Emergency Exit App incorporated with Arduino.

Final Report (semester 2)

Kacper Woloszyn

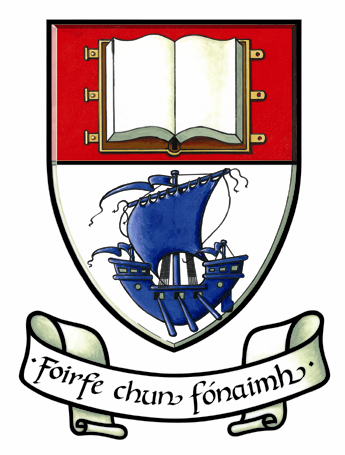
20071494

Supervisor: Caroline Cahill

Second Reader: Lucy White

BSc (Hons) in Applied Computing

Emergency Exit Mobile Application with Arduino



By Kacper Woloszyn, 20071494.

Contents

[Introduction 4](#_Toc7049579)

[Diagrams 5](#_Toc7049580)

[Use Case (Easier to Read Horizontally) 5](#_Toc7049581)

[Domain Model 6](#_Toc7049582)

[Splash Screen 7](#_Toc7049583)

[Technology used in my assignment 8](#_Toc7049584)

[Arduino 8](#_Toc7049585)

[Firebase – Log In / Register / Change Password / Deactivate 10](#_Toc7049586)

[Language 13](#_Toc7049587)

[Google Maps API 15](#_Toc7049588)

[Route finding – Dijkstra’s algorithm implementation 16](#_Toc7049589)

[Canvas Android 28](#_Toc7049590)

[How security is dealt with? 28](#_Toc7049591)

[Plan for the application after the final year project 29](#_Toc7049592)

[Draft user manual 29](#_Toc7049593)

[Learning curve 29](#_Toc7049594)

[Citations 29](#_Toc7049595)

[Minutes from meetings – Semester 1 31](#_Toc7049596)

[Minutes from meetings – Semester 2 32](#_Toc7049597)

# 

# Introduction

I have decided to go with a mobile application for my final year project. I have developed as a student educationally over the four years in Waterford Institute of Technology and at Universidad de Malaga, to finally be able to put my own project idea combining all the things that I have studied, and the things I would like to learn, not necessarily all the things that I had learned at WIT, but also individually.

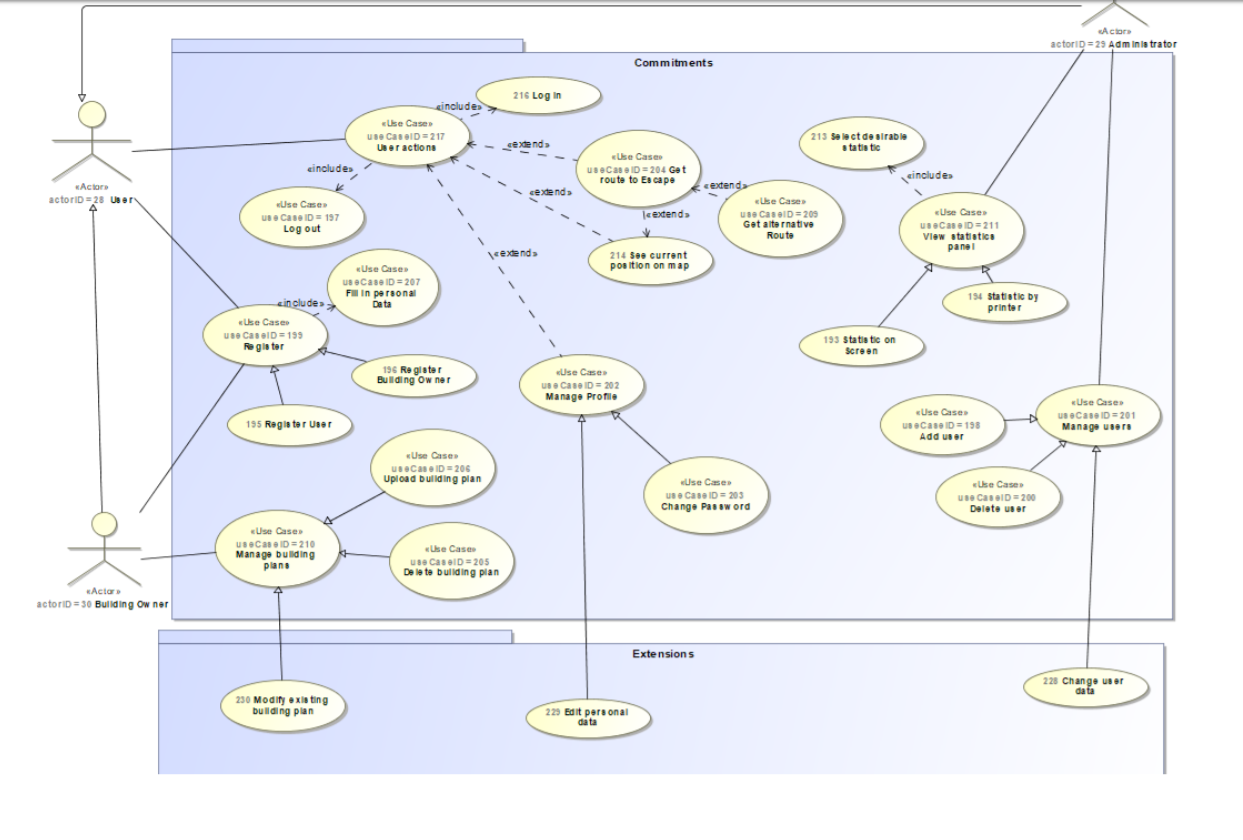
I feel the mobile application I am doing for my final year project may be very important for future generations as each of us nowadays does things on our phone more times than not. I feel there is a need in the market for an application that displays emergency exits to people in public areas and buildings in case of an emergency.

In the project I use an Arduino, the Arduino will be connected to my application and I programmed a buzzer and a sound sensor in Arduino. The buzzer will sound, and an emergency exit route is displayed on the screen, with some additional information, such as the path length or the time taken to find the route. The application is for android users, but as the projects developed in the future it could be brought to IOS devices. The application has a database of users that is stored in firebase so a connection to the internet would he a requirement. An algorithm is used in the background to find the quickest route out of the building. The algorithm used is called Dijstras. It works on having the start node, and the end node, and It picks the unvisited vertex with the lowest distance, calculates the distance through it to each unvisited neighbour, and updates the neighbour's distance if smaller. I used the knowledge from my Erasmus course in Intelligent Systems, where I learned how to programme route finding algorithms such as A\* Algorithms and Dijkstra algorithm or genetic algorithms and neural networks. I take specific measures to make sure the security of the user is not breached, and that the application works as desired, by allowing logins so that I can keep track of who is on the application, and the location is not saved, so just in case someone was trying to steal the data it would be hard.

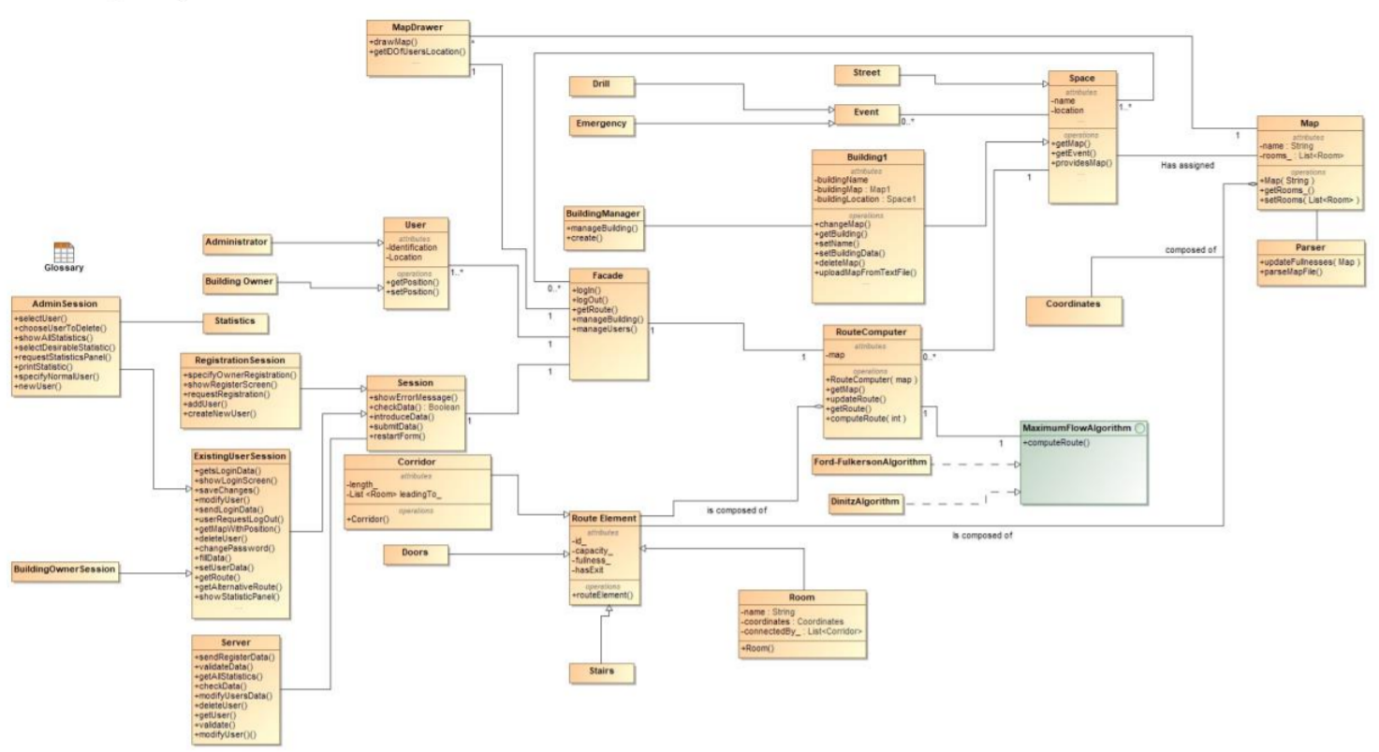
I have used a tool called magic draw to develop my diagrams which show the functioning of the application.

