

# Drop Solutions



**Ruth Doran**

X13331191

**BSc (Honours) in Computing  
Mobile Application Development  
2016/2017**

**Supervisor: Dominic Carr**

## Declaration Cover Sheet for Project Submission

### SECTION 1

<b>Name:</b> <b>Ruth Doran</b>
<b>Student ID:</b> <b>X13331191</b>
<b>Supervisor:</b> <b>Dominic Carr</b>

### SECTION 2 Confirmation of Authorship

I confirm that I have read the College statement on plagiarism (summarised overleaf and printed in full in the Student Handbook) and that the work I have submitted for assessment is entirely my own work.

Signature: \_\_\_\_\_ *Ruth Doran* \_\_\_\_\_ Date: 10/5/17 \_\_\_\_\_

NB. If it is suspected that your assignment contains the work of others falsely represented as your own, it will be referred to the College's Disciplinary Committee. Should the Committee be satisfied that plagiarism has occurred this is likely to lead to your failing the module and possibly to your being suspended or expelled from college.

# Table of Contents

Executive Summary .....	4
1.1 Background.....	5
1.2 Technologies.....	5
2 System .....	6
2.1 Requirements .....	6
2.1.1 Functional Requirements .....	7
2.1.1.1 Requirement 1: Sign Up.....	9
2.1.1.2 Requirement 2: Sign In .....	10
2.1.1.3 Requirement 3: Change Password .....	12
2.1.1.4 Requirement 4: Change Email .....	13
2.1.1.5 Requirement 5: Send Reset Email .....	15
2.1.1.6 Requirement 6: Sign Out .....	17
2.1.1.7 Requirement 7: Maps.....	19
2.1.1.8 Requirement 8: Important Contacts .....	20
2.1.1.9 Requirement 9: Group Chat .....	22
2.1.1.10 Requirement 10: Push Notifications.....	24
2.1.2 Data Requirements.....	26
2.1.3 User Requirements.....	26
2.1.4 Environmental Requirements.....	27
2.1.5 Usability Requirements .....	27
2.2 Design and Architecture .....	28
2.3 Implementation.....	28
2.3.1 Our Depots Map Page and Functions.....	28
2.3.2 Sign In Page and Functions.....	28
2.3.3 Group Chat Page and Functions .....	29
2.3.4 Our Staff Page and Functions .....	30
2.3.5 Important Numbers Page and Functions .....	31
2.4 Graphical User Interface (GUI) Layout .....	31
2.5 Testing .....	38
2.6 Customer Testing.....	39
3 Conclusion .....	40
4 Further Development or research.....	41

5	References.....	42
6	Appendix.....	43
6.1	Project Proposal .....	43
6.1.1	Objectives .....	43
6.1.2	Background.....	43
6.1.3	Technical Approach .....	44
6.1.4	Technical Details.....	44
6.1.5	Evaluation .....	44
6.2	Project Plan.....	45
6.3	Monthly Journals.....	46
6.4	Testing Materials .....	54
6.4.1	Firestore Issues List .....	54
6.4.2	Firestore Issues Graph.....	54
6.4.3	Firestore Test Lab Results.....	55
6.4.4	Customer Testing Survey Results .....	56
6.4.5	Individual Customer Survey Responses.....	59

## **Executive Summary**

The idea behind my project came about when a family friend who owns a delivery company approached me and asked was there any applications on the market that would fix the problems he was having. The main problems were communication issues as well as finding the fastest route to several locations. There were other minor problems such as making it easier for new employees to be familiar with certain numbers and staff within the company.

I researched and was unable to find any application so I decided that it would be possible for me to make it and have the required functionalities as well as others that would help overall. I decided to create a mobile application as it will allow ease of access to the application. I called the application DropSolutions, I wanted to give the user access to functionalities that will aid them in their day to day tasks. The application is designed for all employees within a delivery company to use.

The main problems that the application will aid in areas such as communication, directions and overall provide the user with information on the company. Such as the other depots locations, the staff in each depot as well as important numbers that the user may need. These functions are especially useful for new employees as it helps them get familiar with the company overall.

I approached my family friend and asked would it be possible for me to test my application in one of his depots. I wanted to do this as it would allow me great insight from my targeted users. They agreed to test the application for a week this gave them enough time to get use to the application and rate the functionality.

DropSolutions is an application which could be modified slightly to fit other delivery company's needs and then be launched within the company. As there is a large amount of applications out there that fix problems that delivery companies encounter but I have not found one application that has all the functionality in the one app. I feel that DropSolutions could be a solution for this as I have created specific functionalities to resolve the delivery company's common problems.

## 1.1 Background

The purpose of this document is to give a good understanding of the development of my application called Drop Solutions. It was designed for company use with delivery drivers in mind to aid them in their day to day tasks. It will allow the user access to a number of functionalities. Each functionality is designed to assist the user with the common problems delivery companies and especially drivers encounter on a daily basis. Overall I hope that my application will fix and help with a lot of the problems that delivery companies currently have.

- One of the main problems in a delivery company is communication. This happens when the office staff are trying to communicate with the drivers if they ring them it is likely that they are driving and won't answer. But the message they are trying to tell them is time sensitive. This means that the office staff will just need to keep calling the driver every so often until they answer. This is time wasting as that employee could have been answering calls from clients or other depots during this time.
- Another problem that they face is that when they hire new employees especially drivers they may not be familiar with the area. Meaning they might not know the fastest routes to all their delivery's this will lessen productivity. They will also not be familiar with the location of the other depots within the company, which they may need to visit to collect special delivery's that need to be done.
- The final common problem that I have found is a lot of new employees or even existing employees may not be familiar with people's numbers. They may need to contact them for a number of reasons such as call in sick or contact about a delivery etc.

I decided to create an application that would address the issues above and resolve them. I created the application using android for a number of reasons the first being that Android devices is a huge market that is constantly growing. The second is that I am most familiar with creating android applications, in contrast to IOS or Microsoft. This is because two of my fourth-year modules allowed me to learn a greater understanding on android applications and development of them.

## 1.2 Technologies

The main technologies used in my project are as follows. Android Studios is the application that I used to code all the application. It allowed me to completely customise every page through XML, to my specifications and add certain content and functionality to each page using Java, it would not have been possible to make my application the way I did I feel without the use of Android studios. Although there are other alternatives of applications that can be used to create mobile applications such as eclipse. I chose Android Studios as I was more familiar with it and felt because of this I thought it would be easier to create my application on this rather than an alternative.

Another technology that I used was Firebase I utilized a number of the services that firebase offers, I used firebase authentication for the user login section. There are alternatives such as using SQL for the database instead of the Firebase database. I chose to use Firebase as I had initially decided to use SQL but in the mid-point presentation I was advised that Firebase would be a much better alternative. I researched firebase after my mid-point presentation and found that it was much better in comparison as it had a number of functionalities that I could implement. This is why I choose to use Firebase over other alternatives as it allows me to use multiple functions while being able to manage them all from the one console.

Firebase allows me to store the users details as well as manage the users that connect to the application. For example, if a user has left the company it is possible that the admin login to the firebase console and delete that users account. Firebase authentication ensures the user is entering the correct email and password combination to login (See Reference 8). I also used Firebase messaging and push notifications to allow the users to receive a pop up when they receive a message sent from the admin. The firebase console also allows you to send custom push notifications that pop up on user's devices with custom messages. This will allow the admin to send messages directly to all the users (See Reference 10). Through the firebase console in authentication section it is also possible to reset specific user's passwords, if they are unable to do it themselves.

I have also used Firebase Database alongside Firebase cloud messaging for the group chat functionality (See Reference 7). The Firebase Database stores the information of each message, it stores the username, time and message that the user has sent. Each message sent gets a unique identifier that it is stored as in the database, within it is all the information associated with it, username, time and message.

## **2 System**

### **2.1 Requirements**

To use this application, you must be a member of the company and have set up an account in the application. I have made the application very user friendly and easy to use. Due to this I hope training for this application will not need to be too extensive due to the setup of the application. Users should know all of the functionalises and how to navigate each after 3-4 days of training. The users after 2-3 uses will be familiar with all the functions of the application. Due to the user-friendly design errors made by the user will be very limited. This is because all the options are very straight forward and ensure nothing is too complicated, which would confuse the user. This leads to fewer errors made by the user. The training requirements will be more extensive for the admin account due to its additional functionality's that they have access to such as the firebase console. Therefore, there training will range between 1-2 weeks this time can be increased if the user still feels they need extra training in using the functionalities.

The requirements have evolved throughout the creation of this application some requirements have changed completely due to time constraints and limits. Some of the requirements that I have no longer included into the application are admin creation of new users. I changed this as I now allow the user to create their own login making it less work for the admin and allowing easier creation of new users. I also removed driver tracking I did this as after doing further research I found that some drivers may find it too much an invasion of their privacy.

The main change that was made was the map requirements I was unable to get turn by turn navigation or to get the route to snap to the roads. I then decided to change the function as I was unable to get the map to the same level or close to the functionality google maps offers. Therefore, I decided to use google maps for the route generation with the click of a button it now opens google maps through an intent instead.

I also changed the login and sign up functions instead of the admin creating usernames and passwords the user can create their own. This will be much more useful as the user can manage their own account and reset their password or change and email if they require.

I also added a number of functionalities that I feel overall enhance the whole application and address a number of problems. I will go into further details on all of the functions and how each work in Functional Requirements.

### **2.1.1 Functional Requirements**

Functional Requirements identify what the system must do. They show the processes that the system must perform. In this section I will discuss the functional requirements for my application.

