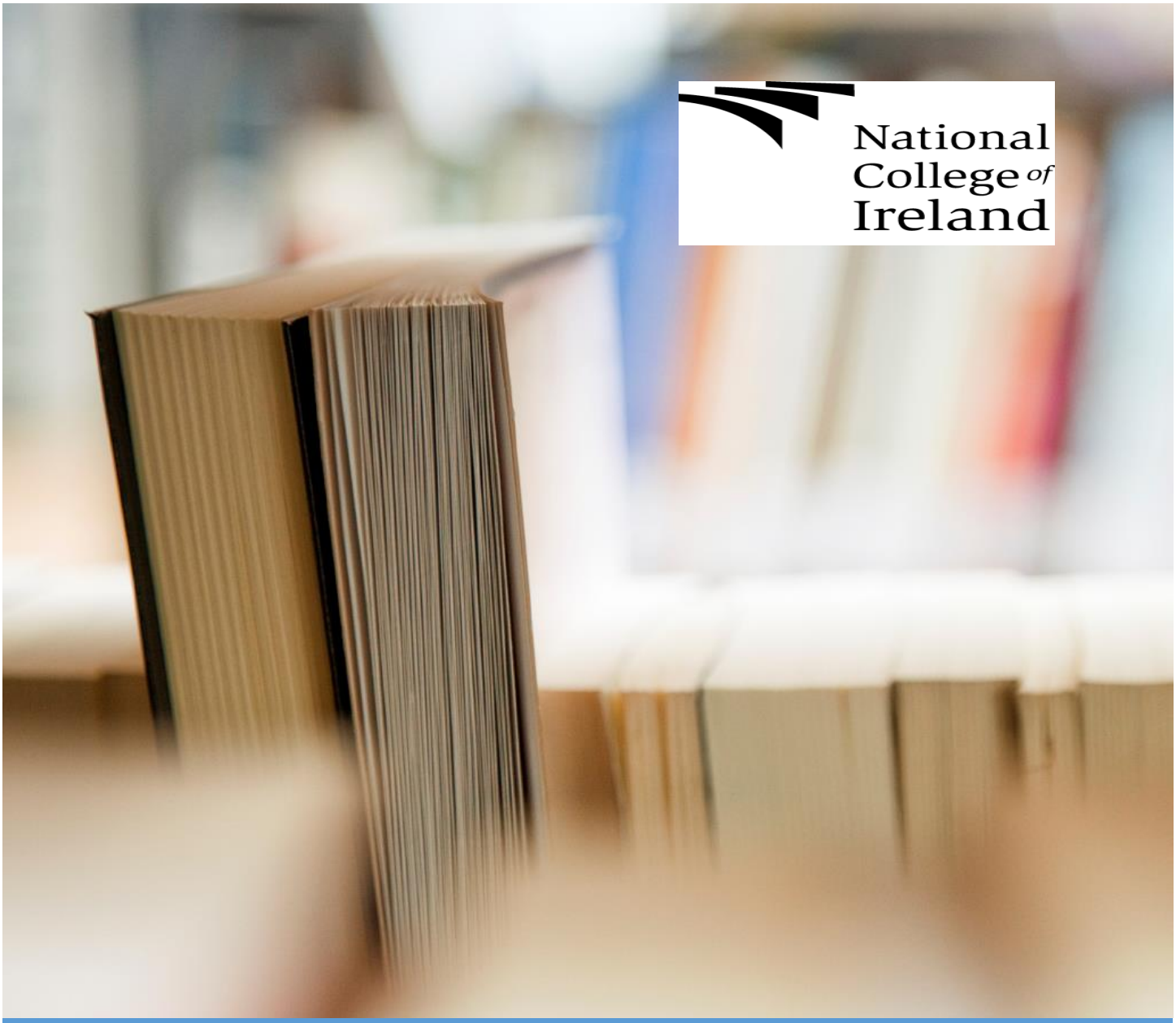


National College of Ireland

Final Technical Report

BangOnTaxiApp

X13728111

Maricel Costel Buiciuc **5/10/17** **BSCH Software Stream**

Table of Contents

1 Executive Summary 3
2 Introduction 4
2.1 Background 4
2.2 Aims 5
2.3 Technologies 5
3 System 6
3.1 Functional Requirements 6
3.1.1 Requirement 1 <User Registration> 7
3.1.2 Requirement 2 <User Log In> 8
3.1.3 Requirement 3 <Displaying Road Info Activity> 10
3.1.4 Requirement 4 <Speech Recognition Activity> 11
3.1.5 Requirement 5 <Points and Position on the Leaderboard> 13
3.1.6 Requirement 6 <User Profile> 15
3.2 Non-Functional Requirements 17
3.2.1 Performance/Response time requirement 17
3.2.2 Availability requirement 17
3.2.3 Recover requirement 17
3.2.4 Robustness requirement 18
3.2.5 Security requirement 18
3.2.6 Reliability requirement 18
3.2.7 Maintainability requirement 18
3.2.8 Portability requirement 18
3.2.9 Extendibility requirement 18
3.2.10 Resource utilization requirement 18
3.3 Data Requirements 18
3.4 User Expectations 19
3.5 Environmental Requirements 20
3.6 Usability Requirements 20
4 Design and Architecture 21
4.1 G.U.I. Graphical User Interface 22
4.1.1 Welcome Activity 22

4.1.2 Register Activity 23
4.1.3 Login Activity 24
4.1.4 Main Menu Activity 25
4.1.5 Street Map Activity 26
4.1.6 Road Info Activity 27
4.1.7 Accidents Activity 28
4.1.8 Traffic Activity 29
4.1.9 Checkpoints Activity 30
4.1.10 Ranks Activity 31
4.1.11 User Profile Activity (Update Profile) 32
4.1.12 User Profile Activity (Update Password) 33
4.1.13 Leaderboard Activity 34
4.1.14 Speech Recognition Activity 35
5 System Architecture 36
5.1 System Architecture Diagram 36
5.2 Class Diagram 37
5.3 Flow Diagram 38
6 Analysis and Design 38
6.1 Purpose of the Product Design Specification 38
7 Overview and Design Guidelines/Approach 39
7.1 Assumptions / Constrains / Standards 39
8 Architecture Design 39
8.1 Logical View 39
8.2 Hardware Architecture 39
8.3 Security Architecture 40
9 System Design 40
9.1 Application Algorithms and Methods 40
9.1.1 Registration 40
9.1.2 Login 41
9.1.3 Street Mapp Geolocation 41
9.1.4 Leaderboard 42
9.1.5 Speech Recognition 43
9.1.6 Accidents 43

9.1.7 User Profile (Update User Details and Password) 44
10 Testing 45
10.1 Think Aloud Test 45
10.2 Think Aloud Tasks 45
10.3 Results for Think Aloud Test 45
10.4 Effectiveness 47
10.5 Efficiency 47
10.6 Scope for Further Improvements 47
10.7 Testing Conclusion 47
10.8 System Usability Scale 48
11 System Evaluation 48
12 Conclusions 49
13 Appendix 49
13.1 Project Proposal 49
13.2 Monthly Journals 52
13.2.1 September 2016 Reflective Journal 52
13.2.2 October 2016 Reflective Journal 53
13.2.3 November 2016 Reflective Journal 54
13.2.4 December 2016 Reflective Journal 55
13.2.5 January 2017 Reflective Journal 56
13.2.6 February 2017 Reflective Journal 57
13.2.7 March 2017 Reflective Journal 58
14 References 59

1 Executive Summary

The purpose of this project is to create a mobile app for all taxi drivers in Dublin, which might want to interact, to each other by using this app. Every taxi driver would be very important because she/he can bring some relevant information to this app and make it more efficient. This application will be unique because will have a speech recognition functionality integrated which allows drivers to create text messages without typing any word.

Part of this app will be also related to “Gamification” which is a concept that awards the users in a way that increases interactivity with the application and records their progress. It can take the form of creating one or many messages, and all these activities/messages will be counted and the results will be posted for other users to view.

Awards are gained by the users in the form of, total number of points achieved and a ranking place on a leaderboard. The gamification aspect of this application will be based on messages on each section of the app. The user’s activity/results will be displayed on a leader-board, allowing them to compare themselves with other users.

This mobile app would be available to work on both known operating systems, iOS and Android. The framework chosen to build this project is the cross-platform named Ionic framework and the scripting language is AngularJS. This Ionic framework allows programmers to build hybrid applications without having to write code twice for each operating system. This AngularJS scripting language gives the ability to control and create interactions within the app. The Ionic framework uses also Cordova library which will bundle the web code and compile it into native applications for both mobile platforms.

For the database storage, Firebase was chosen to be used because it is a free platform owned by Google and is very helpful to develop real-time applications. It uses NoSQL database structure that stores data in a JavaScript object notation and this makes it easy to work with AngularJS Apps.

2 Introduction

2.1 Background

The taxi businesses in Dublin are owned by a few different people, which have in place and use their own personalised app. Motivated by this objective, I thought that this is a very good hint to build an app which might be a real help for all the taxi drivers in Dublin city. The idea is to build an app which can be used by any taxi drivers no matter what company they belong to, but also to be used by any private taxi drivers in town.

At present, all taxi driver uses their own devices and any social media network to interact with each other in relation to road information, which is not a very good practice.

As Dublin became a very busy city and the transport is vital for the people, the idea to build a new app for all the taxi drivers in a necessity. This is supposed to be an app that has more advanced functionalities than any existent apps on the market.

Now, this would be the only taxi app in Ireland that allows drivers to interact with the application by using the speech recognition functionality. The drivers would be able to leave voice messages which will be converted into text messages straight away and displayed on the app for other drivers to view. The taxi drivers will be delighted with this idea of real time information functionalities and the app is going to be a real benefit for all taxi drivers.

2.2 Aims

The scope of the Ionic project is to develop a cross-platform mobile app allowing for the application to be used by any taxi drivers that have either an Android or iOS mobile device. This application will be unique because it allows drivers to create text messages without typing any word.

This is supposed to be an attractive app with a user-friendly look and especially with better and more advanced functionalities than any existent taxi apps.

The functionalities on this mobile app should have a huge impact on the taxi drivers and should be a real benefit to all taxi businesses in Dublin.

One interesting thing about the app is the gamification aspect. This can attract users to interact more with the app, as taxi drivers have plenty of time to use their devices. The leaderboard will rank the users depending on how many activities/messages they create on the app which can be more fun for them.

2.3 Technologies

This project will be developed using the Ionic framework which works fine with AngularJS. This AngularJS scripting language gives the ability to control and create interactions within the app. Ionic also uses Cordova library which will bundle the web code and compile it into native applications for both mobile platforms iOS and Android. Also, “Annyang” Speech Recognition library will be integrated into the app to transform the voice messages into text messages.

For styling, some CSS will be used but also Sass will be integrated automatically into the project because is used by the Ionic framework to create its styles, which is a CSS pre-processor that has a lot of abilities on top of CSS. Ionic also uses

Gulp.js to manage the project, processing Sass, and handle other tasks. Also, this app will use some additional API's for example, Google maps.

For the database storage, Firebase will be used because it is a free platform owned by Google. This tool is very helpful to develop a real-time application. It uses NoSQL database structure that stores data in a JavaScript object notation which suits AngularJS Apps. One of the main features in Firebase is, the ability to handle different kinds of registration automatically.

3 System

The system structure of this technical report will be outlined and described in this section. There is the presentation in detail of the requirements for this cross-platform Ionic framework application. Also, the Design and Architecture of the entire System will be presented as well with all components bonded together to form the general working environment.

3.1 Functional Requirements

- The app shall allow the new user to register with username and password, but also the taxi policy number. To use the app and its content, all users/drivers must log in with a valid username and password.
- The app shall give the user the ability to change the password and update the profile details.
- The app shall open its activities in reasonable time and be accessible from iOS and Android mobile devices
- The app shall direct the user to the activities he is looking for. All activities will be linked together to ensure that the user gets to the right place without any difficulties.
- The app should allow user to use its speech recognition functionality to create voice messages which will be converted into text messages.
- The app should have Google street maps integrated, where user can view traffic information's which happens on different roads in Dublin.
- The app should display fresh information where live Checkpoints/radar cameras are in Dublin.
- The app should display fresh information where the traffic is getting busy in Dublin.
- The app should display fresh information about the recent accidents in Dublin.
- The app should display information about taxi rank stations if they are already occupied.

3.1.1 Requirement 1 <User Registration>

3.1.1.1 Description & Priority

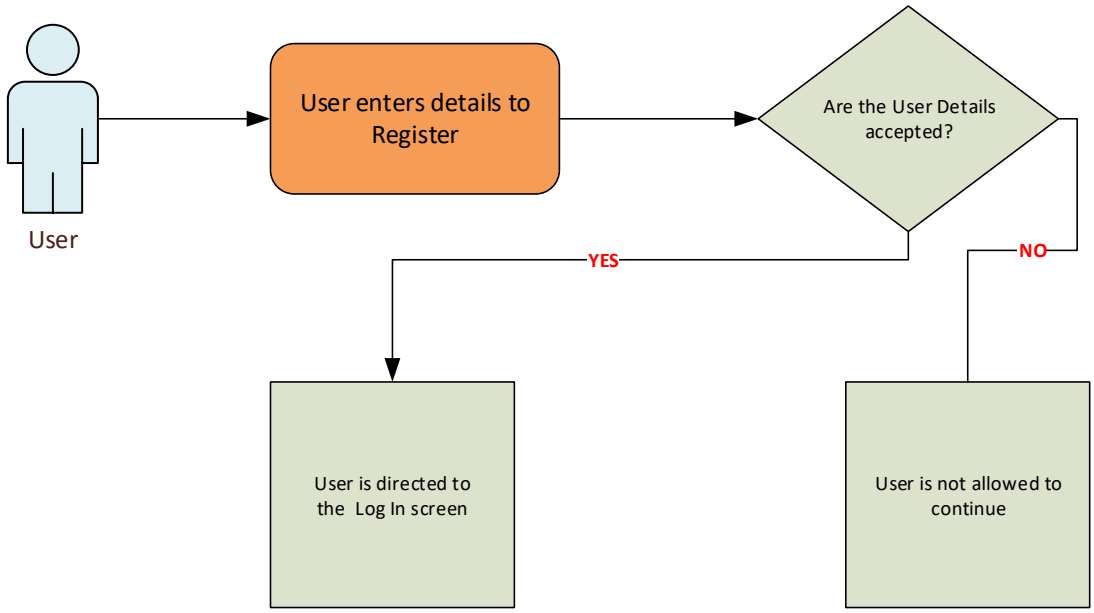
This use case represents a major piece of the system functionality. Here a user, can register to become a member. After a user is signed in successfully he/her can use the entire application system.

3.1.1.2 Use Case

Scope

The purpose of this use case is to show the steps when the user creates an account.

Use Case Flow Diagram



Precondition

Firs the user fills the required details in the correct fields

Activation

This use case starts when the user selects registration from the provided submit button.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1. User fills the required details for registration	1. The system accepts the registration request and saves it into the database 2. Application starts the Main Menu activity

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User fills invalid or already used details	1. The system can't accept the registration request 2. Application remains on the same register activity

Termination

The system presents that the user is successfully registered and the flow is finished after the details are saved.

Post condition

The system records user's details and redirects him to the Main Menu activity as the application reaches the "wait state" mode.

3.1.2 Requirement 2 <User Log In>

3.1.2.1 Description & Priority

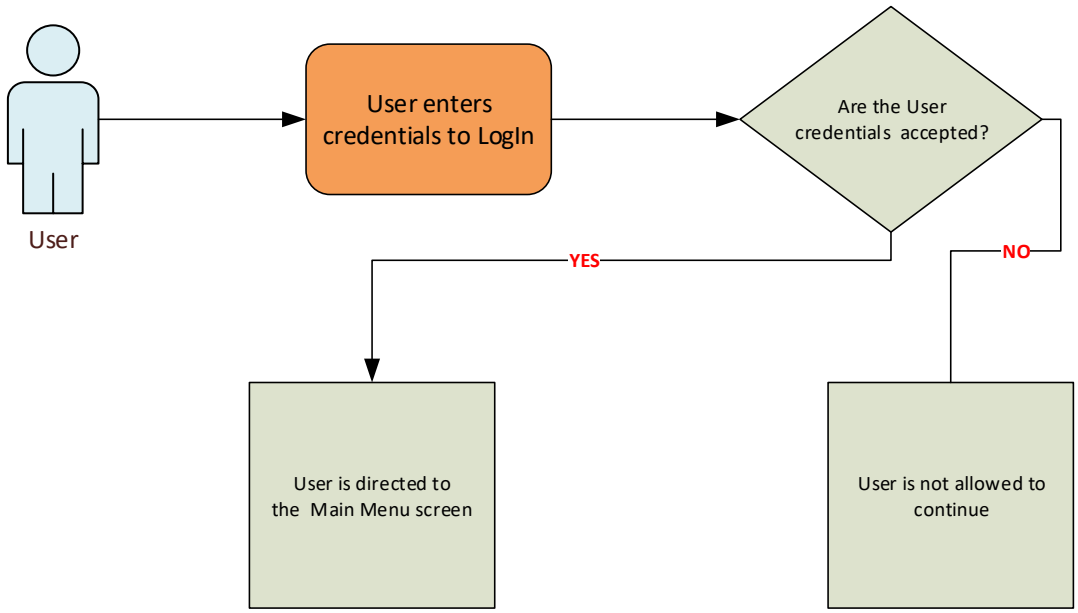
This use case represents an important piece of the system functionality. Here a user, can Login and only after that step he can use the entire application system.

