

**OFFICE FINDER**

Team 23

### 

# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No | Version Date | Revised By | Description |
| 1.0 | 30/10/2017 |  | * Document Purpose/Executive Summary * Project Purpose |
| 1.1 | 14/11/2017 |  | * Deliverables * Team Roles |
| 1.2 | 28/11/2017 |  | * Project planning * Task Breakdown |
| 1.3 | 13/12/2017 |  | * Hardware and Software requirements for End Users |
| 1.4 | 28/01/2018 |  | * Domain Analysis * Overview of what the software will do and not do |
| 1.5 |  |  | * Functional requirements * Non-functional requirements * Assumptions * Definition of terms |
| 1.6 | 09/03/2018 |  | * High-level overview of Functionality and responsibilities of the system * Package and Class diagrams. * Sequence diagrams * GUI, Human Interface Views * Test plan |



# Table of contents

Version History Page 2

Table of contents Page 3

Document Purpose/Executive Summary Page 4

Domain Analysis Page 4

Project Purpose Page 5

Deliverables Page 5

Project Planning Page 6

Team roles Page 7

Project task breakdown Page 8

Hardware and Software Requirements Page 9

References Page 10

Definition of terms Page 10

Functional Requirements Page 11

Non-Functional Requirements Page 13

Assumptions Page 13

Constraints and Dependencies Page 13

Software Design Page 14

Test Plan Page 27



# 

# Document Purpose/Executive Summary

The purpose of our project specification is to completely break down every little thing that needs to be considered and then later accomplished to fully complete development of the application.

The specification provides the team with complete analysis of the application and its development process. It forecasts all the errors and both technical and ethical problems that the team might run into while developing the app.

All of the sections are edited by team members which are experienced and knowledgeable in their respective fields. This lets the team be fully operational and as efficient as possible, knowing that most of the issues in every single stage and field of development are already known to the team even before the development starts.

If everything mentioned in the specification is successfully accomplished, it will make the application very efficient, flexible and reliable to the greatest extent possible.

# Domain analysis

The apps we are looking at for Inspiration are ‘Google Maps’, ‘Windows Maps’, and ‘Apple Maps’ as these apps as we wanted to get a range of different map/route finding apps, including those not on Android, so we chose ‘Apple Maps’ from IOS, ‘Google Maps’ from Android, and ‘Windows Maps’ from Windows phones

We looked at the similarities between the apps to see what features would be essential to this kind of app, all three included a way to specify both the end location and starting location of the route the user is trying to find.

As this is an essential part of these types of apps we made sure to include it when discussing features, we are going to implement.

All three apps also have a 2D map that is displayed to the user, this allows the user to look over the route before deciding to use it, we agreed that a 2D map in our app would increase usability, we also decided to allow text instructions, for people who want to reserve phone power in case they are low on battery life.

allows the user to look at the route/Map from the perspective of someone walking/driving along the route. We considered implementing a feature similar to Google Street View as we believe it would allow the user to be able to understand the directions more easily. However, due to time constraints, we believe this task would also be too difficult to complete in the current time frame.

All three of these apps included some kind of location service that can be used to direct the user to their chosen destination from where they currently are, without having to enter their current location. After researching this we concluded that it would be difficult to implement as the building is not very large compared to the maps used by the apps we looked into and would face difficulties such as being unable to find the user’s location in a suitable range and would take too much time to implement.

Although we are taking inspiration from these apps, they are finding routes outside, using roads, not inside of buildings like our app is. As these apps are not exactly like the app we are developing, we took considerations when thinking about the design and functionality. As our app has to deal with different floors, we decided to have each be a different map the user can swap between when looking at their route.

# Project Purpose

Large buildings that have a lot of offices and facilities are very convenient in space but when it comes to getting around one, that's when they can become troublesome. The main issue with them is that you can get lost easily or end up in a place you weren't supposed to. This project's goal is to simplify the task of finding your way around an enormous building with the ease of a few clicks on your phone.

The application is aimed to help people who are not familiar with places in the Urban Sciences Building. It will be able to direct people to the location they want in steps and it will also be able to tell users their current location if they are lost.

With the building being new, no one is yet fully aware where everything is situated and there can be lots of confusion. Not only that, but people from industry or visitors would benefit from the instructions that the app will provide. For example, if there was a business meeting or an open-door event, then the person attending can just type in the meeting, lecture room or office to find the needed location.

# Deliverables

**Android application**

The main deliverable of our project is the office finder android application which we will create. We will test the application throughout the implementation, as well as afterwards, according to our test plan. We aim to finish the implementation of the application by mid-March 2018, which will leave us over a month for testing and amendments before the final deadline on 23rd April.

**SQL database**

In parallel with developing the Android application, we will create an SQL database to supply and store information for the application. It will be hosted on the web space given to us by the university, and we will use phpMyAdmin (or an alternative provided by the university) to maintain the database.

**Website**

We will create a website with the aim of marketing the application and providing information about it, aimed at the end-user. The website will also include a user manual which will give full instructions for use of the application.

**Poster**

On the 16th March, we will hand in a poster used to help promote our app at the trade fair.

**Presentation**

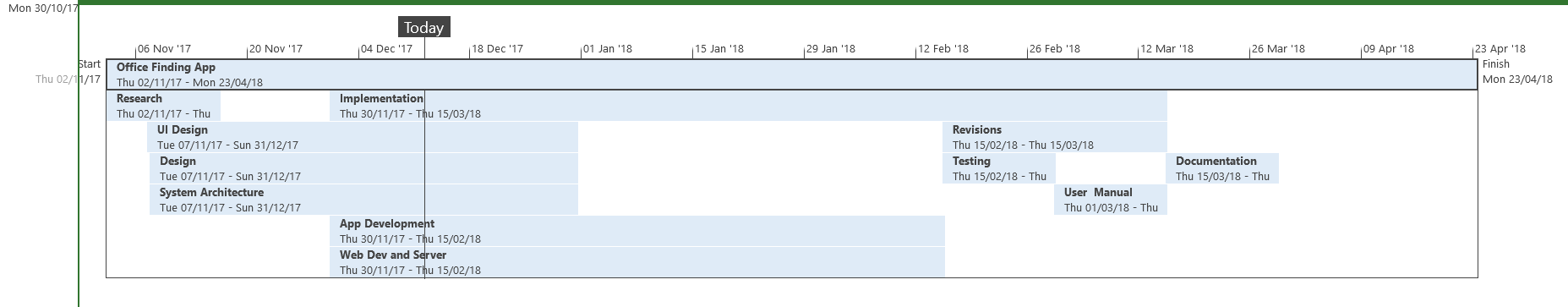
We will give a final presentation on 10th May, for which all team member will be present, to showcase our work and provide a technical demonstration of the application we have developed.