# Diagrams

## Use Case (Easier to Read Horizontally)

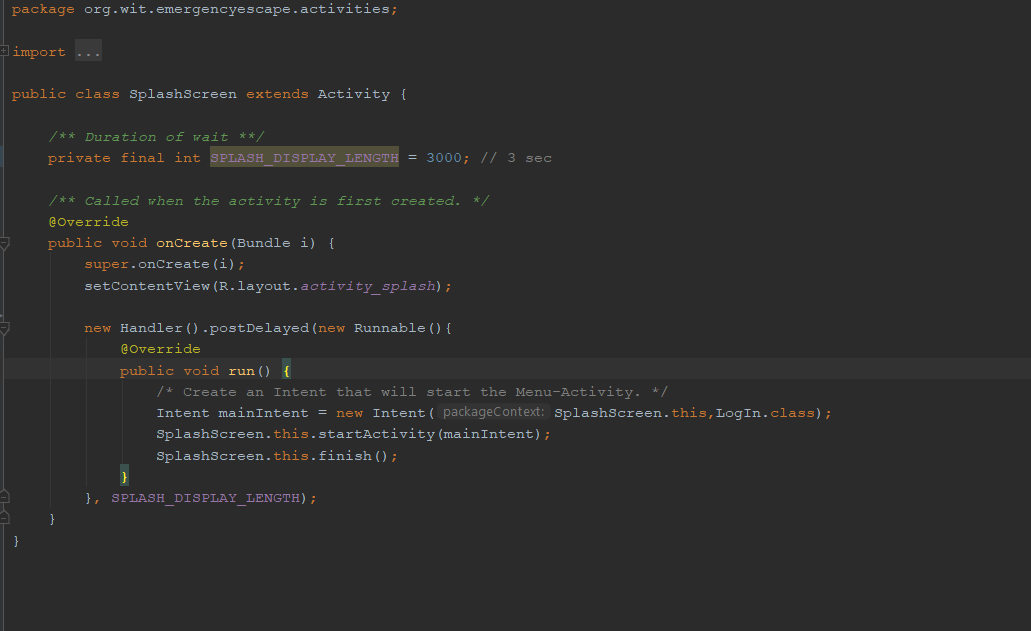


## Domain Model



## Splash Screen

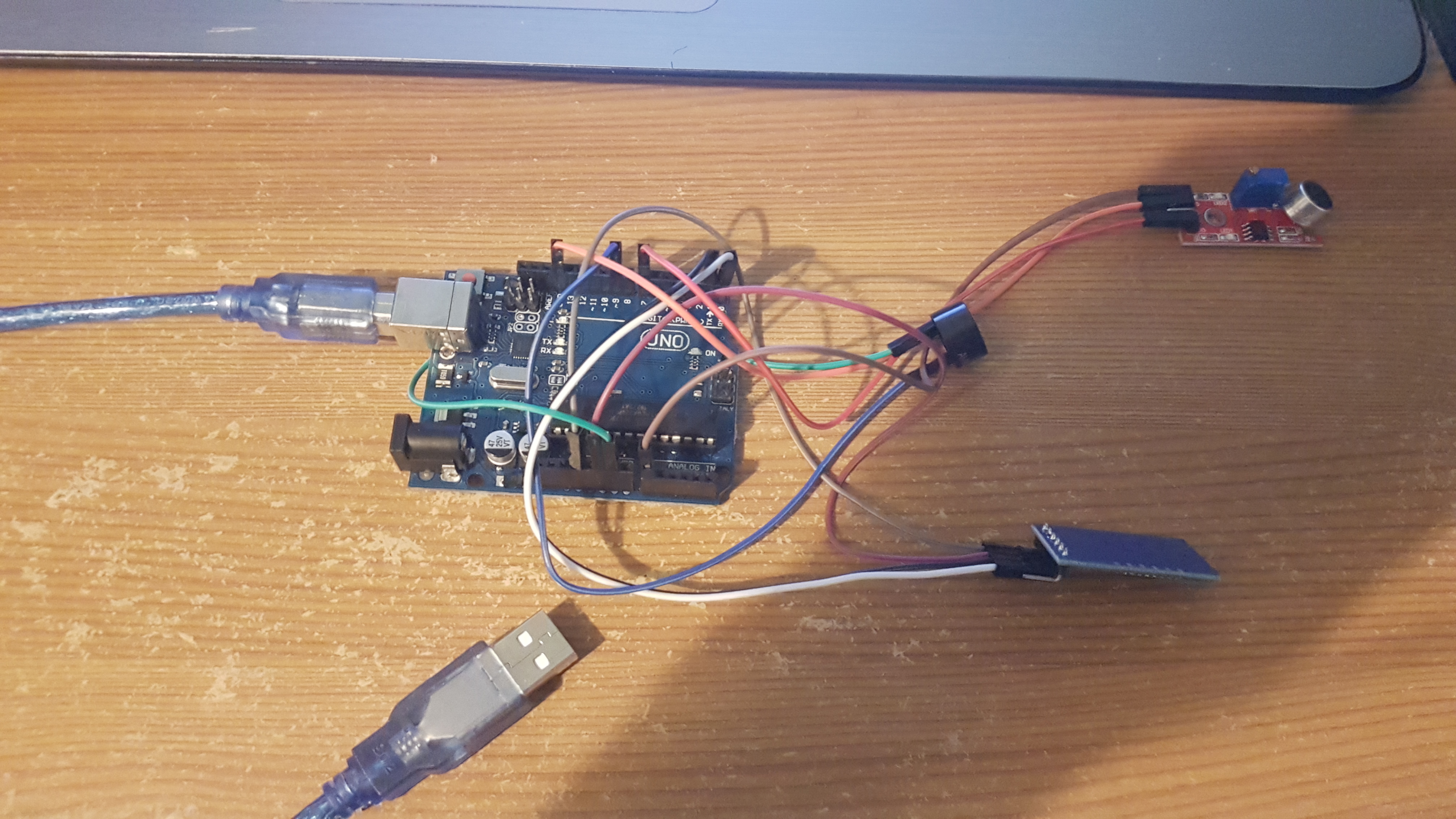
A splash screen is a graphical control element consisting of a window containing an image, a logo, and the current version of the software. Splash screens have their pros and cons, but since this is a personalised application, I’ve decided to design my own splash screen. If I were to place this app commercially on the Google play app store, or to develop it further, I would not include the splash screen as it steals valuable seconds from a user, to get the results he or she wants. I put in a splash screen, as many commercial applications have one, whilst the application is connecting to the internet. I had developed my own splash screen, and I’ve set its time to be 3 seconds just for visual purposes (Livewire,2018).





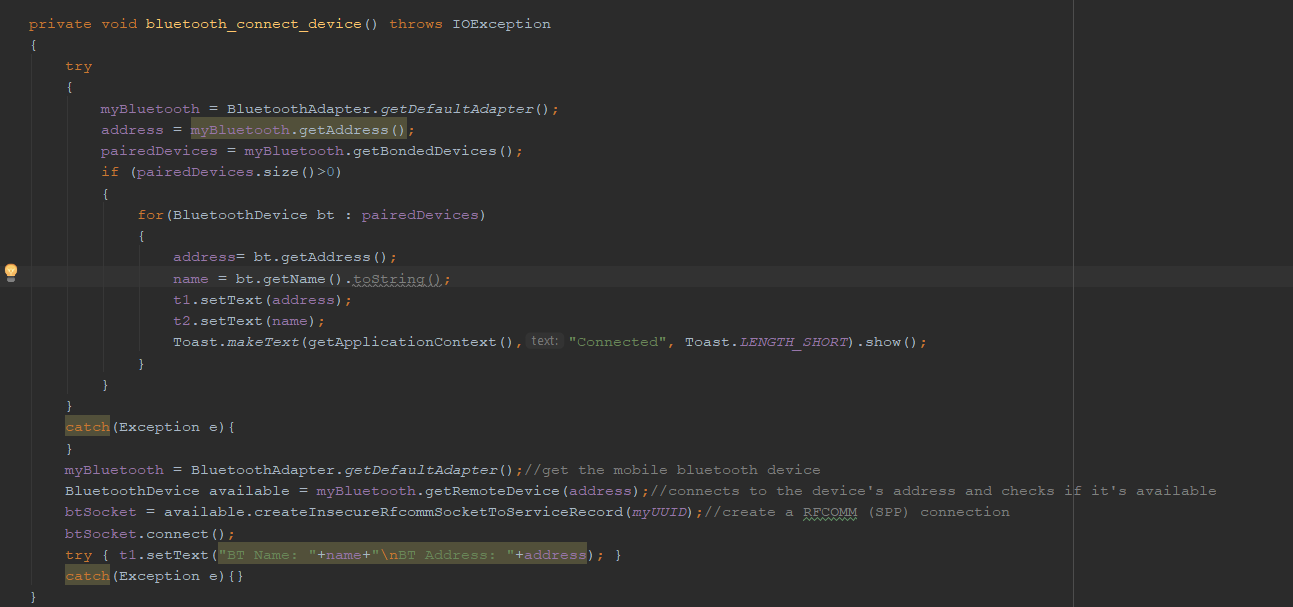
# Technology used in my assignment

## Arduino



I have an Arduino board with 3 modules I had to obtain separately along with the female-male wires, that did not come with the board. I have connected the Bluetooth module to the 3.5 V entry Ground, RX and TX as specified in a tutorial online on how to connect and wire up the module. The module I obtained was a cheap one, for this Project, and if I were to do this to a bigger scale, I would get more reliable and more documented hardware, as it was quite a challenge to know what to do with the module once obtained. The buzzer is connected to a digital pin 9, as specified also in the code below and ground. The sound sensor, which is the red item, is connected to ground, Analogue pin 0, as it has to pick up all of the sound ranges and not only 0 and 1’s like the buzzer on the digital pin. It also has to be connected to a digital pin 7 as there could be a noise signal that was digital perfectly.





I have a method called Bluetooth\_connect\_device, that connects to a device, and creates a connection that’s stored in a variable. I then set the name and address fields of the application to be that of the device that has been connected. Arduino uses mostly C++ for programming, and I have programmed a connection program, that I flash the board with and a buzzer programme.

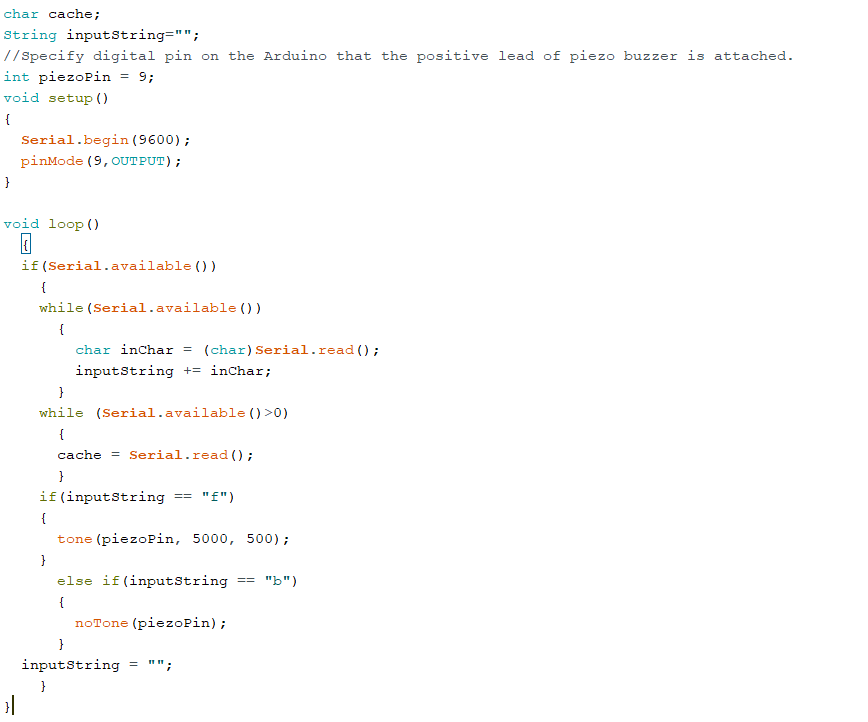


The programme above basically checks, what is connected to the pins on the Arduino board, and it flashes the board with a programme to allow for the Bluetooth module to work properly. It writes a set baud rate to make sure everything works properly and so that the Arduino board is not overheated and overclocked. To use Bluetooth features in my application, I had to declare two permissions. The first of these is BLUETOOTH. I need this permission to perform any Bluetooth communication, such as requesting a connection, accepting a connection, and transferring data. The other permission that I had to declare is ACCESS\_FINE\_LOCATION. A location permission is required because Bluetooth scans can be used to gather information about the location of the user.

I use a Bluetooth API, that scans for Bluetooth devices, query’s the local Bluetooth adapter and establish RFCOMM channels.

Arduino is an open-source hardware and software company that develops boards and kits for building digital devices and interactive objects that can sense and control from both the physical and digital worlds.

There are DIY kits that I use to make a buzzer sound whenever the application sends out a notification about an emergency. I could use that in the future iterations of the project so that whenever an alarm sounded the application would open, so build a sensor inside the application, but then security comes into play as the application would be listening over time.

A buzzer works like a drum, with a membrane that is stuck by a drum stick causing the device to vibrate and produce noise, but in the Arduino, the membrane is an electric current. To connect the device, firstly I connect the positive leg of the buzzer and connect it to a positive digital pin on the board. The negative leg was connected to ground on the Arduino. Tone command plays command, with two parameters, the digital pin, and the frequency of the noise. There is also a no tone command, which I would use in my assignment when there was no clear emergency. 