3.1.2.2 Use Case

Scope

The purpose of this use case is to show the steps when user logs in on the app.

Use Case Flow Diagram



Precondition

First the user files the required details in the correct fields.

Activation

This use case starts when the user selects login from the provided submit button.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1. User fills the required details for log in	1. The system recognizes and accepts the login request 2. Application starts the Main Menu activity

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User fills invalid details to Log in	1. The system does not recognize the login details

	2. Application remains on the same login activity

Termination

The system presents that the user is successfully logged in and that the flow is finished, after that the user is redirected to the main menu activity.

Post -condition

The system accepts user's details and redirects him to the Main Menu activity as the application reaches the "wait state" mode.

3.1.3 Requirement 3 <Displaying Road Info Activity>

3.1.3.1 Description & Priority

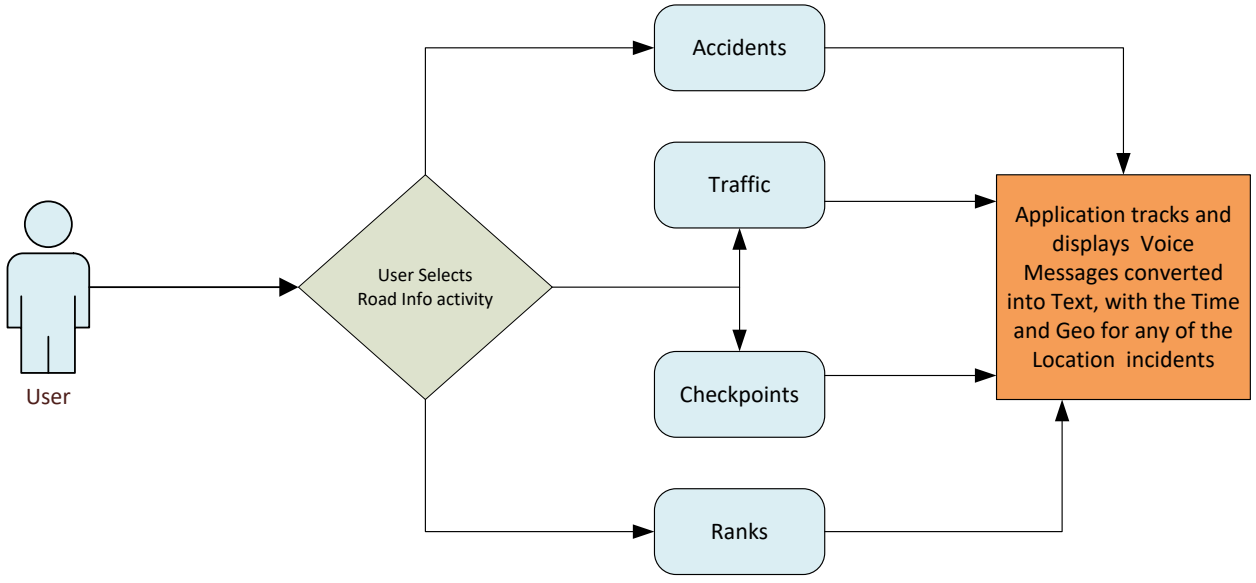
This use case represents an important piece of the system functionality, because the application's main goal is road information activities and if there is no option to display road information the rest of functionalities make no sense.

3.1.3.2 Use Case

Scope

The purpose of this use case is to show in detail the processes involved in displaying the Road Info activity.

Use Case Flow Diagram



Precondition
The system is in the initialization mode and shows that some other use case has already run to set it up.

Activation
This use case starts when a user starts one of the Road Info activity options.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1. User selects one of the Road Info options activity 2. User is starting that specific activity	1. Application starts to display the activity content chosen by the user

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User is not selecting any of the Road Info options activity	1. Application does not display any Road Information 2. Application remains in the initialization mode

Termination
Once the Road Info option is being displayed to the user successfully the flow is finished.
Post condition
Road Info option is displayed by the application and the system goes into a wait state.

3.1.4 Requirement 4 <Speech Recognition Activity>

3.1.4.1 Description & Priority

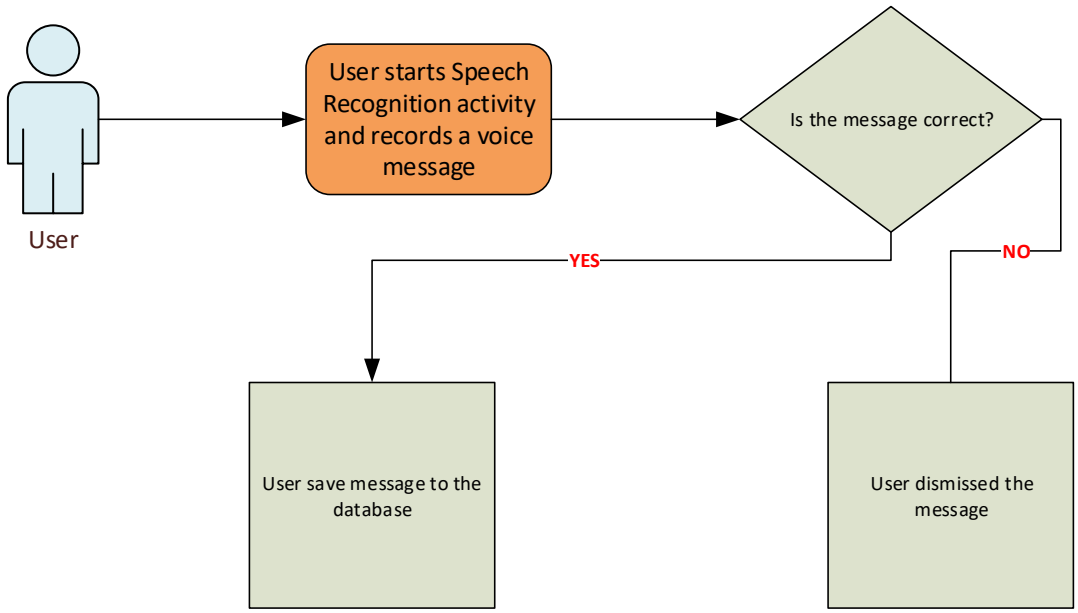
This use case represents a vital piece of the system functionality, because it makes the application unique by allowing the user to create text messages without typing.

3.1.4.2 Use Case

Scope

The purpose of this use case is to display the process of how the user creates a voice message and saves it.

Use Case Flow Diagram



Precondition

The user must start Speech Recognition activity.

Activation

This use case starts once a user started the Speech Recognition activity.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1. User start Speech Recognition activity 2. User leaves a voice message 3. User saves message	1. Application starts Speech Recognition activity 2. Application takes the message 3. Message is saved in into database as text

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User starts the Speech Recognition activity 2. User leaves a voice message 3. User does not save his message and does not make a new attempt 4. User Stop his activity	1. Application starts Speech Recognition activity 2. Application takes the message 3. Application does not save message and does not create a new one 4. Application stops Speech Recognition activity

Termination
Once the user saves or dismisses the message successfully the flow is finished

Post condition
The user voice message is saved into database as a text message and displayed on the app message list and google map.

3.1.5 Requirement 5 <Points and Position on the Leaderboard>

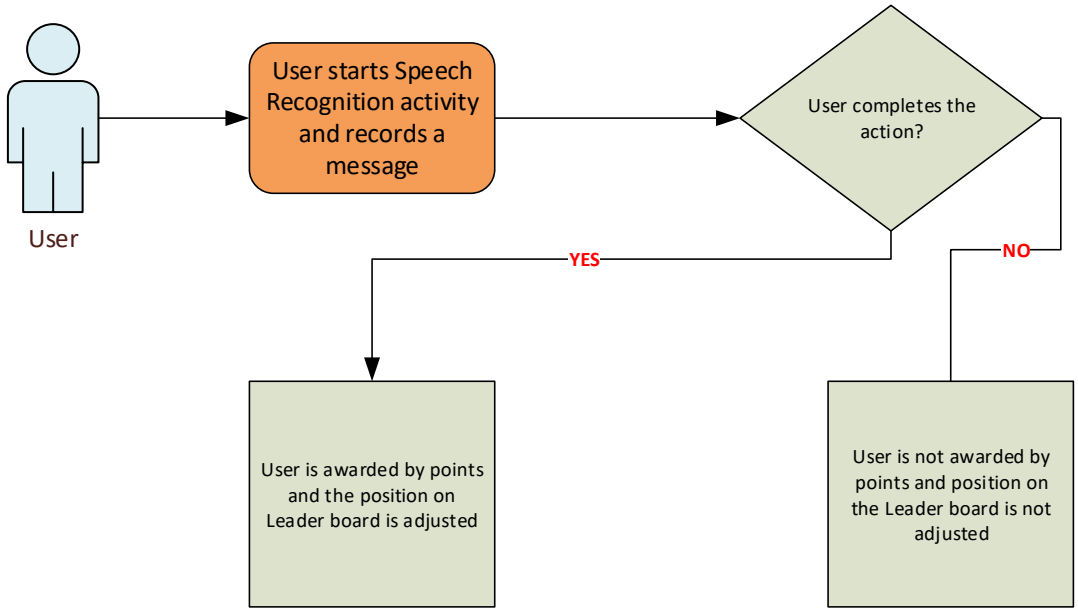
3.1.5.1 Description & Priority

This use case represents the “gamming” aspect of the system functionality, because it allows the user to gain points and to increase the position on the leaderboard after contributing with messages to the app.

3.1.5.2 Use Case

Scope
The purpose of this use case is to display the process how the user can get points and a better position on leaderboard.

Use Case Flow Diagram



Precondition

The system is in the wait state and the user must start and complete the action.

Activation

This use case starts once a user saved the message created on the Speech Recognition activity.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1.User is on the started Speech Recognition activity 2.User leaves a voice message 3.User saves the message	1.Application starts Speech Recognition activity 2.Application takes the message 3.Message is saved into database as text 4.Application adds one point to the user and increases his position on leaderboard

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User is on the started Speech Recognition activity 2. User leaves a voice message 3. User stops his activity during the process 4. User does not save his message and does not make a new attempt	1. Application starts Speech Recognition activity 2. Application temporarily takes the message 3. Application stops Speech Recognition activity 4. Application does not save message and does not create a new one

Termination
Once the user saves or dismiss the message successfully the flow is finished

Post condition
The user message is saved into database and the user has achieved a point. The position on the leaderboard has increased after his contribution with a message to the app

3.1.6 Requirement 6 <User Profile>

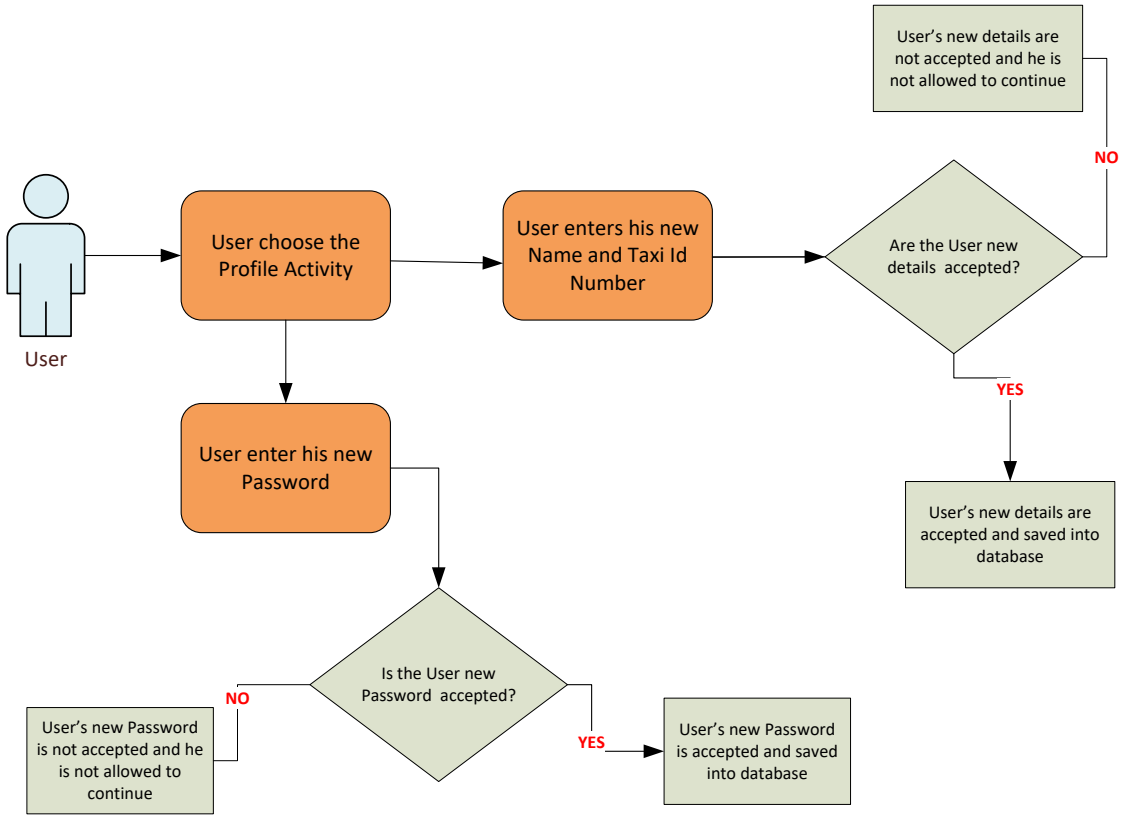
3.1.6.1 Description & Priority

This use case represents a very important part of the system functionality, because it allows the user to change or update his Name, Taxi Id Number and the Password.

3.1.6.2 Use Case

Scope
The purpose of this use case is to display the process how the user can change or update the credentials on his Profile.

Use Case Flow Diagram



Precondition

The user must start User Profile activity.

Activation

This use case starts once a user started the User Profile activity.

Main flow

Flow Description process	
Actors action	**Reply of the System**
1. User starts the Profile activity 2. User fills the new details for the Profile 3. User updates/saves the details	1. Application starts User Profile activity 2. Application accepts the new details 3. Application saves the details into database

Alternate flow

Alternative flow description process	
Actors action	**Reply of the System**
1. User starts Profile activity 2. User fills invalid or already used details 3. User is not allowed to continue 4. User Stops his activity	1. Application starts User Profile activity 2. Application does not accept the user's new details 3. Application does not allow the user to continue 4. Application stops User Profile activity

Termination

Once the user saves or does not save his new details successfully the flow is finished

Post condition

The user's new details are saved into database and the user can log out and then log back into the app with his new credentials

3.2 Non-Functional Requirements

3.2.1 Performance/Response time requirement

As this application is about real-time traffic information the app should launch in no more than 5 seconds and should always be able to display updated information with as little error as possible.

3.2.2 Availability requirement

The application will be available online 24/7 but will still have minimal functionality while the device is offline.

3.2.3 Recover requirement

If the system fails or becomes unresponsive the users should be notified about the issue. The system should be available back to the users as soon as possible and the users should not be left without using the application for more than 5 to 7 hours.

3.2.4 Robustness requirement

A failure to connect to the database will not cause the system to terminate. It will display an error message stating the error to the user.

3.2.5 Security requirement

A secure login and user profile will be encrypted and stored on the database with all the user details safe from unwanted intrusion.

3.2.6 Reliability requirement

All user interaction will not conflict with the application and each function will be error free from the expected range of user inputs. The data that is being transmitted to the database will not conflict with existing stored data.

3.2.7 Maintainability requirement

Each component of the system can be repaired or updated without the rest of the system being affected. If an error presents itself in the application, it will be easily rectified by correcting the specific component that needs attention. i.e. if there is a problem with the database connection the user will still be able to use the app.

3.2.8 Portability requirement

The app shall open its activities in reasonable time and be portable to iOS and Android mobile devices.

3.2.9 Extendibility requirement

The app will have the opportunity to extend new features to the finished product, without any compatibility issues.

3.2.10 Resource utilization requirement

The app should execute an action or command in no more than 1 second.
The updated information of a user should be carried out in the background with no strain on the performance of the front-end GUI.

3.3 Data Requirements

On the next section of this document is the explanation of the data requirements which are vital when it comes to the implementation of the key functions functionalities mentioned earlier in this report.

- **Google Firebase:** All the data of this application for example, traffic information records, user profile information, etc., will be stored in a non-SQL database.
- **The Database System:** will be linked with the application using the API keys and dependencies provided by Firebase documentation.
- To create a user profile on this application, user email and user name is a requirement and will be stored into the database
- **User Geolocation:** user's geolocation will be captured by the application to be able to track exactly the address of any incident reported by the users
- **User Activity Records:** application will be saving and storing all the user's messages.