#### **1. Sign Up:**

The user must setup an account they do this by clicking the sign-up button they choose their own password and email address. This is done with the use of Firebase authentication.

#### **2. Sign In:**

The user can then sign in once they have signed up for the application. They enter the email address that they signed up with along with the password they choose and they then click sign in button. They will then be signed into drop solutions. Once the user is signed in the phone remembers the credentials until the user sign's out.

#### **3. Change Password:**

The user can also change their password once signed in if they go to the manage account page. The user clicks the change password option and an input box will appear for the user to enter their new password. Once they have entered the password they click the

change password button and the user will be told if the password has been successfully change or if it wasn't accepted.

#### **4. Change Email:**

The user can also change their email which they use to login with once signed in if they go to the manage account page. The user clicks the change email option and an input box will appear for the user to enter their new email Once they have entered the new email they click the change email button and the user will be told if the email has been successfully change or if it wasn't accepted.

#### **5. Send Reset Email:**

If the user forgets their password and is not signed in they can click the password reset button and an email will be sent to the email they use to login and they can reset the password through the email that has been sent. Once this is complete the user will then be allowed to login to the application again with the new password they have chosen.

#### **6. Sign Out:**

The sign out allows the user to log out of the application completely. Without clicking the signing out button the user will remain signed into the application even if they close it.

#### **7. Maps:**

The maps page is broken into two parts. One part is a map built with the use of Google maps API with set locations to show examples of depots within the company. The map will also allow the user to see their location on the map once they click the location button located in the top right corner. The second part of the map page is the get route, here google maps application is opened in an intent, the user can then input up to ten locations and the map will then generate.

#### **8. Important Contacts:**

This page will allow the user to see a list of important contacts that they will then be able to click the call button beside it. Once clicked the button will then open an intent to open the number in the dial screen. This allows the user to then make a call to the number associated with the button clicked or add it to their contacts.

#### **9. Group Chat:**

The group chat functionality allows all the users of the application to have send and receive messages in a group conversation. Each message that is sent shows what user sent it with their email address being displayed alongside the time they sent it at, with the message below it. The user will input a message into the text input area and then click the envelope button to send the message. The messages are stored in a firebase database, and every message that has been sent previously will be able to be viewed by scrolling up.

#### **10. Push Notifications:**

The push notifications function allows the admins to send a message of their choice to all users of the application. The users will receive a pop up of the message text that was inputted. This allows important messages of information to be received straight away.

The messages are sent through the firebase console and displayed as a notification on the device. Once clicked it will open the DropSolutions application.

### 2.1.1.1 Requirement 1: Sign Up

The requirement is Sign Up. The user clicks the sign-up button and inputs an email address and a password of their choice.

#### 2.1.1.1.1 Description & Priority

The requirement is Sign Up, this allows the users to login to the application. This requirement is essential to the application as it allows the users to sign-up and the be able to login to the application. Without signing up the user will not be able to access the application.

#### 2.1.1.1.2 Use Case

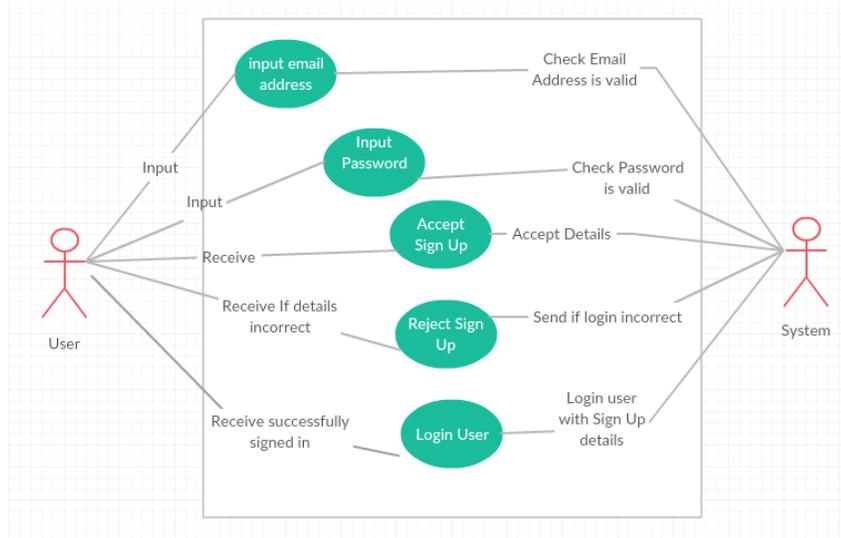
##### Scope

The scope of this use case is to allow the user to sign up to the system with an email and password of their choice.

##### Description

This use case describes the sign-up page of the application.

#### Use Case Diagram for User Sign Up Page



### **Flow Description:**

#### **Precondition**

The user must use a valid email address and a password that has 7 characters or more.

#### **Activation**

This use case starts when the user clicks the sign-up button on the sign in page of the mobile application.

#### **Main flow**

1. The system identifies the sign-up button being pressed and directs the user to the sign-up page where the user must the input an email address and password that is 7 characters or more in length.
2. Once the details are entered the user clicks the sign-up button.
3. The system checks that the details are valid, if they are valid the user will then be signed in.

#### **Alternate flow**

1. The system identifies the sign-up button being pressed and directs the user to the sign-up page where the user must the input an email address and password that is 7 characters or more in length.
2. Once the details are entered the user clicks the sign-up button.
3. The system checks if the details are valid, if they are not valid the user will be told the problem and will be allowed try again.

#### **Termination**

The system presents the user with the main navigation page.

#### **Post condition**

The system goes into a wait state.

## **2.1.1.2 Requirement 2: Sign In**

The requirement is Sign In. The user inputs their login credentials they chose for in the sign-up section.

### **2.1.1.2.1 Description & Priority**

The requirement is sign in, this allows the users to sign in to the application. This requirement is essential to the application as it allows the users to sign in without signing in the user will not be able to access the application.

### **2.1.1.2.2 Use Case**

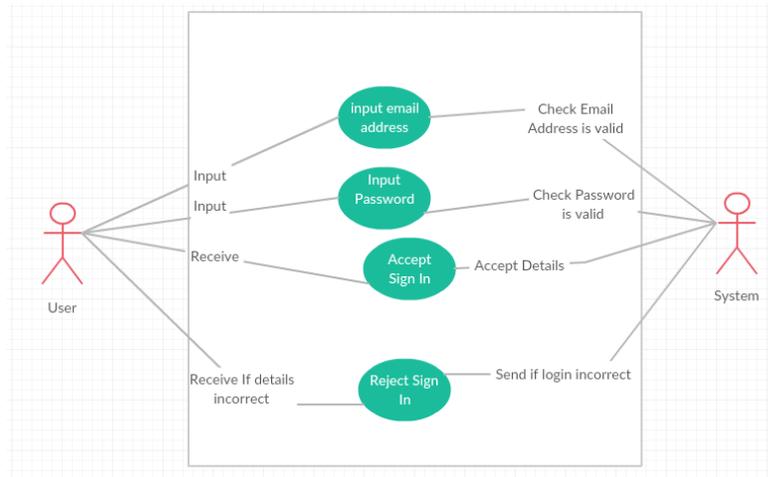
#### **Scope**

The scope of this use case is to allow the user to sign in to the system with the email and password they chose when they signed up.

### Description

This use case describes the sign in page of the application.

### Use Case Diagram for User Sign In Page



### Flow Description:

#### Precondition

The user must have signed up to the application by using a valid email and a password 7 characters or longer.

#### Activation

This use case starts when the user opens the application and has not signed in before or has signed out.

#### Main flow

1. The system identifies that no user is signed in and loads the sign in page where the user must then input their login details.
2. Once the details are entered the user clicks the sign in button.
3. The system checks that the details are valid, if they are valid the user will then be signed in.

#### Alternate flow

1. The system identifies that no user is signed in and loads the sign in page where the user must then input their login details.
2. Once the details are entered the user clicks the sign in button.
3. The system checks if the details are valid, if they are not valid the user will be told the problem and will be allowed try again.

#### Termination

The system presents the user with the main navigation page.

**Post condition**

The system goes into a wait state.

**2.1.1.3 Requirement 3: Change Password**

The requirement is Change password. The user once signed in can navigate to the manage account page and clicks change password.

**2.1.1.3.1 Description & Priority**

The requirement is Change password. The user once signed in can navigate to the manage account page and clicks change password. They then input the new password into the text box and clicks the change password button.

**2.1.1.3.2 Use Case**

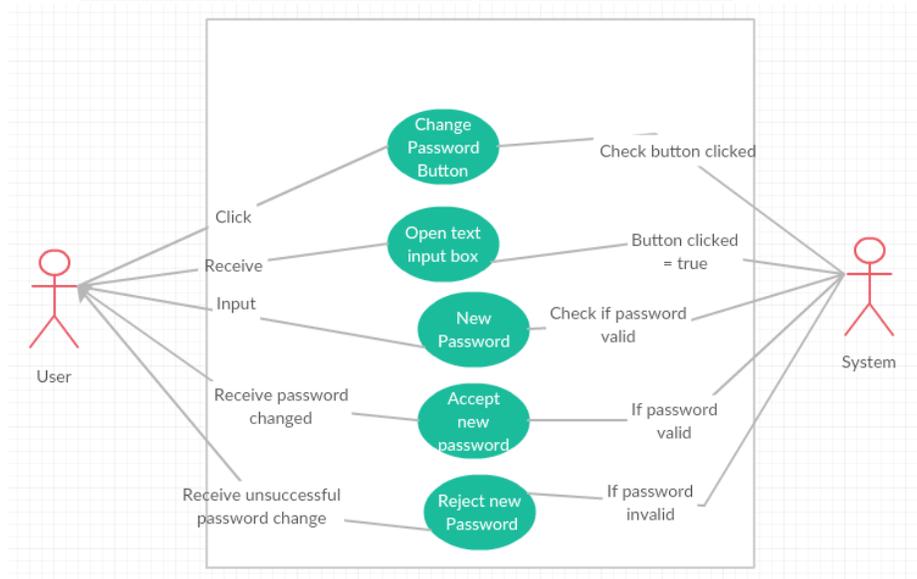
**Scope**

The scope of this use case is to allow the user to change their password once signed in.

