assisting the lead roles with the development of their respective field, be it programming, documenting, etc. These include: Junior Programmer, Junior Web Designer, Assistant Designer and Assistant Documenter.
* Team Leader – Oversees all the different areas to make sure everything is on track, also organises meetings for the team outside of our scheduled weekly meeting and ensures each team member is doing their part.

### Deliverables

At the end of our project, we aim to have created a number of different things for our clients to use. These are:

* Tour guide application for the USB, feature complete including maps of each floor of the building, navigation options to find specific rooms and accessibility options for disabled people as a minimum set of requirements. We also aim to have a design that is accessible for colour blind people and simple for all users to navigate
* Marketing website for the application, detailing the features of the app, where you can download it, and some information about the Urban Science Building’s design and how we incorporated that into the design of our app. The website should also be easy to navigate so it can effectively market the app to our target audience of teenagers and students.
* Documentation on how the application functions on a technical level and a user level. This should be in depth, covering every method within the app and covering all the different features that a user might access during the use of the application

## Project Plan



## Hardware and software platforms to be used for developing and running your solution

#### Software

Android Studio - Used to develop and test app. Includes powerful APIs and tools used to speed up and efficiently build the app. It is available for free meaning it is widely accessible to all team members. Its main programming language is JAVA which all members of the team have experience with.

GitHub - To share files and collaborate as a team. Used to create repositories and keep project backed up. Team members can access GitHub via the web or with the GitHub software which is free to download.

Microsoft Teams - For communication between the team. It is a free download and can be accessed via the web. Includes integration with other Microsoft products such as Word, PowerPoint and Excel which is beneficial for documentation and design.

#### Hardware

Android Devices - For testing we will use a range of Android devices between version 5 and 9 of various sizes. 89% of the Android market has version between this range meaning we can exploit the APIs used in version 5 in our app.

Computers - We’ll each be using various computers that have the capability to run Android Studio and to emulate android devices. These computers will be used to program and compile the app as well. They will be connected to the internet so that we can use software such as Microsoft Teams and GitHub where we can share files and communicate with each other.

## Solution requirements

### Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Priority**  **(H, M, L)** | **Supplier**  **Compliance**  **(Full, partial or will not be delivered)** | **Supplier**  **Comment** |
| **Data** | | | |
| 1.1. The app should allow the user to view the number and specification of computers in each cluster room | L | Full |  |
| 1.2: The app must include a menu for the Café within the USB. | L | Full |  |
| 1.3: The app must include the locations of the vending machines in the USB, either on the floor plan or as a part of a separate page. | L | Full |  |
| 1.4. The app must display a three-dimensional fully rendered floor plan of the building | L | Will not be delivered | The memory requirements outweigh the functionality gained |
| 1.5. The app must display a news feed of events taking place in the USB | L | Will not be delivered | The news feed of events provided by the university is not updated and thus would be a pointless inclusion |
| 1.6: The app must provide details on the opening and closing times of the Café. | M | Full |  |
| 1.7: The app must include a tour guide for open days, with all the necessary information within. | M | Full |  |
| 1.8: The app must include the locations of the and toilets in the USB, either on the floor plan or as a part of a separate page. | M | Full |  |
| 1.9: The app must include both locations and descriptions of various architectural features of the USB. | M | Full |  |
| 1.10: The app must include a Wheelchair-accessible navigation guide. | H | Full |  |
| 1.11: The app must provide details on the opening and closing times the USB. | H | Full |  |
| 1.12: The app must include information for the standard fire procedure of the USB, including meeting locations. | H | Full |  |
| 1.13: The app must include colour-coded floor plans for navigations, including further details on tutor rooms, computer rooms, teaching spaces and details for out-of-hours access. | H | Full |  |
| 1.14: The app must include Tutor Profiles which include names, details and the location of their tutor rooms. | H | Full |  |
| **Features** | | | |
| 2.1. The app should display data from the building sensors | L | Will not be delivered | This is not required and would be an unnecessary feature. |
| 2.2: The app must include a tour guide detailing the architectural features of the USB. | M | Full |  |
| 2.3: The app must give the user directions both to and from the USB and other locations such as the main campus. | M | Full |  |
| 2.4: When the room is found, the app must give the user directions to said location. | H | Full |  |
| 2.5: The app must include a tour guide section, offering a guide for visitors to rooms, features of the building and nearby landmarks. | H | Full |  |
| 2.6: The app must give the user the ability to search for specific tutors and their rooms. | H | Full |  |

### 

### Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Priority  (H, M, L) | Supplier  Compliance (Full, partial or will not be delivered) | Supplier  Comment |
| **Design Requirements** | | | |
| 3.1: The app must have a menu that can be swiped in from the left-hand side of the screen. | L | Full |  |
| 3.2: The app must include photos of the building’s interior and exterior as a part its design. | M | Wil not be delivered | Storage requirements outweigh the benefits of inclusion. |
| 3.3: The app must have a menu that can be accessed on every page. | H | Full |  |
| 3.4: The app must have an appropriate search button in order to use the search function of the app. | H | Full |  |
| 3.5: The app must have a title section with appropriate menu icons for navigation. | H | Full |  |
| 3.6: As the floors of the USB are colour coded, the floors of the app must be colour coded using the same choice of colourings as the building uses. | H | Full |  |
| **Accessibility Requirements** | | | |
| 4.1: The app must include text and user interface scaling as an option to improve the user experience for users with impaired vision. | M | Full |  |
| 4.2: There must be an appropriate way to activate step free directions | H | Full |  |
| 4.3: The app must be designed with colour blindness in mind. | H | Full |  |
| 4.4: The app will access the system accessibility of the device and adjust the text size if large text is enabled. | H | Full |  |

## Other considerations

### Assumptions

For this project there are some assumptions that have been made which have influenced the development of the app, those assumptions are listed below.