3.4 User Expectations

After some discussions with a taxi business owner and some taxi drivers, they were delighted with the idea and the functionalities that this app can have and predicted that it would have a real benefit to all taxi drivers. They also mention what their expectation is from that app and the main points are listed below:

- The app shall be portable to iOS and Android mobile devices and to open its activities for the users in reasonable time.
- The app should have a speech recognition functionality which allow users to create voice messages which then are transformed into text messages and displayed for the users.
- The app should have Google maps on it, where users can view the messages about what happened on different roads in Dublin.
- The app should be simple and user friendly which has and displays real-time information only about road traffic in Dublin.
- The app should display fresh information for users about anything related to road traffic in Dublin, for example; Accidents, Live Checkpoints, Busy Roads, Taxi Ranks etc.
- The app should be fun also and keep track of user's messages activity, and to update their position on the leader-board.

3.5 Environmental Requirements

In the next section, there is a description about the environmental requirements. Before starting an application, it is very important to know exactly what is required for building the application. These are the most important requirements that are needed to be able to develop the application.

- **Windows Machine:** For now, this application will be developed on a Windows laptop machine with Ionic framework SDK installed.
- **MacBook**: Later the application will be transferred on a MacBook because now windows machines do not support iOS emulator.
- **Android Device:** An android emulator and device is needed to run and test the application during the development.
- **iOS Device:** An iOS emulator and device is needed to run and test the application during the development.
- **Photoshop:** Photoshop has been used to resize the logo image for the application.
- **Internet Access:** Internet access is needed to connect to the database or to test some specific functions within the application

3.6 Usability Requirements

In this section is a brief explanation of the usability requirements. During the design and development process or GUI process these requirements are delivering some objectives that, should be taken in consideration. The usability requirements were discussed earlier because they are based on the other main requirements.

- **Simple to Manage:** The elements within GUI must be clear and straight forward to allow the user to manage the app with no difficulties.
- **Display the Errors:** In case of internet coverage matters, or user incorrect details entered the app should display errors.
- **Easy to Use:** The application should be easy to use, in a short reasonable time. The user interface, shall direct the users to the activities they are looking for, in an easy way, to ensure that the users get to these activities which they are looking for.

- **Nice and smart look:** The application should look smart and nice to attract users and to catch people's attention. It should have the look of a modern and up-to-date app.

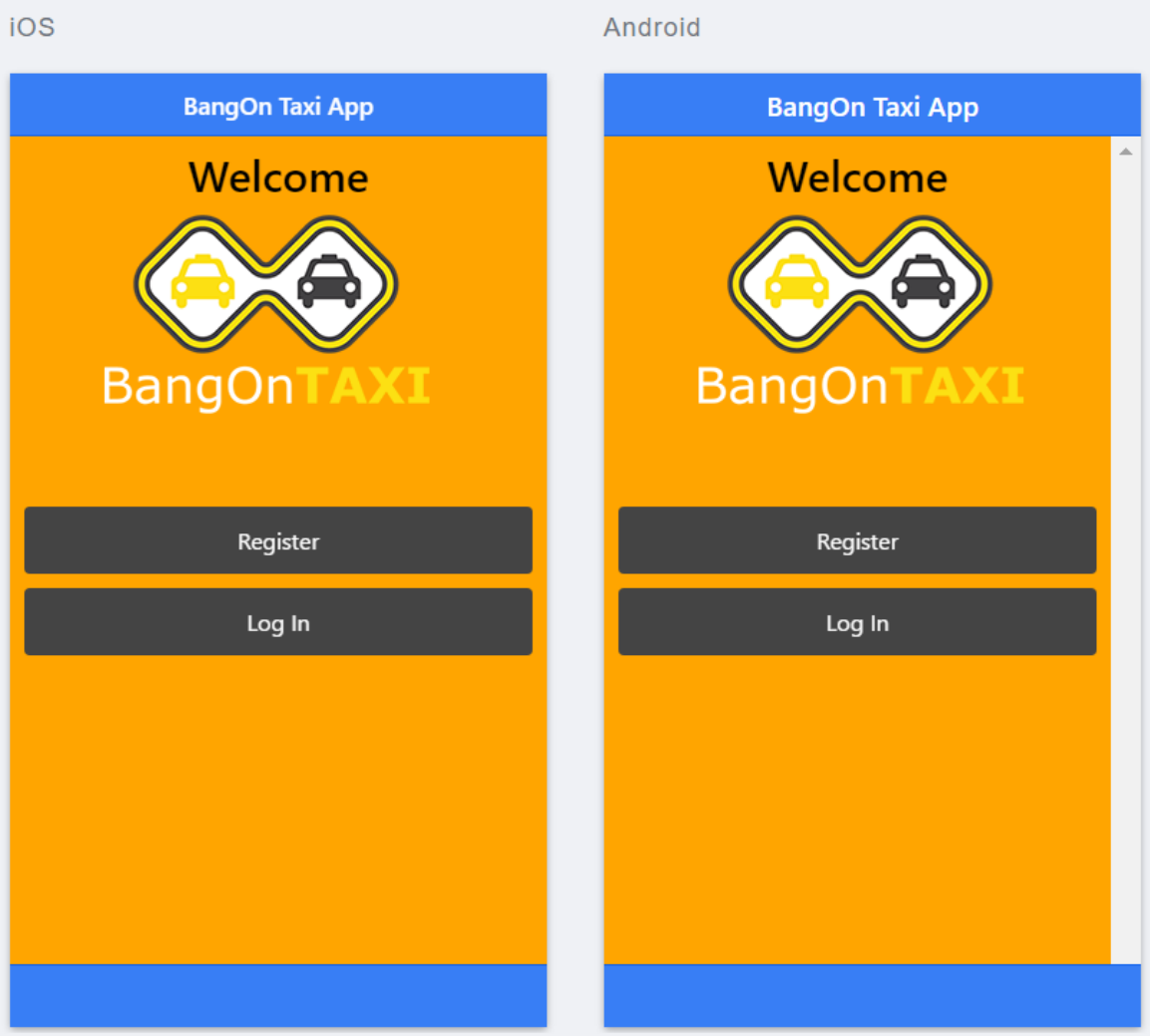
4 Design and Architecture

The system architecture chosen to build the application is a data centred architecture, which separates all main components that allow better flexibility and maintainability of the app. The application will be hosted locally on mobile devices running on iOS and Android Operating Systems. All the user details will be held within a central NoSQL Firebase database. Also, the points of user's activities will be stored into the database. The application will also use Google Map API as a API Services.

The most important architectural goal of the Android and iOS app's is to make sure that, the application does not consume an excessive amount of memory. This is vital because the users must have enough space on their devices to be able to download and install the app. As all other app's this application will be permanently hosted and run on the user's Android and iOS Operating Systems.

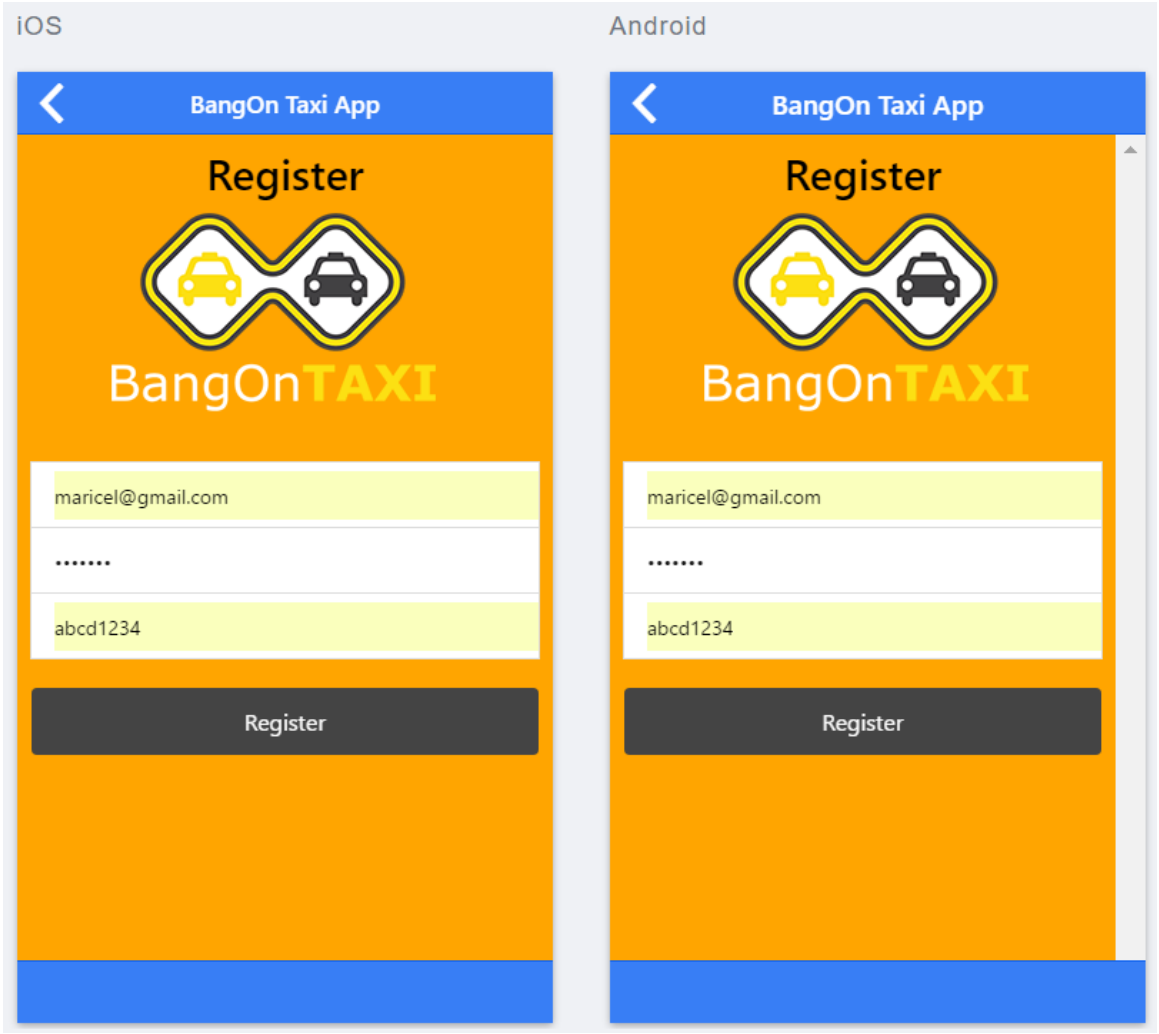
4.1 G.U.I. Graphical User Interface

4.1.1 Welcome Activity



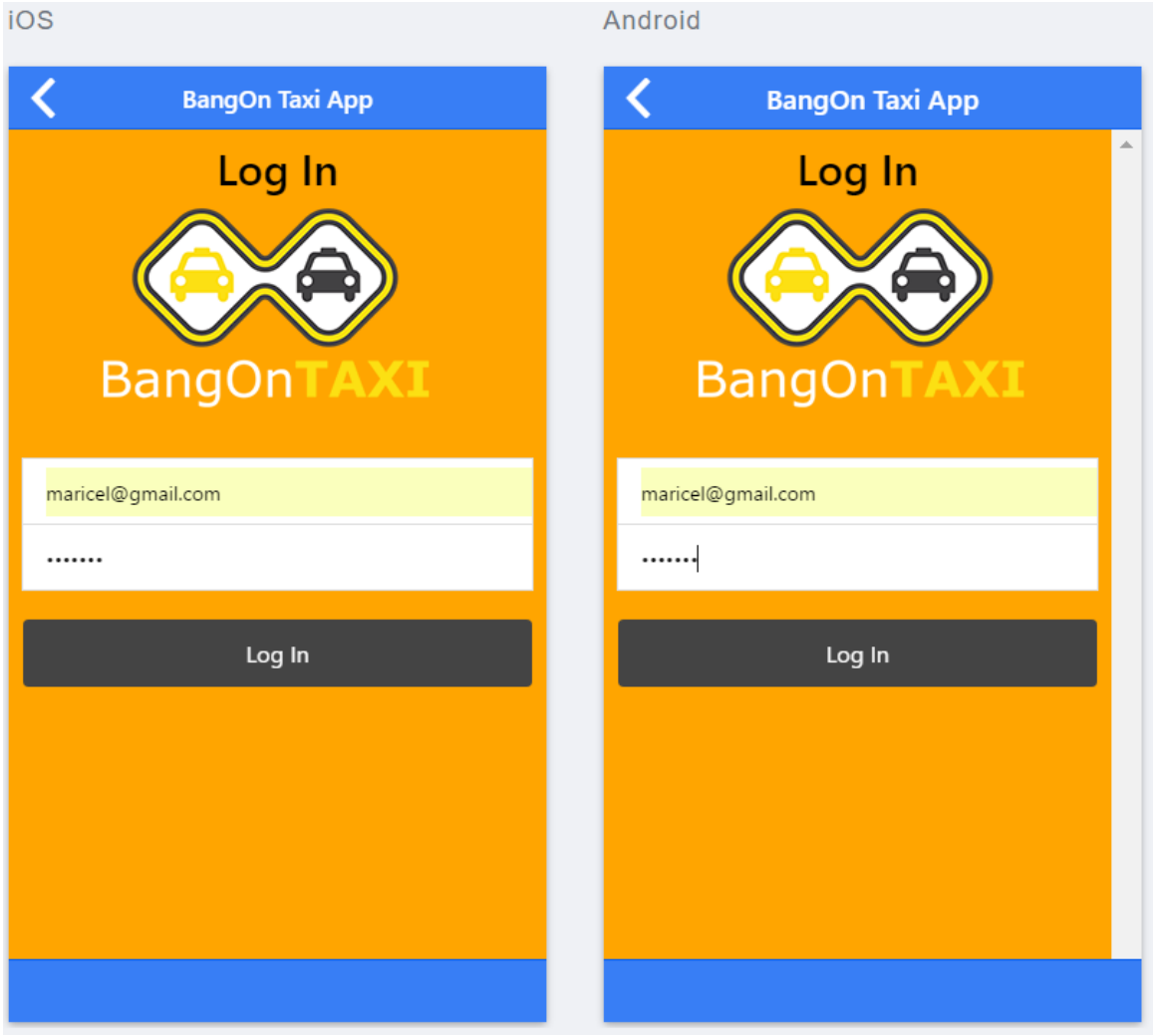
The above image is a screen shot of the Welcome activity opened in both mobile devices iOS and Android, where the user will be redirected when launching the application for the first time. In the header section, it has the name of the Application. On the body section on the top is the message Welcome followed bellow by the logo and the name of the application. Then further down the user has two options, either to “Register” or to “Log in”. By clicking “Register”, the user will be brought into Register activity to fill his details and get registered. By clicking “Log in” the user will be brought into the Log in activity just to enter his user email and password for accessing the app.

4.1.2 Register Activity



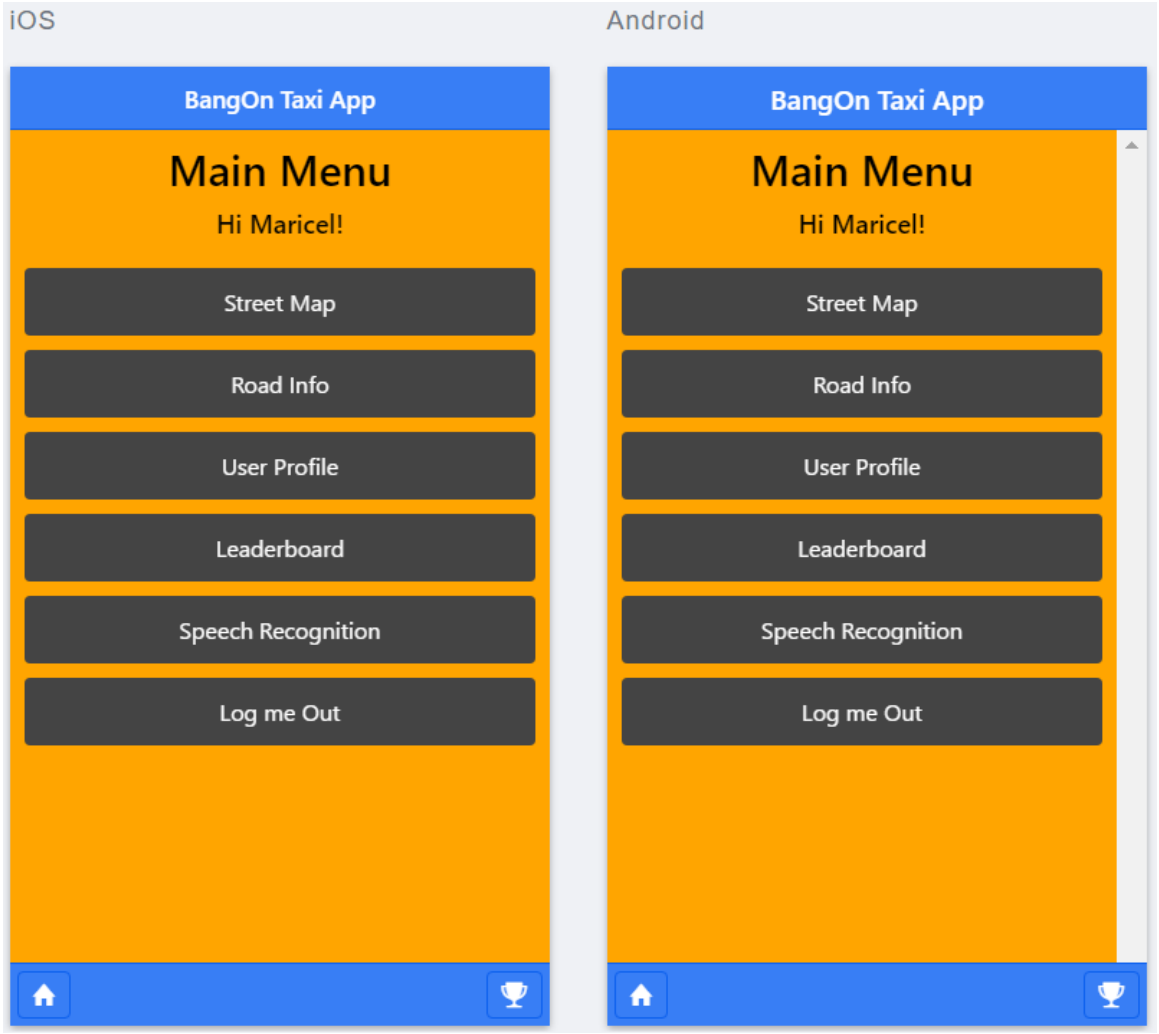
The above image is a screen shot of the Register activity opened in both mobile devices iOS and Android. In here the new user is redirected from the previous Welcome activity to fill all the required fields to create an account. After valid information is inserted in these fields there is only one button called “Register” to be clicked and the new user has created an account. After the new user, has successfully signed up he will be redirected to the “Main Menu” activity to start using the app and its full content.