# Project Planning

## Project Goal

*“To follow the software development cycle and AGILE methodologies, working as a team to develop a mobile application that will enable a user to find their way to offices around the USB building.”*

## Project Schedule

The project schedule is a Gantt chart developed in Microsoft Project. Below is its first iteration:

## Supporting Plans

In order to ensure the efficacy of our team and the success of the project, various measures have been taken:

* We have assigned team roles (please see ROLES section on page).
* Ensured that attendance and minutes are taken at each meeting.
* There is a contingency plan to ensure that any issues are dealt with preemptively.
* We wrote a team contract that was signed by all team members, ensuring that we all work to an agreed set of standards.

## Project Cost and Scope

This project’s main cost will be people power and time. As team members are working on their own time and the technologies we are using are free, there is no large concern for cost other than time.

Future scope of the project beyond our immediate goal includes the app being easily adaptable for use in other environments. This could be in other buildings at the university, other office buildings in the industry or even schools, hospitals and airports to name a few.

# Team Roles

*All members are expected to contribute at every level of the team to ensure maximum efficiency and support all team members.*

**Team Leader**: Charlie Lockwood

*Responsible for setting up meeting agenda, attending team leader meetings and helping out teammates who are struggling.*

**Head of Research**: Marios Kovalenko

*Responsible for researching topics that are introduced in team meetings.*

**Head of UI Design**: Jonathan Sunarja

*Responsible for the design and user interface for the app.*

**Head of System Architecture Design**: Thomas Moore

*Responsible for creating the conceptual model that defines the System design.*

**Head of Application Development**: Liam Piesley

*Must ensure that all departments are working properly for the development of the app.*

**Head of Web and Database Development** Joe Cunningham

*Responsible for developing the web and database for the app.*

**Head of Black Box Testing**: Asen Koparanski

*Must check if the app is working properly without checking the inner classes.*

**Head of White Box Testing:** David Towers

*Testing and checking if all inner methods and classes are working properly*



**Head of Documentation:** Matas Zilaitis

*Responsible for managing the group’s documentation.*

**Minor Roles**: *Responsible for attendance at meetings - David Towers*

*Responsible for taking minutes during meetings - Liam Piesley*

# Project Task Breakdown

As stated above under the section ***Team Roles***, all team members are working and contributing to every aspect of the project. However, as agreed per team contract and meetings, main responsibilities for certain areas of the project are allocated to one or some group members. The table below specifies which subsets of the group are responsible for certain parts and deliverables. For formal role division and definition, see ***Team Roles***.

|  |  |
| --- | --- |
| **Project Task** | **Main responsibility of** |
| Research | Head of Research |
| * Ideas feasibility |  |
| * Web development, server, and database | Head of Web and Database Development |
| * Similar apps |  |
| Design |  |
| * UI design | Head of UI Design |
| * Sketches and mock-ups |  |
| * Application tree |  |
| * Colour coding |  |
| * Test plan for UI | Head of Black Box Testing  Head of White Box Testing |
| * System architecture | Head of System Architecture Design |
| * UML class diagram |  |
| * Pseudocode |  |
| * Test plan for system architecture | Head of Black Box Testing  Head of White Box Testing |
| Implementation |  |
| * Application development | Head of Application Development |
| * Web development and server | Head of Web and Database Development |
| * Document hosting |  |
| * Database |  |
| * Test plan for web development and server | Head of Black Box Testing  Head of White Box Testing |
| Testing |  |
| * Black box | Head of Black Box Testing |
| * White box | Head of White Box Testing |
| * Console | Head of Black Box Testing and Head of White Box Testing |
| Documentation | Head of Documentation |
| * Tests |  |
| * Console commands |  |
| * Implementation |  |
| * User manual |  |
| Final spec | Head of Documentation |
| Submit final spec | Team Leader |
| Final report and log | Team Leader |
| Individual report and log | All team members |

# Hardware and software requirements

The app is not universal and has certain requirements that the user needs to have before the app can be used, the biggest requirement is the User needs a device that runs on Android, as we have developed for Android 4. We are supporting the large majority of Android devices, as Android devices are backwards compatible with earlier versions.

While we want to support most Android devices, the user must use a tablet or phone instead of other gadgets that run Android, for example, we are not supporting any smart watches that run Android as we do not believe the app will be as usable on a smart watch device.

The end user also requires a device that is able to connect to the internet for either the duration of following the root or for the initial download of the route before following. This is because the data for the route is stored on our server, so the user needs to be able to keep a constant connection to retrieve the directions as they are followed or allow the download of the entire route so that it can be followed without constant internet access.

If the user wishes to download the route instead of staying connected to the internet, they must have enough storage space for the route to be stored, if they do not have enough space, the route will not be able to be stored and therefore the app will have nothing to retrieve to give directions.

Finally, the app is not very resource demanding. As such, the user does not need a phone with a fast processor or a large amount of RAM, the app will with a moderate amount of processing and a moderate amount of RAM.

# References

* Eclipse Foundation(2004) Eclipse.org. Available at: http://www.eclipse.org/ (Accessed 20 November 2017).
* Google(1998), jetBrains(2000) Developer.android.com. Available at : https://developer.android.com/studio/index.html.
* phpMyAdmin (1998) phpMyAdmin. Available at: <https://www.phpmyadmin.net/> (Accessed 2 February 2018) .
* MySql(1995) Mysql.com. Available at: <https://www.mysql.com/> (Accessed 2 February 2018).
* Google Maps(2013) Google Maps. Available at: <https://maps.google.com/> (Accessed 10 November 2017).
* Windows Maps (1975) Microsoft Store. Available at: <https://www.microsoft.com/en-us/store/p/windows-maps/9wzdncrdtbvb> (Accessed 12 November 2017).
* Apple Maps(1976) Apple. Available at: <http://www.apple.com/ios/maps/> (Accessed 14 November 2017).