**Description**

This use case describes the change password function of the application once signed in.

**Use Case Diagram for User Change Password Function**



### **Flow Description:**

#### **Precondition**

The user must be signed in to their account, and when choosing a new password, it must be a minimum of 7 characters in length.

#### **Activation**

This use case starts when the user clicks the change password button on the manage account page of the mobile application.

#### **Main flow**

1. The system identifies that a user is signed in and has clicked the change password button on the manage account page. An input box will then appear for the user to type in their new password.
2. Once the user has inputted a password that is a minimum of 7 characters long, they click the change password button.
3. The system checks that the password is valid, if it is valid the user will then be told that their password was successfully changed.

#### **Alternate flow**

1. The system identifies that a user is signed in and has clicked the change password button on the manage account page. An input box will then appear for the user to type in their new password.
2. Once the user has inputted a password that is a minimum of 7 characters long, they click the change password button.
3. The system checks that the password is valid, if it is not valid the user will then be told that their password was not changed and they can try again.

#### **Termination**

Once the user has successfully changed the password the system accepts the changes and this function ends, until the change password button is clicked again.

#### **Post condition**

The system goes into a wait state.

### **2.1.1.4 Requirement 4: Change Email**

The requirement is Change Email. The user once signed in can navigate to the manage account page and click change email.

#### 2.1.1.4.1 Description & Priority

The requirement is Change Email. The user once signed in can navigate to the manage account page and clicks change email. They then input the new email into the text box and clicks the change email button.

#### 2.1.1.4.2 Use Case

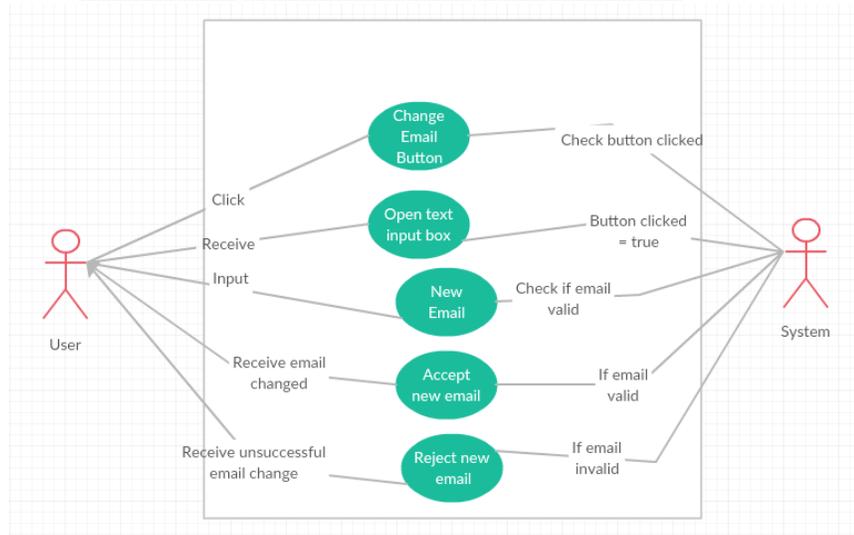
##### Scope

The scope of this use case is to allow the user to change their email once logged in.

##### Description

This use case describes the change email function of the application once signed in.

#### Use Case Diagram for User Change Email Function



##### Flow Description:

##### Precondition

The user must be signed in to their account, and when choosing a new email, it must be a valid email address.

##### Activation

This use case starts when the user clicks the change email button on the manage account page of the mobile application.

##### Main flow

1. The system identifies that a user is signed in and has clicked the change email button on the manage account page. An input box will then appear for the user to type in their new password.
2. Once the user has inputted a new email, they click they change email button.

3. The system checks that the email is valid, if it is valid the user will then be told that their email was successfully changed.

#### **Alternate flow**

1. The system identifies that a user is signed in and has clicked the change email button on the manage account page. An input box will then appear for the user to type in their new password.
2. Once the user has inputted a new email, they click they change email button.
3. The system checks that the email is valid, if it is not valid the user will then be told that their email was not successfully changed and will be allowed try again.

#### **Termination**

Once the user has successfully change the email the system accepts the changes and this function ends, until the change email button is clicked again.

#### **Post condition**

The system goes into a wait state.

### **2.1.1.5 Requirement 5: Send Reset Email**

The requirement is Send reset email. The user can get their password reset by email if they are not signed in and have forgotten it.

#### **2.1.1.5.1 Description & Priority**

The requirement is Send reset email. The user can get their password reset by email if they are not signed in and have forgotten it. The user will be on the main sign in page and can click the reset password button. They will then be taken to another page which they then input their email address into it and will be sent an email with instructions on how to reset their password.

#### **2.1.1.5.2 Use Case**

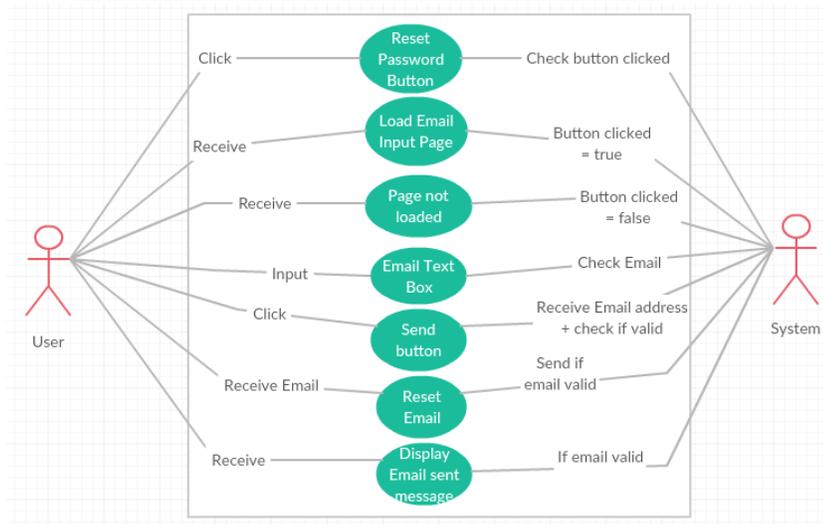
##### **Scope**

The scope of this use case is to allow the user to change their password without being signed in.

##### **Description**

This use case describes the change password function of the application when the user is not signed in.

## Use Case Diagram for Send Reset Email Function



### Flow Description:

#### Precondition

The user must not be signed into the application and must have forgotten their password.

#### Activation

This use case starts when the user clicks the password reset button on the main login page.

#### Main flow

1. The system identifies that a user is not signed in and has clicked the reset password button on the main login page.
2. Once the user clicked the button they are brought to a new page.
3. The user must then input their email address and click the password reset button.
4. The system checks that the email is valid, if it is valid the user will then be told that their reset email was sent successfully.
5. The user will then receive an email and they can click the link and then input their new password for their account.

#### Alternate flow

1. The system identifies that a user is not signed in and has clicked the reset password button on the main login page.
2. Once the user clicked the button they are brought to a new page.
3. The user must then input their email address and click the password reset button.
4. The system checks that the email is valid, if it is not valid the user will then be told that their email was not successfully sent and will be allowed try again.

### **Termination**

Once the user has successfully changed their password through the reset email the user will be able to sign in. This function will terminate until the reset password button is clicked again.

### **Post condition**

The system goes into a wait state.

## **2.1.1.6 Requirement 6: Sign Out**

The requirement is Sign Out. The user once signed in can navigate to the manage account page and click sign out.

### **2.1.1.6.1 Description & Priority**

The requirement is Sign Out. The user once signed in can navigate to the manage account page and click sign out. They then can click the sign out button and they will then be told they are signed out. The main login page will then be loaded.

### **2.1.1.6.2 Use Case**

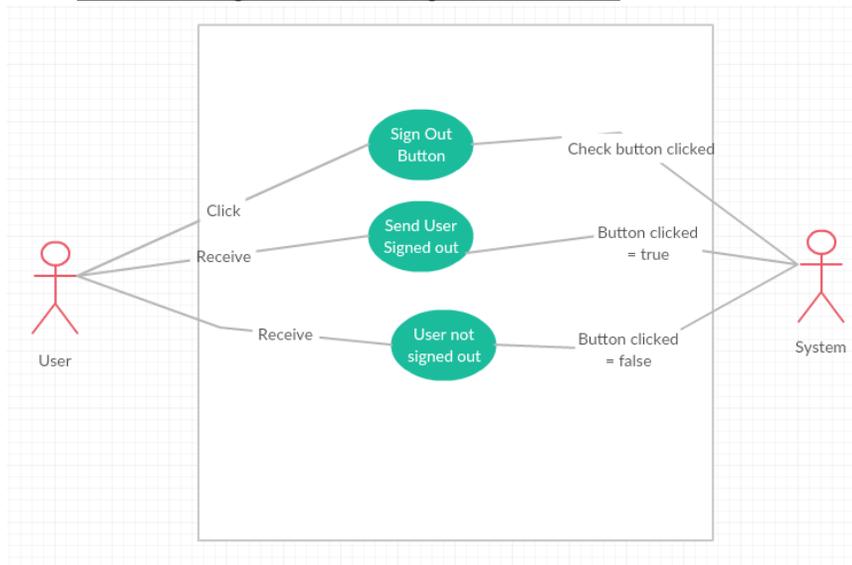
#### **Scope**

The scope of this use case is to allow the user to sign out of their account once signed in.