4.1.3 Login Activity



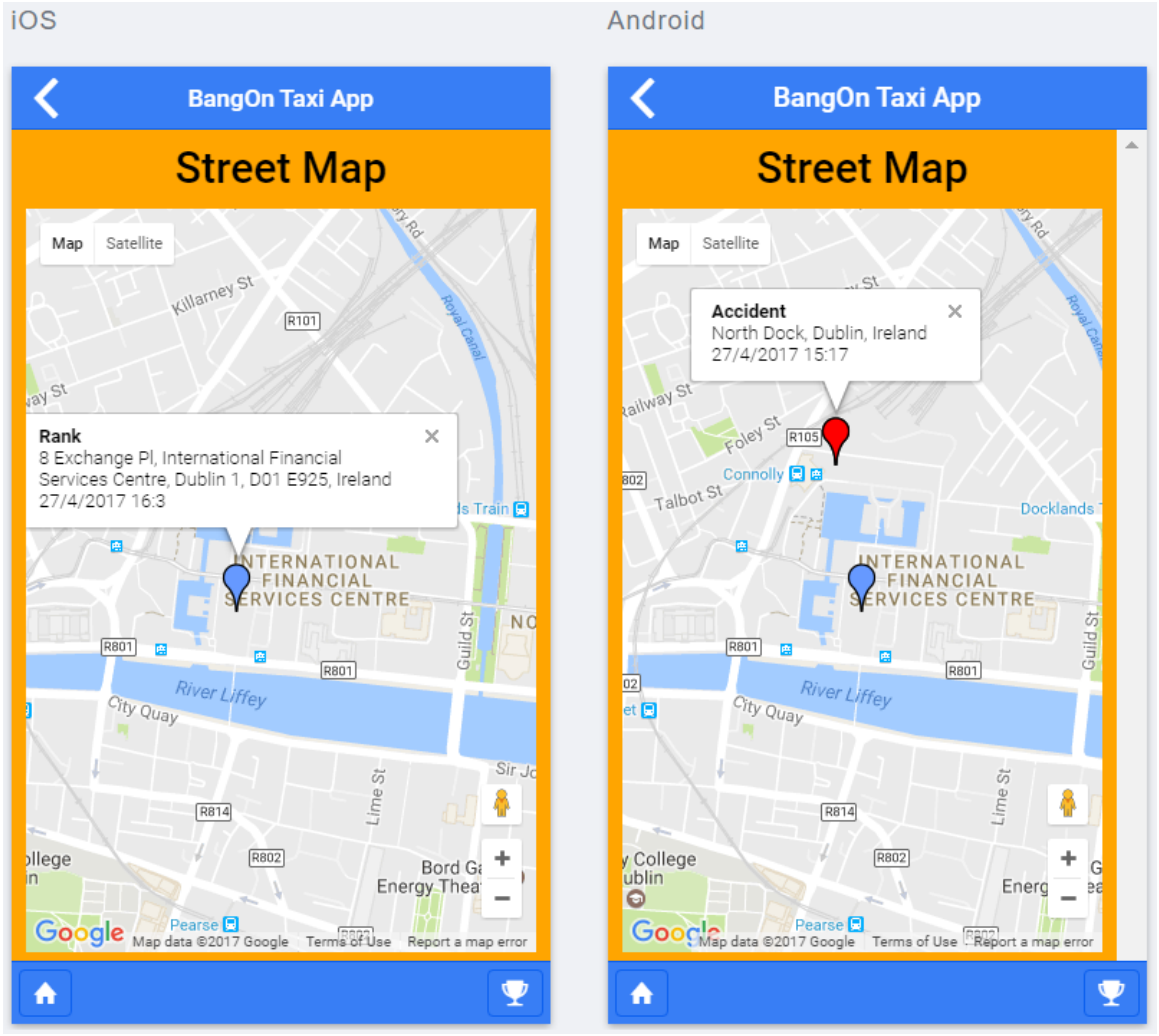
The above image is a screen shot of the Log in activity opened in both mobile devices iOS and Android. In here the user is redirected from the previous Welcome activity to fill all the required fields to login into his account. After valid information is inserted in these fields there is only one button called “Log in” to be clicked and the user has full access to the app. After the user is successfully logged in he will be redirected to the Main Menu activity to start using the app and its full content.

4.1.4 Main Menu Activity



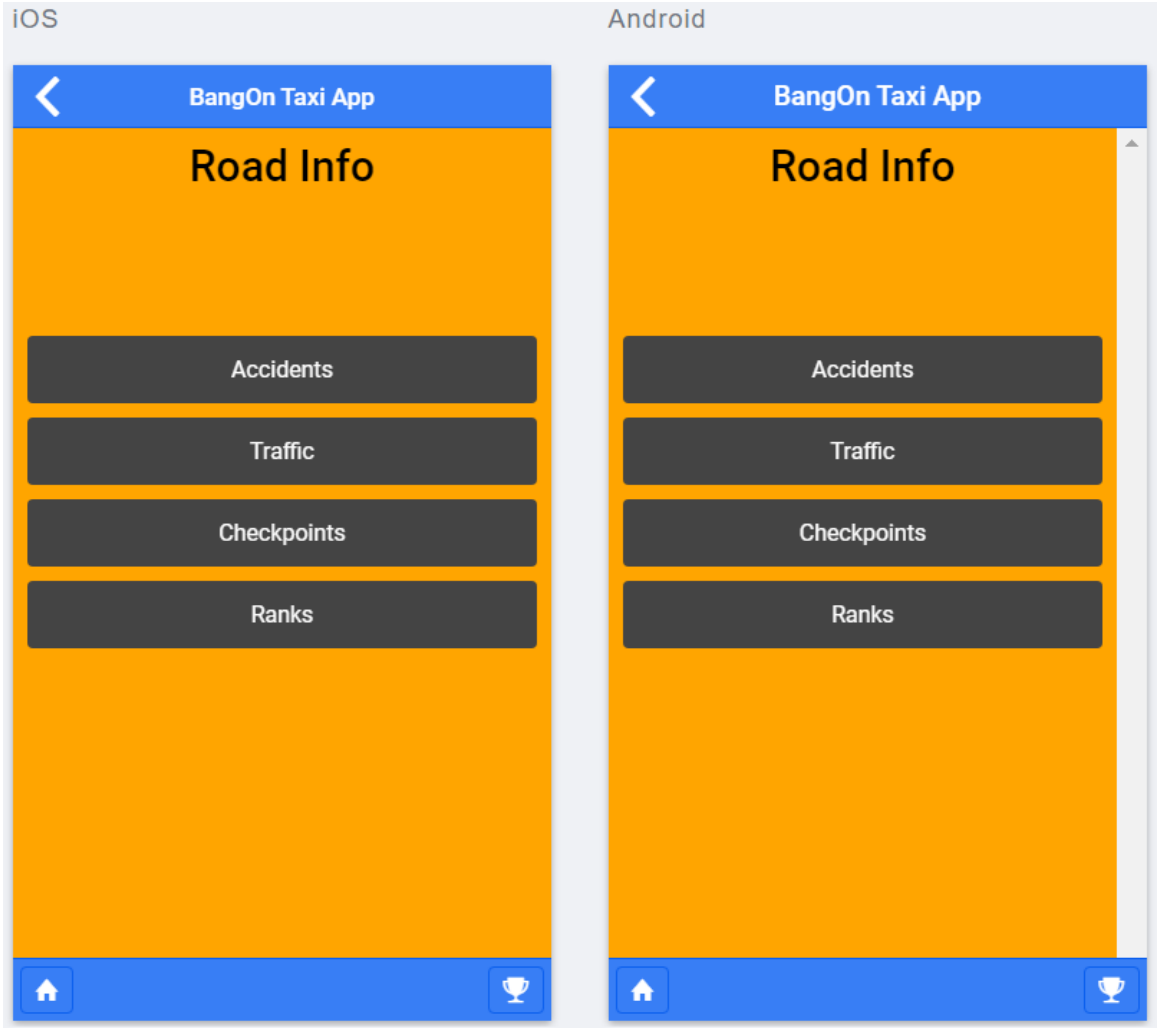
The above image is a screen shot of the Main Menu activity opened in both mobile devices iOS and Android. The user is redirected in here after the Register or Log in process and has a variety of new options where to go and choose from. After a user is clicking one of the options the required activity will open and then user can go even further in there to search for information and to take actions. User can logout straight away by clicking the bottom “Log Out” button which is located at the bottom.

4.1.5 Street Map Activity



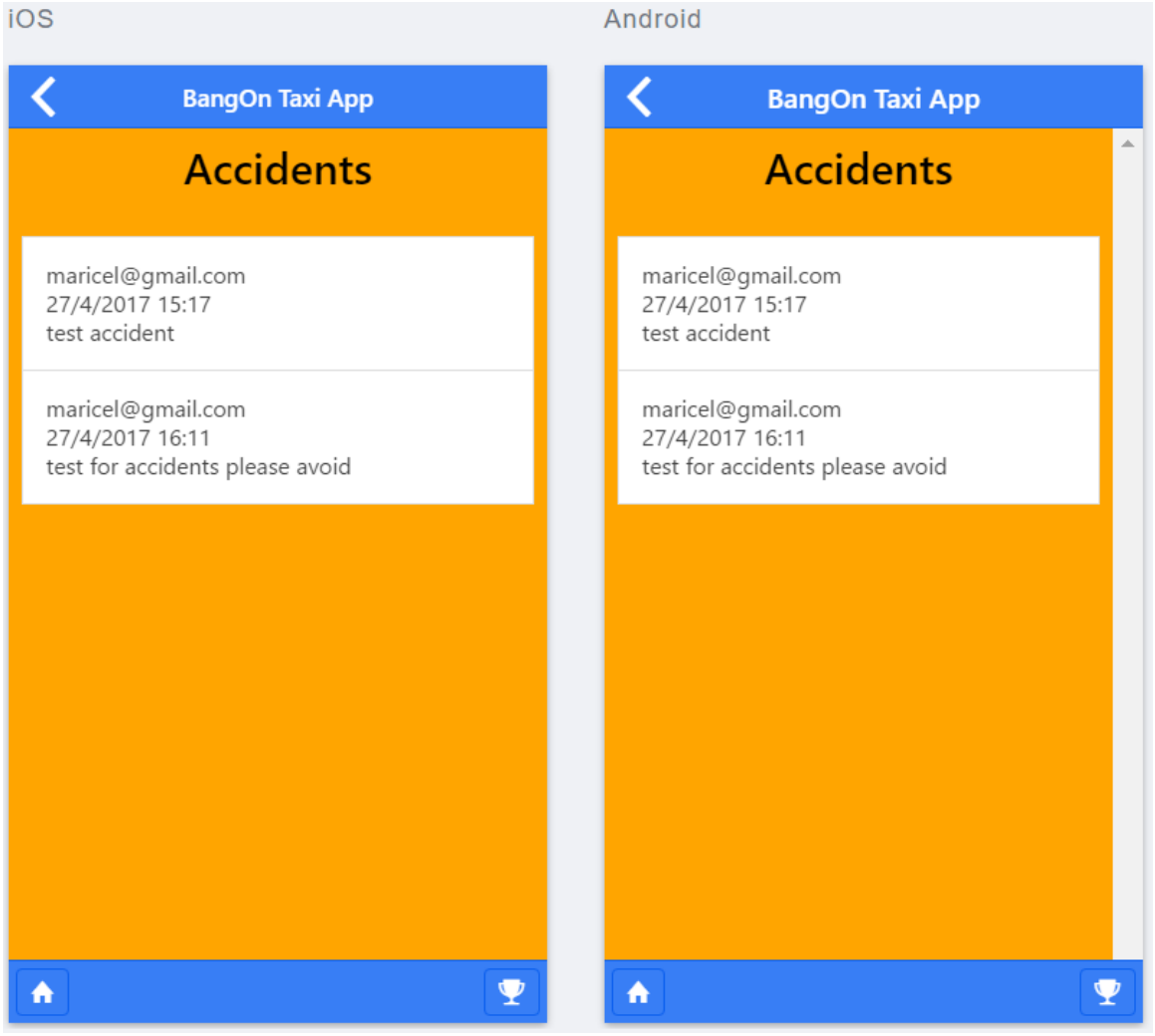
The above image is a screen shot of the Street Mapp activity opened in both mobile devices iOS and Android. The user is redirected in here from Main Menu activity and has the google map ready with different color pins. Each pin contains information such as, type of incident, location/address, date and time. User can click on any of them to find out exactly what is happening on Dublin streets. From this activity, the user, can go back to previous activity by clicking on the top left hand side arrow. Also, user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at the bottom in the footer section.

4.1.6 Road Info Activity



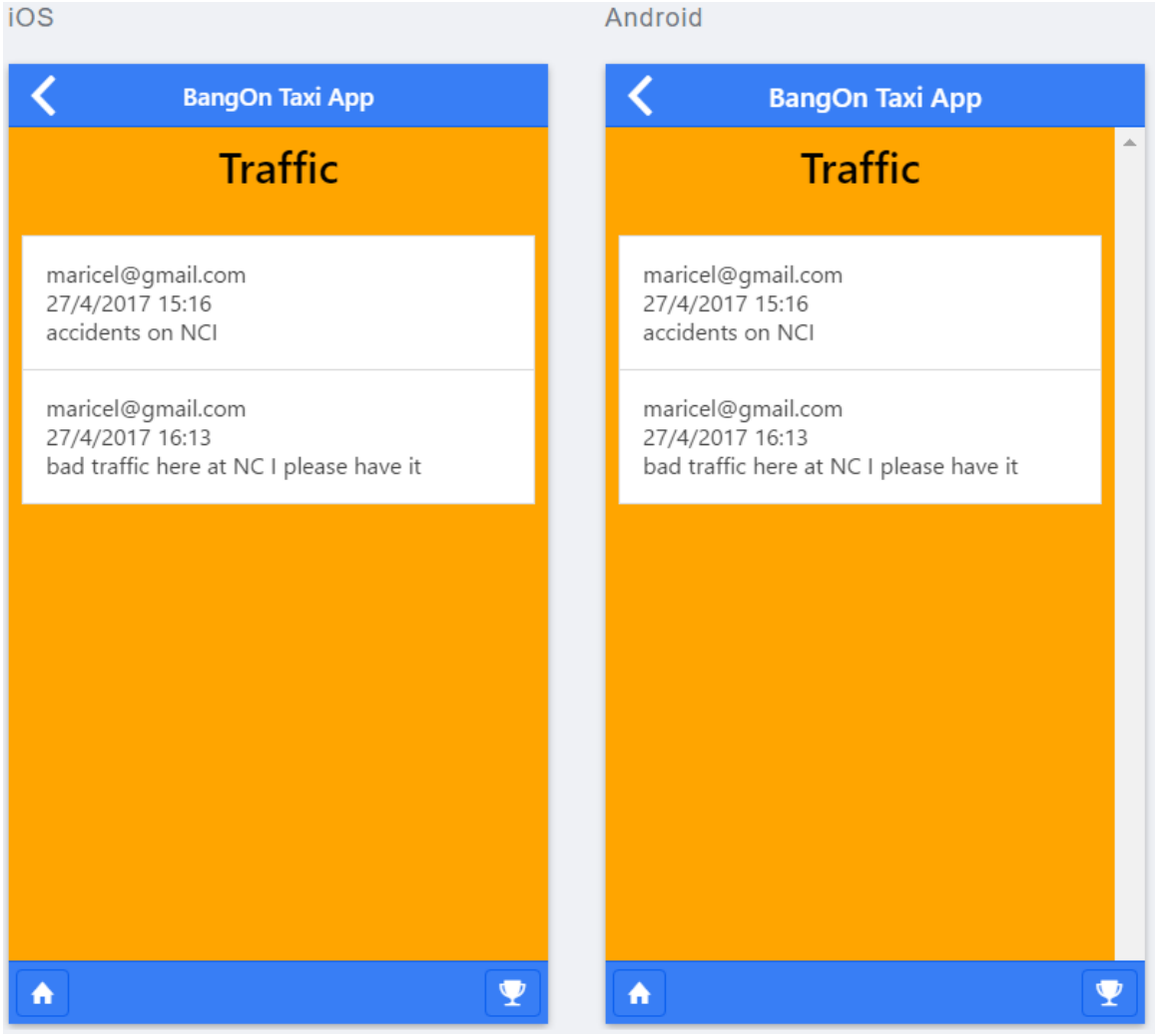
The above image is a screen shot of the Road Info activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Main Menu has multiple road information options to choose from. After a user is clicking on one of the options the required activity will open and the user can view information related to that specific topic. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at the bottom in the footer section.