# Definition of Terms



|  |  |
| --- | --- |
| **Term** | **Definition** |
| Page | When discussing the mobile app, we use the word page, to describe the different views the user sees and can change between |
| Route | For our purposes, a Route is a path with a starting location and a destination, that has directed instructions on how to get from the starting location to the destination |
| Map | We are using ‘map’ to mean a graphical representation of the building/route. |
| Out of Hours | The building we are making the app for does not yet offer 24-hour access, Out of Hours represents the period of time after the school closes, but when students are still allowed to access most of the building. This time period is 18:00 - 23:00 |
| User | A User is anyone who may use the app |
| Mobile Data | In this document when we use mobile data we are referring to a 3G/4G that allows a mobile device to access the internet without a Wi-fi Connection |

# Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Importance** | **comments** | **Used by (System or User)** |
| **1 - App** |  |  |  |
| 1.1 - App will find routes between two locations | H | The app will find routes based on entered starting and finishing location | User |
| 1.2 - Route finding restrictions | M | Certain users have access to different areas, the app will determine the best route given these requirements | System |
| 1.3 - App will display the user’s route by a 2D map | L | Allows the user to look at the route beforehand | User |
| 1.4 - App will display text directions for the route | M | Text and map can be switched between | User |
| 1.5 - App will change the route based on the preference of stairs and lifts | M | The app will have a button that allows the user to direct them towards using the stairs or lifts | User |
| 1.6 - App will find a route to a staff members office, from a staff members name | L | The app will not tell the user if the staff member is there, it will only direct them to the office | User |
| 1.7 - Location from photos of room numbers | L | The app will be able to determine the user’s starting location by taking a picture of the room number | User |
| 1.8 - Out of hours access | M | The app will avoid certain paths and rooms during university Out of Hours, due to the closing of rooms | System |
| 1.9 - Floor swapping | H | The app will allow the user to swap between floors to look at the map/route | User |
| 1.10 - Storing and retrieving routes | M | If there is a route the user wants to keep, the route is sent from the app to be stored on the database | User |
| 1.11 - Logging In | M | The app will allow the user to login using their university ID and password | System |
| 1.12 - Accessibility | H | The will allow a user to change the colour scheme to something more suitable for them. This means people who are colour-blind can use this feature to change the colours to those that they can distinguish between easily |  |
| **2 - Website** |  |  |  |
| 2.1 - The website will advertise the app through images of the app | L | The images will be showing off the designs and functionality | System |
| 2.2 - The website will host the database for the app | M | The database will contain information such as staff offices, and user routes | System |
| 2.3 - Logging in | M | The database will allow an admin to log in to manage the database | User |



# Non-Functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Importance** | **comments** | **Used by (System or User)** |
| 1.1 - Route Finding efficiency | H | Routes found by the app will be the shortest route | System |
| 1.2 - Security | H | App and website will be secure, in accordance with the data protection act | System |
| 1.3 - Wi-Fi/Mobile Data | H | To make sure Locations are up to date, like Staff Offices in case of a change of office the app requires an internet connection of some sort | System |
| 1.4 - The app must be fast loading | L | The app should load in under a second | System |
| 1.5 - Transitions between pages must be smooth | L | The user should not notice any stutter when switching between pages | System |

# Assumptions

In our development of the project, for general simplicity we have had to assume the following:

* All users of the application fall into the category of either staff, student or guest
* All students are either undergraduate or postgraduate
* All students are either undertaking a bachelor’s degree, a master’s degree, or a PhD
* A student can only be classified as participating in one of the above degree programs at a time
* All staff have the same access permissions as each other, and they all have access to the second lift
* All users are carrying their smart card at the time of use (this is required to access many areas of the building)
* All users will have an Android smartphone running Android 7.0 (Nougat)
* Users will have constant internet connection throughout application use
* When a user requests directions to a location, they want to find the fastest/most direct route, unless certain parameters are specified (such as no stairs)

# Constraints and dependencies

The application will be limited to Android phones which, according to a recent study, hold a large majority of the global market share in smartphones **1**. The application will be designed to run on Android 7.0 (Nougat), as that is the version of Android running on our test phone - a Moto G5. The app should run without problem on other versions of Android, but it cannot be guaranteed, especially as the Android version goes further away from 7.0. With more time we would have loved to have made an iOS port of the app, especially due to the high number of Apple phone users in the UK **2**, however, we felt that this would have been outside the scope of the project guidelines, and also due to time limitations would not have been possible. It may be necessary for the user to have constant internet connectivity during app use, but this is very likely thanks to the ubiquitous Wi-Fi availability inside the USB. The only time this may pose a problem is when the user is a guest to the university and is, therefore, unable to log in to the Wi-Fi network.

# Software Design

## Overview of what software will and will not do

The software will allow its users to find the shortest available route to any office, teaching room or a facility within the building. The app will help people to navigate around the newly built Urban Sciences Building and will be adaptable to other buildings as well.

The software will not allow users without credentials to use the staff lift, this prevents people who are not authorised from using it.

Every user will have an option to choose stairs or an elevator as their primary travelling method, this feature will mostly help disabled users to navigate around the building.

The software will allow users to change the colour scheme of the app, letting users personalize their experience.

The software will not allow users to search for routes to certain rooms and facilities at certain hours based on their permissions. This feature will make sure that the app is following the official building rules.

The software will allow users to choose between two route options: text instructions and images with arrows for every step. This will help users to choose the information display method that works best for them.

Every user will have an option to search for a room by its number or a staff member’s name. This option will help users to find a room only knowing some details about it.

The software will allow users to choose the starting point to be anywhere in the building, it will make the application more flexible and allow the user to search for a route between any two rooms.



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

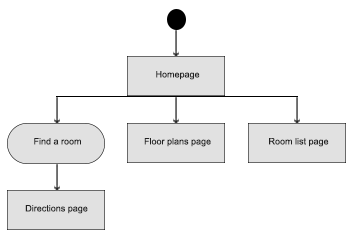
After the Dragon’s Den presentation, we received feedback on our project. One area that they suggested we investigate further was accessibility. As such we decided that the colour scheme should be adjustable to allow colour-blind people to use colours that they can see.

We also investigated what we wanted to do with the app and the time available and concluded that some features would need to be dropped, We originally had plans to have a feature like Google Street View, but when looking at our timeframe we decided that going further with the idea might hinder other more important parts of the project.

When making decisions about any change of direction, we would usually ask team members to do some reflection before discussing our opinions in a team meeting. This ensured that we would hear a variety of informed voices before reaching a final decision.

## High-level overview of the system and how objects interact

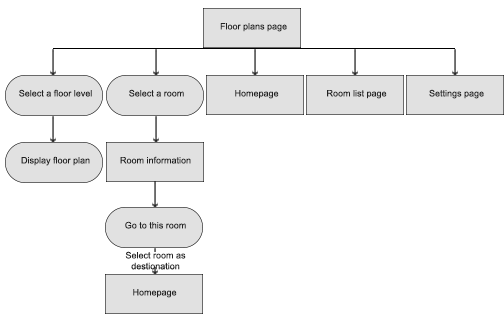
**Homepage**

****

Homepage, also acting as the start page on the app, consists of links to floor plans, room list pages and a user interaction component in the form of find a room box. Performing the latter action will launch directions page as a result.