#### **Description**

This use case describes the sign out function of the application once signed in.

## Use Case Diagram for User Sign Out Function



### Flow Description:

#### Precondition

The user must be signed in to their account and then click the sign out button.

#### Activation

This use case starts when the user clicks the sign out button on the manage account page of the mobile application.

#### Main flow

1. The system identifies that a user is signed in.
2. The user must then navigate to the manage account page and click the sign out button.
3. The system checks that sign out button was clicked and they then sign out the user, and load the main login page.

#### Alternate flow

1. The system identifies that a user is signed in.
2. The user must then navigate to the manage account page and click the sign out button.
3. The system checks that sign out button was clicked and if it was not clicked no action is taken.

#### Termination

Once the user has successfully clicked the sign out button the system accepts the user has signed out and navigates the user to the main login page. The sign out function is then terminated.

### Post condition

The system goes into a wait state until the button is clicked again.

## 2.1.1.7 Requirement 7: Maps

The requirement is Maps. The user can get to the maps requirements from the maps page.

### 2.1.1.7.1 Description & Priority

The requirement is maps. The user can get to the maps functionalities from the maps page. From the maps page, there are two functions that the user can access, the main maps functionality is the our depots. This page uses google maps Api to display the map. Once the user clicks the our depots button a map with six set locations will be displayed as well as the user's current location.

### 2.1.1.7.2 Use Case

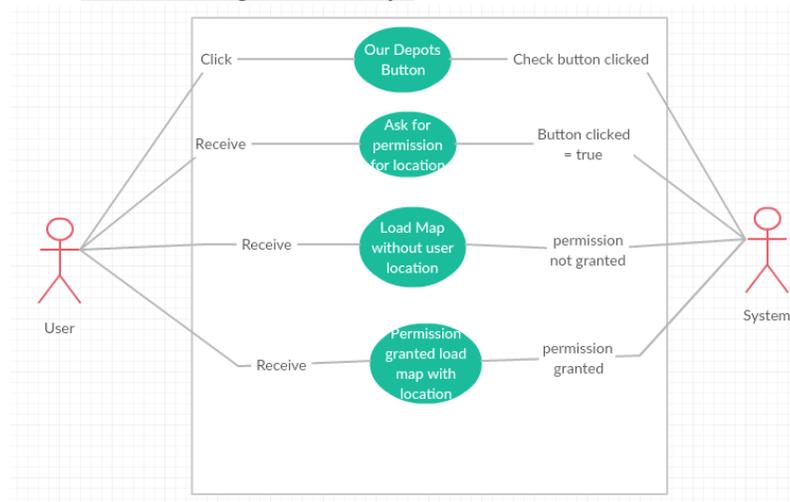
#### Scope

The scope of this use case is to allow the user to see a number of sample depots locations and their current location.

#### Description

This use case describes the map function of our depots page of the application.

#### Use Case Diagram for Maps



### **Flow Description:**

#### **Precondition**

The user must be signed into the application and must have clicked the our depots button from the maps page.

#### **Activation**

This use case starts when the user clicks the out depots button on the maps page.

#### **Main flow**

1. The system identifies that a user is signed in, on the maps page and has clicked the our depots button.
2. Once the user clicks the button they are then brought to a new page and are asked to allow permission to their location.
3. Once the user clicks allow, the page will display a map with six locations on it that are identified by boxes, And their current location.
4. They are then able to click the location button in the top right hand corner and it will zoom into their location.
5. They are also able to click one of the depots and click the directions button which will then open google maps with a route from their current location to the choose depot.

#### **Alternate flow**

1. The system identifies that a user is signed in, on the maps page and has clicked the our depots button.
2. Once the user clicks the button they are then brought to a new page and are asked to allow permission to their location.
3. If the user clicks don't all, the page will load and display the map with the depots but it will not show the current location and that function will not be accessible, until location is allowed.

#### **Termination**

Once the user has clicked back from the maps page it will terminate until the our depots button is clicked again.

#### **Post condition**

The system goes into a wait state.

## **2.1.1.8 Requirement 8: Important Contacts**

The requirement is Important Contacts. The user can call certain numbers that are listed on this page by clicking the call button.

### **2.1.1.8.1 Description & Priority**

The requirement is Important contacts. The user can call certain numbers by simply clicking the call button beside the name and number displayed. They system will then

open the dial screen on their phone with the number typed in through an intent and it will allow the user to ring the number associated with the button that was clicked or store it to their contacts.

### 2.1.1.8.2 Use Case

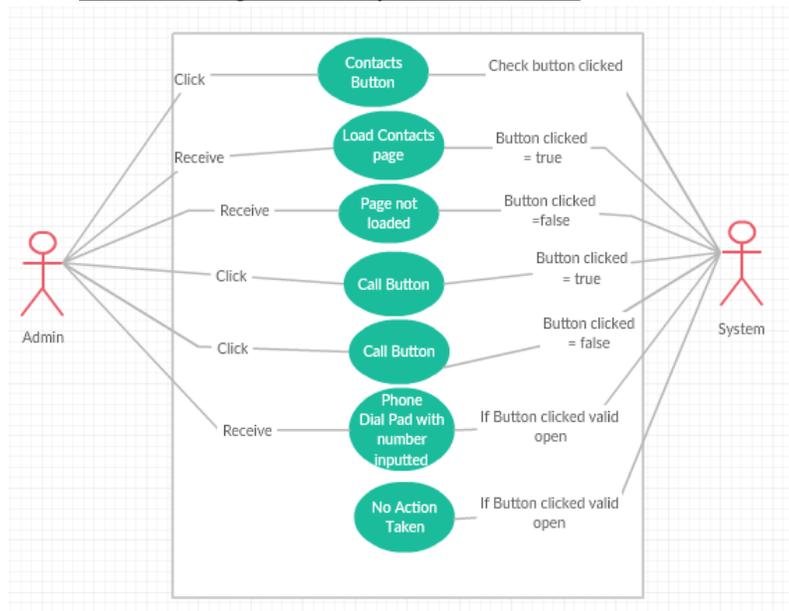
#### Scope

The scope of this use case is to allow the user to open the dial pad with the number that is associated with the button that was clicked. It allows the user to call the number or store it to their contacts.

#### Description

This use case describes the important contacts function of the application.

#### Use Case Diagram for Important Contacts



#### Flow Description:

#### Precondition

The user must be signed into the application and have clicked the important contacts button on the main menu.

#### Activation

This use case starts when the user clicks the important contacts button from the main menu page.

### **Main flow**

1. The system identifies that a user is signed in and has clicked the important contacts button on the main menu page.
2. Once the user clicks the button they are brought to a new page which displays a list of names and numbers as well as call buttons.
3. Once a call button is clicked the system checks what button was clicked and what number is associated with it.
4. It then opens the dial pad with the number displayed from the button that was clicked. It does this by opening the phones dial function through an intent.

### **Alternate flow**

1. The system identifies that a user is signed in and has clicked the important contacts button on the main menu page.
2. Once the user clicks the button they are brought to a new page which displays a list of names and numbers as well as call buttons.
3. Once a call button is clicked the system checks what button was clicked, if no call button was clicked no further action will be taken.

### **Termination**

Once the user has left the important contacts page the function will terminate.

### **Post condition**

The system goes into a wait state until the button is clicked again.

## **2.1.1.9 Requirement 9: Group Chat**

The requirement is Group chat. The user can have a group conversation with other users once signed in.

### **2.1.1.9.1 Use Case**

The requirement is Group chat. The user can have group conversations with all the other users of the application. The user must be signed in and click the group chat button from the main menu page.

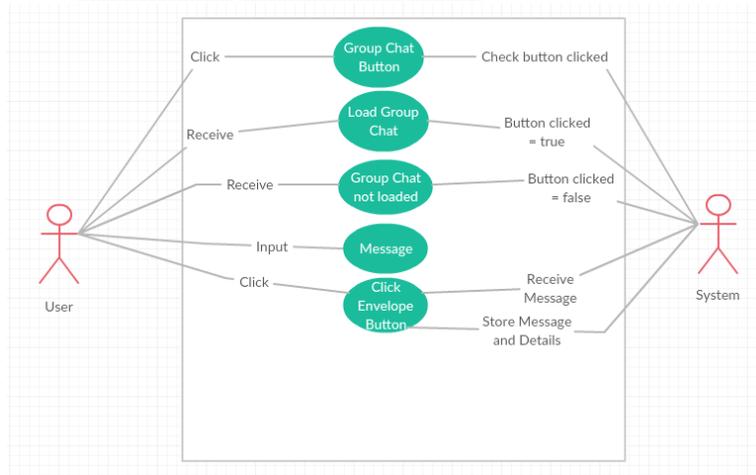
### **Scope**

The scope of this use case is to allow the user to have a group chat with other users of the application.

### **Description**

This use case describes the group chat function of the application that the user has access to once signed in. All of the messages sent in the group chat are stored in a firebase database that is accessible through the firebase console. The information saved in the database is the message, the user who sent it and the time and date it was sent.

## Use Case Diagram for Group Chat



### Flow Description:

#### Precondition

The user must be signed into the application.

#### Activation

This use case starts when the user clicks the group chat button from the main menu page.