4.1.7 Accidents Activity



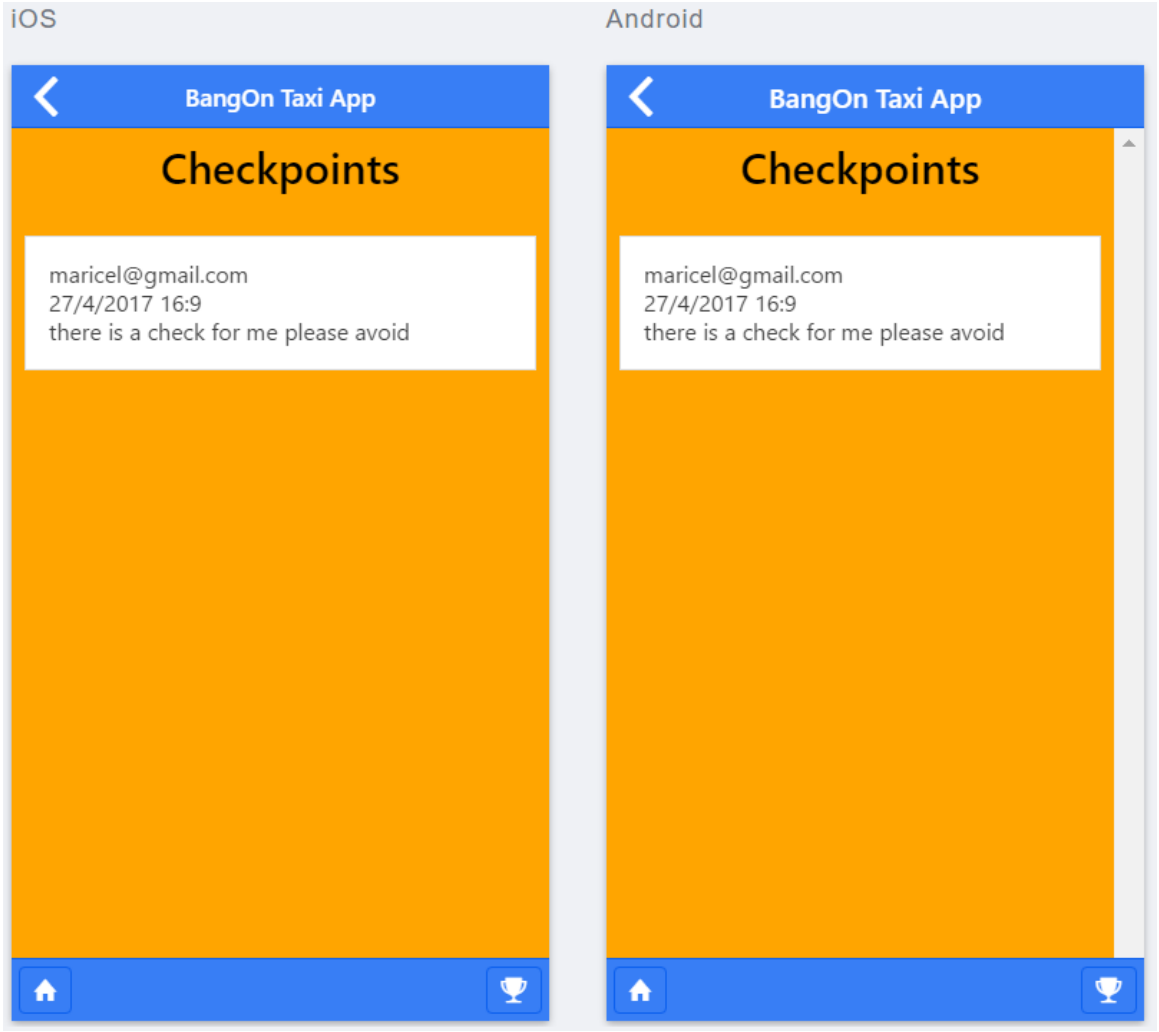
The above image is a screen shot of the Accidents activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Road Info activity has a list of accidents to view. Each message has the user name of the person who created that message, has the date and time, and some specific content. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.8 Traffic Activity



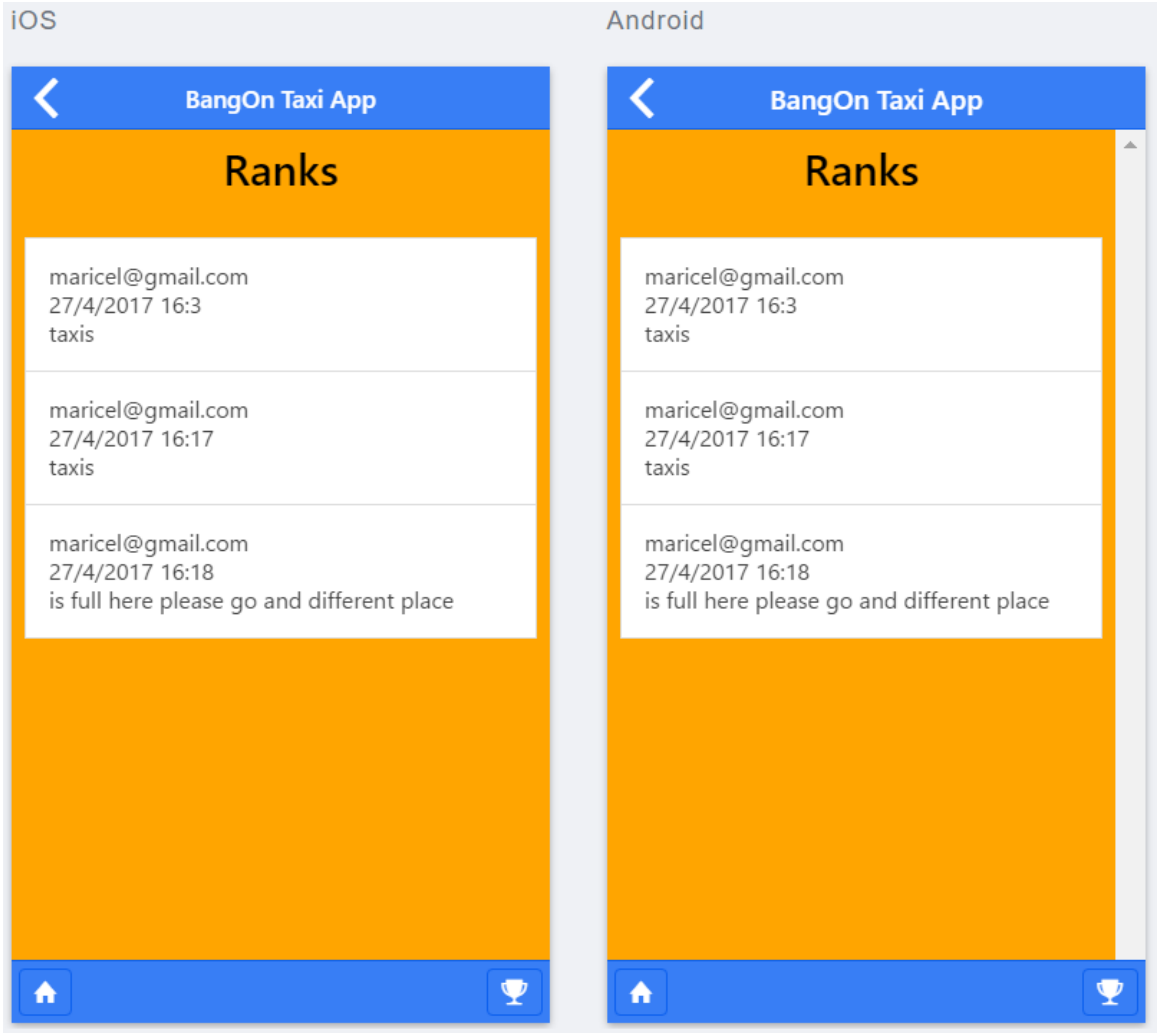
The above image is a screen shot of the Traffic activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Road Info activity has a list of traffic issues to view. Each message has the user name of the person who created that message, has the date and time, and some specific content. From this activity user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.9 Checkpoints Activity



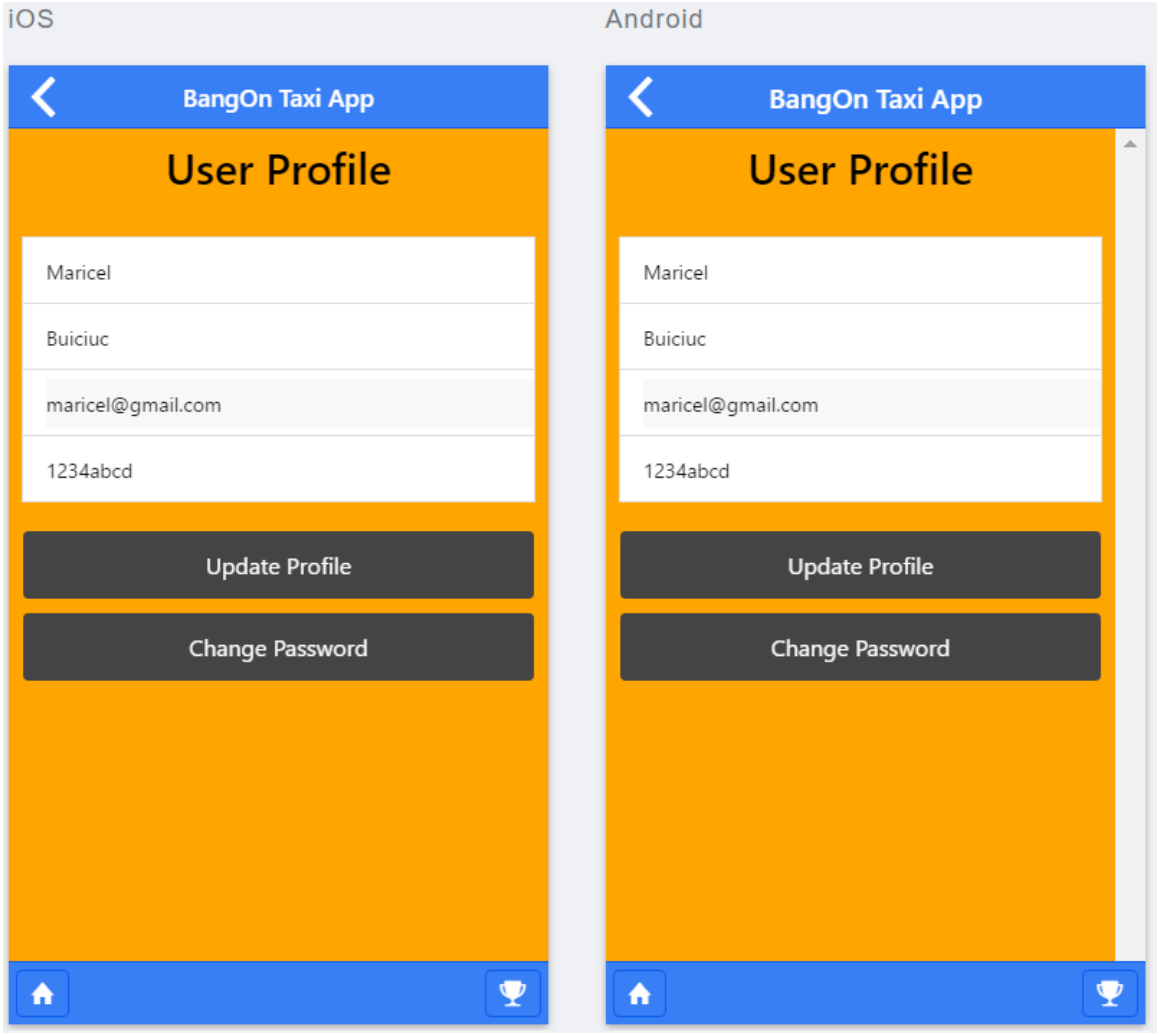
The above image is a screen shot of the Checkpoints activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Road Info activity has a list of Checkpoints to view. Each message has the user name of the person who created that message, has the date and time, and some specific content. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.10 Ranks Activity



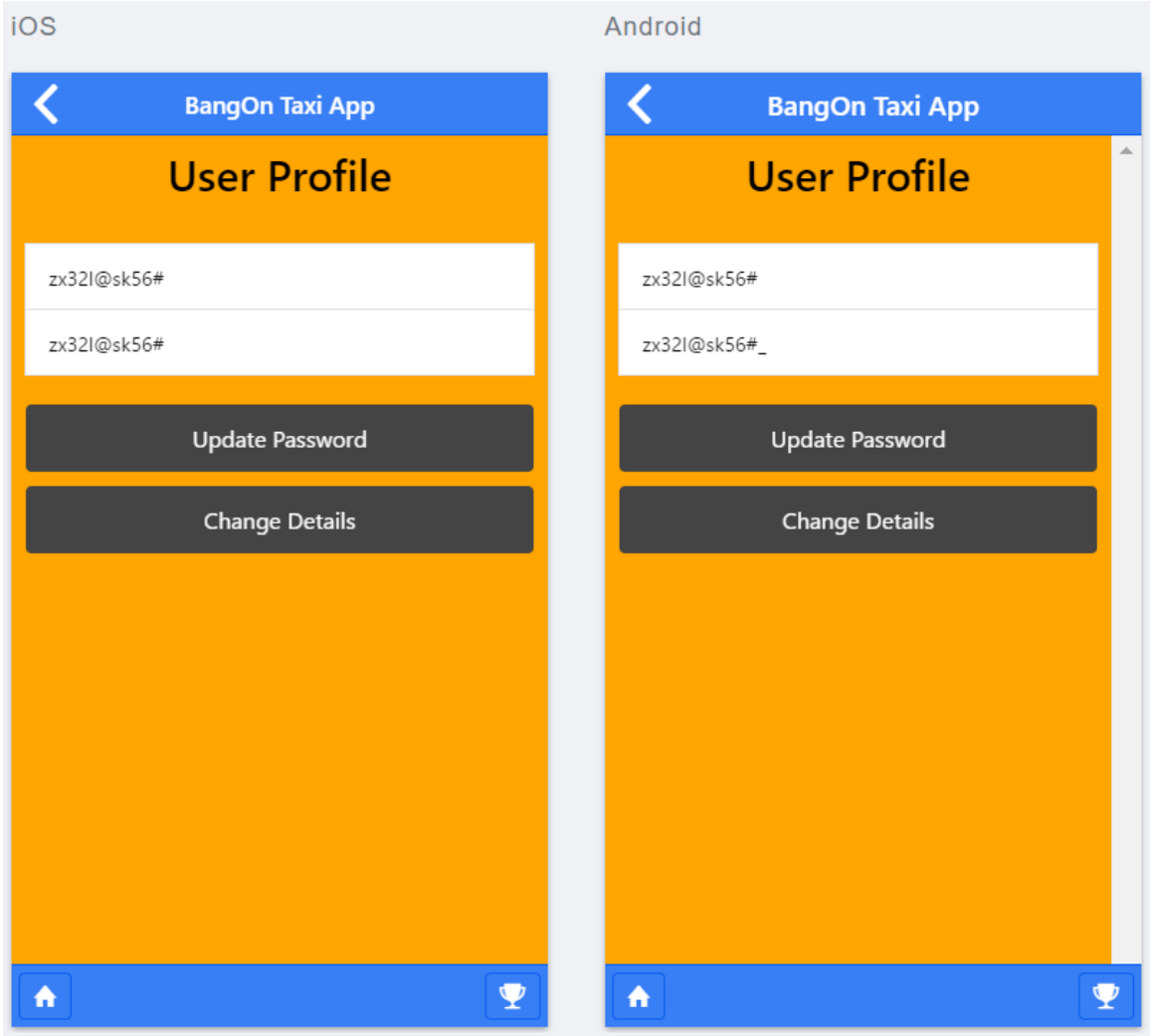
The above image is a screen shot of the Ranks activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Road Info activity has a list of Ranks to view. Each message has the name of the person who created that message, has the date and time, and some specific content. From this activity, the user, can go back to the previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.11 User Profile Activity (Update Profile)