## Firebase – Log In / Register / Change Password / Deactivate

Databases for mobiles need to be:

* Lightweight as storage is limited on mobile devices.
* In a form of library with no or very limited dependency (embeddable) so that it can be used when needed
* Fast and secure.
* Easy to handle through code, and option to make it private or shared with other applications.
* Low memory and power consumption.

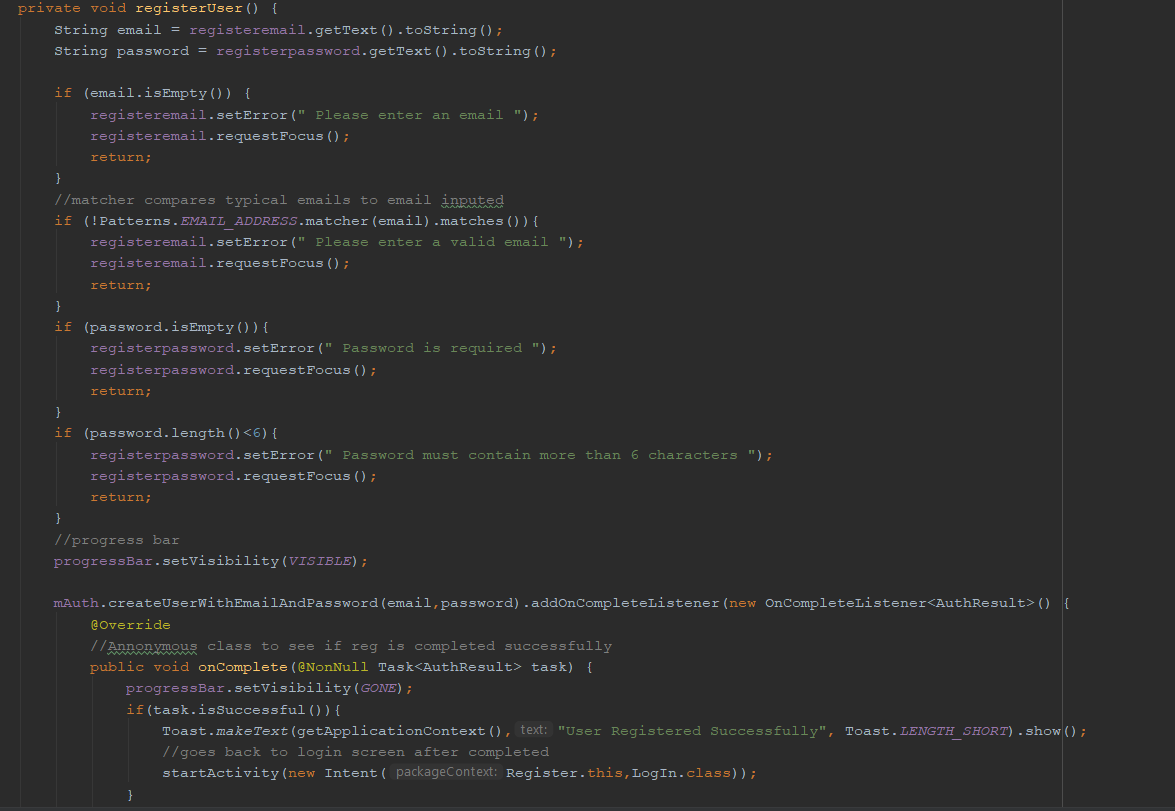
Firebase DB is a mobile app and web app development platform developed by Firebase in 2011, and it has been bought by Google in 2014. I would consider using it as it has been bought by google so it would have many advantages, such as easier accessibility to the google play store. Firebase is used by more than 1.5 million applications (Firebase.Inc, 2018).

Firebase is a free database, hence why it has a big advantage, as probably the application could be free. “Firebase Auth” is a service that can authenticate users using only client-side code, so it wouldn’t need internet to log you in, just would need it to pull an emergency. The company provides client libraries that enable integration with Android / Java and Kotlin, hence why I would consider using it in my assignment.

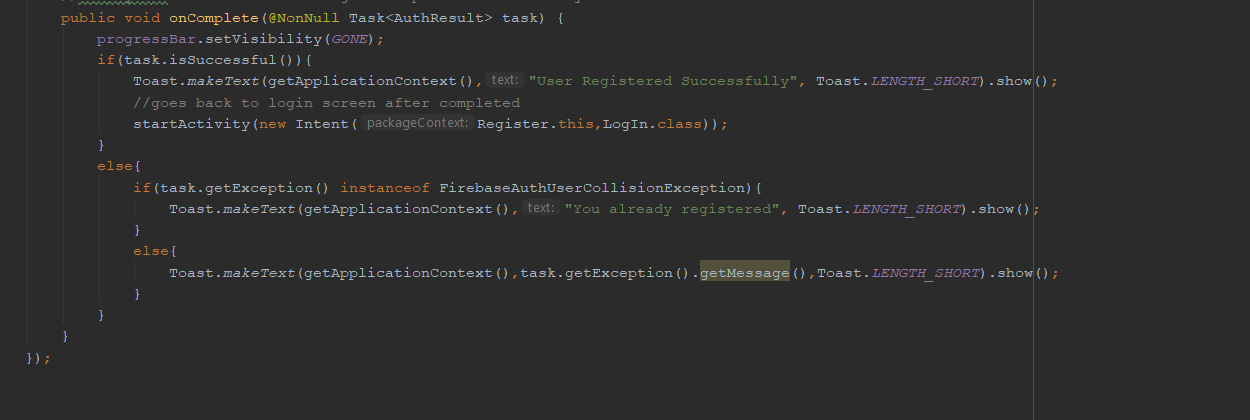
Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content, so if I were to store a Map of a building, it would be very secure, also takin in the fact that internet connection does not have to be amazing to access the service (FirebaseStorage, 2016).

Firebase Notifications is a service that enables targeted user notifications for mobile app developers at no cost, so the user would get notified about an emergency without having to have the app opened.

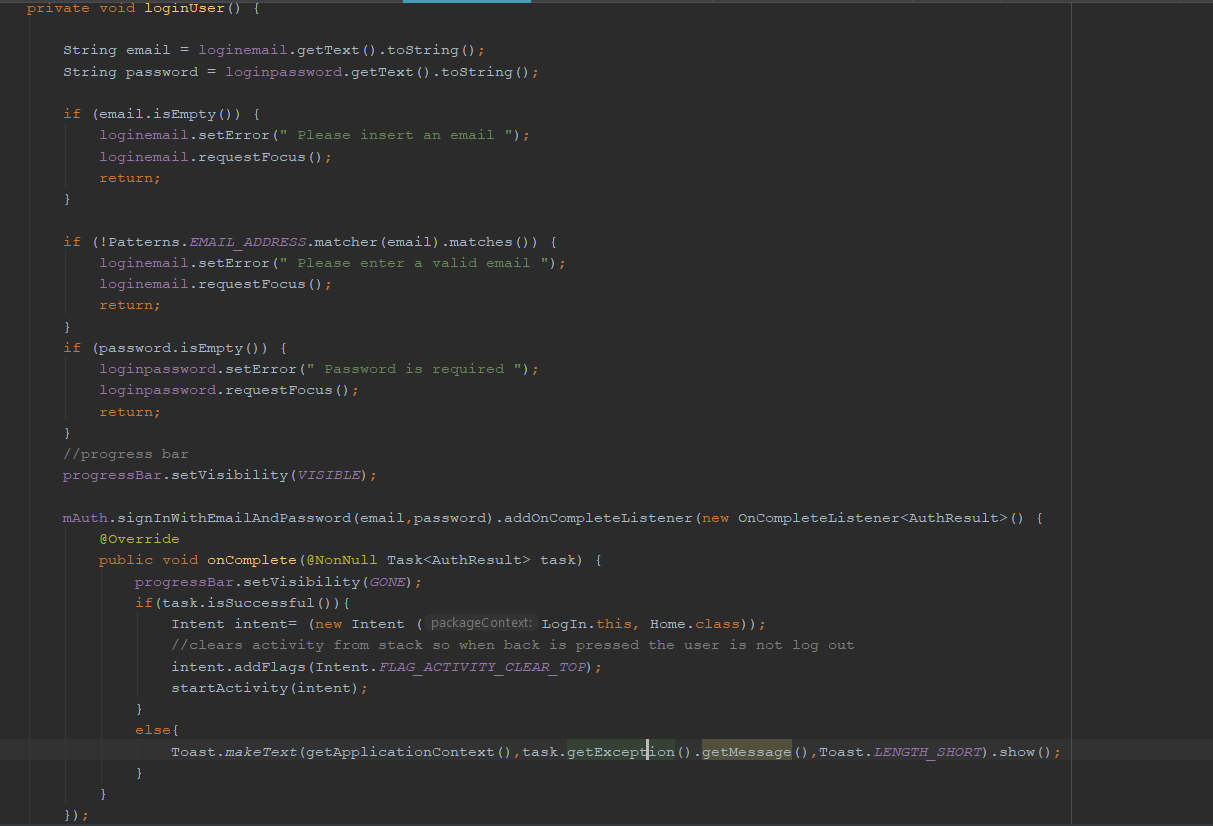
In my mobile application I use firebase auth mostly, as I do not want to store the user’s location, since it would be unnecessary and could lead to very dramatic consequences if someone would hack the application. Firebase auth, basically allows users to log in and register. I do this to make sure I, the administrator of the application, know who is currently using the mobile application to avoid security breaches. Firebase stores the passwords in a salted and hashed manner, so even I do not know what a user’s password is, which is good, as it would be hard for a security breach to happen. Firebase Auth makes my application very secure, so that I know who is using the application at most of the times.



This above is the register method, I check if the email is of correct format, and I check if the password is not empty and it has more than 6 characters, as specified by the firebase authentication.