#### Main flow

1. The system identifies that a user is signed in and has clicked the group chat button on the main menu page.
2. Once the user clicks the group chat button they are brought to a new page, where they are greeted with a welcome message alongside their email address they are signed in with.
3. The user can then see all the messages that have been sent before, and are able to type messages into a text box.
4. The user can then click the send envelope button in the bottom righthand of the screen. Their message will then be sent and displayed underneath the others.

#### Alternate flow

1. The system identifies that a user is signed in and has clicked the group chat button on the main menu page.
2. Once the user clicks the group chat button they are brought to a new page, where they are greeted with a welcome message alongside their email address they are signed in with.
3. The user can then see all the messages that have been sent before, and are able to type messages into a text box.
4. The user can then click the send envelope button in the bottom righthand of the screen. If the user dose not click the envelope button no message will be sent.

### **Termination**

The system terminates when the user exists the group chat by navigating back to the main menu page, or exiting the application completely.

### **Post condition**

The system goes into a wait state.

## **2.1.1.10 Requirement 10: Push Notifications**

The requirement is push notifications. This allows admins to send a notification to the user with a message of their choice.

### **2.1.1.10.1 Description & Priority**

The requirement is push notifications. The admin can log in to the firebase console and then navigate to the notifications section. The admin can then create a new message and input a text message that they wish the users to see. They can determine its priority as well as how long the message will last.

### **2.1.1.10.2 Use Case**

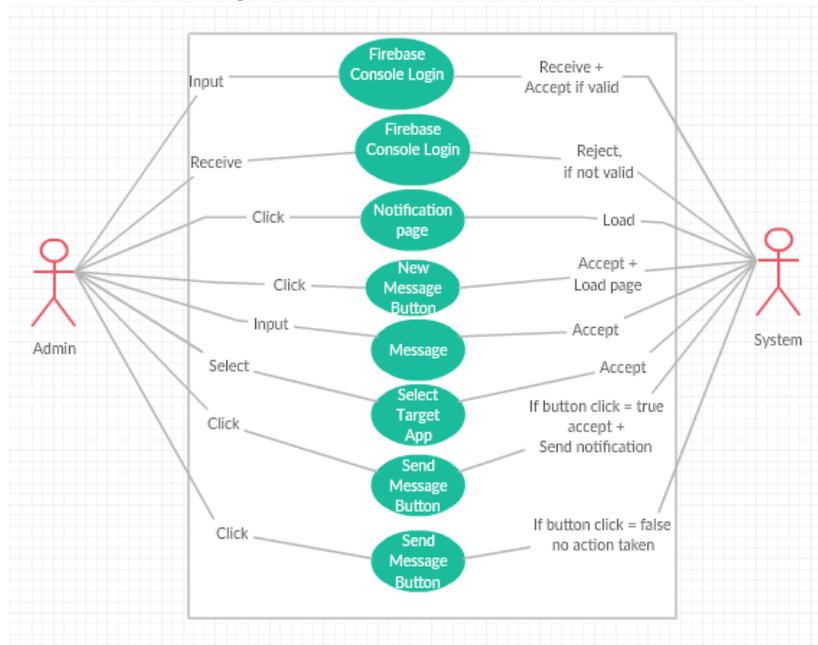
#### **Scope**

The scope of this use case is to allow the admin send important messages to the users. It ensures they can see them as it appears as a notification.

#### **Description**

The use case describes the push notifications function of the application that the admins have access to once logged into the firebase console.

## Use Case Diagram for User Push Notifications Function



### Flow Description:

#### Precondition

The user must be one of the admins as they must have access to the login for the firebase console.

#### Activation

This use case starts when the user has signed into the firebase console and navigates to the notifications page. They then click new message.

#### Main flow

1. The system identifies that a user is signed into the firebase console and has clicked new message in the notifications section.
2. Once the user has clicked new message, they will then input the message they wish the users to receive, and how long they want the message to last.
3. They then click send, they are then able to view how many users it was sent to and how many have opened it already.
4. Once the message is sent all the user of the application will receive a notification displaying the message that was sent.

#### Alternate flow

1. The system identifies that a user is signed into the firebase console and has clicked new message in the notifications section.
2. Once the user has clicked new message, they will then input the message they wish the users to receive, and how long they want the message to last.

3. If the user does not click send, the message will not be sent and the user of the applications will not receive a notification.

#### **Termination**

The function terminates once the users who receive the message click the notification. The message in the console terminates when it has lasted for the set time the admin that created the message assigned to it. For example, two weeks.

#### **Post condition**

The system goes into a wait state.

### **2.1.2 Data Requirements**

The Firebase Authentication stores the users that have signed up to use the application. It stores the email address that the user has used to sign up, in the firebase console you can see what users have signed up and what date. Each user gets a specific User UID, assigned to it. This is unique to every user as it is a unique identifier to each user.

The Firebase Database stores the messages and message data that has been sent in the group chat. Each message is stored under a unique identifier, inside this is the message alongside the text that was sent, as well as the user and email that sent it and the time the user sent it at.

Through the use of the Firebase console it is possible to edit the data that is stored, such as add a user or message. As well as delete messages or users.

### **2.1.3 User Requirements**

There are several requirements for the user in order to use the application.

**Employee or Apart of the Company:** In order to use this application, it is a requirement that the user be an employee or apart of the company that the application is being used in. This is enforced as the application will only be able to be downloaded by employees they will each be sent an APK file. Should any user register for the application that is not an employee or apart of the company their account will be deleted and/or blocked.

**Android Device:** It is essential that the user has an android device as the application will not work on IOS devices just android. The device must also have a GPS this will be needed for the user's location.

**Internet Access:** Internet access is required for a number of functionalities within the application to work properly.

### 2.1.4 Environmental Requirements

This section will discuss the environmental requirements for my application. Without these developments of my application would not be possible.

**Android Studio:** I used Android Studios to create and code my application. Each page is customised by the use of layout pages that are code through XML. The main pages that consist of the functionality are coded in Java language.

**Android Device:** I used an Android device to run the application this allowed me to see my application and see how it functions on a real device.

**Firebase Console:** I implemented firebase in my application and connected it to the firebase console. Firebase offers a number of functionalities and services, I have used a number of theses, Authentication, push notifications, database and crash reporting. Without all of theses a number of the functions that I have in my application would not be possible.

**Windows Laptop:** I used a HP laptop with Android studios installed on it to develop my application.

**Internet Access:** The application requires internet access for a number of the functionalities to work properly. This was required to allow me to test the application properly.

### 2.1.5 Usability Requirements

Usability requirements will aid in the overall design of the application and the interface. This allows you to not stray from what you have decided is a usability requirement. The requirements may change slightly due to what the target users require. My usability requirements altered slightly after speaking with the target users.

**Ease of Use:** The application must be easy to use as if it is not users will not want to use it. The application must also be easy to understand and allow the user to easily navigate through the different functionalities that it offers.

**Design:** The application should have a good design and a set colour scheme throughout the application. The layout of the application will be easy on the eye and easy to read the information being displayed. Through the design, I hope that it will appeal to the user and enhance their experience of the application.

**Operating:** The system should perform all the actions it is required to do error free. Should it encounter an error it should display what it is.

## 2.2 Design and Architecture

My application is built mainly using XML and Java in Android studio. I also used Google maps API and a lot of the functions from the Firebase console. With Google maps API, it allowed me to create a map with custom locations. The Firebase console allowed me to implement functionalities such as login authentication, database storage and push notifications within my application. There is an ease of transition between all the pages in the application and a set design is carried out throughout the entire application. This is done through a set colour scheme, text, and layout.

## 2.3 Implementation

I will discuss some important sections of my application and provide some important code for the functions discussed.

### 2.3.1 Our Depots Map Page and Functions

I wanted to create a map within my application I used Google API's key to allow this to be possible (See Reference 1). I added the key I was given into my application and I was then able to add what zoom level the map would be displayed at (See Reference 6). As well as add custom markers and images to show certain markers on the map itself. I choose to use a box image to display the marker on the map. The code snippet for the maps page shows the markers location that you have chosen in latitude and longitude. The marker created must then have the code to be displayed on the map. This code allows you to decide what image to display the marker as if you wish. As well as what the marker is called and what text the user will see when they click the marker. For this Marker, it is called Depot 1 and it will display Depot 1 as the text when the user clicks the box image as well.

```
LatLng Depot1 = new LatLng (53.3266191, -6.3345646);  
mMap.addMarker(new MarkerOptions().position(Depot1).title("Depot 1")  
.icon(BitmapDescriptorFactory.fromResource(R.drawable.box)));  
mMap.moveCamera(CameraUpdateFactory.newLatLng(Depot1));
```

### 2.3.2 Sign In Page and Functions

The sign in page uses Firebase authentication this ensures the user is signing in with the correct information they signed up with. When the application is opened the login page checks if a user is signed in if they are already signed in it loads the main menu. If there is no user signed in it launches the login page.

```
auth = FirebaseAuth.getInstance();  
if (auth.getCurrentUser() !=null){  
startActivity(new Intent(Login.this,MainMenu.class));  
finish(); }
```

The user's password must also be a minimum of seven characters long. If it is shorter than this it will not allow them to use that password to sign up for the application. In the login page once the user clicks the sign in button it checks the email and password are valid. As well as the password length, as if it is smaller than seven characters it is not a valid password. If the information is incorrect it will show the user a message saying authentication failed and please check email and password and try again. If the sign in is successful it loads the main menu page.

```
if (!task.isSuccessful()) {  
if (password.length() < 7) {  
inputPassword.setError("Password too short, Password must be minimum of 7  
characters!");  
} else {  
Toast.makeText(Login.this,("Authentication failed, please check Email and Password  
and try again!"), Toast.LENGTH_LONG).show();  
} else {  
Intent intent = new Intent(Login.this, MainMenu.class);  
startActivity(intent);  
finish(); }  
}
```