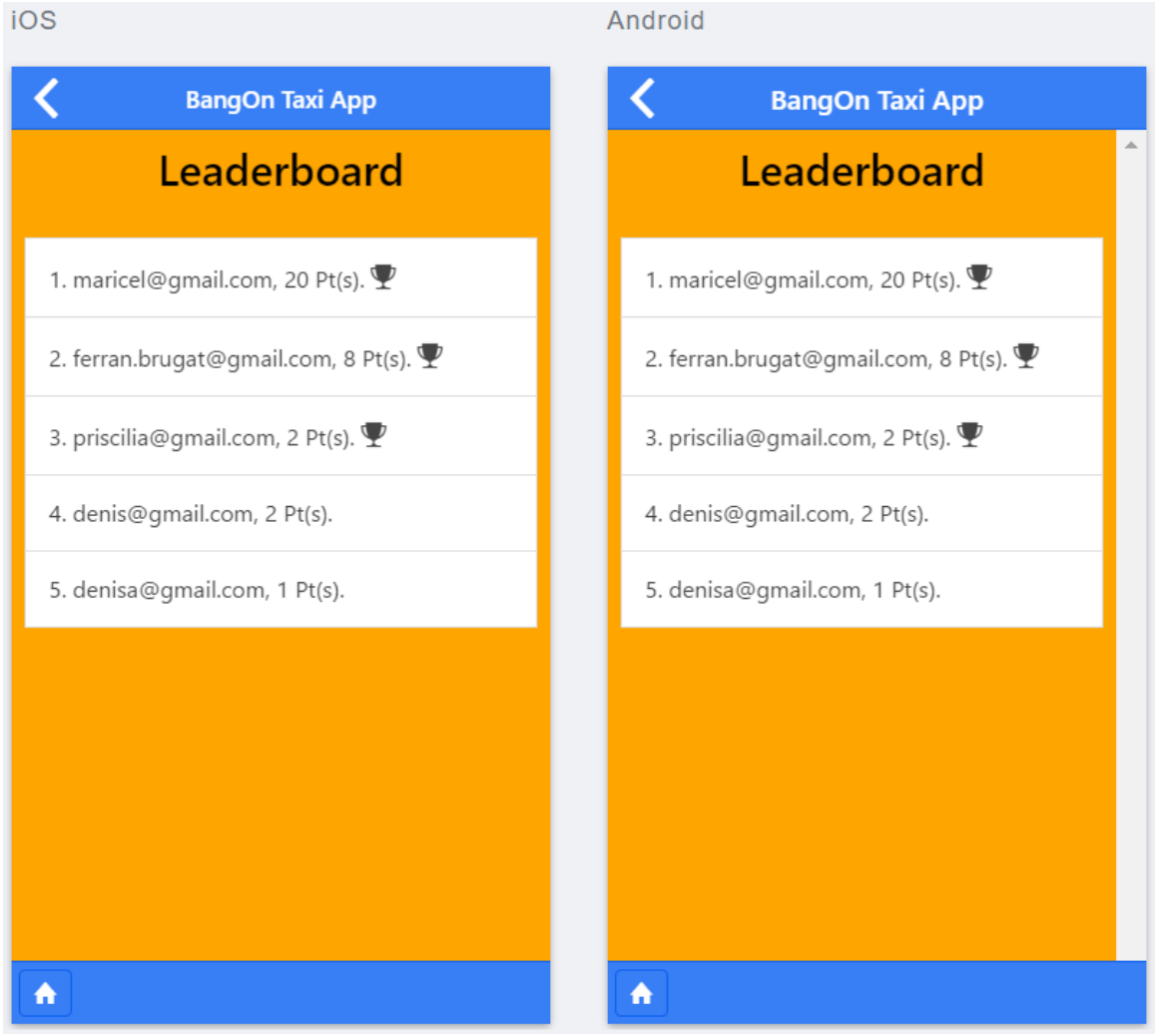
The above image is a screen shot of the User Profile (Update Profile) activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Main Menu activity and has two options to choose from, “Update Profile” or “Update Password”. After a user is clicking on one of the options in this case “Update Profile”, he can enter the new details and click save; after that the new details are saved into the database. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.12 User Profile Activity (Update Password)



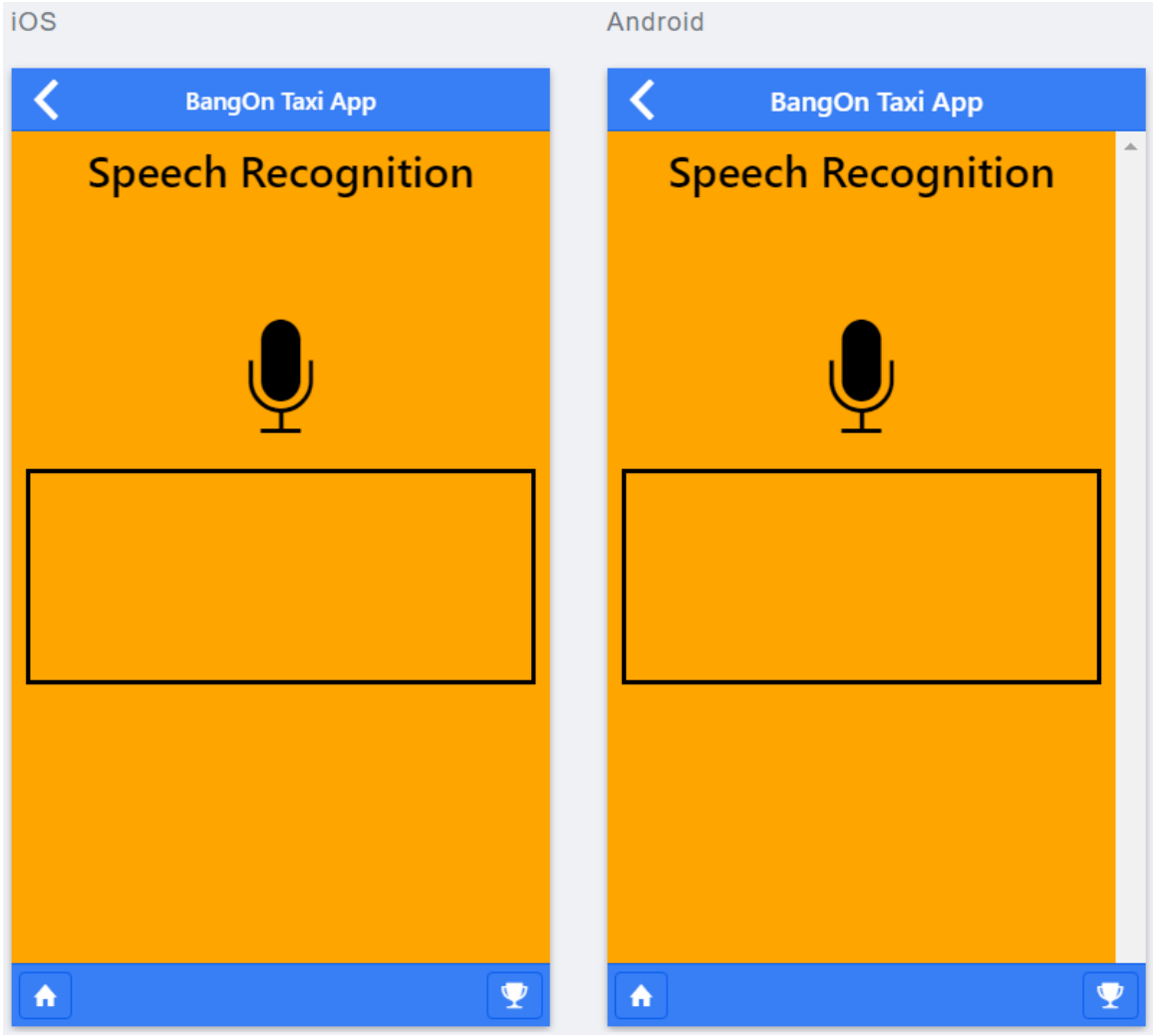
The above image is a screen shot of the User Profile (Update Password) activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Main Menu activity has two options to choose from, “Update Profile” or “Update Password”. After a user is clicking on one of the options in this case Update Password, he can enter the new Password and click save. After the new Password is saved into the database, the user can log out and then log in with its new password. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

4.1.13 Leaderboard Activity



The above image is a screen shot of the Leaderboard activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Main Menu activity has a list of users to view. Trophies are awarded to the first three most active users who always create messages and they will be on top of the list, followed by the less active users. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) by clicking on the home icon/button provided at bottom in the footer section.

4.1.14 Speech Recognition Activity



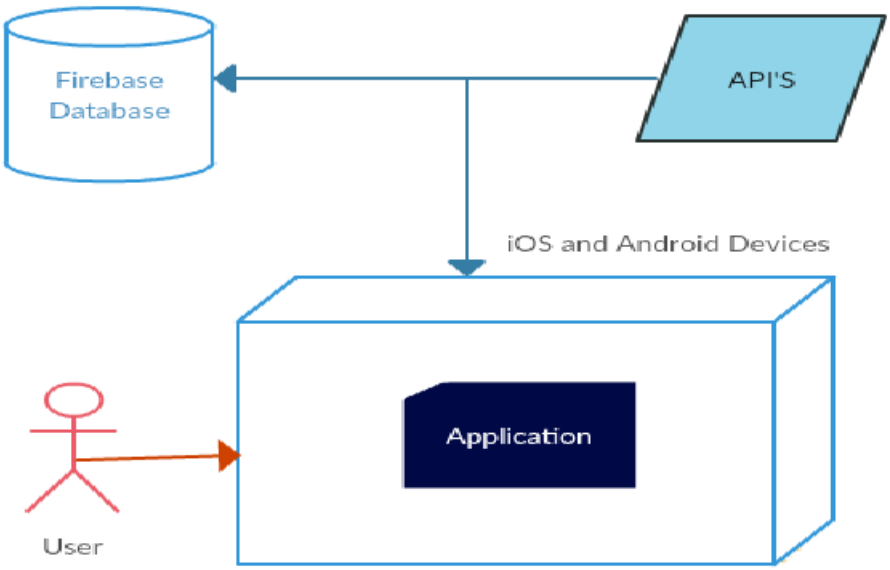
The above image is a screen shot of the Speech Recognition activity opened in both mobile devices iOS and Android. The user who is redirected in here from the Main Menu activity can create a voice message. The voice message will be then transformed into a text message and saved into database. At the same time this will be available to all app users on the view messages sections but also on the Google Map. From this activity, the user, can go back to previous activity by clicking the top left hand side arrow. Also, the user can go home (Main Menu) or to the leaderboard by clicking on any of the two specific icons/buttons provided at bottom in the footer section.

5 System Architecture

The following three architecture diagrams are meant to explain the application's components and their activities.

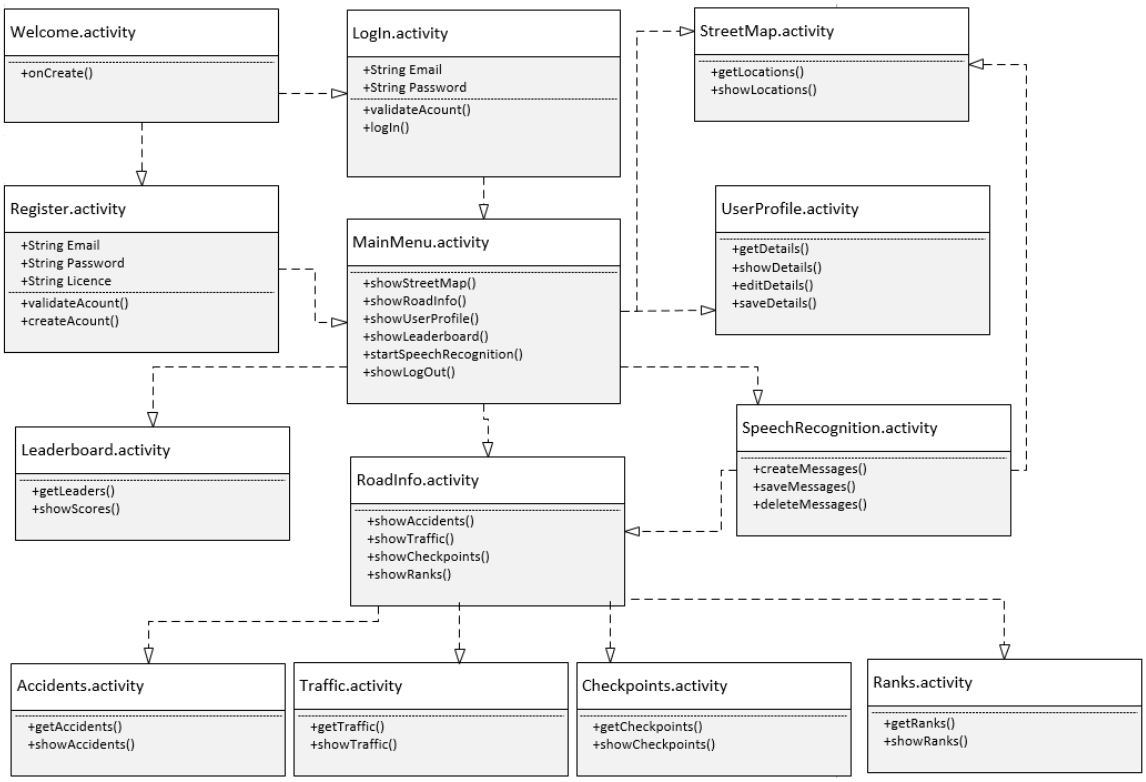
5.1 System Architecture Diagram

The System Architecture Diagram shows that the application will be kept locally on mobile devices running on iOS and Android Operating Systems. All the user details will be held within a central NoSQL Firebase database. Also, the points of user's activities will be stored into the database. The application will also use Google Maps API as a API Services to get the Geolocation.



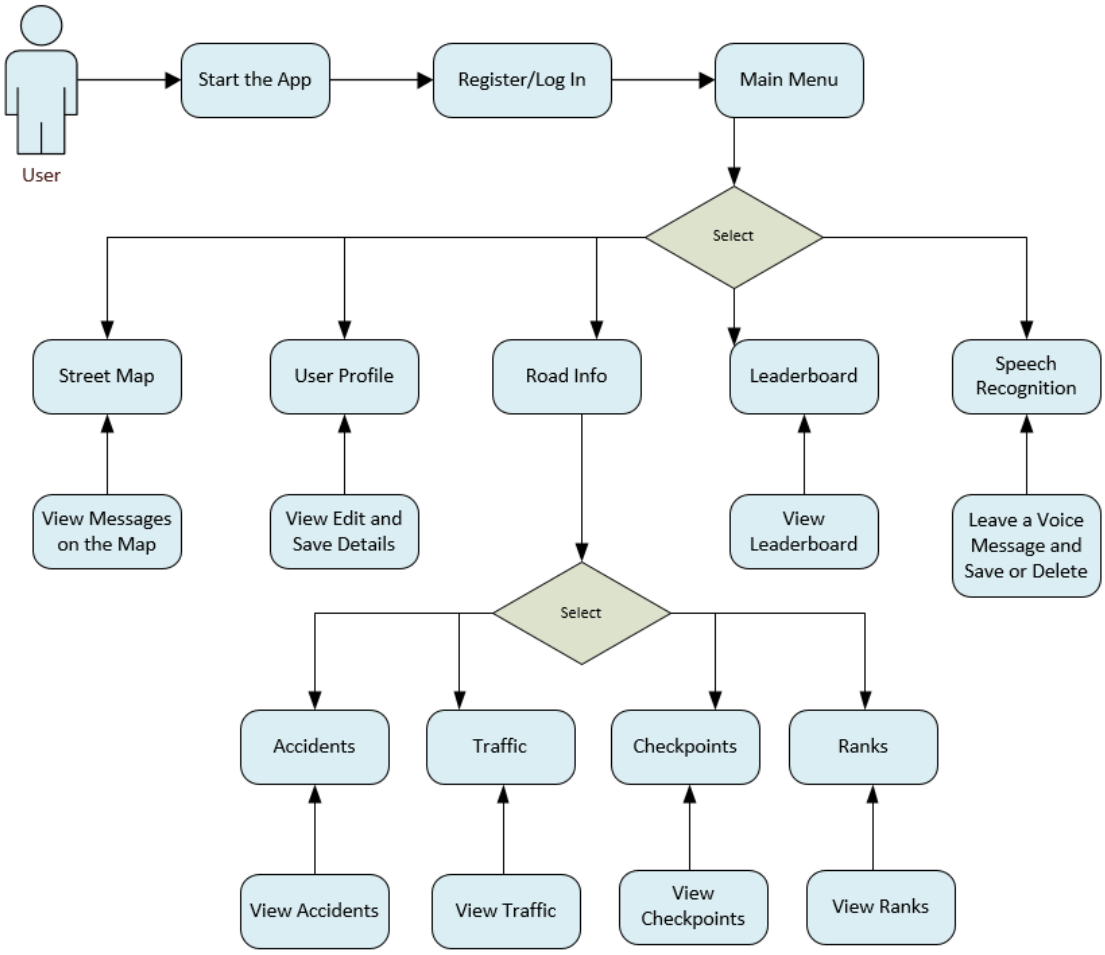
5.2 Class Diagram

The next image shows the class diagram with many classes and demonstrate how the different parts of the BangOnTaxiApp application works between each class. This also, demonstrates high-level interaction among different classes/activities.