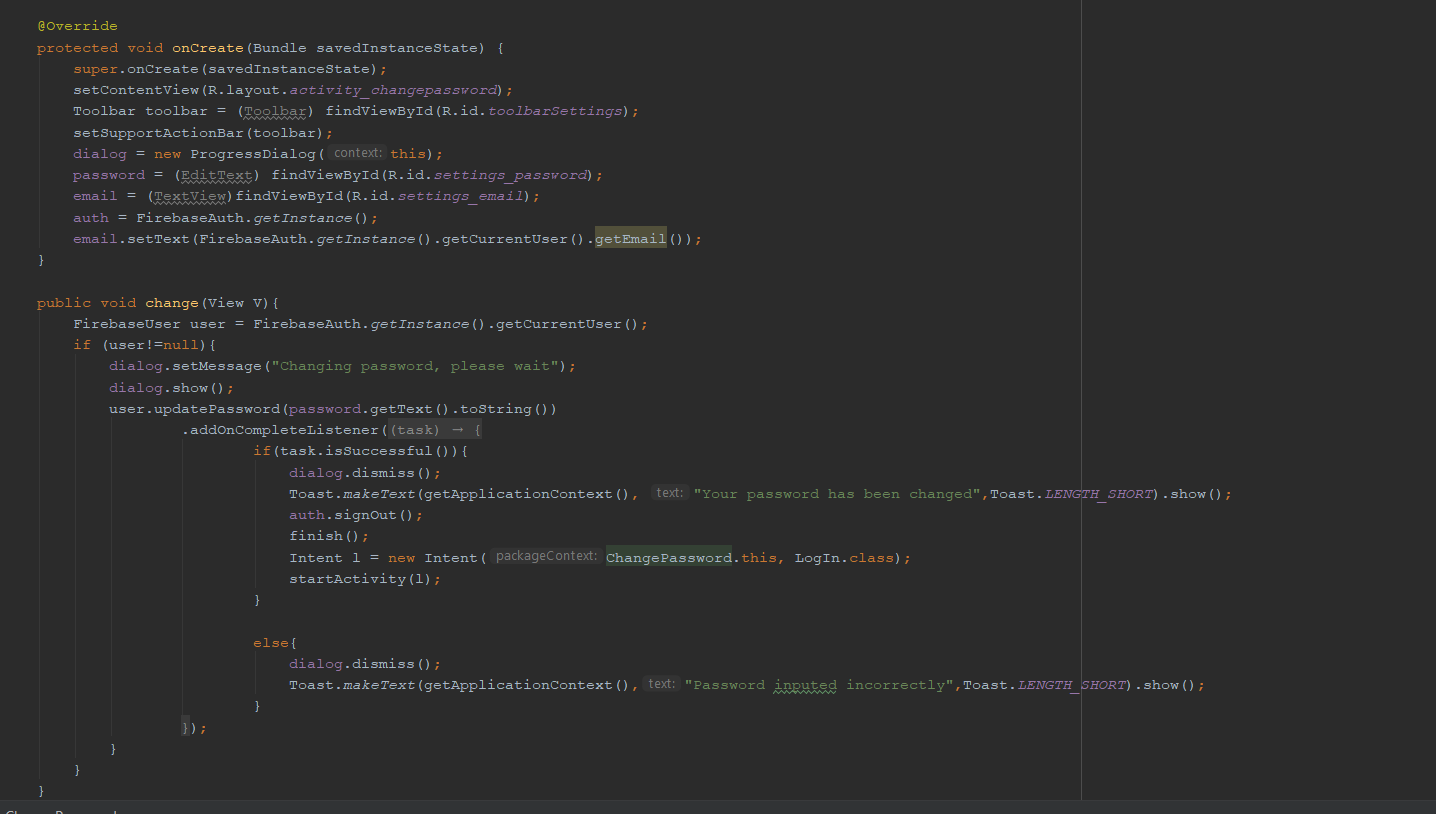


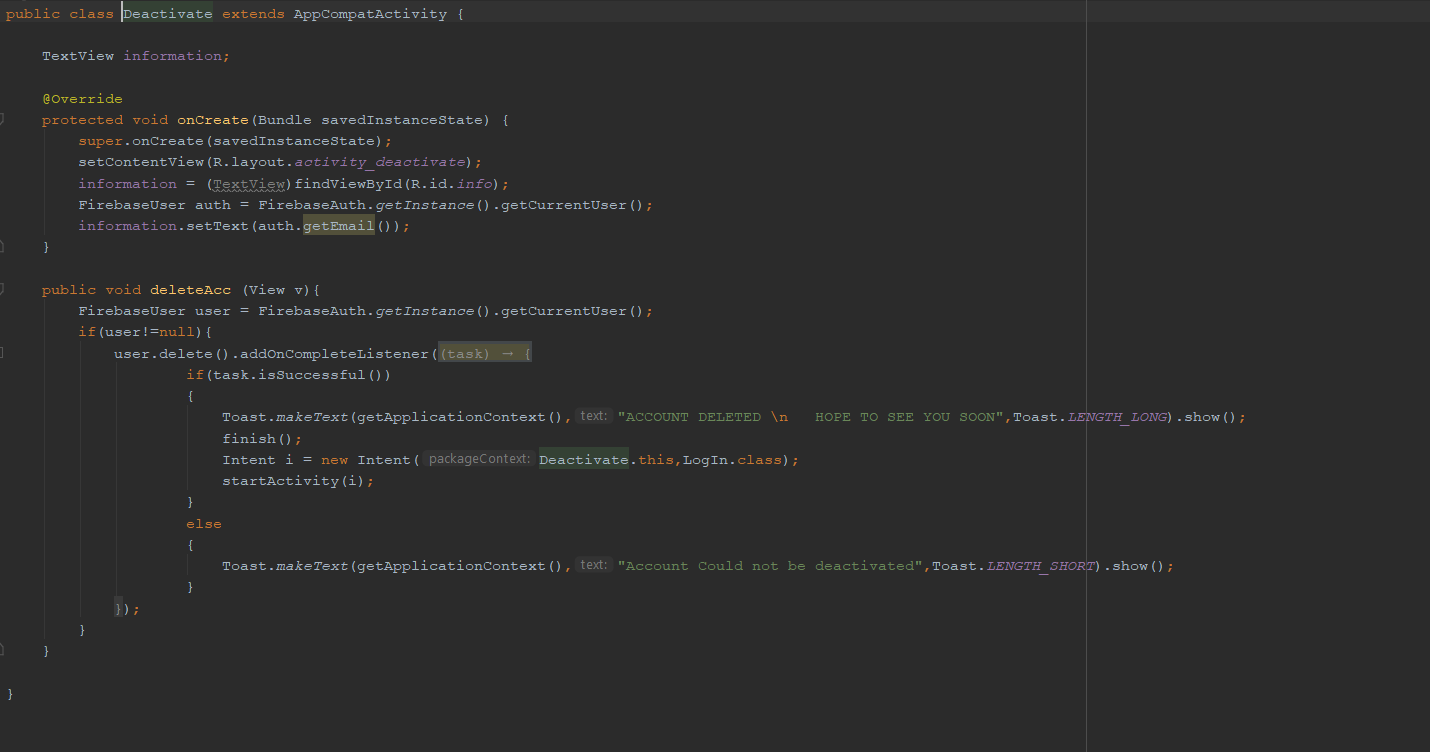
The on Complete is an anonymous class to see if the registration occurred without any errors, if no errors, the user goes to the login screen, if an error with the email already in use, the user gets a message to the screen specifying that he or she is already registered. If firebase error I display the exception.



This is the login method, it checks if the email is empty, also if the email is not valid and if the password is empty. The on Complete checks if the email is correct and if the password is correct, if so it brings the user to the home page, which contains the 3 buttons. If incorrect it gets the exception error from firebase.

As seen below, firebase allows users to also change the password. This is mainly completed in the backend. I let the user see his or her current email, and I’ve a textbox to allow for a new password to get inputted. The change method gets the current user, and if the password input is longer than 6 characters, i.e. it satisfies the constraints of firebase auth, then the user is logged out and brought back to the login screen with a message saying that his or her password is changed correctly, otherwise the password stays the same.



For a user to deactivate his or her account, they press a button, that has a delete Account method that is on Click. The current user is obtained, thanks to firebaseAuth.getInstance.getCurrentUser(). Firebase auth has a method that allows the users to be deleted easily by just calling user.delete() method, which deletes all the user’s entries in the authentication database, and there are no details left about the user.

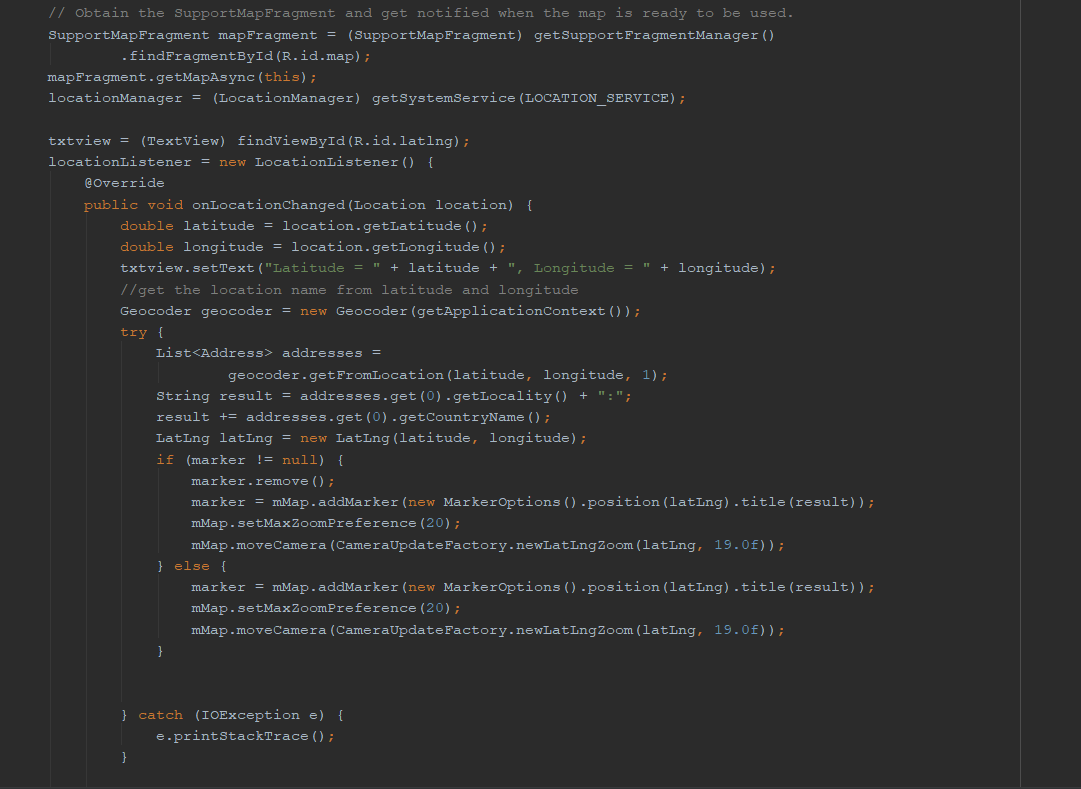
## Language

I wrote the application in Java, as in second semester I learned java with android studio, but also, I had a backup language called Kotlin. Kotlin is a newer version of java, with many things just made easier for the developer to write code in. Java is used, as Kotlin is only a new language, and as much as it simplifies the creation of applications by making some of the methods easier, the language is based on Java. Java was my go to language due to the fact Java apps are more compact – in comparison to Kotlin, Java apps tend to be lighter, which is good for my final year project as I do not know what the phone of the user is going to be. A Kotlin app that includes complex computing processes in its code can turn out to work slowly on user devices with low technical specs. Java also allows top development speed, and faster build process than kotlin according to netguru (netguru, 2018). The build time for my assignment is vital as the quicker the user gets the route, and the quicker the application opens, the more chance the user would have to escape an emergency. According to Netguru Kotlin also has a small developer community, which means limited learning resources and difficulty in finding answers to questions. At looking only on stackoverflow there are only around 8000 questions tagged with Kotlin against 1.37 million questions about Java.

## Google Maps API

An API (application programming interface) is a framework that you can use to write a program. ... For example, web mapping APIs typically include classes for maps and layers so that you don't have to write all the low-level code for displaying an interactive map image and drawing a new layer on it.

The Google Maps API allow for the embedding of Google Maps onto web pages of outside developers, using a simple JavaScript interface or a Flash interface. It is designed to work on both mobile devices as well as traditional desktop browser applications. The API includes language localization for over 50 languages, region localization and geocoding, and has mechanisms for enterprise developers who want to utilize the Google Maps API within an intranet. The API HTTP services can be accessed over a secure (HTTPS) connection by Google Maps API Premier customers (programmableweb,2018).



I have a class called Maps Activity that contains and holds the map. The user of the application must give permissions for the application to get his or her current location. I get the location to be very accurate, as that is important for the user, to use when they want to escape from a specific room, to know exactly which room they are in. Thankfully google has a very accurate map, allowing the user to know which room he is at, when he or she specifies the floor they are at. I tried this feature in WIT and the results I obtained were very good. In the code, I get the location from the get System Service Location Service method. I have an On Location Changed method. This is to make sure that the map updates in real-time with the user. If a user moves, the application will know and present a new location. I have added a co-ordinate block too, in case the user had to specify where he currently is to emergency services, and to make the user more aware of the surroundings.