* From our research we can assume that the majority of users will be using an android device which is running versions of Android between 5.0 and 8.0.
* We can assume all features of the program can be made in Android Studio and written in the Java programming language.
* We can assume that the project with all planned features can be finished within the given timeframe and in-line with the project plan.
* We can assume that the likelihood of the client significantly changing the requirements so that the project requires major restructuring to be unlikely.

### Constraints and Dependencies

|  |  |
| --- | --- |
| **Constraint** | **How the Constraint will be handled** |
| Hardware Constraints | As not all users of the app will be using the exact same phone, the app will have to be designed with that in mind. Due to the fact that it is likely that there will be some users who are using the app with older hardware, so the app will need to be designed to run quickly and efficiently even on devices with less power.  To rectify this potential issue, the app is being designed to function optimally with as few system resources as possible, enabling it to function properly on both older and newer systems alike. |
| Software Constraints | As our research shows, approximately 18% of android devices in active use are still running on Android version 5.0, even though the most recent version, Android version 8.0, is now in active usage. Because of this, the app will have to be designed to be operable with all versions of android between and including versions 5.0 and 8.0.  This, however, comes with the downside of partially limiting the potential functionality of the app as we must forgo some features which are made redundant in later versions of Android or newer features that have been added in later versions. |
| Time Constraints | Creating an app that is both functional and stable is a lengthy process. If not enough time is given to development, some planned features may have to be scrapped as they cannot be finished on time. If not enough time is given to testing, then you are risking releasing a product with serious issues or bugs that either were never found initially or not enough time was given to properly rectify them.  To fix this issue, the project has been planned around how long that each section should take to complete with float time at the end on the off chance that more time needs to be dedicated to the development or testing of the final product. |

## Software Design

### Overview of what the software will do and not do.

Our team’s application is a tour app designed to help people navigate the Newcastle University Urban Sciences building (USB), primarily aimed at new computer science students. The software developed for the project will be composed of a mobile app for Android and a website used to market and provide information for the app. This section will detail what the software will and will not do, the program structure, and architectural designs for the app and website.

#### App

We are programming the App for Android targeting versions 5.0 and up as our research showed that 18.3% of Android users are still on version 5.0. [Google 2019]

Our development environment is Android Studio 3.2.1, we are using the Android SDK and are not using any additional Android frameworks. The entire app will be programmed in Java using the standard library, as that is what the Android UI and API is written in, plus everyone in our team has experience with the language. We will also use the Google maps API in order to give users directions to the USB.

We designed the app to use a simple text file format for stored data, as we wanted it to make sure it was always usable offline and that it can parse and load the data quickly. We will also store images on the device since, again, we want the app to be responsive for the user and do not want to have to communicate over a network to stream images when the user starts the app. This also makes the app easily modifiable, for example you can change the text files and images and the app will work for buildings other than the USB.

The app will allow users to change some settings for accessibility [Nasta, S. 2017]. There will be preferences for options like font size. These settings will also be saved as text files on the device.

#### Website

The website will be developed using modern web technologies such as HTML5, CSS3 and JavaScript. Its primary purpose will be to market the app, and to give users information about why the app exists and how to use it.

### Modifications in light of comments made or changes deemed necessary

Based on the feedback given after our Dragons Den presentation we made several changes to the design of our application.

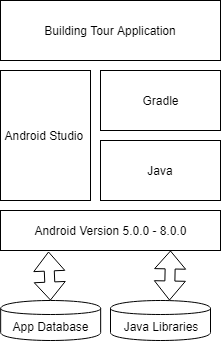
Firstly, is the use of a database. Originally, we had planned to attempt to use text files to interact with data, however based on the feedback given we did research into the use of a database and decided it was the best option. We chose to use SQLite for android studio.

Secondly was the incorporation of walking routes to the local transport links. This will be done by integrating Google Maps into the tour guide section of our app.

The final change made was based on the feedback that our room finder UI needed polish. Based on this feedback we have redesigned the page to be easier to use and more professional in its appearance.

