LabLocator - Functional Spec

Project Title	Lab Locator
Ethan Clarke	19372086
Evelina Prosyankina	19368013
Supervisor	Darragh O'Brien
Completion Date	11/11/2022

0. Table of contents

1. Introduction

1.1 Overview

Lab Locator will be an Android application that will be developed for Android devices running Android 5.0 (Lollipop) and above. The app will be used to find the occupancy of a selected number of rooms within the McNulty building on the Glasnevin campus by identifying what room a student is currently residing in, along with being used to display the daily schedule of the room and predictions of the occupancy using previously gathered data, with the feature for users to announce their location for other users (if they are in or around any of the rooms listed in the app).

While there are methods for students to check room bookings within Open Timetables, there is no way for them to know if the room is occupied or not, which means they may spend time checking rooms that are more than likely already filled with other students who may be potentially disruptive to them doing work. The app will be designed to simplify the process of checking room bookings as well as displaying the room occupancy to the user, so that less time is spent looking for available rooms.

The application will have two main components, an Android application based in Java, and a Python 3 backend that will be used to pass data back and forth between itself and the application while communicating with the SQL database and external sources such as Open Timetable.

1.2 Glossary

McNulty Building	Building housing the computer laboratories on DCU Glasnevin Campus.
OpenTimetable	DCU's classroom timetabling website.
SQL	Structured Query Language. A system of databases used to store and retrieve data.
TensorFlow	A free and open-source software library for machine learning and artificial intelligence.
API	Application programming interface, a software intermediary that allows applications to communicate with one another.

2. General Description

2.1 Product / System Functions

The mobile application will allow students to view data on the congestion of people in computer laboratory rooms in the McNulty building on the Glasnevin Campus. Users will be able to check both the current and predicted future activity of each laboratory room, with the purpose being to help them find a suitable room to use for working on projects and lab work.

There will be a baseline function allowing the user to see when laboratory rooms are booked for specific classes, this will pull from DCU's timetable data and list the times which each laboratory is booked out. Users will be able to visualise a similar timetable to that seen on the OpenTimetable site, but the time it takes to view each room will be faster than it would take to operate OpenTimeable and input the correct settings to bring it to the individual rooms you want to check, as it will be streamlined for the specific needs of the application.

The mobile application will show a live feed of roughly how many people are currently in each individual room, which will be gathered by reading data concerning various Wi-Fi connections in the rooms and surrounding area through the mobile app. We wish to garner enough information concerning the Wi-Fi networks in and surrounding the McNulty building to be able to generate heatmaps showing busy areas and times.

Once we have enough data collected from our own devices, we will be using it to train a machine learning algorithm. The purpose of this algorithm will be to make predictions of the future activity levels and availability of each room, so even without direct data an estimation can be made to recommend rooms based on which should be the least congested for a given day.

2.2 User Characteristics and Objectives

Describes the features of the user community, including their expected expertise with software systems and the application domain. Explain the objectives and requirements for the system from the user's perspective. It may include a "wish list" of desirable characteristics, along with more feasible solutions that are in line with the business objectives.

This mobile application is aimed at DCU students, particularly those with involvements in Computing faculties who regularly make use of the McNulty building and its computer laboratories. The students who would hold interest in this mobile application would want it to be a useful tool to assist them in finding out the best room to pick when they intend to do work using the computer laboratories. Computing students are ideally quite familiar with operating and understanding mobile applications, and should have no trouble with the navigation and operation of LabLocator.

A student using LabLocator might want to know what computer laboratories are more ideal to use for a period of time they wish to do work or study. They should be able to quickly view the most optimal room at the current time. Conversely, students may also wish to know which specific rooms are more crowded or are booked up at certain periods of the day so that they can avoid those rooms and avoid the potential disruption to their workflow that they might experience.

We also foresee a possible use case for professors. Professors who are teaching classes that involve booking laboratory time for students could use LabLocator’s information to check how many people are attending labs, and which rooms they use. A professor may want to know how many rooms are sufficient for a given class, and could drop the booking on a room or add another room based on the statistics of how many students are actually attending labs.