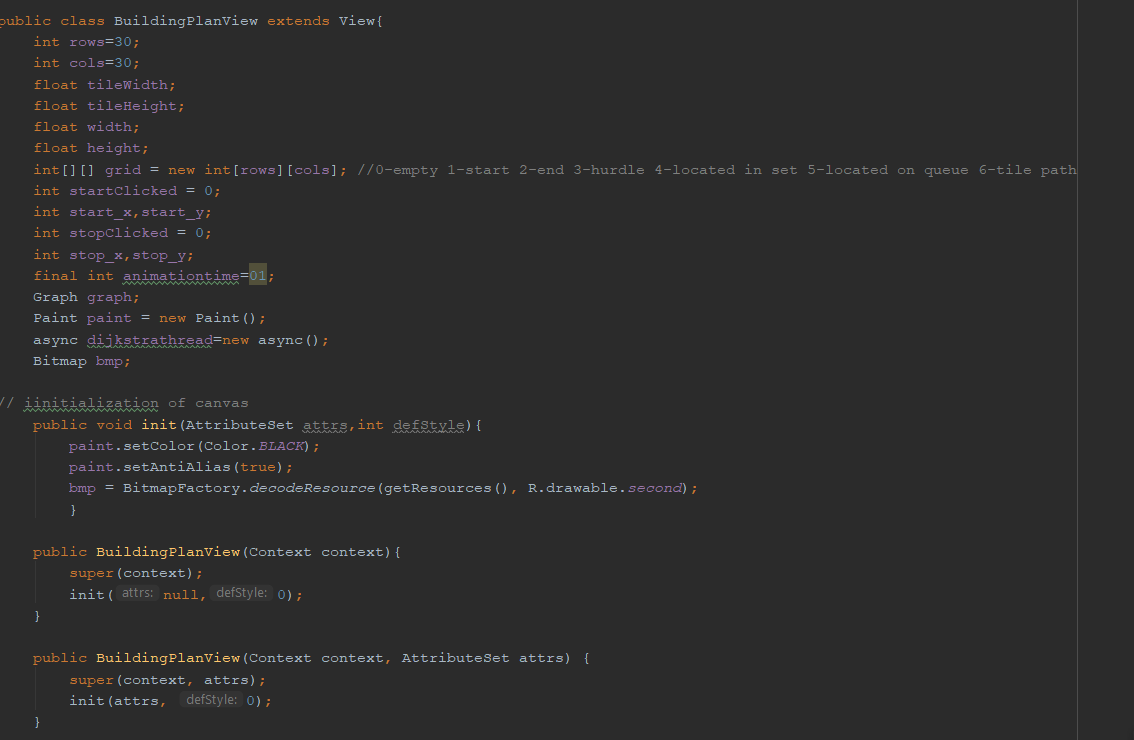


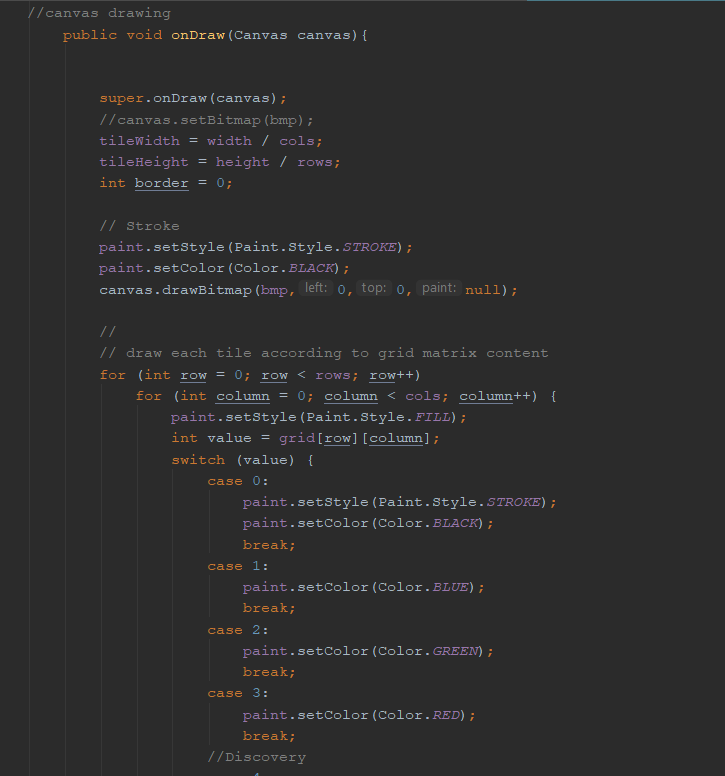
This screenshot was taken before I implemented the code, showing the correct co-ordinates, as I will show in the demo. The user would need this information to specify his/her location on the graph where they start and where they end. I decided not to store this information, as it would be taking security breach risks to a high level.

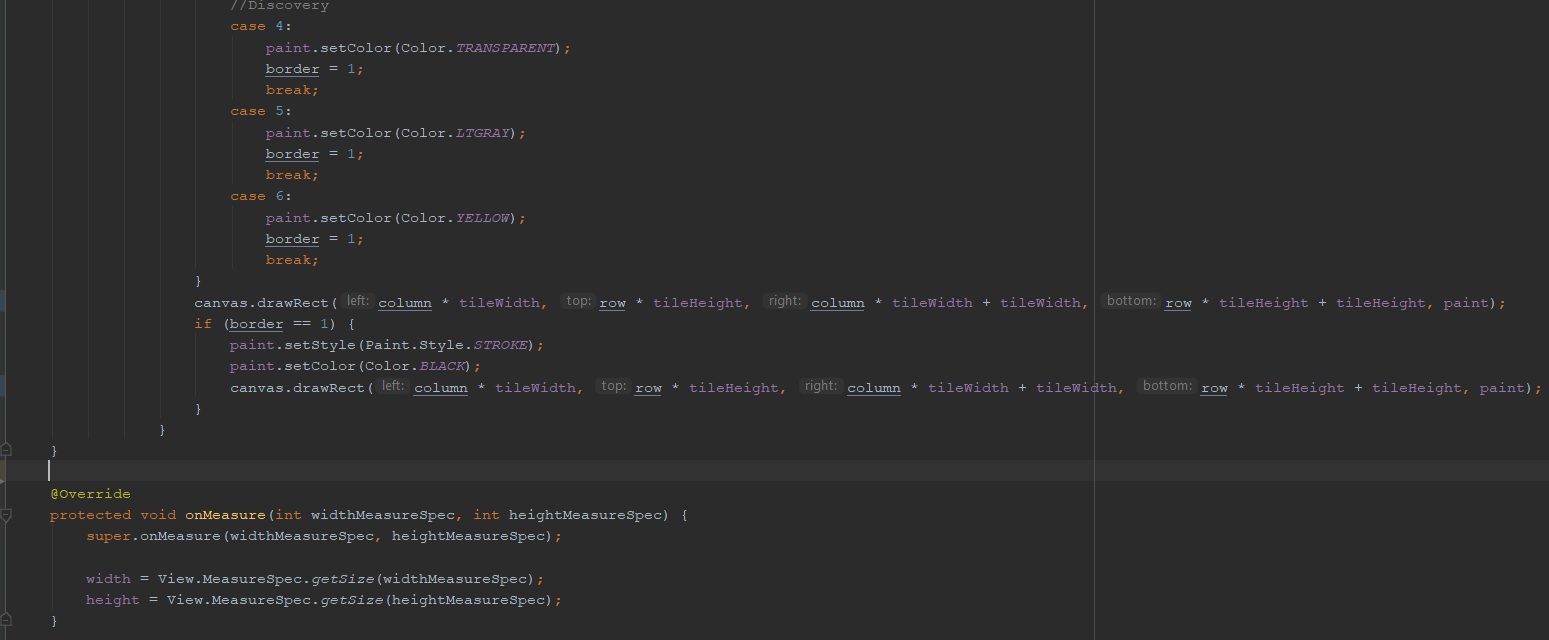
I had problems with API keys, as I’ve created a number of projects previously and had to contact google to get more keys available for my projects, this was a bit of a challenge and a pause in my application development as without the location the user would not know where they are and where to set the start points of the escape to be correctly. The keys were then working but I had to add a package on the google maps API website, as before that, I was just getting a blank map and it was not working correctly. After I completed the map, which was a vital part of my assignment, I decided to design and do out the algorithm, Dijkstra’s and started to look up ways of implementing it. I will talk about this in the next part of the report. Location was in the end obtained accurately and very precisely for the user to use. Not in the video, or the screenshot above, I’ve added a button to save the user from going back to the main screen to get building plan, and instead they are brought to the screen directly from the Maps Activity. This is a user experience addition, as the user would have fresh in his/her mind where they were last time.

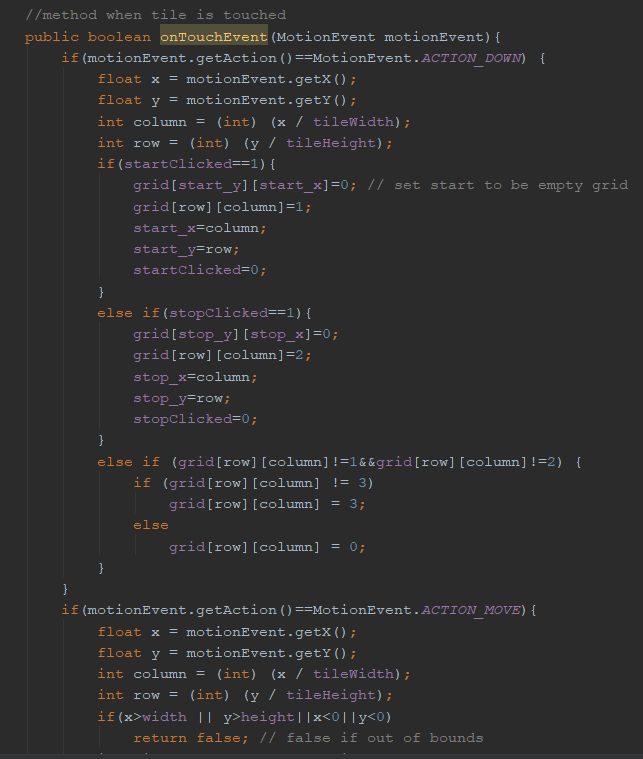
## Route finding – Dijkstra’s algorithm implementation

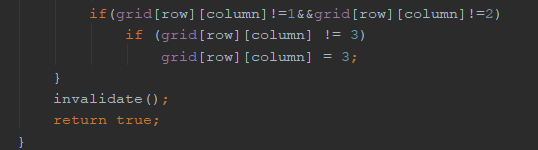
The user is given a route to escape, but how is the route computed. I use one of the algorithms that I learned during Erasmus called Dijkstra which works by starting at one vertex and exploring adjacent nodes until the destination node is reached. Dijkstra algorithm begins with a start node and an "open set" of candidate nodes. At each step, the node in the open set with the lowest distance from the start is examined. The node is marked "closed", and all nodes adjacent to it are added to the open set if they have not already been examined.