5.3 Flow Diagram

The flow diagram shows the basic flow of the user interaction with the application and how the application will respond for different user actions.



6 Analysis and Design

6.1 Purpose of the Product Design Specification

This part of report describes and records the system design and architecture decisions to deliver guidance through the development cycle. This is to aim the

developer of the BangOnTaxiApp system and has been created through Planning Phase.

7 Overview and Design Guidelines/Approach

This part of the report defines the needed strategies and principles, which must be follow when the system was designed and implemented for the BangOnTaxiApp.

7.1 Assumptions / Constrains / Standards

To ensure that everything works fully and as it should be the BangOnTaxiApp application must have an available Internet connection always as all data is saved to Firebase database which is on Cloud.

Another constraint what can appear when developing the application could be out-of-date online coding resources, this makes work harder as most examples on the web are not matching the needs of this app. One big constraint could be, the fact that this framework and the scripting language are totally new and as a developer of this application one must learn nearly everything from scratch.

8 Architecture Design

8.1 Logical View

The BangOnTaxiApp application's logical view will be divided into three major sections:

1. The user interface build using Ionic framework including HTML5, CSS and SASS.
2. The Ionic activities and AngulaJS scripting language will deal with the business logic.
3. For the back-end system, Google Firebase will be used to host and handle all the information and data received from the application.

8.2 Hardware Architecture

During the research for the hardware architecture the market shows that only a few things are needed to be able to host and run this application. First, iOS and Android devices are required with at least 2GB amount of RAM, but also, the versions of the devices and operating systems should be compatible with this application.

8.3 Security Architecture

One important point about the security architecture is to make sure that the application is safe and secured very well. It is vital that only the registered users can login and use the application as the app will allow users to make changes to it. That will require extra functionality to be added on the registration scripting code side such as valid taxi plate number or valid taxi licence. Google Firebase authentication system is very helpful and gives developers the opportunity to be built such a secure authentication system.

9 System Design

9.1 Application Algorithms and Methods

In the next section of the document will be a description of the main algorithms and methods required, to be implemented for the main functionalities mentioned earlier in the report.

9.1.1 Registration

Below is a screen shot with a small part of code which shows the **Register Controller** that allows a new user to Register with a valid email, password, licence number and the details are saved and stored into Firebase database.

```
//this register controller that controls register activity
.controller('RegisterCtrl', function($scope, $state, Firebase) {
  $scope.goBack = function() {// arrow from top that brings user back
    $state.go('welcome');//redirect's user to welcome activity
  };
  $scope.form = {};//this is empty object for the form

  //this is register button when is clicket creates user name and password
  $scope.register = function (form) {
    if (form.$valid) {
      var email     = $scope.form.email,
          password  = $scope.form.password,
          licence   = $scope.form.licence;
      console.log('Email: ' + email);
      console.log('Password: ' + password);
      console.log('Licence: ' + licence);
      firebase.auth().createUserWithEmailAndPassword(email, password)
        .then(function(user) {
          var user = firebase.auth().currentUser;
          // Save the Licence to the database.
          console.log('USER: ' + JSON.stringify(user));
```

9.1.2 Login

Below is a screen shot with a small part of code which shows the **Login Controller** that allows a Registered user to Login with a valid email and password and these details are handled by the Firebase database.

```
.controller('LoginCtrl', function($scope, $state, $ionicPopup, $timeout) {
  $scope.goBack = function() {
    $state.go('welcome');
  };
  $scope.form = {};
  // Check if the user is loged in
  firebase.auth().onAuthStateChanged(function(user) {
    if (user) {
      //if already loged in redirect him to main menu
      $state.go('mainMenu');
    }
  });
//this for the login button when a user try to log in
  $scope.login = function(form) {
    if (form.$valid) {
      var email = $scope.form.email,
          password = $scope.form.password;
      firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error) {
```

9.1.3 Street Mapp Geolocation

Below is an image with the **StreetMap Controller** which allow us to see only a part of the AngularJs code integrated for the Google Maps. This Controller handles a few very important pieces of functionalities for this app such as, Geolocation, Longitude, Latitude and Date and Time of all messages. Also, it manages all the multiple coloured pins which are visible and accessible on the map for user to click and see different incidents messages with title, date and time, address and content.

```
.controller('StreetMapCtrl', function($scope, $state, $cordovaGeolocation) {
  $scope.goBack = function() {
    $state.go('mainMenu');
  };
  $scope.goHome = function() {
    $state.go('mainMenu');
  };
  $scope.goLeaderboard = function() {
    $state.go('leaderboard');
  };
  var options = {//this are options to get the current position using the google maps API
    timeout             : 10000,
    enableHighAccuracy  : true
  };
//we are geting the current position of our browser
  $cordovaGeolocation.getCurrentPosition(options).then(function(position){
    var latLng = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);
    var mapOptions = {
      center: latLng,
      zoom: 15,
      mapTypeId: google.maps.MapTypeId.ROADMAP
    };
```

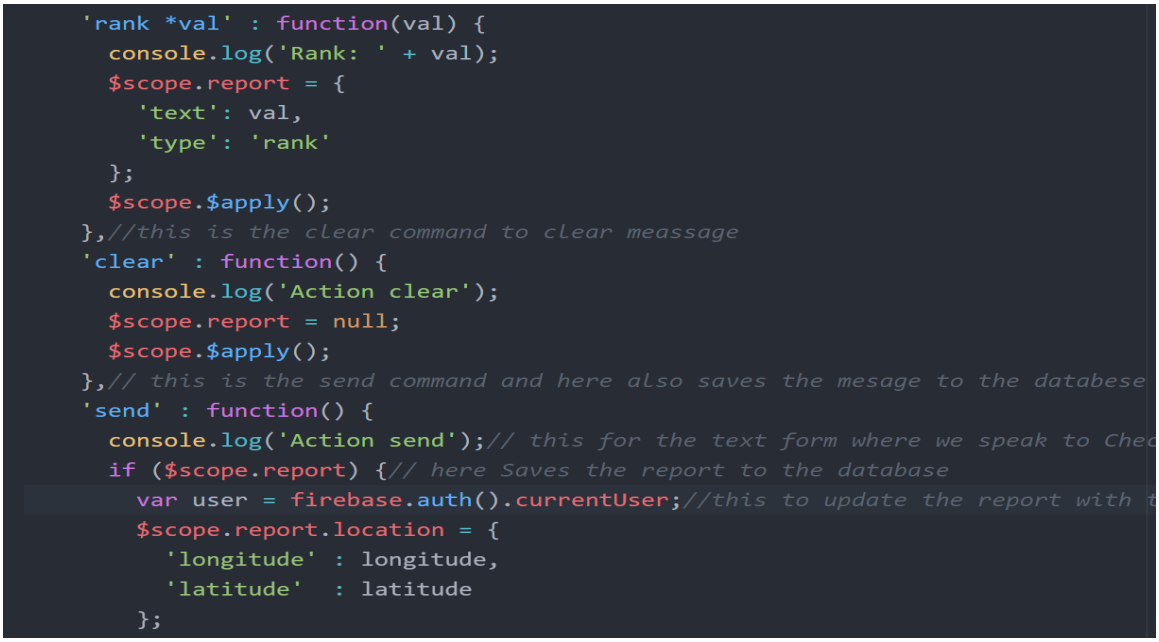
9.1.4 Leaderboard

Below is an image with AngularJs code which shows the **Leaderboard Controller** that manages user's activities. This Controller counts user's messages and then allocates points and updates the ranking position on the leaderboard. All these information's are saved and stored into Firebase database but also the number of points, ranking position on the leaderboard and trophies awards are displayed on the app for all users to view them.

```
.controller('LeaderboardCtrl', function($scope, $state) {
  $scope.goBack = function() {//button to go back on main menu
    $state.go('mainMenu');//button to go back on main menu
  };
  $scope.goHome = function() {//function to go back on main menu
    $state.go('mainMenu');
  };
  var leaderboard = [];
  var ref = firebase.database().ref('messages');//take all the messages from the database
  ref.on('value', function(snapshot) {
    $scope.ranking = [];
    snapshot.forEach(function (childSnapshot) {//for ech message we are taking the value
      var item = childSnapshot.val();
      var email = item.email;
      if (leaderboard[email]) {//we checking if email(user name) exist
        leaderboard[email] += 1;//and adding plus 1
      }
      else {
        leaderboard[email] = 1;//we are adding the email for first time and give the value of 1
      }
    });
```

9.1.5 Speech Recognition

Below is an image with some code which shows the **Speech Recognition Controller** that manages the process of creating and sending/saving the messages and store them into the Firebase database. This controller also, handles the functions which Clears the messages if they are not relevant.



```
'rank *val' : function(val) {
  console.log('Rank: ' + val);
  $scope.report = {
    'text': val,
    'type': 'rank'
  };
  $scope.$apply();
},//this is the clear command to clear meassage
'clear' : function() {
  console.log('Action clear');
  $scope.report = null;
  $scope.$apply();
},// this is the send command and here also saves the mesage to the databese
'send' : function() {
  console.log('Action send');// this for the text form where we speak to Chec
  if ($scope.report) {// here Saves the report to the database
    var user = firebase.auth().currentUser;//this to update the report with t
    $scope.report.location = {
      'longitude' : longitude,
      'latitude'  : latitude
    };
```

9.1.6 Accidents

Below is an image of the **Accidents Controller** which shows part of the AngularJs code that handles the process of displaying accident messages on the app for all users to view them. Similar code is used also for Traffic, Checkpoints and Ranks Controllers to allow users to see all the messages related to these specific points. All the messages were created by many different users and saved them into database and then the app displays them on these specific activities to be available for anyone who is using the app.

```
.controller('AccidentsCtrl', function($scope, $state) {
  $scope.goBack = function() {
    $state.go('roadInfo');
  };
  $scope.goHome = function() {
    $state.go('mainMenu');
  };
  $scope.goLeaderboard = function() {
    $state.go('leaderboard');
  };
//we are saving the current date to compare with the message dates
  var today = new Date();
  var ref = firebase.database().ref('messages');
//we are taking all the messages from the database with type=accident
  ref.orderByChild("type").equalTo("accident").on('value', function(snapshot) {
    $scope.messages = [];
//for each message we are taking their date
    snapshot.forEach(function (childSnapshot) {
      var item = childSnapshot.val();
      var date = new Date(item.date);
//down here we comparing the message date with the current date
      if (
```

9.1.7 User Profile (Update User Details and Password)

Below is an image with some code which shows the **User Profile Controller** that manages the process of updating user details or password and saves them into the Firebase database. After the user is successfully updating his details and password he can log out and then log back in with the new updated details.

```
.controller('UserProfileCtrl', function($scope, $state, $ionicPopup, $timeout) {
//by default we the details form wit this code
  $scope.showDetailsForm = true;
  $scope.showPasswordForm = false;
// we are Geting  the current user and display on the detail user form
  firebase.auth().onAuthStateChanged(function(user) {
    if (user) {
// User is signed in.
//we are going to firebase with these details and filing the form
      var ref = firebase.database().ref('users/' + user.uid);
      ref.on('value', function(snapshot) {
        var obj = snapshot.val();
        $scope.form = {
          id         : user.uid,
          name       : obj.name,
          surname    : obj.surname,
          email      : obj.email,
          licence    : obj.licence
        }
      });
    }
  });
```

10 Testing

10.1 Think Aloud Test

The Think Aloud Test would be one of the best ways to evaluate the app. That shows the user going live through the application and some vital information can be obtained from it. By doing the think-aloud test the developer would have the chance to find out and fix some issues within the app before it becomes operational online for the users. Therefore, some of the main important functionalities from the app were gathered and included to the think-aloud test list. Based on this table of functionalities, typical questions were made up to assess the layout and design of the app.

10.2 Think Aloud Tasks

- Register for the app
- Logout and Login again
- Explore Main Menu Section
- Navigate to Street Map activity
- Click on any pin form the map and check
- Explore Road Info Section
- Navigate to Accidents activity and view messages
- Navigate to Traffic activity and view messages
- Navigate to Checkpoints activity and view messages
- Navigate to Ranks activity and view messages
- Navigate to User Profile activity
- Update your name and taxi plate number
- Update your password
- Navigate to Leaderboard and check points and position
- Navigate to Speech Recognition activity
- Create any given message by voice commands
- Clear or save the message

10.3 Results for Think Aloud Test

Task	Result
Register to the App	The Testers did very well this task with no difficulties. Everyone looked confident and did it in a very efficient way

Logout and Login again	Without hesitation testers were successfully logged out and then quickly with no issues logged back in
Explore Main Menu Section	Testers were landed successfully to the Main Menu activity and explored their buttons / options in a reasonable time.
Navigate to Street Map activity	Here all the Testers were very confident to navigate and find the Map and the coloured pins on it.
Click on any pin form the map and check	Every Tester could click and open any pins from the Map which displayed different messages depending on its colour
Explore Road Info Section	Testers were landed successfully to the Road Info section and explored their buttons / options in a reasonable time.
Navigate to Accidents activity and view messages	Without hesitation testers successfully navigated with no issues to the Accidents activity and viewed some of the messages
Navigate to Traffic activity and view messages	The testers could navigate with no issues to the Traffic activity and viewed some of the messages
Navigate to Checkpoints activity and view messages	Testers started to get used with the app by this time and everything became very familiar to them so there was no problem at all to find and view the Checkpoints messages
Navigate to Ranks activity and view messages	This was the fourth time for the Testers to do the same testing only on different name activity. They were very confident to find the Rank activity and its messages
Navigate to User Profile activity	Without hesitations Testers, could navigate to the User Profile activity very quick and efficiently
Update your name and taxi plate number	Testers had no difficulties to enter another name and plate number for demonstration purpose and update them
Update your password	Testers were entering their new passwords and updated successfully without any issues
Navigate to Leaderboard and check points and position	The Testers could navigate easy to the Leaderboard to view and check the ranking on the table and users points as well
Navigate to Speech Recognition activity	Without hesitations Testers managed very well to reach the Speech Recognition activity

Create any given message by voice commands	It was a bit more difficult here as some Testers had to do the process of creating a message more than one time because they had to speak clearly the commands
Clear or save the message	After creating messages the Testers did clear some messages but also they saved some messages successfully

10.4 Effectiveness

The Testers did not have any problems by using and navigating through the app. They were testing the app and accomplished their tasks in a reasonable time with slight or no hesitation. As soon as the testers were aware what they have do, there was no problem for them to go step by step and gain the anticipated results.

10.5 Efficiency

There was no limit time allocated to the Testers when they were asked to do their tasks, in fact, most of them were finished within 20 to 40 minutes. Obviously, this gives a very good impression about the app, taking in consideration the fact that not every tester was from IT area. This kind of testing gives developer's a much better understanding about the app quality. The feedback can be used to fix or improve the application if it's necessary with not much time or financial investment.

10.6 Scope for Further Improvements

The participants to the app Testing did not find any issues regarding the functionalities of the actual application. However, there were some Testers which came with some very interesting ideas and suggestions for the future version of the app. All these suggestions and feedback were taken into consideration and will be discussed with the developers and some decisions will be made in future.

10.7 Testing Conclusion

Most of the testers completed their tasks with no difficulties in an impressively short time. It means that the application is user-friendly and very efficient. The testers did not have to communicate too much or to put a huge effort when they were doing the testing tasks on the app and this show's again high performance and efficiency. When the test was finished, many Testers expressed their thoughts and suggestions about some parts and functionalities of the app. All these suggestions and feedbacks were taken into consideration and will be analysed and some

decisions will be made for the next version of this app. Overall all the Testers were positive, pleased and satisfied with the app.

10.8 System Usability Scale

Along with the usability testing the Testers were given to fill and answer the System Usability Scale (SUS) which has been used to evaluate the general usability of the app. Therefore, ten question where given to each Tester with the possibility of scaling any answer from 1 to 5. For example: 1 means Strongly Disagree or 5 means Strongly Agree, etc. Below are the ten questions answered as following:

1. I like the app very much and my plan is to use it always
2. I discovered that the app is not necessarily complex
3. I was thinking that the app would be easy to use
4. I believe that I might need some technical support to use this app
5. I discovered that most of the functions in this app have been well integrated
6. I thought there has been too much inconsistency in this app
7. I believe that most users would learn very fast how to use this app
8. I discovered that this app is very cumbersome to use
9. I was very confident when I used this app
10. I had to learn some technical skills to be able to start using this app

After examining the data which was completed and submitted through the Usability Scale Calculations, it was revealed that user experience of most Testers who used to test the app was very positive. The average score of System Usability Scale (SUS) was 80 which in relation to the industry standards is a little bit more than the average. This clearly shows that the application met high efficiency and effectiveness standards.

11 System Evaluation

A very important functionality which must be implemented on future versions of this app is related to the user password. In case that a user forgets the password, he should be able to create a new one. Also, another functionality which can be implemented in future versions of this application is the gamification aspect. Usually taxi drivers are getting bored by waiting for the next customer and it will be a good idea to implement something more funny and attractive to play, maybe a quiz related to Dublin city places, roads etc. This will keep the drivers active as much as possible within the app.