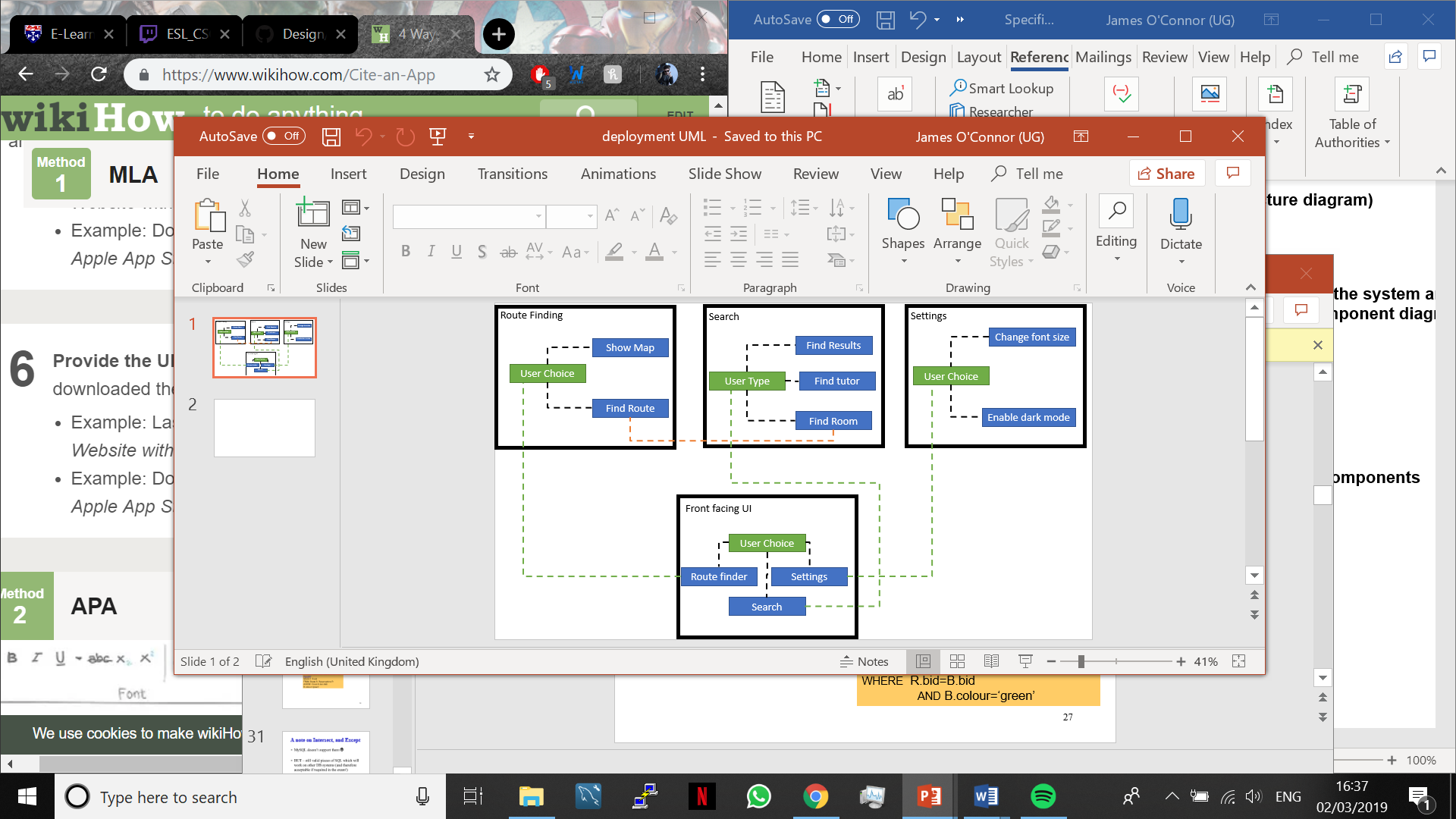
### System Architecture

**Architecture Diagram**



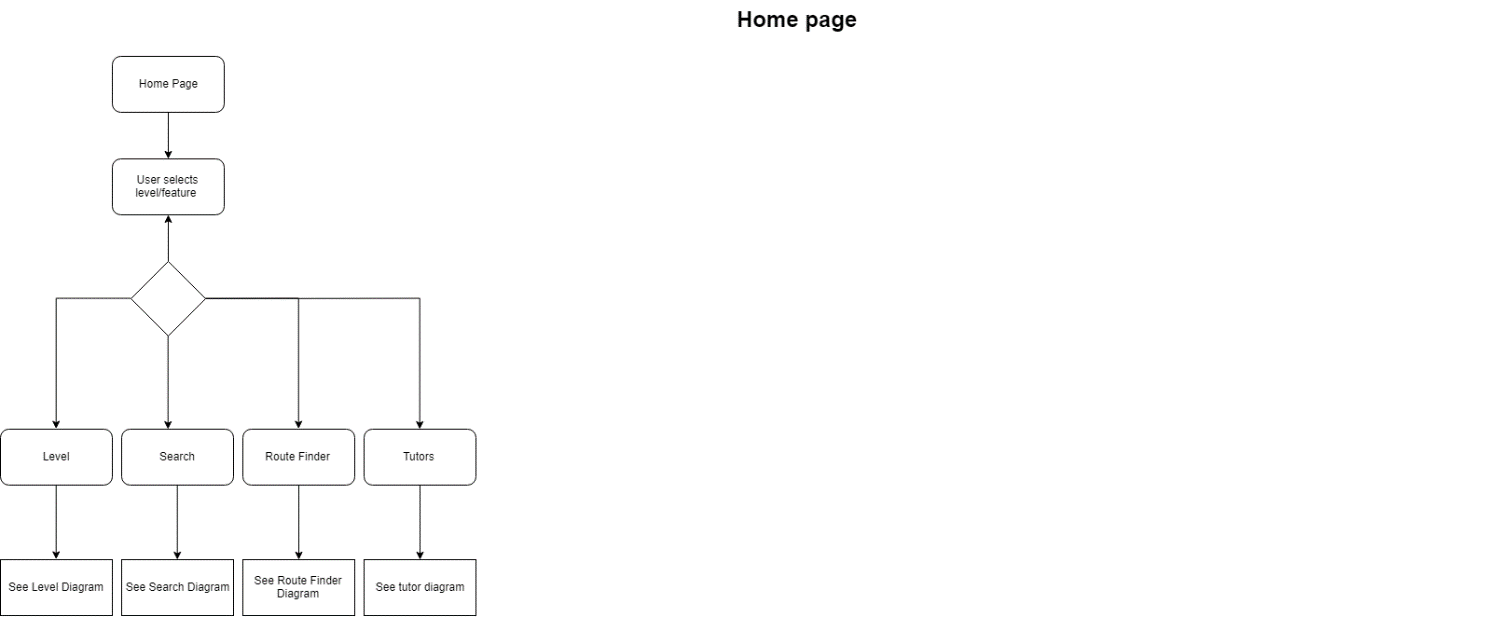
The Architecture diagram Shows the foundations that our app will be built upon. For the building tour application this starts with the android software versions 5.0.0 to 8.0.0 as these are what versions can run our app. On top of that is android studio which is where we will create our app. Within that is Java which is the programming language and a gradle which is what allows it to run on a mobile device. To allow all this to run we have an app database with room locations and information on the building as well as all the java libraries that we will require to allow the app to work.