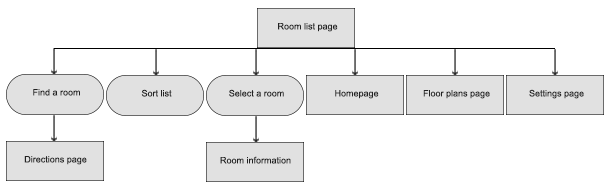


**Floor plans page**



Floor plans page consists of links to homepage, room list page and settings page. On the top left area of the screen, the user can select a floor level through the use of a drop-down menu and its floor plan will be shown. Clicking a particular room on the floor plan will call room information to be displayed.

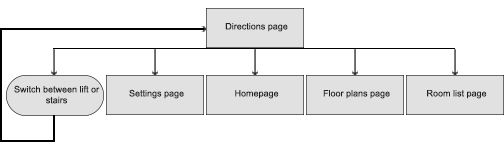
**Room list page**



Room list page consists of links to homepage, floor plans page, and settings page. On the top area of the screen, the user can use the find a room box to select or type in starting point and destination. Below this box, a list showing staff names and their corresponding room numbers can be seen. The user can sort the list in ascending or descending alphabetical order using the sort button next to the title of each column.

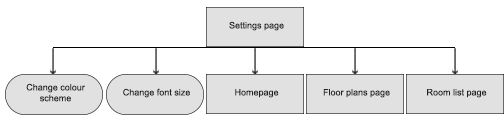


**Directions page**



Directions page consists of links to homepage, floor plans page, room list page and settings page. This page simply gives directions to users to get from chosen starting point to chosen destination. The user may wish to switch between using the lift or stairs, so directions given can be suited accordingly by clicking on the toggle just below the room name and the page will refresh.

**Settings page**



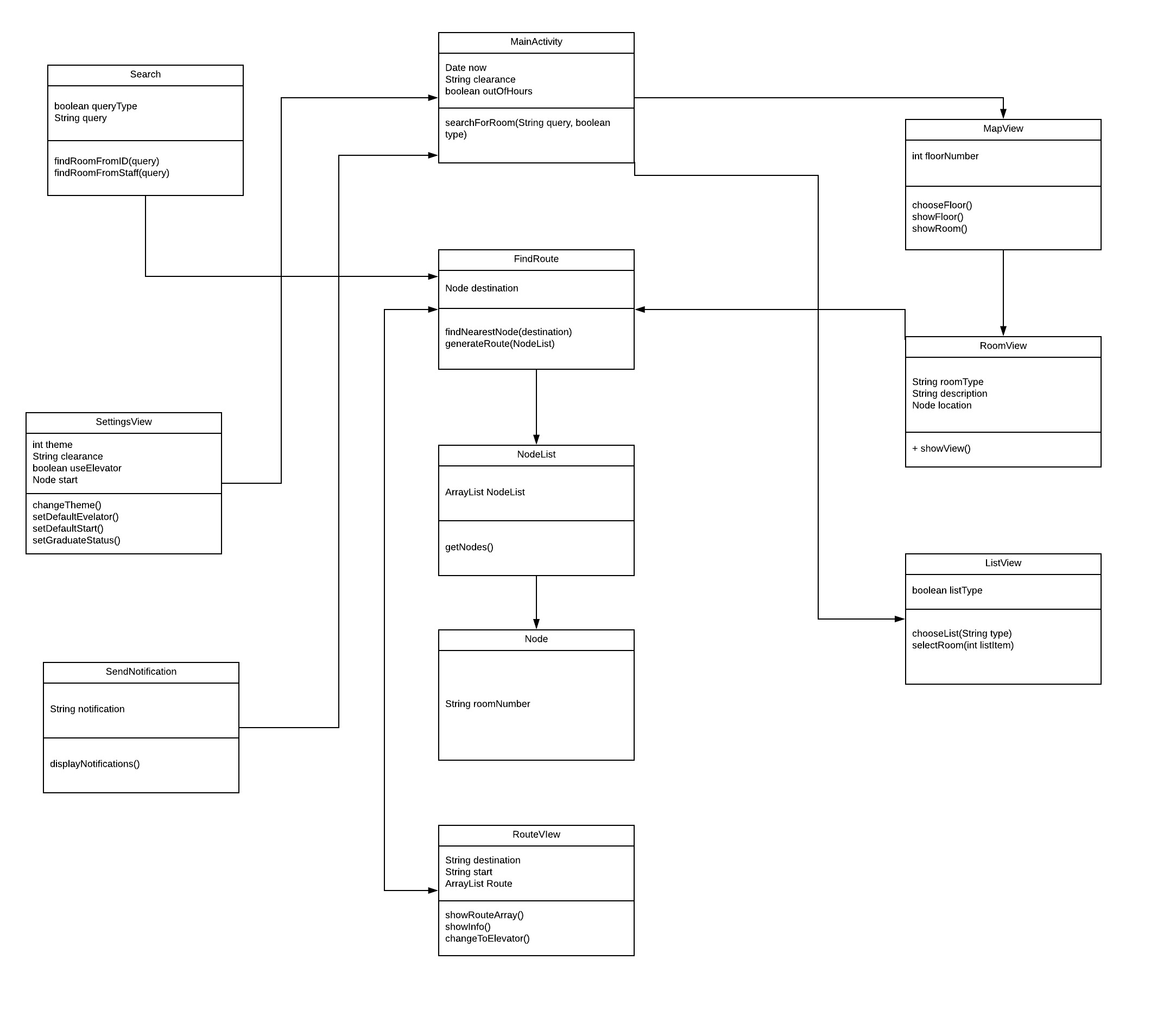
Settings page consists of links to homepage, floor plans page and room list page. On this page user can perform two actions: change the colour scheme and change font size.

## 

## Package and Class diagrams which show dependencies between components

## Package Diagram

## 

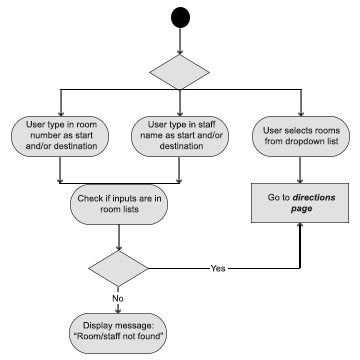
**Class Diagram (UML Diagram)**



## 

## Dynamic behaviour of the system

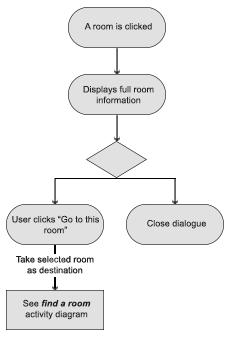
Apart from switching between pages, there are 4 main activities on this app (explained below with a diagram and description):



**Find a room**

Find a room is an action that user can perform while accessing the homepage or the room list page. The user can choose a starting point and destination from dropdown lists of rooms or simply by typing in a room number or a staff name for starting point and destination. If the user chooses to type, the system will check whether inputs typed in matched with any room number or staff name stored. Valid inputs will launch the directions page. Meanwhile, in the case of invalid inputs, the system will display an error message telling the user that room number or staff name cannot be found.