2.3 Operational Scenarios

2.3.1 Students

Students who wish to find a computer lab for work or study can use our app to view information about the various computer rooms in the McNulty building. They can use this information to decide the best room based on the likelihood that they will be disturbed in their work. They may also wish to seek out the more populated rooms in order to find friends and classmates.

2.3.2 Lab Supervisors and Lecturers

Lecturers, Lab Supervisors, and anyone else who manages booked lab times for classes can use our app to view statistics on laboratory attendance. Professors could easily see how many students attend a lab, and in which rooms they go to. They can also see the trend of how long students appear to attend the lab, being able to see if students typically leave early or stay for additional time. This could prove useful in planning lab assignments and booking a sufficient amount of computer lab rooms.

2.4 Constraints

2.4.1 Hardware

The use of our app will require a smartphone running a minimum install of Android 5.0 (Lollipop), this will be our primary target platform due to the relative ease of development. This is mostly due to a large majority of Android development tools being open source. We do not intend to support IOS, as we already own Android smartphones.

2.4.2 Timing

We have numerous time constraints to work around for this project. The final submission is due April 2023, which we have to have everything completed by. This effectively gives us several months of production time to work on the project, and we will have to use this time wisely to complete a minimum viable product. We also have additional features we can implement if we successfully reach our goals early.

2.4.3 Knowledge

This project will involve technology stacks that we are relatively new to working with. In particular, mobile app development and using Android Studio is a new experience for us that we will need to become accustomed to quickly in order to avoid problems down the road. Upskilling in areas such as this will be a requirement for us if we intend to execute our plans in good time.

2.4.4 Data Acquisition

In order for our application to work, we are going to need to capture live data to get accurate, up to date readings. We intend to populate as much of this with our own scans and tests as possible, but ultimately we will need to use some information from other users of our app. We intend to be careful with what we use and how we use it, and we want to ensure that we use it only when necessary. Careful consideration will be required in order to achieve this.

3. Functional Requirements

3.1. Retrieve lab information via backend

Description	The app should be able to connect to the backend and be able to retrieve data for each of the available rooms
Criticality	Highly critical - the app would be unable to function without the ability to retrieve data on the available labs
Technical Issues	Lab information must be manually inputted, which leaves the risk of information being entered incorrectly and being displayed incorrectly as a result
Dependencies with other requirements	N/A

3.2. Retrieve lab schedule

Description	Retrieval of the schedule for a specified room from Open Timetables
Criticality	Moderately critical – The app would be able to function as normal without this function, although it is still important that it exists to give information on any bookings or classes
Technical Issues	As Open Timetables lacks a proper API for retrieving data, the retrieved data may end up being parsed incorrectly, and may not display well as a result, alongside the possibility of Open Timetables having technical difficulties and being unable to return any data
Dependencies with other requirements	Retrieval lab information via backend

3.3. Retrieving lab occupancy

Description	Using information collected from the routers found within the labs, the occupancy will be decided via the number of devices connected to the router, as well as outputting the data to a table within the database
Criticality	Highly critical - the app would be unable to function as intended as it would be missing one its core functionalities, as users would be unable to decide the occupancy of the room they are viewing
Technical Issues	Creating the necessary implementation to retrieve information on lab occupancy may be more complicated to implement than what is thought, which means as a result, this part of the project is at a risk of an increased number of bugs and issues that can affect the overall performance of the app
Dependencies with other requirements	Retrieve lab information via backend

3.4. Predicting future lab occupancy

Description	Using data gathered of room occupancy within set times, an approximation of future room occupancy for the day will be shown to the user
Criticality	Mildly critical
Technical Issues	The implementation of a predictive model requires for there to be a large dataset to be trained from, and as the dataset is to be constructed by us, the risk of the dataset being insufficient or compiled in a way that supplies inaccurate results can make the implementation of this feature more difficult
Dependencies with other requirements	Retrieve lab information via backend

3.5. Adding and managing lab information via admin portal

Description	Via the admin portal, administrators can create and manage lab information without interacting directly with the database
Criticality	Moderately Critical
Technical Issues	N/A
Dependencies with other requirements	Admin Login