12 Conclusions

One of the main advantages of this project is that the application will run on both platforms iOS and Android devices and both type of users will benefit from that. One disadvantage is that there is not enough open source API's specific for this type of taxi app. But there are great opportunities with the amount of open source materials, platforms, tools and NoSQL database. The limits for this project will be the fact that we can't run an iOS emulator on the windows laptop or desktop machines. For that reason, later nearly to the end a MacBook will be provided to be able to run both emulators iOS and Android for the demonstration and presentation.

13 Appendix

13.1 Project Proposal

BangOnTaxiApp

Objectives

The project I have decided to create is a mobile app for all taxi drivers in Dublin, which might want to interact, to use it and to make their work easier. Every taxi driver will be very important because she/he can bring something relevant to this app and make it more fun and efficient.

Part of this app will be also related to "Gamification" which is a concept that awards the user in a way that increases interactivity with an application and records their progress. It can take the form of creating and saving messages and the results will be posting for other users to view.

Awards are gained by the users in the form of, total number of points achieved and a ranking place on a leaderboard. The gamification aspect of this application will be based on messages on each section of the app. The user's activity/results will be displayed on a leader-board, allowing them to compare themselves with other users.

Background

The taxi businesses in Dublin are owned by number of different people, which have and use their own personalized app. Motivated by this objective, I thought that this is a very good hint to build an app which might be of a real help and needed in taxi business in a very busy Dublin city. Recently, I got an idea to build an app which

can be used by any taxi company/driver, but also by any private taxi drivers in town.

At present, every taxi driver uses their own company app or social media networks to interact with other drivers in Dublin and it is not easy to do that all the time.
The manager of a big taxi company in Dublin, recently asked me to build a new app for all the taxi business and drivers in the city. This is supposed to be an app with a more professional look and especially advanced functionalities than any existent apps on the market (a mobile app to be a huge help and benefit to all taxi drivers and taxi business).

After some discussions with a taxi business owner and some taxi drivers, they were delighted with the idea and functionalities that the mobile app would have and predicted that it would be of a real benefit for the taxi business.

Technical Approach
To have an effective impact on the taxi business, and to be a benefit for all its drivers, a mobile app that would work on both known operating systems, iOS and Android that would suit best. With time constraints and my limited knowledge in all these areas I looked at some cross-platform frameworks that would make this possible without having to develop a separate code for each of the existing operating systems.

Through all my research I came across the Ionic framework which uses AngularJS. This AngularJS scripting language allow me to control and create interactions within my app. Ionic uses also Cordova package / library which will package the web code and compile it into native applications for both mobile platforms. Cordova also gives us access to native functionality, like the accelerometer or a camera through a series of JavaScript APIs.

Ionic also uses Sass to create its styles. Sass is a CSS pre-processor that gives us a lot of capabilities on top of CSS. Now like many other projects Ionic uses Gulp.js to manage projects, processing Sass, and handle other tasks.
Ionic uses a few other utility libraries to operate, including things like Node.js for handling installation, Git for version control, and Bower for installing some of its components.

For the database storage, I will use Firebase which is a free platform owned by Google. This tool is very helpful to develop a real-time application. It uses NoSQL database structure that stores data in a JavaScript object notation and this is easy

to work with AngularJS Apps. One of the main features in Firebase is, the ability to handle different kinds of registration automatically for us.

The rest of the implementation may change as times goes on until the end of next semester.

Functionality

One of the main functionalities in this app will be voice recorder (Speech Recognition). For example, if a taxi driver is on a specific road and an accident causes a huge traffic or blocks a road, by touching a button on the app he will be able to leave a voice message. This message should be available in text for all the other drivers which have the app activated on their devices. The same thing can be applied for any incidents like, checkpoints, radars, busy traffic, etc.

Another main functionality will be, to give information about the taxi ranks on the streets. For example, if the rank on O'Connell Street is full, the drivers should go somewhere else for parking without driving all the way to check if the rank is full or not. This can be done by looking on the app to see, if there are recent messages saying that the rank is full, if yes, there is no point to drive to this rank for parking.

What the application should do:

1. Allow the users to login with username and password. To view the app and its content, all users must login in with a valid username and password.
2. Opening the app and its activities in reasonable time. The app will be built by a specific SDK tool which is available on iOS and Android platforms.
3. Directing the users to the activities they are looking for. The application will have specific activities which will be linked together to ensure that the users get to the right place in no more than three steps without any difficulties.
4. Displaying right information related to the heading of that activity. The app will be checked and validated to ensure that the content from every activity correspond to the heading of that activity.
5. To be responsive and accessible by the users on mobile smart phones like Android and IOS.
6. Keep track of user's activity, points and see position on a leader-board.

What the application should not do:

1. Can't give the users any other information other than those relating to traffic on the Irish roads. The application will specifically focus on the traffic roads and nothing else.

2. Can’t warrant the upload time, as the client side connection is dependent on the upload performance.
3. A user’s profile can only be stored in a database or via a social media account.

Technical Details

Implementation language and principal libraries

Frontend

- Ionic Framework
- Cordova Library
- HTML5
- Sass
- Gulp.js
- Git
- Bower

Server side

- AngularJs
- NoSQL Firebase for database connectivity
- REST architecture

DBMS

- NoSQL

Evaluation

How the system will be evaluated with an end user:

- Discussions with taxi manager
- A taxi manager discussing the app and its features with drivers
- Using of feedback forms available for the drivers

13.2 Monthly Journals

13.2.1 September 2016 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

This is the first journal for my biggest software project this year, and I am looking forward to deliver the best work for my application.

This month, I managed to spend more time talking and thinking about a good project idea, and I did a lot of research in this area as well. After all, at the end, I

decided to build a new Taxi Mobile App especially for the Dublin taxi drivers but in future, I may extend this for taxi drivers throughout Ireland.
My contributions to the project included also, some personal meetings with two of the NCI's teachers in the college. I explained my project idea to Sam Cogan and he accepted as a good and valid one.

My Reflection

I felt, everything worked well for me at the start of my 4th year in NCI, the preparation for this semester and to have the idea for this big project sorted out already, for me means a lot. Of course, I was spending a lot of time and I was working hard to achieve this project plan.

Intended Changes

Next month, I will try to find out and decide exactly what kind of software I'll use for the App, what platforms, tools, scripts, languages, etc. I intend and hope to develop an App not just for the android smartphones but also for Apple Phones IOS too.

Supervisor Meetings

After I will get the approval for my project to go ahead, I should find an available teacher that can be my supervisor for the whole year.

Date of Meeting: 09 - 09 - 2016
Items discussed: Several project ideas.
Action Items: I decided to go ahead with the BangOneTaxiApp.

13.2.2 October 2016 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

Now, this is my second reflective journal, and I am very glad to say that, my project idea was approved by the college committee, who queried me about the application that I want to build. So, this month my achievements were:
First, I prepared for the presentation Pitch and then I presented my project idea in front of the college committee.
Second, I started and completed the project proposal and then I have uploaded it to Moodle on time. Third, I also had a meeting with Frances Sheridan which is my supervisor for this year software project, but also with other students which are supervised by her at the same time.

My Reflection

I think, that most of the things worked well for me, in relation to this 4th year project on a very busy second month at NCI. Of course, things can be done much better, but with six modules to deal with this first semester, it is hard to stay on top and have everything under control. But it is still possible by putting the maximum effort on it and spending extra time at home to study; so, there should be no problem.

Intended Changes

After I talked to the committee jury and to my supervisor, I decided to make a change to my app. They advised me to implement a voice recorder in my application instead of typing on the app all the time. This will make it easy and more safe for the drivers to communicate and create voice messages when they must do so, especially while on driving time.

Supervisor Meetings

As I mentioned, I already had a meeting this month with Frances Sheridan my supervisor and we discussed again about the project idea a little bit but, also the best ways to approach the work on the project.

Date of Meeting: 26 - 10 - 2016
Items discussed: Project idea, the steps that must be followed to be more efficient.
Action Items: To start working on the requirements as soon as possible but also on the prototype too.

13.2.3 November 2016 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

Now this is my third reflective journal and the things are getting very busy with 6 modules to be handled and slowly we come closer to the midpoint presentation. On top of this Christmas is nearly there as well. My achievements this month are as follows:
First, I completed the project requirements specifications and uploaded them to Moodle. Second, I started the Technical Report and I have that nearly finished ready to be uploaded on Moodle. Third, I started my prototype and did some progress on it. I also met Frances Sheridan my supervisor for this software project, and have some feedback from her relating to the project.

My Reflection

This month I think, the things worked a bit harder for me, because of so many CA's and projects to be completed. But I still think some nice work was done and I made good progress overall. With maximum effort, I hope I will be able to upload everything on time and have prototype and PowerPoint slides ready for the presentation.

Intended Changes

I did not make any major changes to my application but I still think about changing the implementation of voice recorder. I know that it is a very important functionality to have on this type of app, but I am afraid that I will not be able to implement it on time, with so much stuff to do on other modules.

Supervisor Meetings

Also, this month, I had a meeting with Frances Sheridan my supervisor and we discussed again about the progress on the Project and how to make sure everything goes in the right direction.

Date of Meeting: 05 - 12 - 2016
Items discussed: Technical Report, Midterm Presentation Slides and Prototype.
Action Items: To finish the technical report and the power point slides for the midpoint presentation but also to do as much as possible on the prototype.

13.2.4 December 2016 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

This is the last journal for 2016 which is mainly based on the midpoint presentation. The achievements this month were as follows:
First, the Technical Report for the project was completed and uploaded on to Moodle. Second the PowerPoint presentation slides for the prototype were done as well. Third, some more work was done on the prototype to ensure it is ready for the presentation. Both Technical Report and presentation slides were sent to Frances Sheridan to be checked, and had some feedback from her before the presentation.

My Reflection

This month was a bit difficult, because of so much stuff to be completed on all modules. But overall some good progress was made and the main thing was that the midpoint presentation was done successfully.

Intended Changes

After the midpoint presentation feedback, there are not any major changes to be done on this application. The only thing what was suggested by the jury team during the presentation was to continue working on the voice recorder functionality and try to implement this on the app.

Supervisor Meetings

This month, there was no meeting with Frances Sheridan my supervisor because we had one on the 5th of December and shortly after that it was the presentation and then the break before the exams. We did collaborate and discussed through emails about the progress on the Project and how to make sure everything goes in the right direction.

Date of Meeting: 05 - 12 – 2016
Items discussed: Technical Report, Midterm Presentation Slides and Prototype.
Action Items: To continue the work on the prototype and to do as much as possible for the entire project.

13.2.5 January 2017 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

This is the first reflective journal for 2017 and is giving the impression that there is not much time left until the end. The achievements this month were as follows: First, the app was connected successfully to the Firebase database where drivers can now register with email, password and taxi policy number. Also, Log in works, for example after a driver is registered, it needs only email and password to log into the app. Second major achievement was the Speech Recognition implementation. This functionality is the most important for this app and is implemented, but needs a bit more work to be done in this area until it will work properly. Third a Log out button was integrated on the app to allow the drivers to leave safety when they wish.

My Reflection

This month was a bit more relaxed with the small break after the exams, but, there was no way to stop, and the extra time was spent working on the project. Overall, it was good because a huge progress was made on the app. The main thing is that the app is connected and can communicate with the Firebase database.

Intended Changes

After the last reflective journal, there are not major changes done on this app. The only thing what was changed is on the Main Menu activity, where a Log Me Out button was added.

Supervisor Meetings

This month, again, I had a meeting with Frances Sheridan my supervisor and we talked about the project evolution stages and priorities. We also did collaborate and discussed through emails about the Project.

Date of Meeting: 30 - 01 – 2017

Items discussed: The progress made since midpoint presentation.

Action Items: To continue to work and implement the Speech Recognition functionality on my project.

13.2.6 February 2017 Reflective Journal

Student name: Maricel Costel Buiciuc

Programme: BSc in Computing

My Achievements

Being the second reflective journal for 2017 it is obvious that the time is flying very fast and the project looks more relevant than ever before. This month the project was improved by adding into it, the Google Street maps API and geolocation. Now when a driver leaves a voice message about any incident on the road, firstly the message is transformed into text then goes to the database and after that is displayed on the app messages required section. Also, the message goes automatically into google street map and based on its category it will add a different color marker for it on the map. For example, a red mark will be allocated for accidents and if a driver clicks on it an accident message pops up with the address where that happened. This month the work for the show case was started as well.

My Reflection

This month started to be busier again, because a few more projects and CA’s for other modules need to be done as well and the time was running out. Overall a

vast progress was made on the app by having the most important functionalities up and running.

Intended Changes

After the last January, reflective journal, no major changes have been made to this app other than some colours and styling.

Supervisor Meetings

This month again, I had a meeting with Frances Sheridan and we discussed about the project evolution. We also did collaborate and discussed through emails about the Project. She has been added as a collaborator on my Git Hub account.
Date of Meeting: 20 - 02 – 2017
Items discussed: The progress made since January meeting, and Git Hub.
Action Items: To continue to work and implement the Google street map and geolocation functionality on my project.

13.2.7 March 2017 Reflective Journal

Student name: Maricel Costel Buiciuc
Programme: BSc in Computing

My Achievements

This is the last monthly reflective journal for the project and most of the hard work was carried out with only minor adjustments remaining to be made. This month user profile was integrated into the app, where the user can go into it to edit or update user name and password. The Leader board was integrated as well, where users who participate most to the app by leaving relevant messages about road traffic, will be displayed on top, followed by other users with less contribution. Also, a logo image was prepared and integrated into the app which will be displayed on the Welcome, Register and Log In activities. Another thing which has been done for the project was the poster for the final presentation.

My Reflection

This month was a very tough time again, as the amount of work became huge because of other projects deadline approaching. Overall a very good progress was made on the app because all the requirements and functionalities were implemented.

Intended Changes

After the last February journal, no major changes have been made to the application. Working to implement the original requirements and the new thing added to the app was the logo image.

Supervisor Meetings

This month was maybe one of the last meetings with Frances Sheridan at the end of the semester before the final presentation and we discussed the project progress. We also disused about the last uploads and how to prepare for the presentation.

Date of Meeting: 03 - 04 – 2017
Items discussed: The progress made since the last meeting, the final documentation, presentation and testing my Angular JS code.
Action Items: To continue to work on the project by implementing the Leader board functionality and updating the messages every 24 hours. Also, to do the testing.

14 References

Angular.io. (2016). *Angular*. [online] Available at: https://angular.io/ [Accessed 8 Dec. 2016].

App.pluralsight.com. (2016). [online] Available at: https://app.pluralsight.com/player?course=building-mobile-apps-ionic-framework-angularjs&author=steve-michelotti&name=building-mobile-apps-ionic-framework-angularjs-m7&clip=0&mode=live [Accessed 8 Dec. 2016].

Docs.angularjs.org. (2016). *AngularJS*. [online] Available at: https://docs.angularjs.org/api [Accessed 8 Dec. 2016].

Dublin's Taxi App. (2016). *Home | Lynk - Dublin's Own Taxi App*. [online] Available at: http://www.lynk.ie/ [Accessed 8 Dec. 2016].

Healy, S. (2016). *Dublin's biggest taxi companies are joining forces against apps like Uber and Hailo*. [online] TheJournal.ie. Available at: http://www.thejournal.ie/taxis-dublin-ecab-mobile-app-2381423-Oct2015/ [Accessed 8 Dec. 2016].

Ionic Framework. (2016). *Ionic Framework*. [online] Available at: https://ionicframework.com/docs/ [Accessed 8 Dec. 2016].

Lynda.com - from LinkedIn. (2016). *Welcome*. [online] Available at: https://www.lynda.com/AngularJS-tutorials/Welcome/368920/386098-4.html?srchtrk=index%3a0%0alinktypeid%3a2%0aq%3abuilding+ionic+app+

with+angular+js%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2 [Accessed 8 Dec. 2016].

Maddalone, J. (2016). *AngularJS Speech Recognition with annyang*. [online] YouTube. Available at: https://www.youtube.com/watch?v=Ds5tm-6Nc9g [Accessed 8 Dec. 2016].

Renaux, J. (2016). *Create a Mobile Application Using WordPress, Ionic, and AngularJS*. [online] Code Envato Tuts+. Available at: https://code.tutsplus.com/tutorials/create-a-mobile-application-using-wordpress-ionic-and-angularjs--cms-24170 [Accessed 8 Dec. 2016].

Team, W. (2016). *Hailo*. [online] Hailoapp.com. Available at: https://www.hailoapp.com/ie/?gclid=Cj0KEQiApqTCBRC-977Hi9Ov8pkBEiQA5B_ipdzkZO4XpZu77hBD39_U0_zqUmFBhNDQt4mEFEa-xacaAqrt8P8HAQ#?campaign=637125313&label=&creative= [Accessed 8 Dec. 2016].

Wilken, J. (2016). *Getting Started With Ionic: Cordova*. [online] Code Envato Tuts+. Available at: https://code.tutsplus.com/tutorials/getting-started-with-ionic-cordova--cms-27023 [Accessed 8 Dec. 2016].