Software and Hardware Requirements for Humber Campus Navigator

**Anoopjot Kaur Dhallu**

**Bettin Jacob**

**Ishan Khuttan**

**Humber College, Toronto**

**Winter-2019**

Table of Contents

[List of Illustrations 5](#_Toc536113344)

[Introduction 7](#_Toc536113345)

[Additional Comments 7](#_Toc536113346)

[Project Description 9](#_Toc536113347)

[Requirements Specification 10](#_Toc536113348)

[Software 10](#_Toc536113349)

[Hardware 12](#_Toc536113350)

# List of Illustration

# Introduction

Humber Campus Navigator is a project made specifically for navigation within a group of buildings or a campus, because for a group of building’s like a college campus, it is really hard to navigate through it and find the room numbers, washrooms etc. We are trying to solve this issue in Humber North Campus with an Android application. For instance, when we were new to this college, we used to roam inside the campus to find our destination point within the campus. It was tiresome and frustrating. There was no much help as we don’t know who are new and who know the way to our destination. Moreover, everyone are busy too. Even in this semester it was difficult for us in finding the location of NB part of the college. To solve that, we are making an application which will work within the campus. As a sample project, we are considering the Humber Campus, and to give it a try for a complete set of building, we are first trying to take five points and trying to navigate within them. After taking Professor’s suggestions, time available and our experience into consideration, we decided to do try and solve this real-world problem. Once we are successful in completing that, we have a plan to expand it to the entire campus level, it will be very helpful for Humber students and its visitors, and it will be much easier for anyone to find a point or destination on campus.

## Additional Comments

# Project Description

We are designing an app to make the navigation within the Humber campus, easy and convenient. For students, especially who are new to campus, it is hard to navigate within the campus. This application will help students to explore and navigate within the campus from any point to any point in short time and less distance. The app works with the help of the QR code and finding shortest path using Dijkstra’s Algorithm. Unique QR codes are pasted to each door of the classrooms or washrooms, or where ever we want within the campus. The user needs to scan the QR code nearest to him, and type in or select the destination point from the map or from a drop-down list and the application will show the easiest route possible and this will be accomplished with the help of Dijkstra’s Algorithm. In campus navigation is the main function of the app, other than that it tells the temperature of current point where we are standing, and the normal average of temperature at that point. It also tells the current and historic average temperature of the destination point where the user wants to go. All values are taken with the help of our sensors and are be stored within our database.

The internet connected hardware will include a Raspberry Pi with a custom PCB with the following sensors and actuators

1. LSM9DS0 9-axis accelerometer, Gyroscope, Magnetometer and Temperature sensor.
2. 1.8 TFT Color Display ST7735
3. Tmp-36 Temperature sensor

The database will store the temperature readings from LSM9DS0. We need to design a PCB board which can stack all the sensors on a single board, so that the integration is complete. So the temperature readings from the LSM9DS0 sensor will also be displayed on ‘1.8 TFT Colour Display ST7735’ screen and the temperature readings from LSM9DS0 will be verified by the temperature readings from Tmp-36 Temperature sensor, which also will be displayed on the ‘1.8 TFT Colour Display ST7735’ screen alongside. The temperature readings from LSM9DS0 sensor stored in database are made use by the android application also.

## Requirements Specification

### Software

#### Mobile Application

Bettin Jacob will be responsible for this part. We are developing the application in Android platform with a support for API key 21 and above which supports 95% of android users worldwide. The SDK we make use of is Android studio of Google for developing the application. The high point of this application is that, it will work without GPS, because instead of using GPS we have used Dijkstra’s Algorithm, which is the backbone of this project. Another important part includes QR code scanning and the path showing. In scanning QR code and finding the current location we are using camera function, for finding the shortest path we are using Dijkstra’s algorithm and for displaying the path, we have used the path array coming from the algorithm. Another attraction of the app is that, in case of an emergency, we can directly call security, as there is a specific option to make a call named “call security” on home page. Then, as an extra point, we have an address book which have phone numbers of all the major academic schools and facilities, so a student or a visitor or any interested person can get the number easily and they can just call and take the information of their interested areas. This application will help in promoting Humber College and people and will help to explore Humber Campus easily in a single platform.

In the application, First the user has the option whether he want the navigation, or he want to call security, or he want the access to phone book of major schools or he want to check the temperature values at that point.

For navigation, the user need to scan the nearest QR code through which the app identifies the current location of the user and the user can then type in the destination point, and on clicking a button, it will take the user to the next page, which highlights the shortest/easiest path to reach his destination, along with the other available paths as well. If the user wants to go using another way or the user can just follow the highlighted route on the map, it is up to the user to decide.