### High level overview of how the functionality and responsibilities of the system are partitioned and assigned to components (deployment diagram, component diagram)

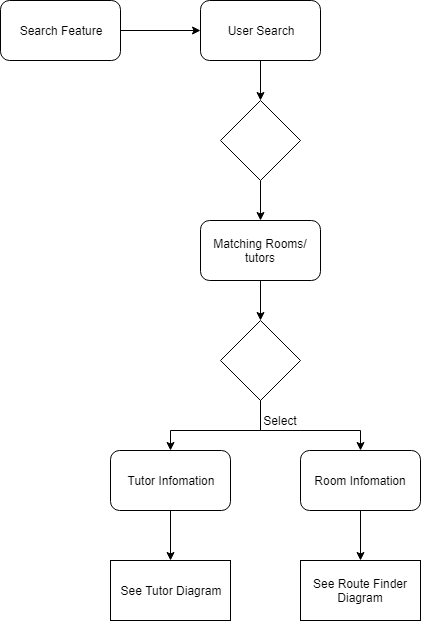
**Deployment Diagram**

The deployment diagram shows how the user can interact with each part of the system and what happens when they make a choice. The user will either choose route finder, search or settings. If they choose route finder, they then must choose show map or find route which will complete the action of which they chose. For search they must choose find results, find tutor or find room, which again will complete the action that they choose. Finally, for settings they much choose font size or dark mode to make changes to the app.

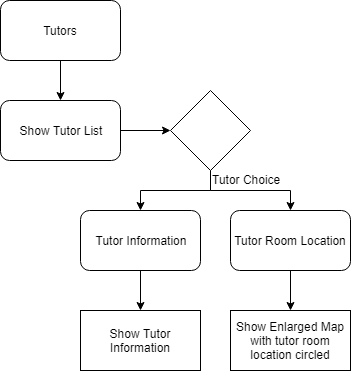
**Component Diagram**



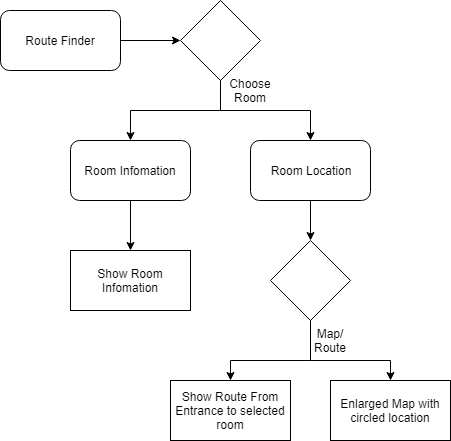
Component diagram for the home page of the application shows that the user must select a feature of the app and depending on which of the four options that they choose, level, search, route finder or settings, they will be taken to a different feature of the application.



Component diagram for the user search feature of the application shows that the user must enter a search criterion and the application will return matching rooms and tutors that fit the criteria given. With the return information the users must then select to get the room information or tutor information, and this is what will be returned when they select which of the options they require.

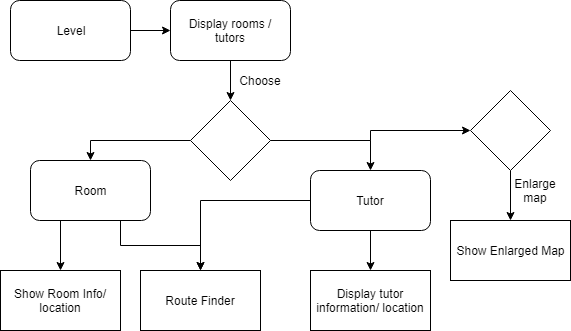


Component diagram for the Tutors feature of the application shows that a list of tutors will be given to the user and they are then required to make a choice of whether they would like the tutor information or the tutor’s room location. If they would like the information, they will be shown the tutors information including their name, room number, speciality etc. or if they chose to get the tutor’s rooms location then an enlarged map will appear with the tutor’s room location circled on it so that the user knows where to go.



Component diagram for the route finder feature of the application shows that the user must have already chosen a room to get to this point, either form the search feature or level feature. They will then choose to either see the room information, in which case the information of the room will be returned such as the number, type of room etc. or they will choose to see the room location. At this point they will have two options to either just see where it is on the map, which will bring up an enlarged map with the location of the room circled or they can choose a step by step route which will be given to them to direct them to the room they chose.