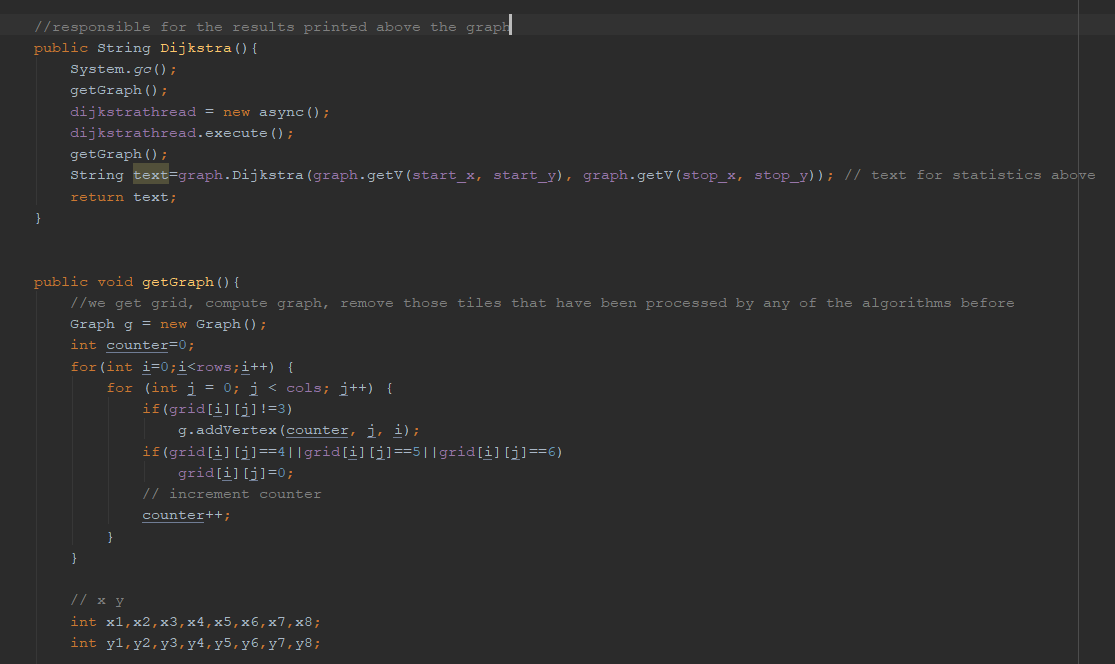


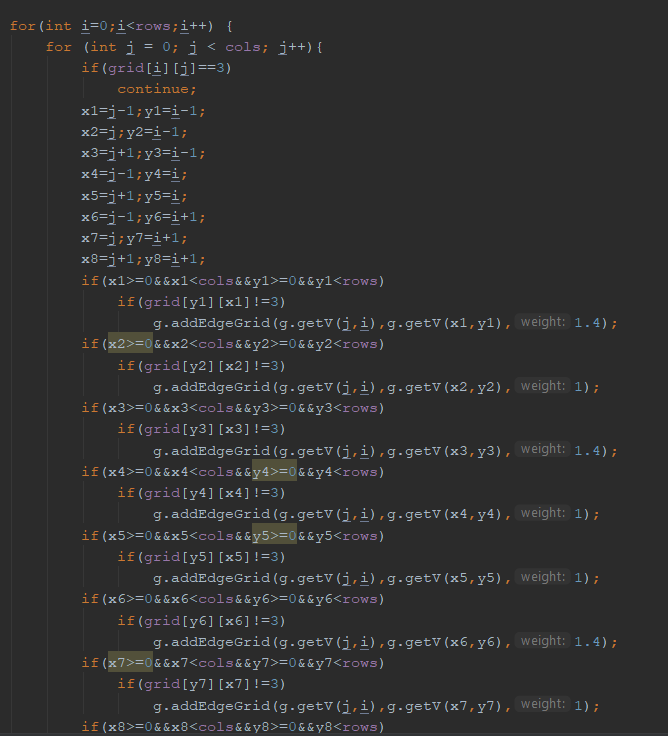


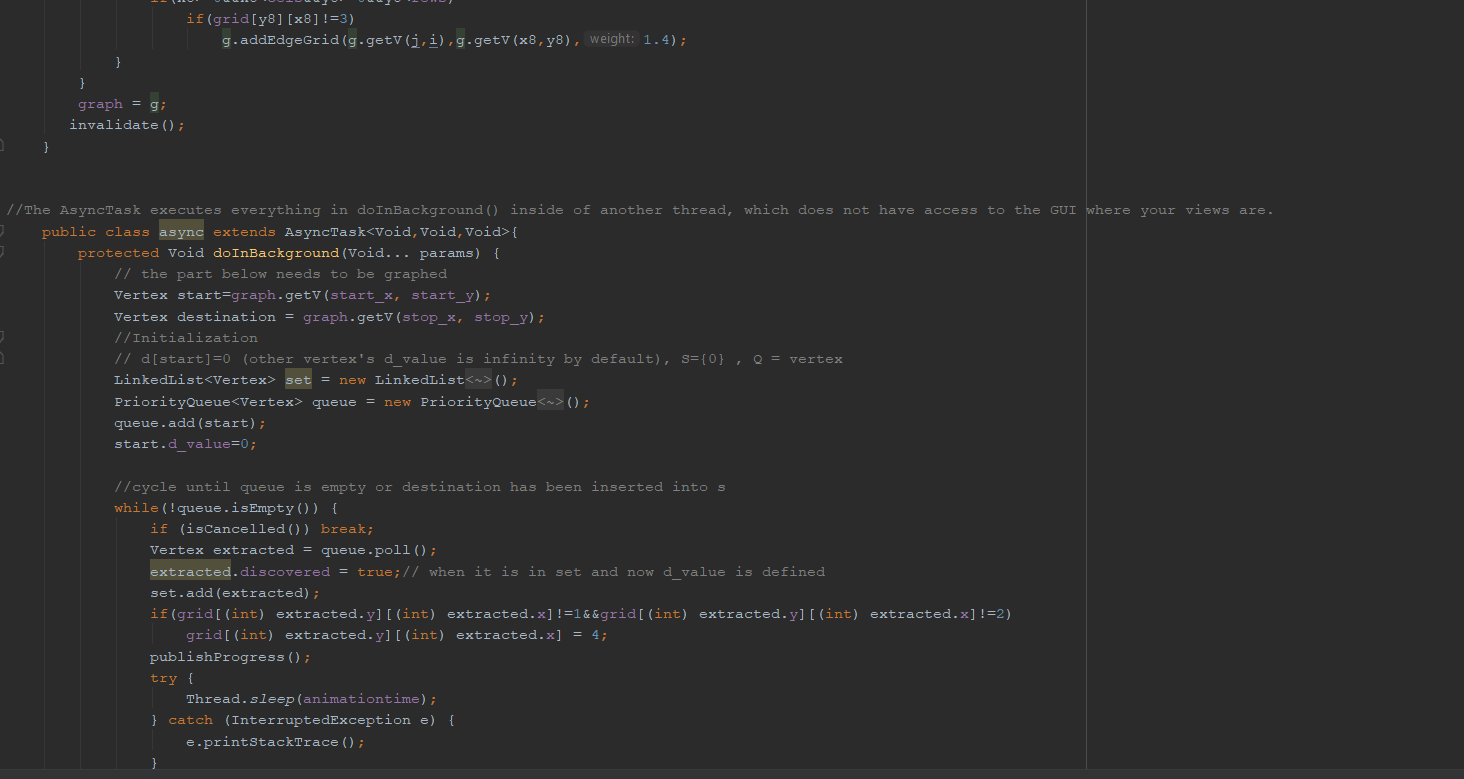


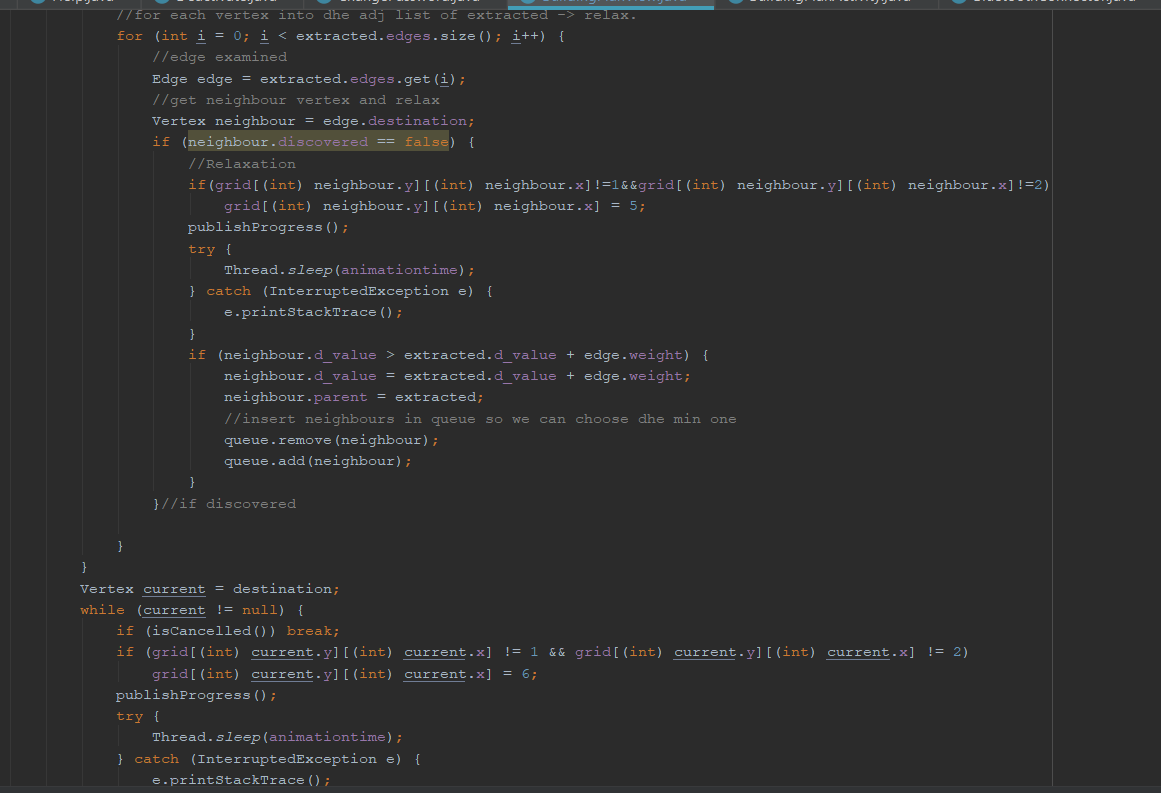