While using the navigation, the user can have a look of the temperature values of the place where he stands and the temperature of the room/point where he wants to go.

If user clicks ‘Call Security’ button from the main page, app will take the user to a page, where the app needs a kind of confirmation from the user, if he really wants to call security, and if user again taps ‘call security’ button on that page, it will call security directly, else user can go back to main page. The app also displays the number to which the user is going to make the call.

On Selecting the ‘Phone Book’ option from the home page of the App, it shows the names of all the major schools, and by tapping the interested school, it will display the phone number of that school, and if the user taps on that number and select the call icon from the bottom, it will call selected school. All extensions are included within that number, so the user need not type in any of the extension.

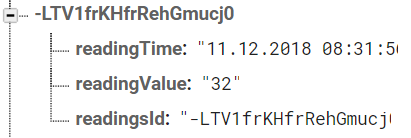
For Temperature values of a room, we have button named “Temperature Readings”. If the user scans a QR code, the application will display the temperature of that room and also the historical average of temperature of that room.

At last we have ‘Quit’ button, which will quit from app completely.

#### Database

Bettin Jacob will take care of the Database. We make use of the Firebase database provided for free by Google. Two data are handled by the database. One is the data from the sensor in each room and other is the data about the nodes. Sensor data has a time stamp with room number, date and time in 24-HR format with precision to a second. This time stamp will carry a corresponding reading from the sensor of each room so that we can understand what the temperature readings were at each point of time was. The other data that is held by the application are the information about all the nodes of the graph. The nodes will be room numbers and each room number will carry a unique id for the application to identify the nodes.

A sample screen shot of temperature reading recorded in database,



#### Firmware

We make use of Raspberry Pi to stack our sensors and the raspberry pi uses of Raspbian as the operating system. The readings from the sensor are handled using Python programs installed in Raspberry Pi. The Python programs converts the signals from the sensor into human readable decimal values and that values are being written into the database. We can decide the interval between two readings to be recorded to database. It can be read every 10 minutes or every hour or however we want.

### Hardware

#### Development Platform

The Hardware development Platform for Humber Campus Navigator is Raspberry pi 3 model B+, which is the latest production of Raspberry pi 3 featuring a 64-bit quad core processor running at 1.4 GHz and 1 GB RAM. It supports Wi-Fi and Bluetooth connectivity, along with these features, it can support other devices using the interfaces, I2C and SPI. In Humber Campus Navigator project, we are connecting our sensors (LSM9DS0, TMP36 & ST7735 1.8 TFT Color Screen) with Raspberry pi using i2c interface. The main function is to display the temperature of a particular place using LSM9DS0 sensor on the android application. LSM9DS0 is a 9-axis accelerometer and Temperature sensor which reads temperature and the reading are sent to Raspberry pi, then the python code will take the readings and will convert them into human readable format and records into a database and also displayed on ST7735 1.8 TFT Color screen. We are cross checking our temperature readings, with the help of TMP36 sensor. Raspberry pi is acting as a platform here, to record, convert & display the reading and store them in the database.

#### Interface boards and sensors

Interface Board

Anoopjot Kaur Dhallu and Ishan Khuttan will be responsible for designing the PCB which will hold all the sensors. The PCB will be directly attached to Raspberry pi. LSM9DS0 and ST7735 is interacting directly with raspberry pi, but TMP36 needs a microcontroller (ATtiny85) to interact with raspberry pi, as it an analog device, it cannot interact directly with raspberry pi. For TMP36, TMP36 is connected to ATtiny85 over SDA & SCL CLK lines and ATtiny85 is connected to Raspberry pi. The three sensors, LSM9DS0, TMP36 & ST7735 are connected on single platform, Raspberry pi and they will be interacting, using i2c interface.

Sensors

LSM9DS0 9-axis accelerometer, Gyroscope, Magnetometer and Temperature Sensor will take temperature readings and display on ST7735 and these readings will be recorded in Database also.

ST7735- It is a 1.8 TFT Color Display Screen, on which both the temperature readings will be displayed.

TMP-36 Temperature Sensor -> It is an analog sensor, it will be verifying the temperature readings with LSM9DS0 sensor.TMP36 sensor will take the readings, and send them to ATtiny85 and the ATtiny85 will convert them into human readable format and will send to raspberry pi to display on screen, upon request.

#### Other Accessories and Enclosure

The development platform will require a power source, display and network connection.

The enclosure should be designed in CorelDraw and laser cut from acrylic (Responsibility: Ishan Khuttan). It may be further improved by 3D printing (responsibility: Mechanical Engineering Technology collaborators- if available).