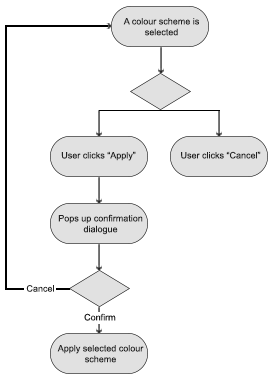


**Showroom information**

The user may also click on a room on a given floor plan or in the room list which will tell the system to display full room information. This dialogue can be closed simply by clicking on the close button at the top right of the window. However, if the user wishes to go to this room, clicking the “Go to this room” button will copy the room number and put it as a destination in the ***Find a room*** activity above.

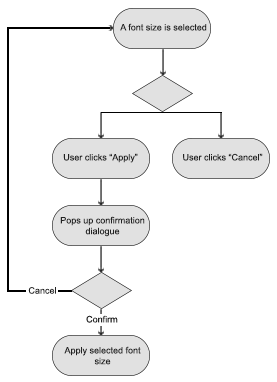


**Change colour scheme**

On the settings page, it is possible for the user to change the colour scheme used in the app from a selection of colour schemes. After selecting a colour scheme, the user needs to click “Apply” to change it. A confirmation dialogue will ask the user to confirm the selection and user simply needs to click “Confirm” to apply the change. Clicking “Cancel” takes the user back to the settings page with the aforementioned colour scheme still selected. To cancel any changes made, the user can click “Cancel” on the settings page. 



**Change font size**

Another action that can be done on the settings page is to change font size. After selecting a font size, the user needs to click “Apply” to change it. A confirmation dialogue will ask the user to confirm the selection and user simply needs to click “Confirm” to apply the change. Clicking “Cancel” takes the user back to the settings page with the aforementioned font size still selected. To cancel any changes made, the user can click “Cancel” on the settings page. 