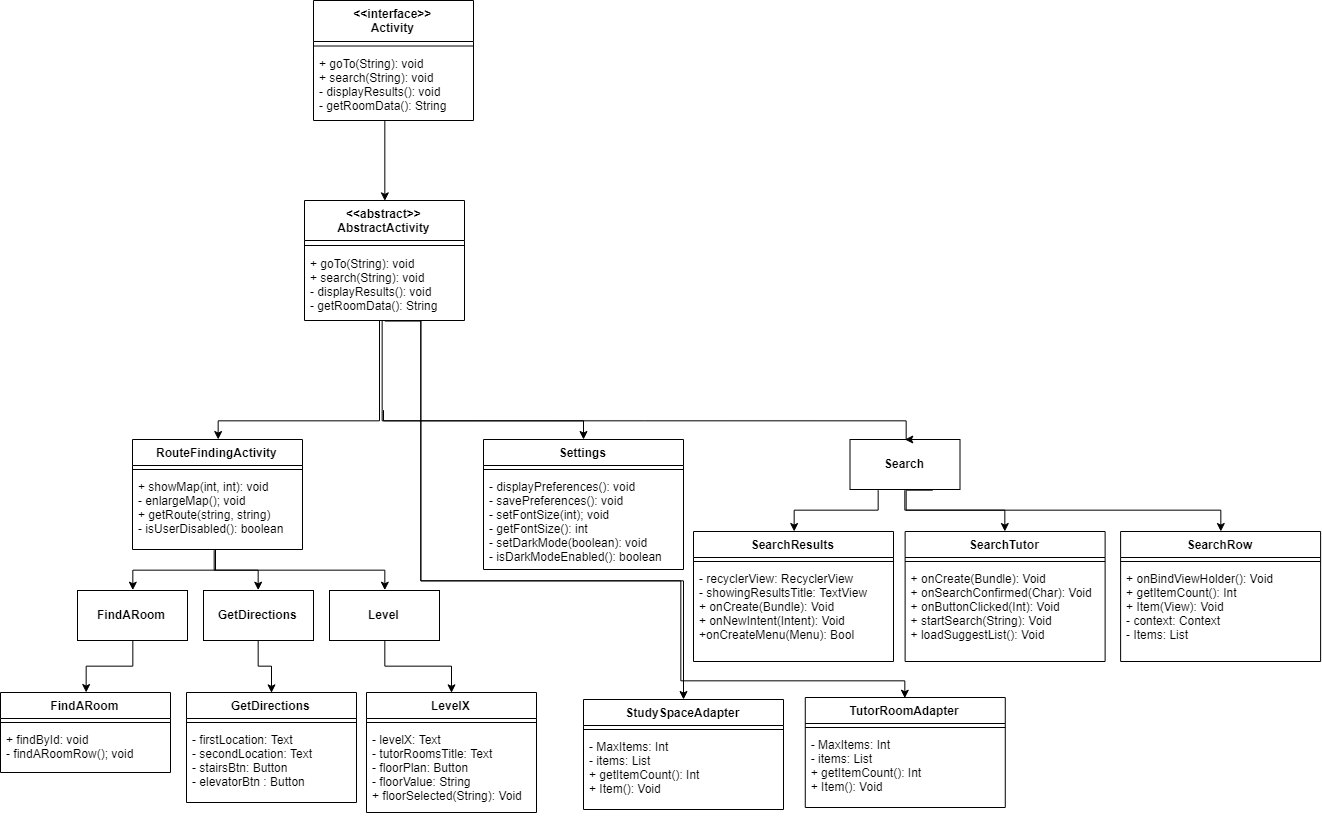
Component diagram for the level feature of the application will be the same for each of the levels in the USB and will display the rooms, study spaces and tutors of that level. The user must then choose a room or tutor. If they choose a room, they will either see the location and information of the room or be taken to the route finder feature to find the room depending on what the user will choose. If they choose a tutor the information of the tutor will be shown or the location of their room again depending on the users choice. The user has one more ability on the levels of the room which is to click to view an enlarged map of the level that they are on to see where the rooms are and the layout of the level.



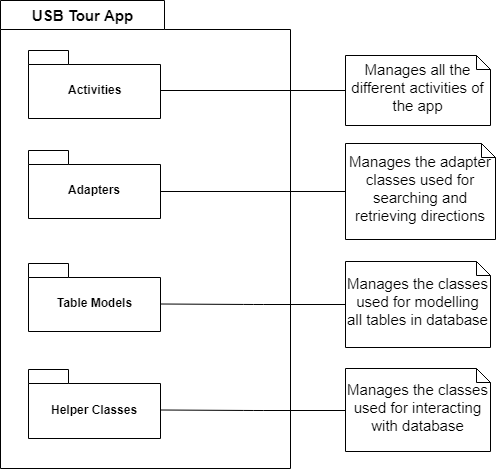
### Package and Class diagrams which show dependencies between components

**Class Diagram**

The class diagram shows all the classes that will be used to make our building tour application. We will start with two interfaces which will be what all our classes come under in order to keep the app together. We will then have a route-finding class which will hold the algorithms to find the rooms and routes within the building. This will then be a parent class for the find a room, get directions and level classes which will be what allow users to get directions to a room or see the rooms on a certain level. This will also include giving the enlarged diagrams of the levels and showing where all the rooms are within that level. Our app will also have a settings class to deal with the apps settings and how the user can change then which include font size and dark mode enabling. Finally, we will have all the search classes, these include search results which outputs the search results for the criteria given by the user, the search tutor class which will output the results when searching for a tutor which includes their name and information about them and finally the search row class. We will also have two adapter classes within the application to find and display all study spaces that users can use to study in and all tutors’ rooms which they can use to find their tutors,