### 2.3.3 Group Chat Page and Functions

The group chat allows users to communicate by messaging each other all in the one group chat. Once the group chat button is clicked the system checks that a user is signed in and then gets the email address they are signed in as. This is vital as the email address is used to distinguish what user sent what message. The code snippet below shows this.

```
if (FirebaseAuth.getInstance().getCurrentUser() == null) {  
} else {  
Toast.makeText(this, "Welcome" +  
FirebaseAuth.getInstance().getCurrentUser().getEmail(), Toast.LENGTH_LONG).show();  
}
```

Once the system has got this information it will then load the group chat page. A greeting message will also appear for the user saying welcome alongside their email address. All the messages that have been previously sent will then be visible on the screen. The messages are displayed in a list view and are all stored on a Firebase Database. The code snippet below shows the code for the information stored in the list view and the connection for the firebase database.

```
private void displayMessage() {  
ListView textList = (ListView) findViewById(R.id.textList);  
adapter = new FirebaseListAdapter<Message>(this, Message.class, R.layout.item,  
FirebaseDatabase.getInstance().getReference()) {
```

```

@Override
protected void populateView(View v, Message model, int position) {
    TextView text = (TextView) v.findViewById(R.id.text);
    TextView time = (TextView) v.findViewById(R.id.time);
    TextView username = (TextView) v.findViewById(R.id.username);
    text.setText(model.getText());
    username.setText(model.getUsername());
    time.setText(DateFormat.format("dd-MM-yy (HH:mm)", model.getTime())); }
}; textList.setAdapter(adapter);

```

The text list is the XML layout for the message list items this allowed me to show what order they appeared and what position on the screen the information was displayed. The time is displayed in the format for example 07-05-17 and the time is show as the hour and minute. This is displayed to the right of the username and it shows what time the message was sent at.

### 2.3.4 Our Staff Page and Functions

The our staff page displays a number of the staff alongside their name and role and what depot they are from. The information is displayed in a recycler view and each item is in its own card view within the recycler view (See Reference 9). The information that is stored in the Employee view holder is what every item in the recycler view must have. I have four pieces of information that each user must have this is job, name, depot and image. Three of which are text and the other is an image resource. The card view is within the employee view holder and stores the information for each employee item in its own individual card view. In the code snippet below it shows the employee view holder being declared with the RecyclerView adapter and what fields of information each employee must contain.

```

EmployeeViewHolder(View itemView)
{ super(itemView);
    cView=(CardView)itemView.findViewById(R.id.cView);
    employeeJob= (TextView)itemView.findViewById(R.id.EJob);
    employeeName =(TextView)itemView.findViewById(R.id.EName);
    employeeDepot = (TextView)itemView.findViewById(R.id.EDepot);
    employeePhoto = (ImageView)itemView.findViewById(R.id.EPhoto);
} }

```

The information for each employee is stored in an array list each employee must have a value for each of the required field. As all the fields bar the employee image is a text field they are all strings and must have text within them. The image field must reference a drawable resource. Below is a code snippet showing the array list and three of the employees added to that array and their information.

```

employees = new ArrayList<>();
employees.add(new Employee("CEO", "David Green", "depot 1", R.drawable.male));
employees.add(new Employee("Director", "Emma Green", "depot 1 & 2",

```

```
R.drawable.female));  
employees.add(new Employee("Manager", "John Ward","depot 1", R.drawable.male));
```

### 2.3.5 Important Numbers Page and Functions

The Important numbers page allows the user to click a call button from the important numbers page. It then opens the dial pad on the phone with a number inputted in it. The number inputted in the dial pad is associated to the button that was clicked. Each button has a different number associated with it. The code snippet below shows the function that happens when the call button 1 is clicked and what number is associated with it.

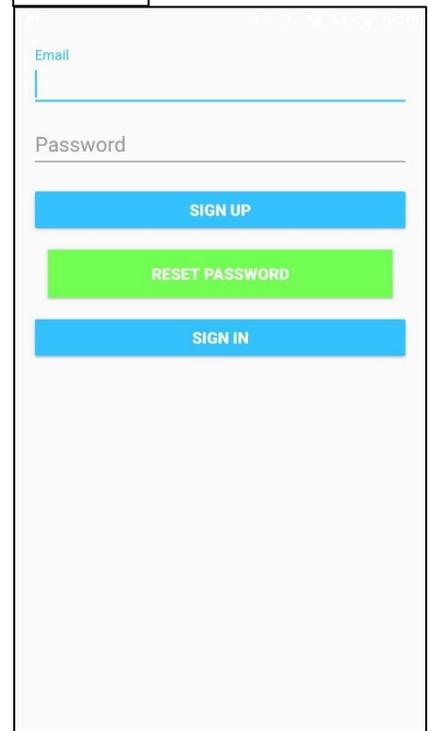
```
CallBtn1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent callIntent = new Intent(Intent.ACTION_DIAL);  
        callIntent.setData(Uri.parse("tel:014498500"));
```

## 2.4 Graphical User Interface (GUI) Layout

There are several pages that are important in my application each for different reasons. Below I will show screenshots of each page and have a brief explanation of each alongside it.

**Sign up Page:** The sign-up page is visible in Figure 1. It allows users to sign up for the application. Once the user has signed up once they will not need to use the sign in page again. The Firebase checks that the password meets the set requirements so a minimum of seven characters. It also checks the email is valid and not already in use by another user. Once both the email and password is valid it will allow the user to be signed up to the system. There details will then be stored in the authentication page of the firebase console.

**Figure1**



**Figure2**

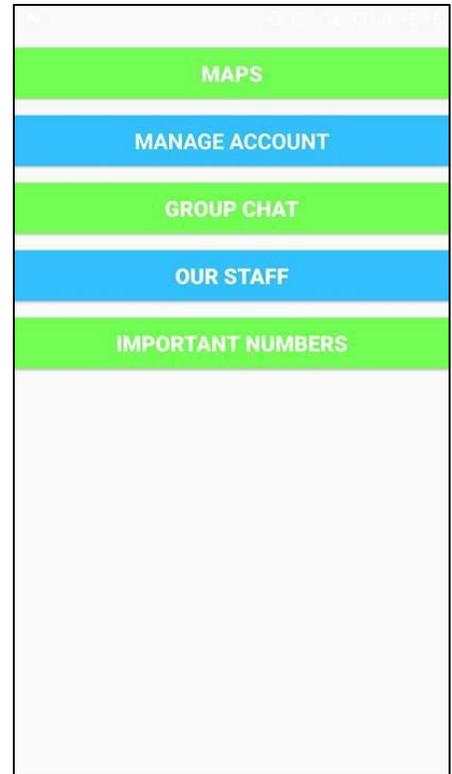
The form consists of two input fields: 'Email' and 'Password'. Below the fields are three buttons: 'SIGN IN' (blue), 'SIGN UP' (green), and 'RESET PASSWORD' (blue). The form is enclosed in a black border.

**Sign in Page:** The sign in page visible in Figure 2. Allows the user to sign in to the application. The sign in page allows the user to sign in to access all of the functionalities of the application without signing in the user can not see any functionalities other than sign up or reset password.

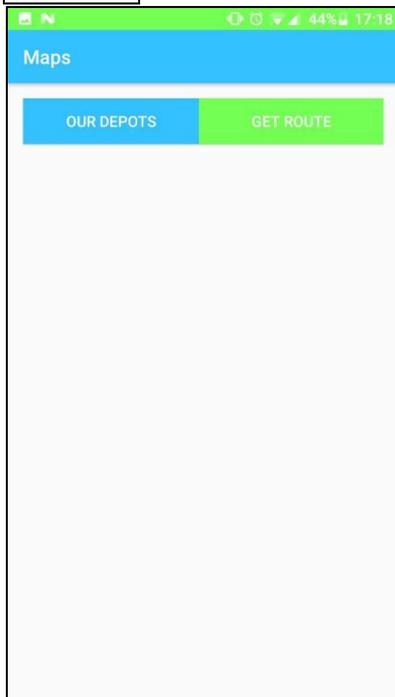
**Figure3**

**Main Menu Page:** The Main menu page is visible in Figure 3. This is the main navigation page of the here the user can navigate to a number of functionalities. It is essential as it shows the user all the main pages they have access to.

Once the user has successfully signed in it is then brought to the main navigation page. This page shows some of the options that the users can avail of, such as manage their account, maps, group chat and the staff page. The navigation page is essential as it allows the user to easily go to the page they require.



**Figure4**



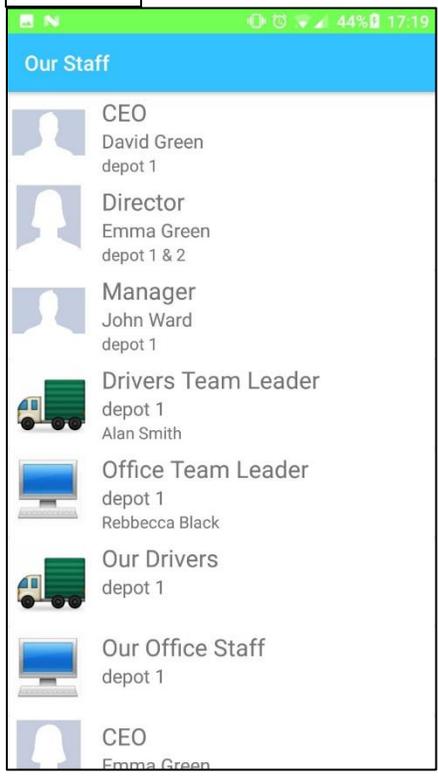
**Maps Page:** The maps page is visible in Figure 4. It displays two buttons, each with a different function. The get route button opens google maps through an intent.