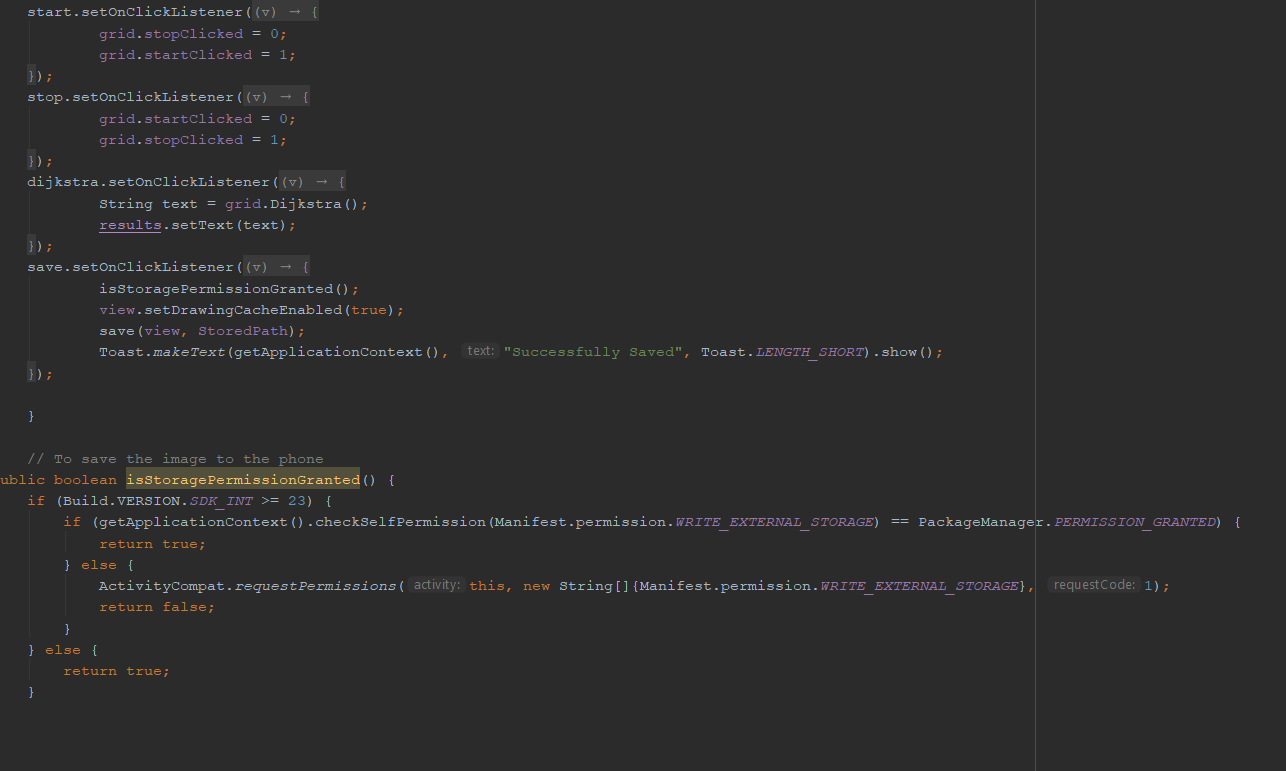


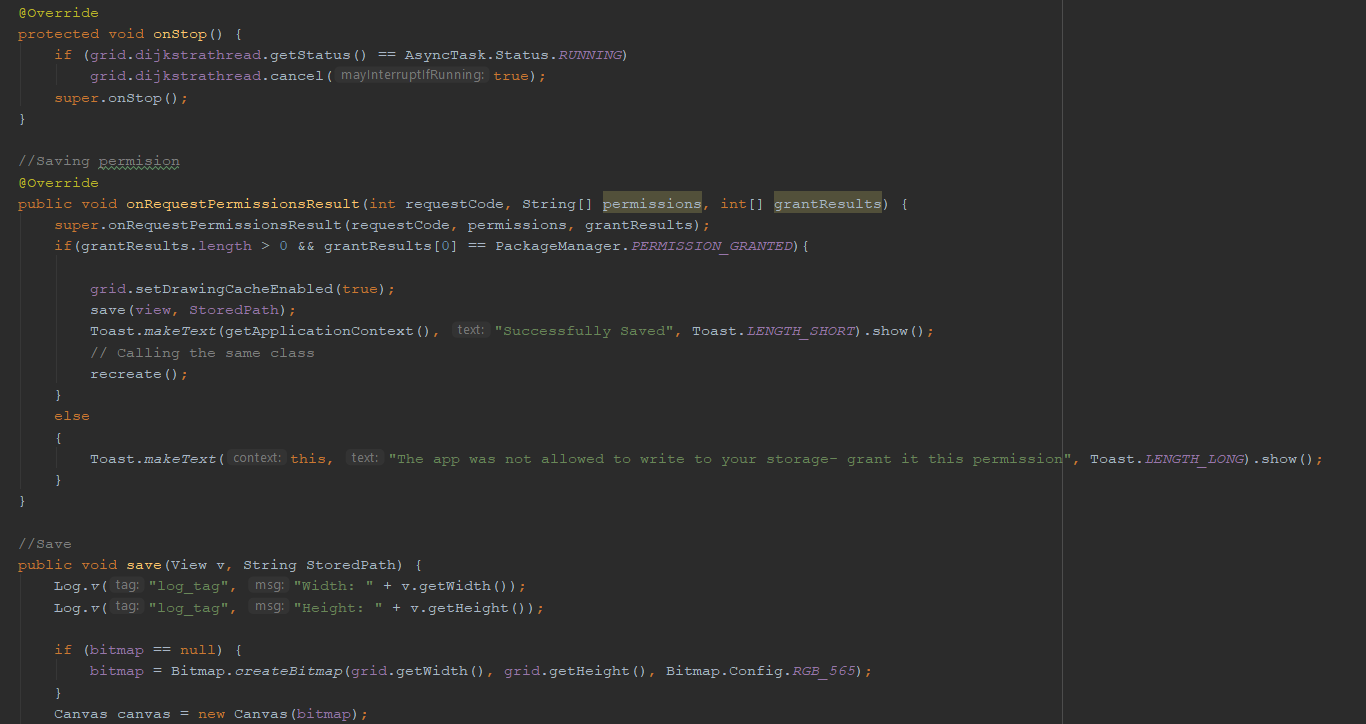
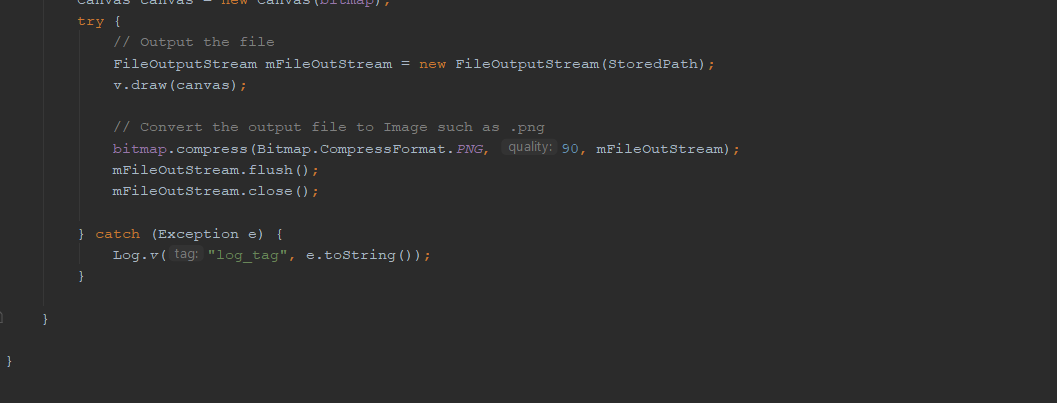