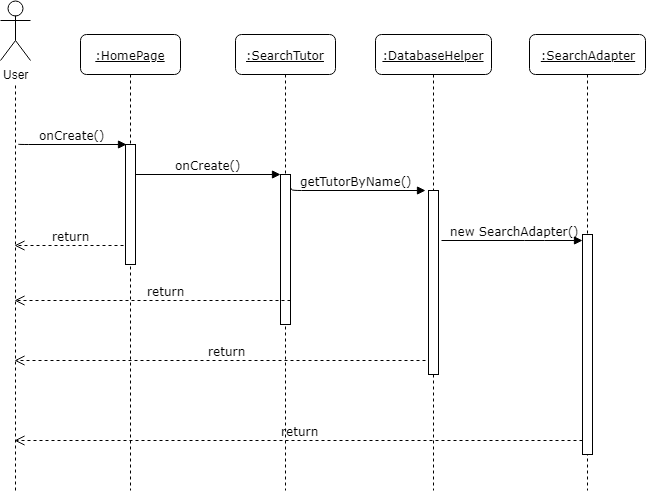
**Package Diagram**



The package diagram outlines the structure and description of all the packages in our project. Our app will consist of the main package that holds the classes for all activities in the app, this includes important classes such as LevelX, FindARoom, SearchResults and GetDirections. The adapters package consists of classes that help to display search results in a recycler view and directions to rooms in a recycler view. A table models package is used to store classes that model the tables in the app database, these classes store important information about each entity in our database including Tutor details, Room details and various Route details. Finally, we have a helper classes package which has a few classes which deal with creating, accessing and modifying our app database and with showing images of a room circled on a map.

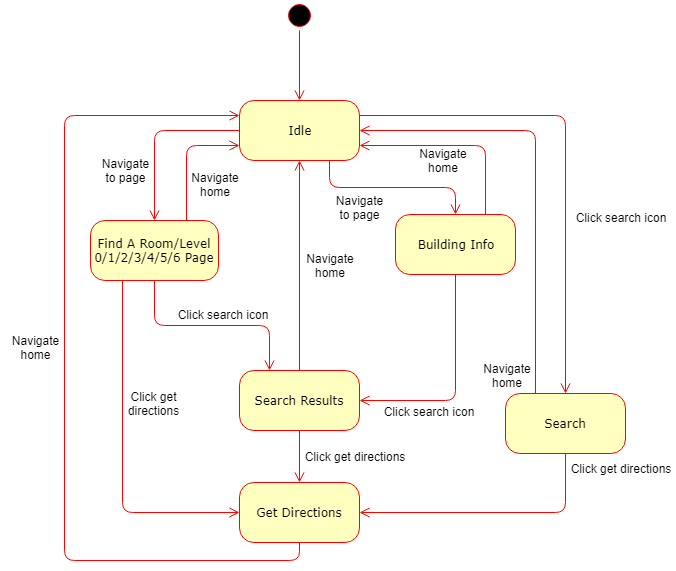
### Dynamic behaviour of the system

**Sequence Diagram of Searching for a Tutor**



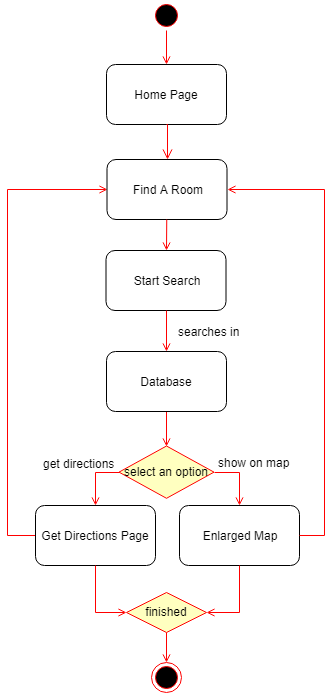
This sequence diagram shows the sequence of object interactions during extended searching in the app. The user can access the home page, then when they click on the search icon, the search tutor class will be instantiated. Then, when the user types in a search query, the database helper class will retrieve the results from the database. A search adapter object is then created to display these results in a recycler view. The user of the app can go back to the home page at any point within this sequence.

**State Machine Diagram**

****

This state machine diagram shows how the app can go from an idle state to any other state. It shows the different pages of the app that can be reached. From the home page the user can navigate to one of: building information page, search page, find a room page and the level page. From the level and building information pages the user can reach search results which will allow them to get directions to different rooms. Throughout the whole application, the user can always navigate back to the homepage to restart their use of the app. The red arrows indicate how the user can navigate between pages, including back to the idle state.

**Activity Diagram of Searching for Rooms**

****

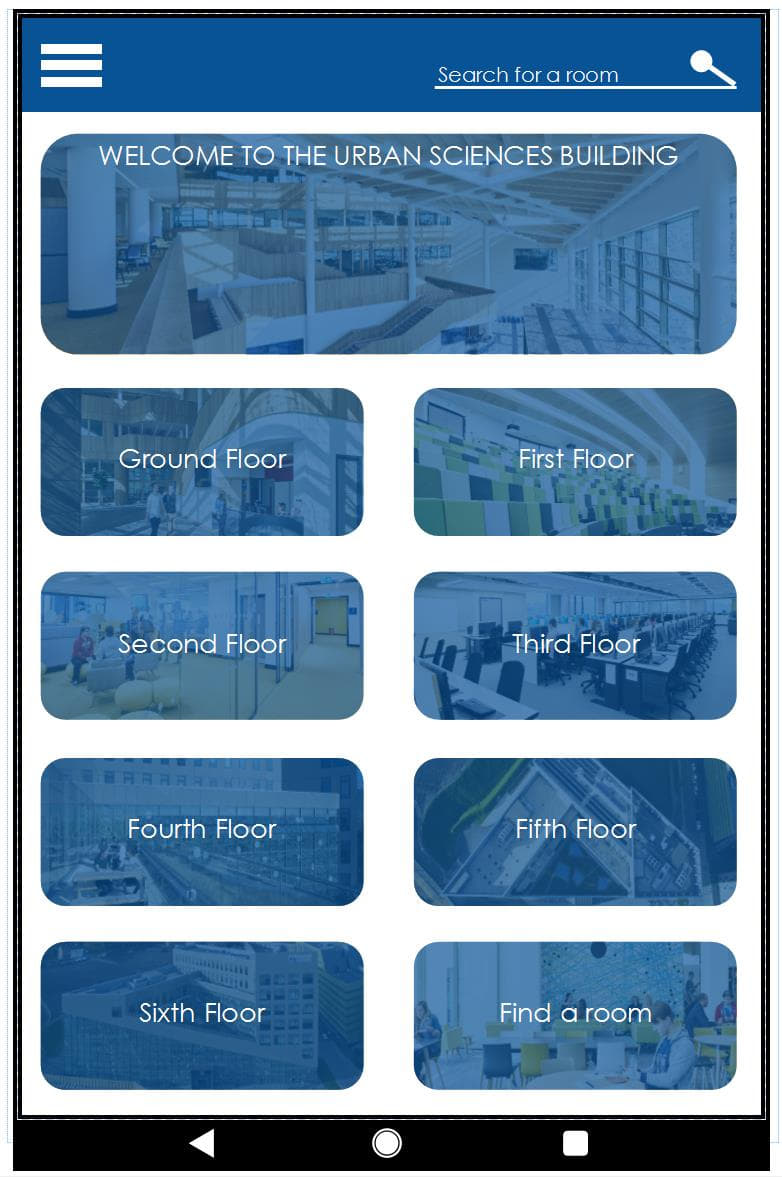
This activity diagram shows the path the user can take in order to reach search results. The user starts from the home page and can navigate to the find a room activity where they can input a search query which will be searched for within the database via the database helper class and the relevant results will be returned. From here the user has two choices: either get directions or enlarge the map to view the location of the room. The user can always navigate back to find a room in order to restart the search.