## 

## 

## 

## 

## 



## GUI, Human Interface Views – Describe how the user accesses functionality

Figure 9.7.1

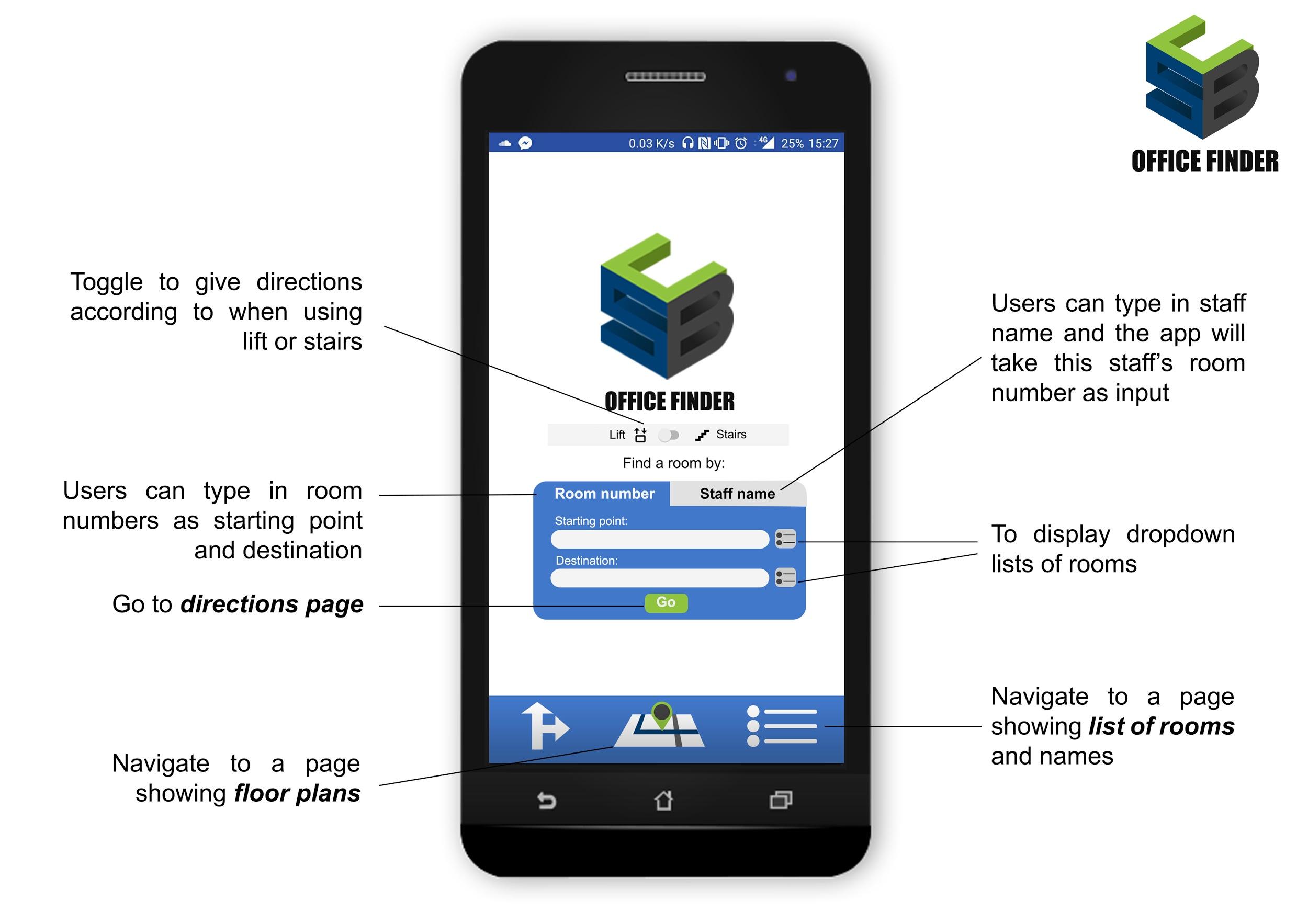


Figure 9.7.2

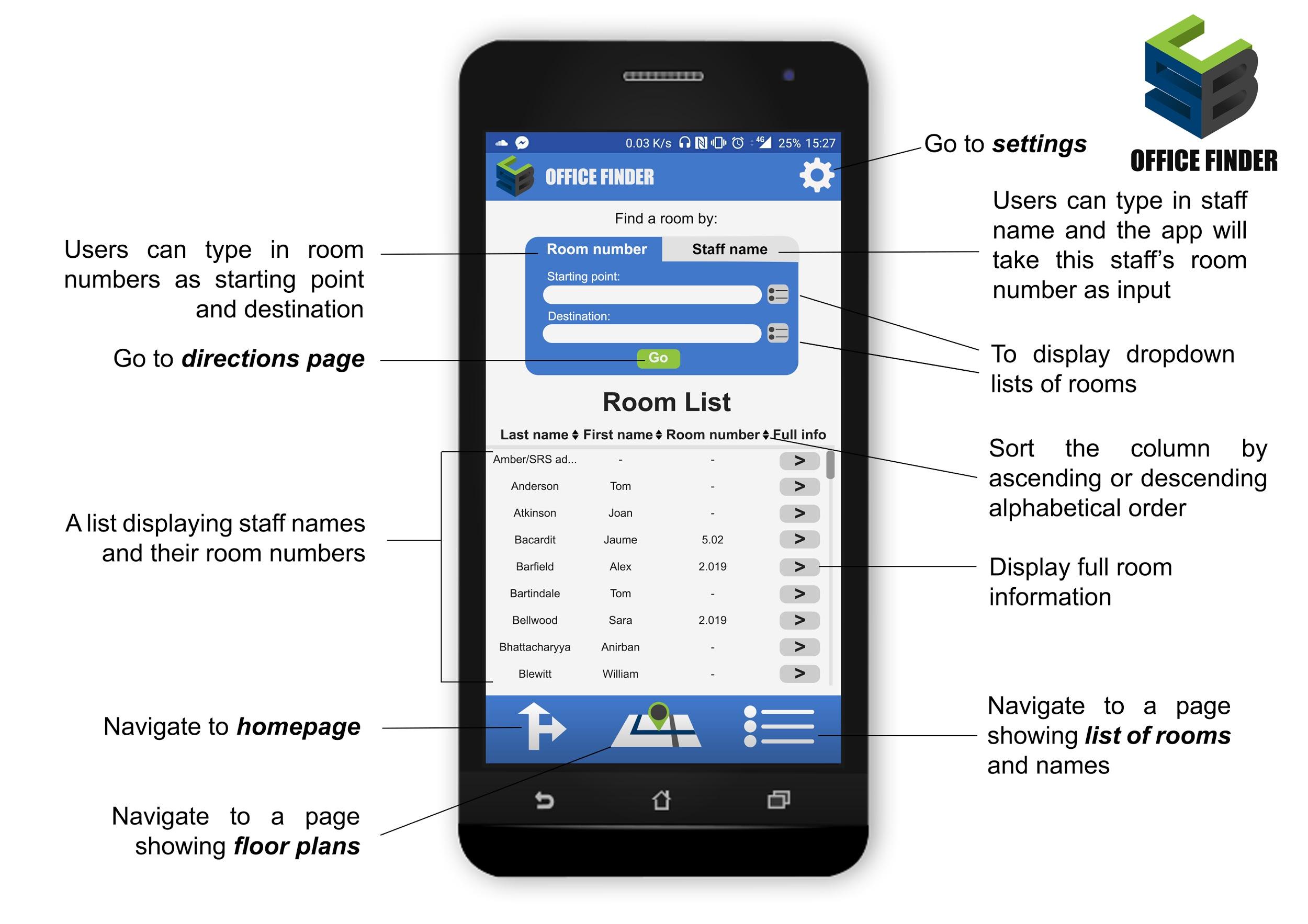


Figure 9.7.1: Is the main page of the application.The user has options to choose if he wants to use stairs or lift in the middle of the screen, lift is set as default because elderly people or people with disabilities might not be able to Use the stairs. In addition, the user can search for a room or find it by searching the name of a staff member in the search boxes in the middle of the page. On the Bottom right of the page, User can be navigated to another page where there is a list of Rooms (Figure 9.7.2). Additionally, on the middle bottom, a user can be navigated to a floor plan page (Figure 9.7.5).



Figure 3.2: Shows the Room List Page Where User can find a room manually and use the arrows on the middle of the screen to list names in alphabetical order based on last name, first name or arithmetic order for Room Number. Each Room has information button which transfers the user to a page containing information regarding the specific room shown in Figure 9.7.6. Settings option also available on the top right corner which will take the user to the settings page (Figure 9.7.4). In addition, User can use the options on the bottom of the page to navigate to the homepage (Figure 9.7.1) and floor plans page (Figure 9.7.5).