3.6. Admin Login

Description	Given existing credentials, admins will be able to login in via the admin portal to create
Criticality	Highly critical
Technical Issues	Authentication and authorization may be difficult to implement
Dependencies with other requirements	N/A

3.7. View gathered occupancy via admin portal

Description	Allowing for administrators logged in to the administrator portal to view the gathered occupancy data
Criticality	Moderately Critical
Technical Issues	N/A
Dependencies with other requirements	Retrieving lab occupancy, as without anything gathered, it would not be possible to display any data

4. System Architecture

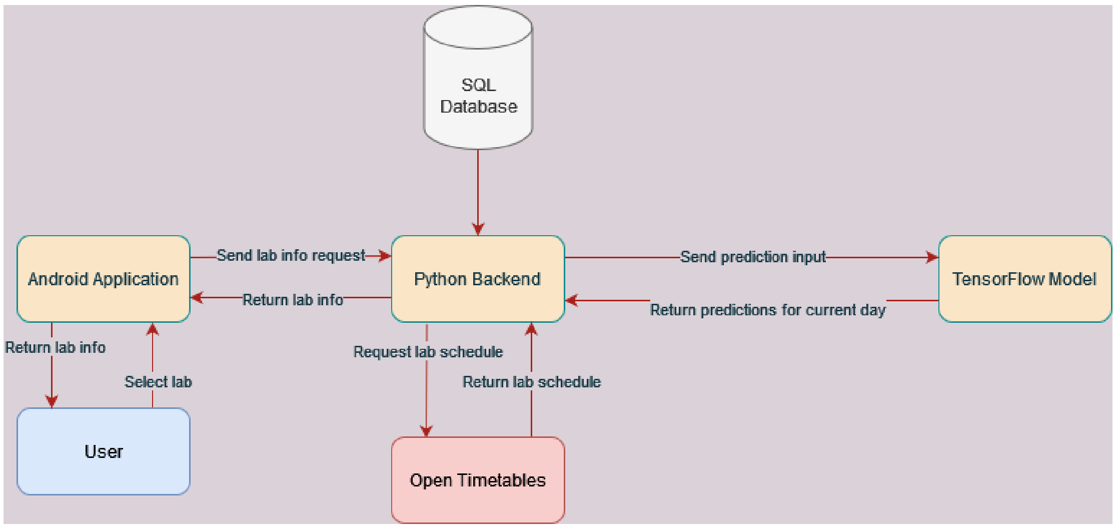


Figure 4.1 - High-level overview of the system architecture

User - Will use the application to view the available labs within the McNulty building as well as view the day's schedule for a specific room and be able to signal their location by selecting a room to signal from

App - Will communicate with the backend to retrieve and display all the available labs, lab occupancy of each lab, lab schedules and predicted occupancy.

Database - Will be used to store lab information, as well as store data gathered occupancy data to be used for the machine learning model.

Python Backend - Will be used to take in the results from the model, pull data from the database and compile the data into a readable format to be displayed on the app.

TensorFlow model - A model created using TensorFlow that will be used to approximate the future occupancy of a room based on data collected previously on each room.

Open Timetables - Will be used to retrieve the schedule for the labs to be displayed within the app.