**Figure5**

**Our Depot Page:** The Our depot page is visible is Figure 5. It Display a map with six depot locations and the user’s current location. As you can see the depots are displayed with a parcel image. A blue dot displays the user’s current location.

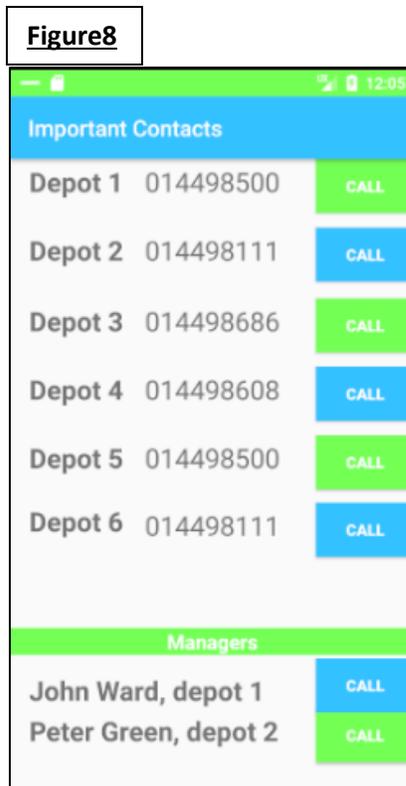
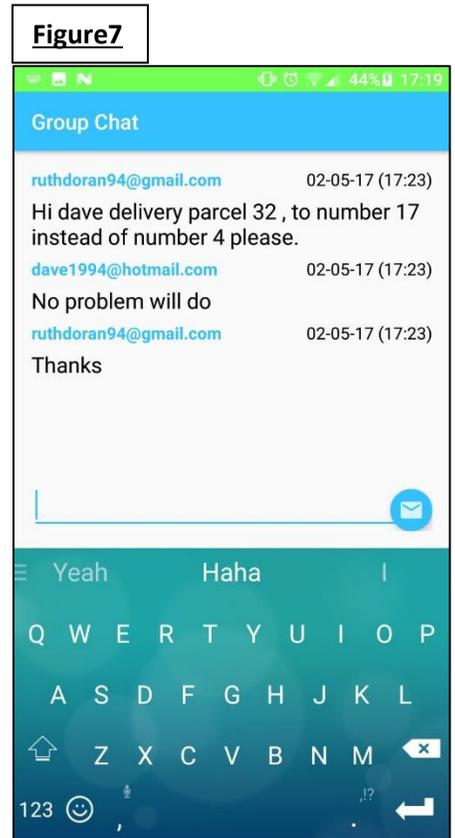


**Figure6**



**Our Staff Page:** Our staff page is visible in Figure 6. It displays the staff from every depot in a recycler view. It shows the role alongside a photo and the person’s name. The recycler view is used in android to create lists and display a number of card views, all on the one page in a list, it is much more complex than a standard list view. See more information on recycler view in reference 9.

**Group Chat Page:** The group chat page is visible in Figure 7. It allows the users to communicate through messaging. It allows every user that is signed in to write their own message and send it to the group. All the messages sent are stored in a database, and then displayed every time the chat is loaded. The messages can be deleted or changed through the database section where they are stored in the firebase console. The chat page checks that the user is signed in and with what email address they are signed is as. When the user clicks the envelope button to send the message the email address they are signed is as is then displayed above the message alongside the date and time it was sent at.



**Important Numbers Page:** This page is visible in Figure 8. It allows you to call some of the important numbers that employees may need. The user simply clicks the call button beside the person they want to ring. It will then open the dial pad with the number inputted already. It then allows the user to ring the number or store it if they wish. This data is coded directly in the application and the numbers are assigned to the buttons. The developer can easily change the number or add a number if necessary.

**Manage Account Page:** This page is visible in Figure 9. It allows the user to manage their account. From this page, the user can:

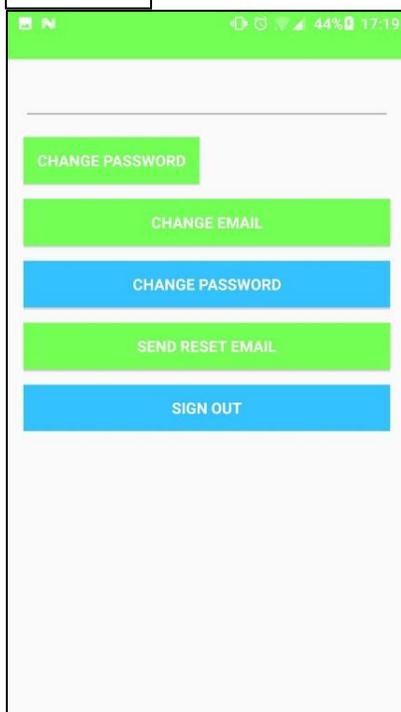
- Change their email address (Figure 10)
- Change their password (Figure 11)
- Send an email to their account to reset their password (Figure 12)

Below is then showing the GUI's of the functions the buttons offer once clicked.

**Figure9**



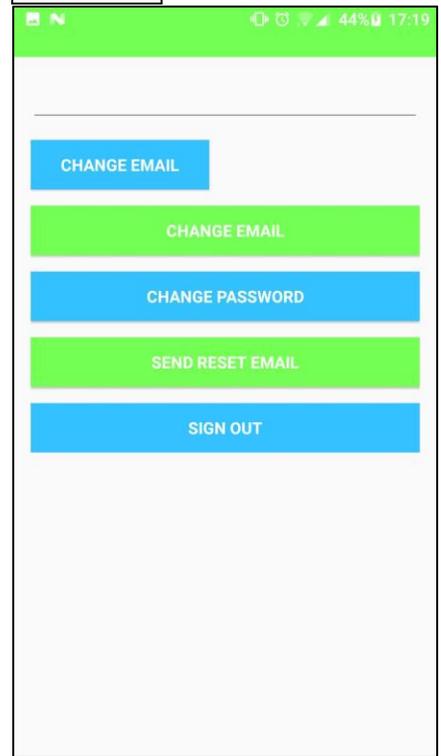
**Figure10**



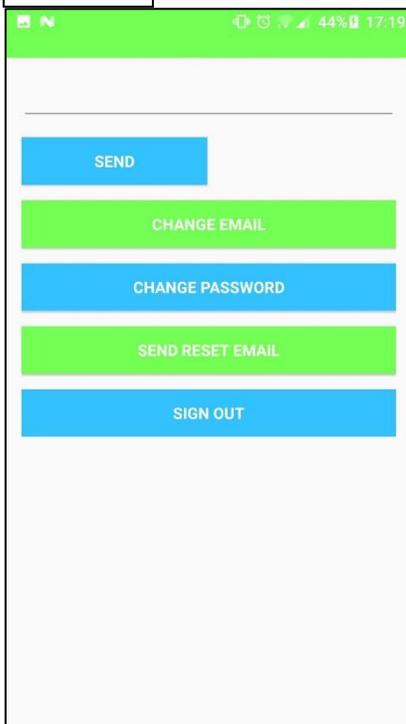
**Change Password Function:** This function is visible in Figure 10. It shows the function that the change password button performs. It allows the user to enter a new password into the text box and then click the change password button. Their password will then be changed, if the information provided is valid.

**Change Email Function:** This function is visible in Figure 11. It shows the function that the change email button performs. It allows the user to enter a new email address into the text box and then click the change email button. Their email address will then be changed, if the information provided is valid.

**Figure11**

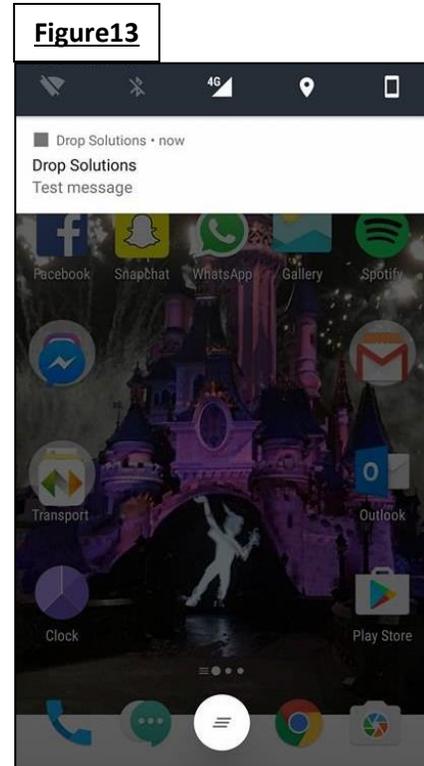


**Figure12**



**Change Password by Reset Email:** This function is visible in Figure 12. It shows the function that the send reset email button performs. It allows the user to enter their email address into the text box and then click the send button. The user will then receive an email and can change their password through the link provided in the email. This function is made possible through the use of the Firebase console with the use of authentication. You enable password reset by email and can customise certain parts of the email the user will receive. Once the button is sent the Firebase console receives the email address that wants the reset email. It then sends the email reset to that email address with a link for the user to click to allow them to change the password.

**Push Notification:** This is visible in Figure 13. It shows the function that allows the users to see a message that an admin has created and sent using the firebase console. It allows users to receive important messages in real time. The message is displayed as a notification on the user's phone. As the user has installed the application and since the application is connected to Firebase. They then just need to include FCM dependency in the gradle file of the application, once this is set up push notifications are able to be sent. By simply going to the notifications section of the firebase console.