### GUI, Human Interface Views

The requirements that each page fulfils are show as the requirement number. E.g. [1.1].

Requirements 3.5, 4.3 and 4.5 are delivered on all pages.

**Home Page:**



This page is the first one that the user will see. It is used to all them to navigate to any other part of the app.

The page is a selection of buttons which are:

1. For each of the floors in the USB that links to the level page.
2. A Welcome button that will link to a building information page.
3. The Find a room button which will link to a searchable list of all the rooms within the USB.

There is also options for the user to access the side drawer pull out through the hamburger icon.

There is also the option to search, which has now been updated to allow the user to search a tutor by name. [3.4]

The side menu and search button are consistent throughout all pages within the app.

**Levels:**

**A screenshot of a cell phone

Description automatically generated**

The levels page is a multi-use page that is updated based on what level the user has selected. The colour is also updated based on the USB’s floor colour scheme. (See section 2.2) [3.6]

The page consists of 2 areas:

1. The floor plan. This loads an image of the level’s floor plan over the current page as a pop up. [1.13]
2. The lists of rooms. This is split up into sections to allow for ease of use. The sections shown are Tutor Rooms [1.14] and Study Spaces, however this has now been updated to include another for Other Rooms. [1.8]

**Room finder:**

A screenshot of a social media post

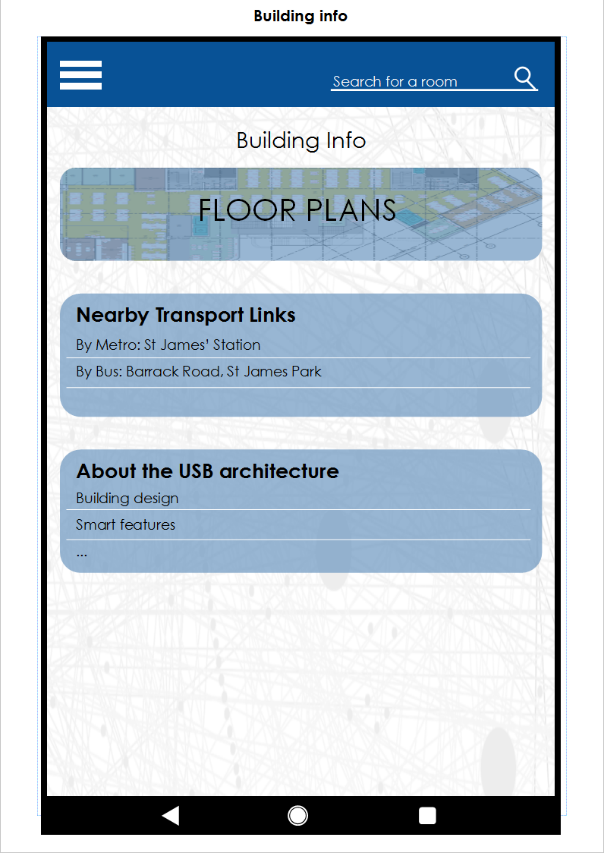
Description automatically generated

The room finder is a page used to allow a user to view all the rooms within the USB. The description of the room is shown along with the room number.

The user can then choose to either:

1. Get directions which will link to the Get Directions page
2. Or use the new feature we have added in of view on map. This will display a pop up of the floor plan with the current selected room circled.

**Building information page:**



The building information page is used to display information about the USB that is more general such as:

* Opening times [1.11]
* Café location and menu [1.2], [1.3], [1.6]
* Details of cluster sizes [1.1]
* General details about the USB [1.12]

This has been extended based on Dragons den feedback to allow the user to get navigation (See section 8.2).