Figure 9.7.3

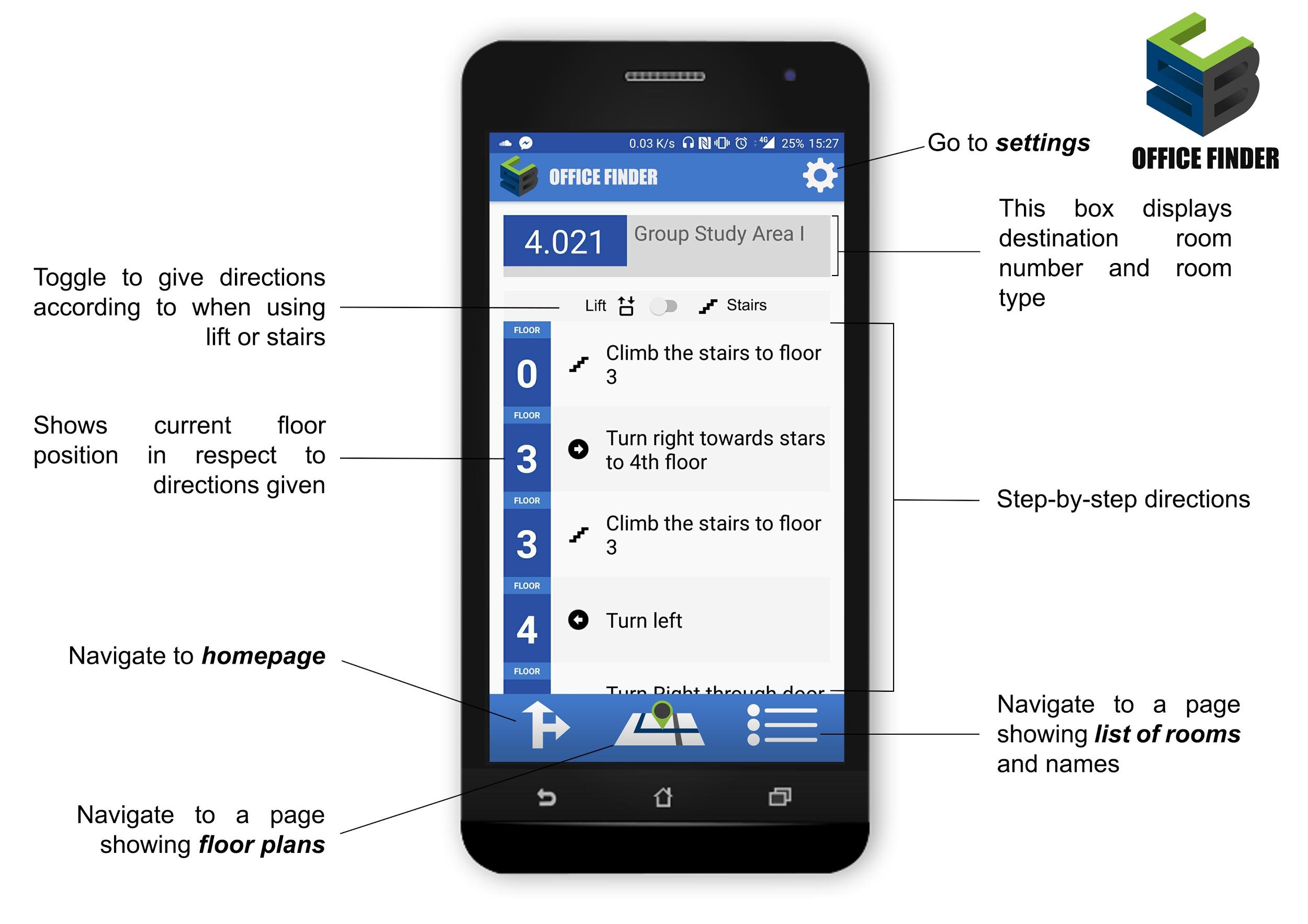


Figure 9.7.4

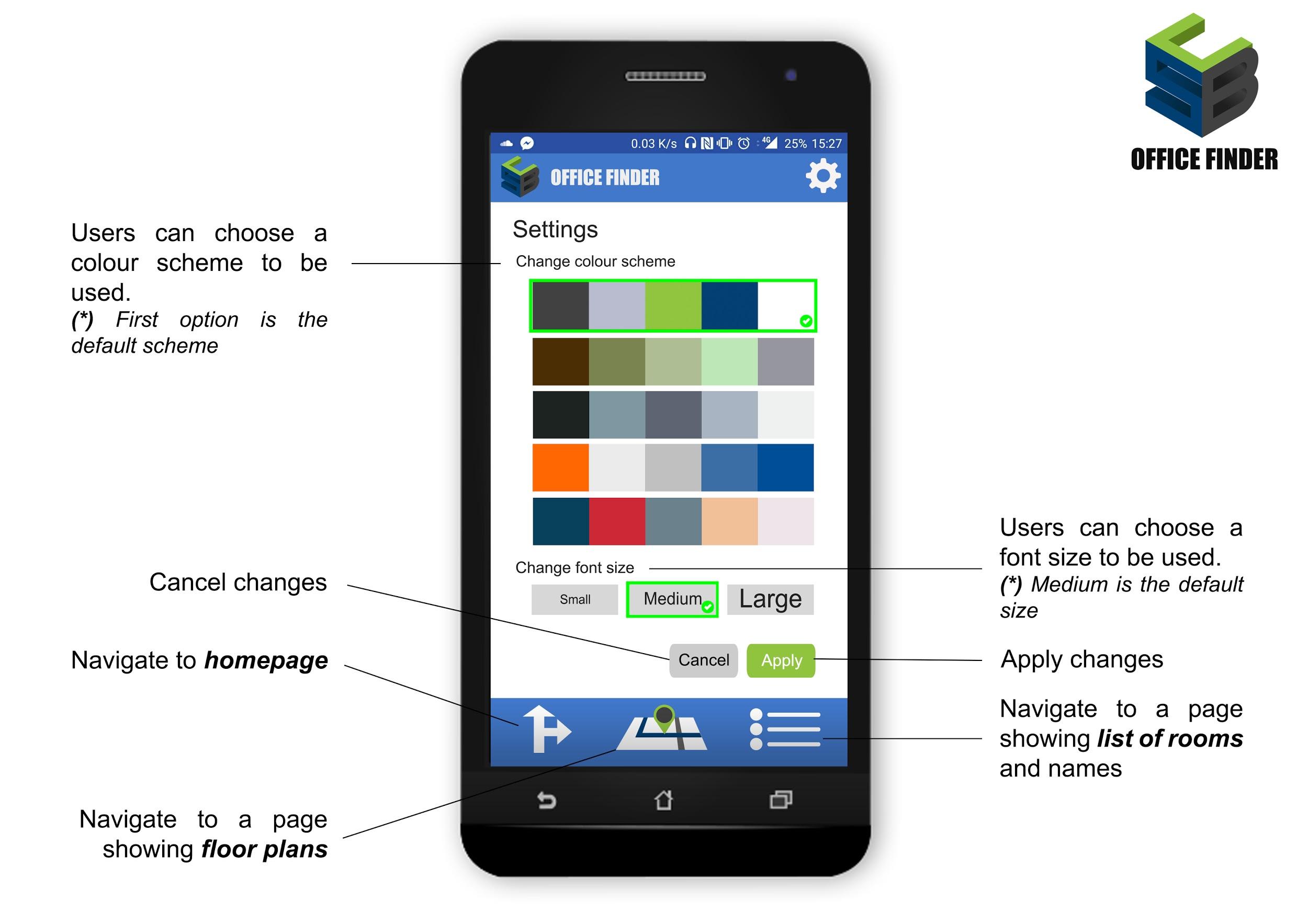


Figure 9.7.3: Page that is user access when searching for a Room. On the top, the destination and its details are displayed. Below step by step directions are displayed and at the bottom, the user has the option to navigate to the homepage shown in (Figure 9.7.1), floor plans shown in (figure 9.7.5) and list of Rooms pages (Figure 9.7.2). Lastly on the top right corner, a user can navigate to the settings page (Figure 9.7.4).

Figure 3.4: The settings page gives the option to User to change the colour, font size and choose to apply or cancel any changes made. Default colour is set to blue as it is the university colour and font size medium. Additionally, on the button of the page, a user can navigate to the homepage (Figure 9.7.1), floor plans (Figure 9.7.5) and list of rooms pages (Figure 9.7.2).