5. High-Level Design

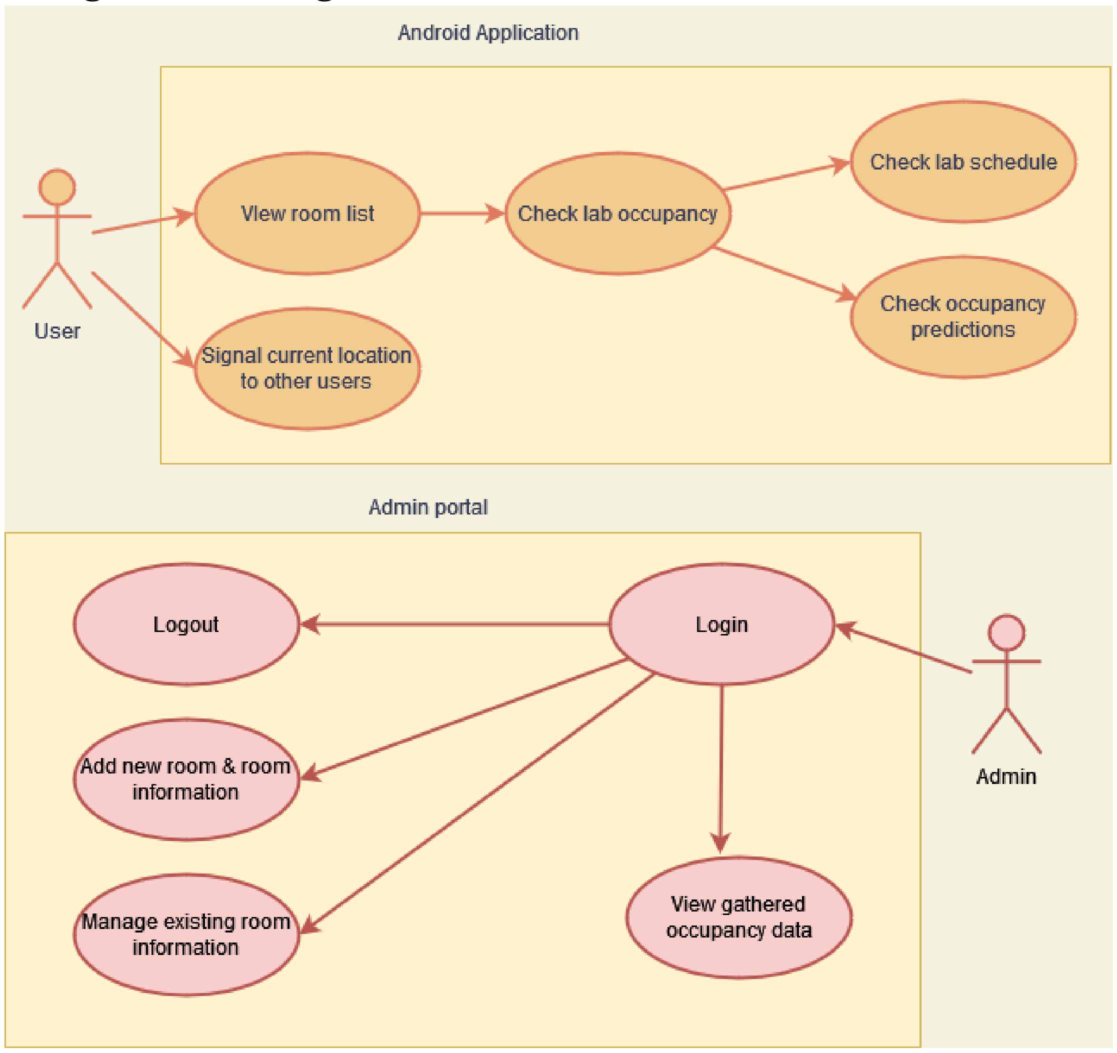
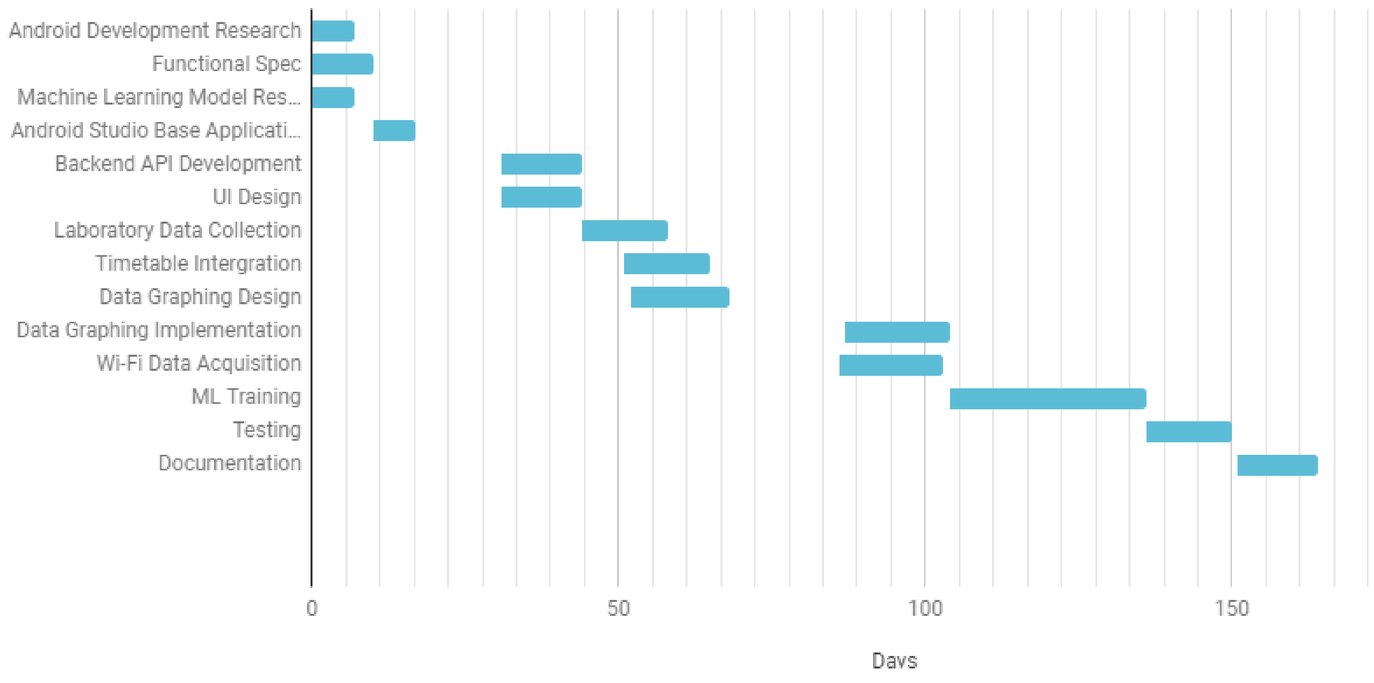