**Search/ Search results:**

A screenshot of a cell phone

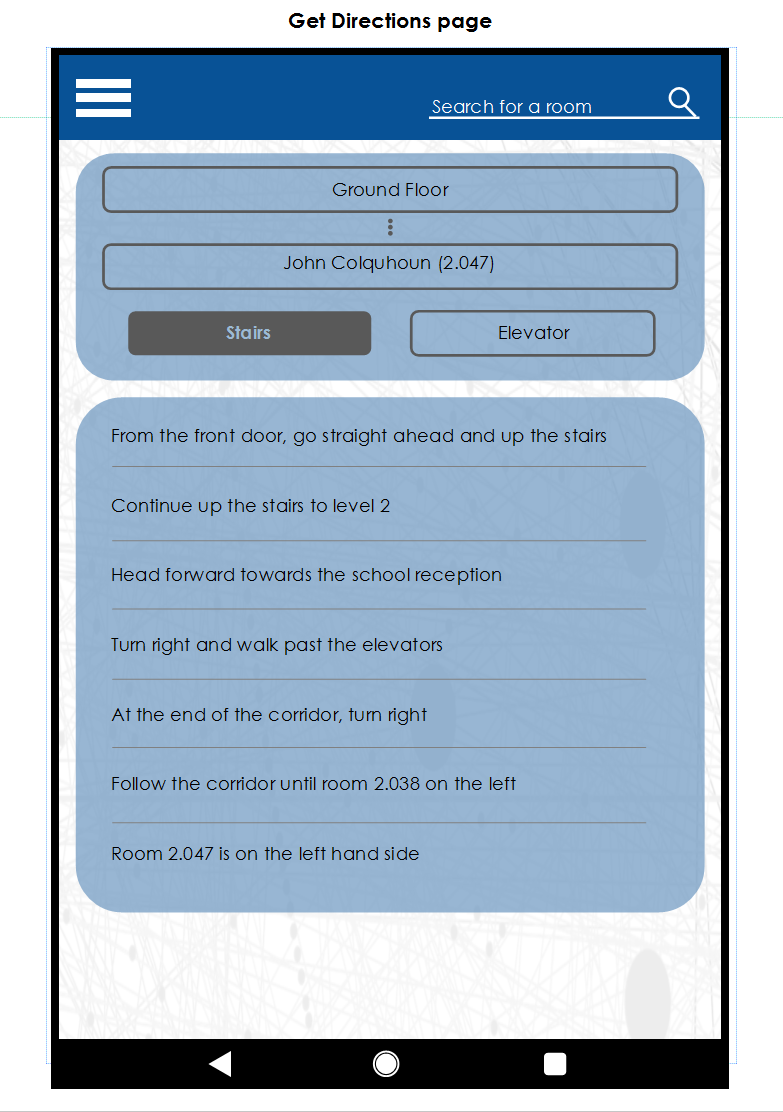
Description automatically generated

The search results when first loaded displays a list of all tutors that are based in the USB. [2.6]

From this point the user can either:

* Select one of the current options and get directions or show on map (both buttons functioning the same as they did on the room finder)
* Use the search button to reduce the results by searching for a tutor by name. Any matches would then be displayed.

**Get Directions:**



Get Directions is the page that allows the user to select two locations and the type of route they need (stairs or elevator) and then view the directions for each step of the route. [1.10] [2.4] [4.2]

**Side Navigation Bar:**

**A screenshot of a cell phone

Description automatically generatedBuilding Tour Guide**

The side navigation bar can be viewed from all pages. It links to all pages in the app. [3.1] [3.3]

[1.7] [1.9] [2.2] [2.3] [2.5]

**Settings Page**

[4.1]

**A screenshot of a cell phone

Description automatically generated**

## References

* Activ, Hello (2018). *Newcastle Castle*. (Version 1.0), [mobile application software], Accessed from Google Commerce Ltd.
* Collinge, R. (2017). *How to Design for Color Blindness*. [online] Usabilla Blog. Available at: https://usabilla.com/blog/how-to-design-for-color-blindness/ [Accessed 23 Nov. 2018].
* Google (2019) *About the platform, Platform versions.* [online] Android Developers. Available at: https://developer.android.com/about/dashboards/ [Accessed 6 Feb. 2019].
* Nasta, S. (2017). *Common Mobile Application Accessibility Issues to Avoid (Checklist)*. [online] Microassist. Available at: https://www.microassist.com/digital-accessibility/mobile-application-accessibility-part-2-of-2/ [Accessed 23 Nov. 2018].
* Show app, Action (2016), *Action tour Guide. (*Version 1.18), [mobile application software] Accessed from Google Commerce Ltd.
* Vusiem Limited (2015). *Natural History Museum Guide (Version 2.7.6).* [mobile application software]. Retrieved from Google Commerce Ltd.

## Definition of terms

No terms are used in this document that require further definition.

**Contribution matrix:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Section | Will Comber | Edward Phillips | Ethan Goodchild | Imran Mohammed | James O’Connor |
| Purpose | CM |  |  |  |  |
| Background and analysis |  |  |  |  | CM |
| Roles and deliverables | M |  |  |  |  |
| Project plan |  |  |  | CM |  |
| Hardware and software platforms |  |  | CM |  |  |
| Solution requirements | CM | M | M | M | M |
| Other considerations |  | CM |  |  |  |
| Software design |  |  |  |  |  |
| References | CM |  |  |  |  |
| Definitions |  |  |  |  |  |

Key:

C = Create

M = Modify

R = Review

T = Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Section | Joe Lawrence | Joe Lowthian | Oscar Gilley | Titas Alvikas |
| Purpose |  |  |  |  |
| Background and analysis |  | CM |  |  |
| Roles and deliverables |  |  | CM |  |
| Project plan |  |  |  |  |
| Hardware and software platforms |  |  |  |  |
| Solution requirements |  | M |  | M |
| Other considerations |  |  |  |  |
| Software design | CM |  |  | M |
| References |  |  |  |  |
| Definitions |  |  |  |  |