Figure 9.7.5



Figure 9.7.6

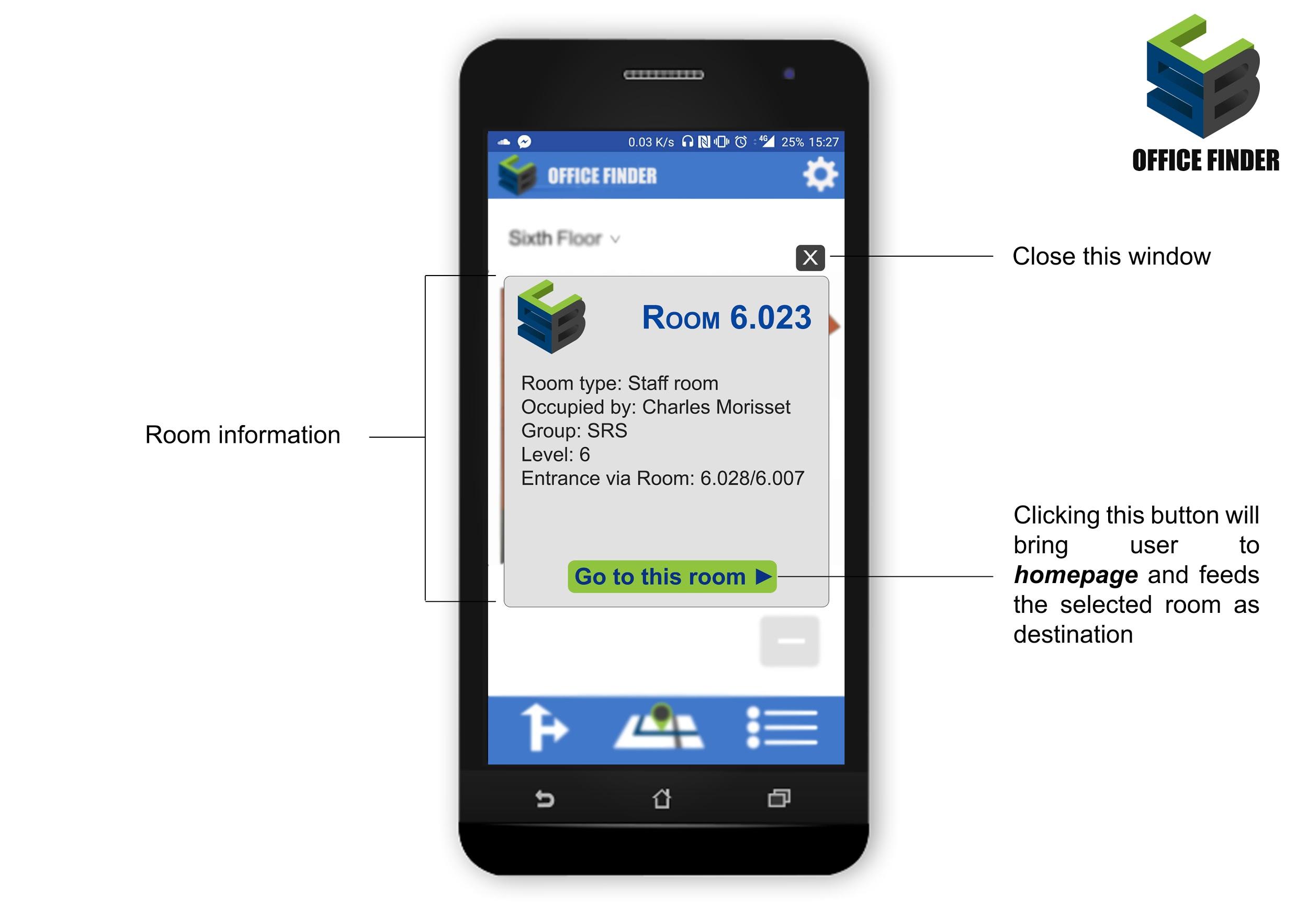


Figure 9.7.5: The Floorplans page, User can select on the top right the floor map he wants to inspect, the map is displayed in the middle and a user can zoom in and out. In addition, a User can go to the settings page (Figure 9.7.4) using the settings button on the top right corner and on the bottom can navigate to the Homepage (Figure 9.7.1) and list of rooms pages (Figure 9.7.2).

Figure 9.7.6: Information window that shows information regarding the room that a User selects. A user can navigate back to the homepage with the current room as a destination.



# Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| **Test no.** | **Test Area** | **Test Description** | **Expected Result** |
| **1** | **App** |  |  |
| 1.1 | App Installation | The app should be able to run immediately after a fresh install. | After a fresh install, the app should run without a crash and bring the user to the log in screen. |
| 1.2 | App Login Area | The user should be able to log into the app using their student number and password. | The user can enter their username and password and log in |
| 1.3 | User Saved Information | The app should display to the user and saved routes they have saved. | After logging in the user should see any routes, they had saved on any device where they logged in. |
| 1.4 | Route Finding Using Room locations | The app should find a route between two locations given both a start location and an end location. | Upon being given two locations a start and an end, an efficient route is found and displayed to the user. |
| 1.5 | Route Finding Using Staff Name | The app should find a route from a starting location, to an office belonging to a member of staff by using the name of the staff member. | When given a starting location and the name of a member of staff the app finds an efficient route to the correct staff member’s office. |
| 1.6 | Access specification | The app should avoid lifts/stairs depending on the preference of the user. | Upon being given a route that involves ascending multiple floors, the app changes the route to a route that matches the preference criteria set by the user |
| 1.7 | Security | A user should be able to access another user’s settings or preferences without their username and password. | Users should not be able to access the app without logging in, and when logging in only see the required information. |
| 1.8 | Out of hours test | Certain rooms and pathways are not accessible during out of hours. | When finding a route out of hours, the app should avoid areas that are restricted. |
| 1.9 | Restricted lift | PhD students, staff and students with disabilities are able to use a lift which goes to floor 5 and 6, which other students can’t, the app should use that lift when prompted by one of these users. | When finding a route if you have access to the lift the app should allow you to use it, using it preferably to the other lift, while users without access should never be directed to use it. |
| **2** | **Website** |  |  |
| 2.1 | App logo and name | The website should contain the app logo, name and description. | The app logo, name and description should all be in the appropriate places for marketing purposes. |
| 2.2 | Database storage | The database must be properly stored and linked to the website. | The database must be stored on the website, so that any changes made while accessing it from the website, must reflect in the database of the app and apply to the users. |
| 2.3 | Admin login | The admin should be able to log in to the database and change information in it. | When logged on, the admin can update any information in the database accordingly. |
| 2.4 | User manual | The website should store all the needed information and guides on how the app can be used. | When a user goes to the website they should easily find information regarding the purpose of the app, how it works and how to use it. |