Android Application
User
View room list
Check lab occupancy
Check lab schedule
Check occupancy predictions
Signal current location to other users
Admin portal
Logout
Login
Admin
Add new room & room information
Manage existing room information
View gathered occupancy data

Figure 5.1 - Use case diagram for the application and the admin portal

6. Preliminary Schedule

TASK NAME	START DATE	END DATE	START ON DAY	DURATION	TEAM MEMBER
Android Development Research	10/1/2022	10/8/2022	0	7	Ethan
Functional Spec	10/1/2022	10/11/2022	0	10	Ethan & Evelina
Machine Learning Model Research	10/1/2022	10/8/2022	0	7	Evelina
Android Studio Base Application	10/11/2022	10/18/2022	10	7	Ethan
Backend API Development	11/1/2022	11/14/2022	31	13	Ethan
UI Design	11/1/2022	11/14/2022	31	13	Evelina
Laboratory Data Collection	11/14/2022	11/28/2022	44	14	Ethan & Evelina
Timetable Integration	11/21/2022	12/5/2022	51	14	Ethan
Data Graphing Design	11/22/2022	12/8/2022	52	16	Evelina
Data Graphing Implementation	12/27/2022	1/13/2023	87	17	Evelina
Wi-Fi Data Acquisition	12/26/2022	1/12/2023	86	17	Ethan
ML Training	1/13/2023	2/14/2023	104	32	Evelina
Testing	2/14/2023	2/28/2023	136	14	Ethan & Evelina
Documentation	3/1/2023	3/14/2023	151	13	Ethan & Evelina


Android Development Research
Functional Spec
Machine Learning Model Res...
Android Studio Base Applicati...
Backend API Development
UI Design
Laboratory Data Collection
Timetable Intergration
Data Graphing Design
Data Graphing Implementation
Wi-Fi Data Acquisition
ML Training
Testing
Documentation
Days