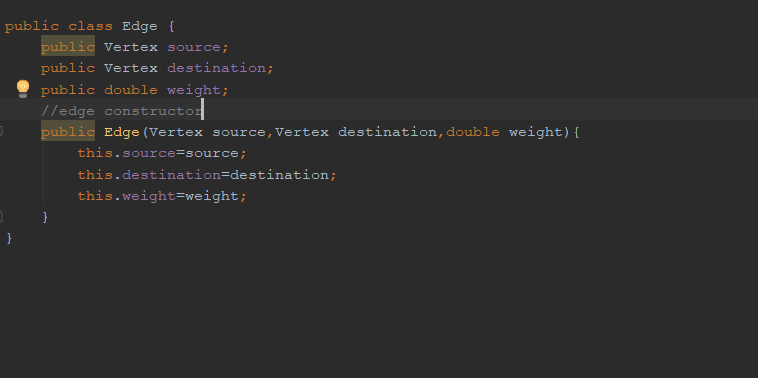
\

Activity

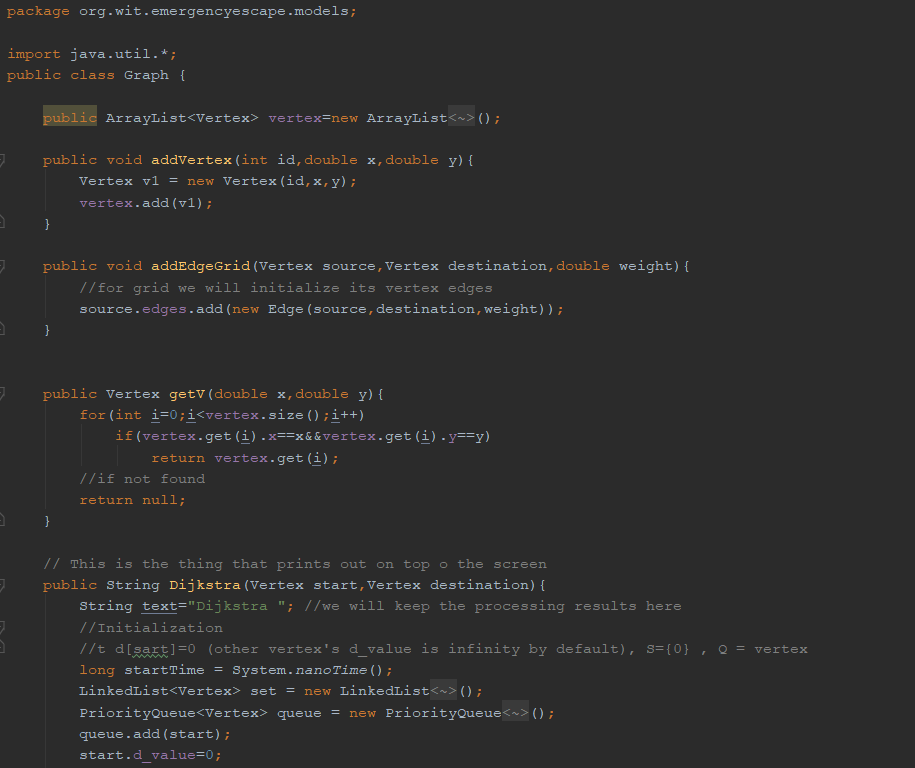
 

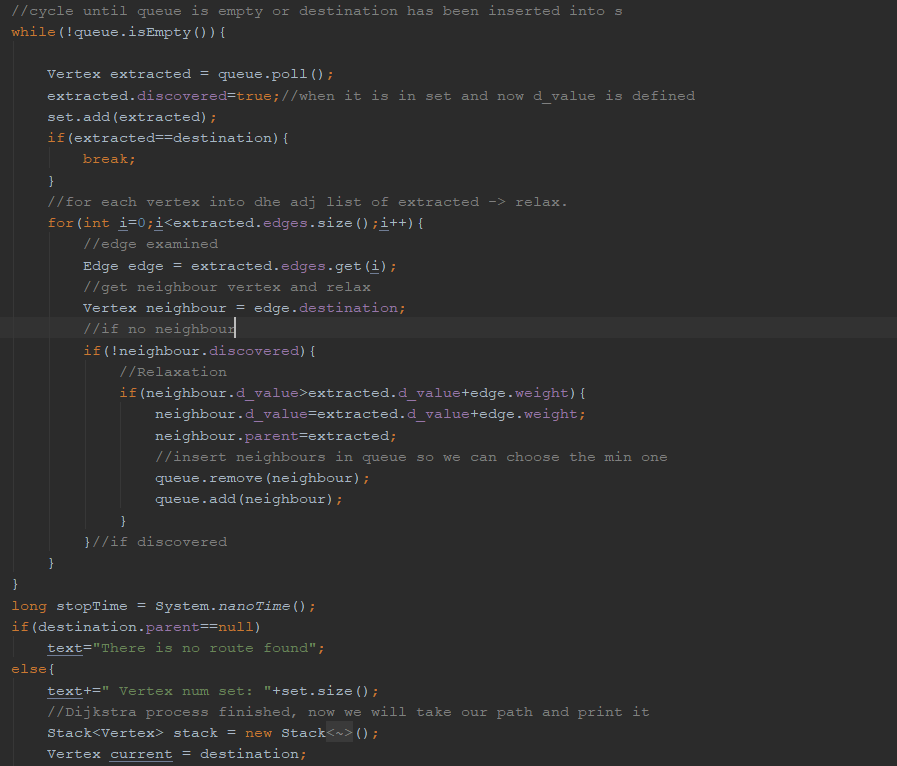
 

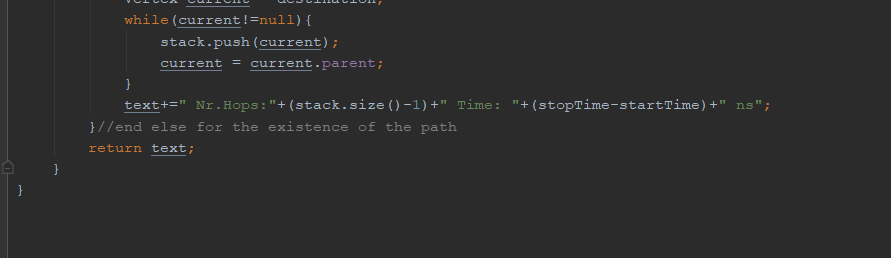
Edge



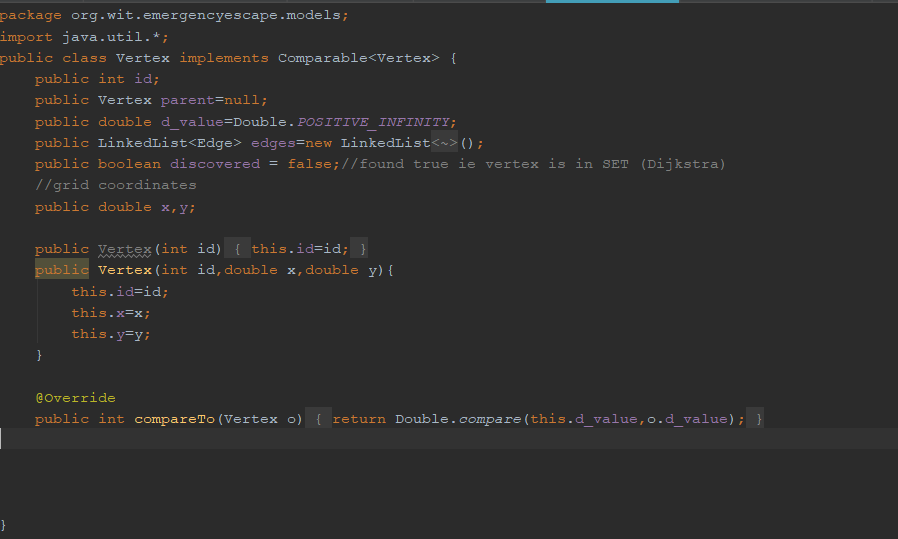
Graph







Vertex



As my application would develop I could use neural networks to find the best

route, if the number of users was enough, to test if the network would be enough. The users would be inputs to the neural network, and the neural network second layer could be different locations and based on the current location of the user the route would be shown. This would be an overkill for my final year project at this stage though as I do not have many users.

## Canvas Android

I will have a map showing the location of the user, and a building image plan that will display the emergency exits and the current location of the user. The application will draw a route thanks to the route-finding algorithm found above. The map of the building i.e. a building plan is going to have to be an image, and there are two formats of the image I consider. PNG is the one I plan to use, as it is a lossless compression file format, which makes this format very good for line drawings, texts and it is a small file size, which makes it better for the loading time of the application which is vital. I could use a jpg format to use it, which is a lossy compressed file format, but for storing line drawings and iconic graphs at a smaller file size PNG is better, so in case there is errors for loading PNG I will use JPG, but I plan to use PNG (labnol, 2014).

# How security is dealt with?

Security will be important for this project, as each person will be having to specify their own location. I will make sure the variables for latitude and longitude are kept in a private safe database, and that each user would have to log in, and when logged in they would only be able to see their own private location. No other person’s location could be given away, and personal information such as name and surname and other details given at the register stage could not be given away. The log in/ register feature is there to provide security to avoid many people entering the system and trying to hack the system. Thanks to the use of my database storage, firebase, I could allow users to log in with different accounts from Facebook / Gmail and those passwords are hashed and salted.

# Plan for the application after the final year project

I plan to get Arduino connected to the application, this will be done using a java connector. The application then will be written with the connector taken care of. The app would have a database connection which I will take care of by connecting the firebase database, to keep information of users log in details and the current location.

The application would then be tested manually to see if a route is given for a user when an emergency occurs, and then if the buzzer sounds to check the connectivity.

Firstly, I plan to get the administrator and building owner rights in onto the system, admins can add building owners, and building owners can add or delete maps. The user would not have any access to adding or deleting maps, or owners. The user will only have a map of the building, and when an emergency happens the user will be able to see the path. In building the application, I would make sure basic functionality such as login into the database, are met as security of data will be

vital in my project, due to the user providing the location he is currently at.

The application at its final stage would be connected to the amazon web services to be deployed in the cloud for many users.

# Draft user manual

The user would download the application from the app store. The user then would have to register for safety reasons and for the information to be used by the application regarding location, to keep a database of the user. The user would then log into the application where both a map would be displayed, showing the current location and a building plan. If an emergency is sent from the server by the administrator to the app, the quickest route is displayed using an algorithm and graphs in java on the PNG image file. The buzzer also sounds from the Arduino board to notify the user. The app then draws a navigation path to the nearest emergency exit and that is displayed on the app.

# Learning curve

This is the first time I plan to do this project, where I must link things that I have learned throughout different subjects and combine them together to get a fully working effect. The things I learned at doing this report are vital for my future as an IT graduate. I had researched the things that I plan to do, and the things that I will do if things do not work out for me. I plan to continue this approach in different areas of my project, and I plan to use my problem-solving skills to the best of my ability. The project is a challenge to show my time keeping skills, my programming skills and my way of working around problems that will arise in the project. I will also develop an app in a language I didn’t work in before, and I will use Arduino, which is going to be an interesting challenge.

# Citations

Aptuz Technology Solutions. 2015. 6 Reasons Why We Chose AWS | Aptuz Technology Solution. [ONLINE] Available at: http://aptuz.com/blog/6-reasons-why-we-chose-aws/. [Accessed 01 December 2018]

Digital Inspiration. 2018. JPEG or PNG - Which Image Format Offers Better Quality. [ONLINE] Available at: https://www.labnol.org/software/tutorials/jpeg-vs-png-image-quality-or-bandwidth/5385/. [Accessed 01 December 2018].

Firebase. 2018. Cloud Storage | Firebase. [ONLINE] Available at: https://firebase.google.com/docs/storage/. [Accessed 01 December 2018].

Lifewire. 2018. Pros and Cons of Splash Pages in Web Design. [ONLINE] Available at: https://www.lifewire.com/splash-pages-pros-cons-3469116. [Accessed 03 April 2019].

Name year Dummy's Codes: Using Java RxTx library for Serial Communication with Arduino - Part 1. Available at: http://dummyscodes.blogspot.com/2014/08/using-java-rxtx-library-for-serial.html. [Accessed 10 November 2018].

Name 2018 Kotlin vs. Java: Which One You Should Choose for Your Next Android App | Netguru Blog on Kotlin. [ONLINE] Available at: https://www.netguru.co/blog/kotlin-java-which-one-you-should-choose-for-your-next-android-app. [Accessed 01 December 2018].

ProgrammableWeb. 2019. Google Maps API | ProgrammableWeb. [ONLINE] Available at: https://www.programmableweb.com/api/google-maps. [Accessed 23 April 2019].

SourceForge. 2018. Java-Arduino Communication Library download | SourceForge.net. Available at: https://sourceforge.net/projects/javaarduinolibrary/. [Accessed 09 November 2018].

The Firebase Blog. 2018. The Firebase Blog: What’s new at Firebase Summit 2018.

Available at: https://firebase.googleblog.com/2018/10/whats-new-at-firebase-summit-2018.html. [Accessed 01 December 2018].

# Minutes from meetings – Semester 1

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Date | Work Completed | Y/N | Work to Do |
| 18/Oct/18 | * Discussed grading scheme of project * Discussed project Idea | Y | * Project proposal start * Technologies that I will use |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Date | Work Completed | Y/N | Work to Do |
| 22/Oct/18 | * Discussed the diagrams of the project and what my project is about | Y | * Write up a report about the technologies that I will use |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Date | Work Completed | Y/N | Work to Do |
| 5/Nov/18 | * Basic report completed * Done over email due to meetings and me working | Y | * Fix the things highlighted by the supervisor * Make the report more aimed at my project |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Date | Work Completed | Y/N | Work to Do |
| 12/Nov/18 | * Report Fixed * Technologies changed and looked up * Fixed Harvard referencing. | Y | * Do minutes for all the reports? * Write about alternatives of the technologies I will use. |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting Date | Work Completed | Y/N | Work to Do |
| 30/Nov/18 | * Minutes fixed * Alternatives written up | Y | * Presentation * Spelling fixes * Citation fixes |

# Minutes from meetings – Semester 2

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 1 | * Refactored login screen * Used proto.io to decide on the look of the application * Meeting time arrangements | Y | * Plan out timetable * UML diagrams * Start looking at W3C guidelines * Splash screen |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 2 | * Planned out timetables * UML Diagrams finalised * Develop a splash screen myself to avoid copy right infringements | Y | * Settings to allow users to see their location * Co-ordinates placeholder * Google Maps API |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 3 | * Settings to allow users to see their location * Co-ordinates placeholder * Google Maps API | Y | * Building plan tab * Drew out a building plan * Classes for coding the Dijkstra’s algorithm * Cognito form for the industry day |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 4 | * Building plan tab * Drew out a building plan * Classes for coding the Dijkstra’s algorithm * Cognito form for the industry day | Y | * Fix library imports * Arduino connection * Sensors and modules in Arduino code * Connecting Arduino with application |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 5 | * Fix library imports * Arduino connection * Sensors and modules in Arduino code * Connecting Arduino with application | Y | * Connectivity checks * GitHub documentation and converting all to Java, as has a few Kotlin classes- as that was my previous module and I had to learn the Java way of implementing something * Buzzer test * Coordinates need to be displayed in the placeholder. |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 6 | * Connectivity checks * GitHub documentation and converted all to Java, as has a few Kotlin classes- as that was my previous module and I had to learn the Java way of implementing something * Buzzer test * Coordinates displayed in the placeholder. | Y | * XML updates to make application look more appealing * Logo of the application will be developed in GIMP * Add help window that helps the users to use the application * Add info tab, for users to contact me the developer with any inquires |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 7 | * XML updates to make application look more appealing * Logo of the application developed in GIMP * Added help window that helps the users to use the application * Added info tab, for users to contact me the developer with any inquires | Y | * Add a Bluetooth file written in Arduino to connect to the application. * Display ip address of Bluetooth * Download nRF Connect on the phone as it gives a debugging tool on the Bluetooth signals from the Arduino. * Add an option to deactivate an account. * Updated Change Password method * Add Parser method, that parses the building plan and adds and creates rooms. * Connect buzzer and button works when I press to buzz. * Add consistent backgrounds in the app and consistent app bar names |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 8 | * Added a Bluetooth file written in Arduino to connect to the application. * The ip address now shows up on the application i.e. they are talking. * Needed to download nRF Connect on the phone as it gives a debugging tool on the Bluetooth signals from the Arduino, so I knew what was going on when I failed to connect. * Added an option to deactivate an account. * Updated Change Password method to keep the email of signed in user, instead of typing it out again * Added Parser method, that parses the building plan and adds and creates rooms. * Connected buzzer and button works when I press to buzz. * Added consistent backgrounds in the app and consistent app bar names | Y | * Dijkstra’s algorithm to obtain a correct graph to display and look at possibly older versions i.e. legacy items to display the thing I would like to display. * Create a poster * Update buttons for a better user experience |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 9 | * Coded Dijkstra’s algorithm gets a correct graph to display and look at possibly older versions i.e. legacy items to display the thing I would like to display. * Created a poster – draft model * Update buttons for a better user experience | Y | * Trying to figure out escape route still, the graph works but only if I specify the nodes statically and calculate the route myself. * Finish the poster * Start the report write up * Test the application |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 10 | * Finish the poster * Start the report write up * Test the application | Y | * Add a better UI * Allow users to save the escape route diagram onto the phone * Report * Video * Print Poster |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 11 | * Printed Poster * Added a better UI with diagrams * Added Save button for users to save the diagram * Report Started | Y | * Report * Further Application Improvements * Video |

|  |  |  |  |
| --- | --- | --- | --- |
| Meeting | Work Completed | Y/N | Work to Do |
| Week 12 | * Tested the application * Report Started * Video Started | Y | * Report * Further Application Improvements * Video End |