## 2.5 Testing

Testing of this application was continuous through the whole creating process. I then followed the normal development process. I did this by running the application on the emulator within Android studios and then running it on a real device as well. Anytime I encountered an error I would fix it straight away if possible, if this wasn't possible I did more research and then fixed the error following what I had learned. I found it very beneficial to constantly test my application as it allowed me to develop certain functions fully. Once the function was working I knew that when I added a new function if there were any problems it would be in that, making the error easier to locate and fix.

This constant testing allowed me to encounter less errors in the future as if I waited and ran the application after adding multiple pages and functions I would not know where the errors were and there could have been multiple. In contrast to the one or two errors that I came across after adding a new function and testing it straight away.

Android studios also allowed me to view some of the errors when they were encountered in the Android monitor. The Android monitor would allow me to see where the application crashed and some of the reasons why It may have crashed. This is a built-in testing tool that Android studio offers. The message gradle build also allowed me to see if my application had any errors or warnings in it. As well as how quick the build time of the application is.

As my application uses firebase there is crash reporting setup for my application. I receive emails should an error occur within the application when it is running, it allows me to see

what the error is and then fix it. The crash report section in the firebase console allows me to see the issues that have occurred overtime and how many users were impacted by the error. The crash reporting page on the firebase console allows you to see the list of all the crash issues the application has ever encountered since being connected to the application (See Appendix 6.4.1). It also shows a graph of the trends showing the instances, users impacted and issues encountered since being connected to the application (See Appendix 6.4.2). This was another help for continuous testing of my application and ensuring its error free.

The Firebase test lab is able to be used from the firebase console. It allows you to choose a test type from two types I chose the Robo test. The Robo test allows you to test your application on a number of devices and it reports any crashes that occur allowing you to fix them. You upload you're APK for the application you wish to test. You then choose from the virtual and physical devices and what API level should be used to carry out the test on each device chosen. For my application, the API level required is 25 due to this there was only six devices that I could choose to do the Robo test on.

The test is limited to four devices per tests. You can then choose the run time of the test the default is five minutes, it also allows you to choose what is the deepest level that it will traverse to within the app the default is 50. Then you click run test and then receive the results for each of the devices you chose to test. It shows the devices I chose to test the application on as well as the results I received from them (See Appendix 6.4.3). When I thought, my application was nearly complete I tested it I received a few errors as the application failed. I found and fixed the errors and ran the test again and passed the test on every device I tested it on. I found this extremely helpful for the overall testing as it provided extremely helpful feedback on how the application runs on other devices as it allowed me to test on a number of devices.

## **2.6 Customer Testing**

I implemented my application into one of the depots of my family friends company, this allowed me to gain valuable feedback from the company. The application was tested in all sections of the depot so the drivers, office staff and managers all had access to the application. They agreed to test the application for a week this gave them enough time to get use to the application and rate the functionality. I then employed a survey on the final day the application was in use for the users to evaluate the application and provide me with feedback. I created a simple six question survey with a simple layout and design. I gained a lot of information on testing through my usability design module I used this in creating my customer testing survey. I used survey monkey to make the survey and it allowed me to see all the responses from the user. I made the survey six questions long as I did not want too many questions as it would discourage the testers from taking part in the survey. I wanted the survey to be to the point while still providing me with vital user feedback. I decided on six questions they are:

1. Did you find this application useful?
2. Would you use this application again?
3. What was best about this application and where needs improvements?
4. Would you recommend this application to other companies?
5. What feature did you find most useful?
6. Is there any changes or functions you would want to add to the application?

Question 1,2 and 4 are simply yes or no options, the user can only click one response. For questions 3,5 and 6 the user is provided a comment box for their answer allowing them to type their response. From the yes or no questions, I was able to view what percentage of the users tested selected yes and what percent selected no. Overall I received mostly positive results with 91.67% of users that completed the survey answer yes to Q1 (See Appendix 6.4.4.1) and 83.33% of users selected yes to Q2 (See Appendix 6.4.4.2). While Q4 received the lowest percentage of 75% which is still quite high overall (See Appendix 6.4.4.3).

The questions with the comment box provide different answers as the user input the text themselves this means it is not possible to get an accurate percentage. As of this I will show some of the responses to questions 3,4 and 5 (See Appendix 6.4.4.4, 6.4.4.5 and 6.4.4.6).

I have also included the full responses of six of the testers. These include both positive and negative responses that I received from the survey (See Appendix 6.4.5).

### **3 Conclusion**

When making an application it is inevitable for things to change along the way, such as functions, this is due to restrictions such as time constraints or limits within the application itself.

One of the main limits that I ran into with my application was the maps function, unfortunately I was unable to create what I had initially planned. This limit let me think of other problems that users may encounter. It ended up being an advantage, as I had to rethink, what is the other main problem delivery company's encounter? I decided it was communication. I added functionalities to fix the communication problem which was much more functionalities then I had initially planned. With testing and receiving user feedback I was then able to suit the application to the target market user's needs. This allowed me to create a well-rounded application with several functionalities that aid the user in several different ways.

Although I did run into a lot of limits and problems with my application this allowed me to create better functionality and fix more problems than I had planned initially. I personally feel that the application I have created is better then what I had originally planned. I feel the limits and problems that I faced made me think of a better solution to different problems, because of this I

think I have made a more unique application overall. I am very happy with my application that I have created and I hope it will fix what I have set out for it to do.

## 4 Further Development or research

There is a lot more development possible with this application. The application can be slightly modified depending on the company and then launched once it meets their adjusted needs. With more time and research, I feel it would have been possible to do a lot more work such as.

- The maps page in terms of getting turn by turn navigation functioning within the application and have a lot more map functionality overall. Such as allowing more than ten stops which is google maps limit. Due to the time constraints, it was not possible to create a map that was equal to or better than google maps that allowed the same functionality this is why I had to change it. I feel with more time and research I could have fixed this and had a map embedded in the application that allowed turn by turn navigation and calculating route time. That would be the main feature that I would have liked to fix had I had more time.
- I would also have liked to have the application running on more devices such as IOS and windows as it would appeal to a larger market.
- I would have liked to include more functionality for example a barcode or QR scanner to allow the drivers scan packages and retrieve information on them. I feel this would be a huge benefit as a lot of the scanners are quite bulky and if it was possible to use a phone it would be much easier and portable.
- Another function that I would have like to have added to the application was a separate map in connection with Google Places API (See Reference 11). It would have allowed the user to see locations such as restaurants, shops and petrol stations closest to them in relation to their current location. I feel this would be a useful tool for the drivers should they be delivery in an unfamiliar area.
- If I fixed the maps function and added a barcode or a QR scanner I would then like to add a function that combines the two. I would need to do a lot more further research to make this possible. As well as research to see is there other applications or software that already has something like this implemented. I would like it to be possible that once the driver scans a package that address is then added to a map. This would then allow a map to be generated with all the addresses of a particular driver with all their deliveries for the day on it. Finally, it would then calculate the fastest route to all the addresses shown on the map.

The application has a large amount of room for growth therefore with more time and resources I feel that I could expand the application. I feel I could implement the functions I have stated above and possibly more. Which would then hopefully allow me to solve all the problems that a delivery company faces.

## 5 References

1. Google 2017. *Google Maps API* [Online]. Available from: <https://developers.google.com/maps/android/> [Accessed September 2016].
2. Stephenmcd 2015. *GoogleMaps/android-samples* [Online]. Available from: <https://github.com/googlemaps/android-samples> [Accessed 4/03/17].
3. Ravi Tamada. *Android Getting Started with Firebase- Login and Registration Authentication* [Online]. Available from: <http://www.androidhive.info/2016/06/android-getting-started-firebase-simple-login-registration-auth/> [Accessed 31/02/2017].
4. EDMT Dev 2016. *Android Studio Tutorial – Chat Application with Firebase* [Online]. Available from: <https://www.youtube.com/watch?v=Xn0tQHpmDnM&t=321s> [Accessed 8/4/17].
5. Anu Sahni 26/01/2017. *Advanced Mobile Application Development, Topic 1, Recyclerview1* [Online]. Available from: <https://moodle.ncirl.ie/course/view.php?id=2775> [Accessed 09/04/2017].
6. Google Developers 2017. *Camera and View* [Online]. Available from: <https://developers.google.com/maps/documentation/android-api/views> [Accessed September 2016]
7. Google Developers 2017. *Set up Firebase Realtime Database for Android* [Online]. Available from: <https://firebase.google.com/docs/database/android/start/> . [Accessed February 2017].
8. Google Developers 2017. *Authenticate with Firebase on Android Using a Custom Authentication System* [Online]. Available from: <https://firebase.google.com/docs/auth/android/custom-auth> . [Accessed January 2017].
9. Android Developers 2017. *Creating Lists and Cards* [Online]. Available from: <https://developer.android.com/training/material/lists-cards.html> [Accessed May 2017].
10. Google Developers 2017. *Send a Notification to a User Segment* [Online]. Available from: <https://firebase.google.com/docs/notifications/android/console-audience> . [Accessed 9<sup>th</sup> April 2017].
11. Google Developers 2017. *Places API for Android* [Online]. Available from: <https://developers.google.com/places/android-api/> . [Accessed February 2017].

## 6 Appendix

### 6.1 Project Proposal

#### 6.1.1 Objectives

The objective of my software project is to create an application that will assist a delivery driver to find the most efficient route to make all their delivery's. It will allow the driver to input a number of addresses into the application. Once the addresses have been entered it will generate the best route for the courier to take. It will choose the most efficient and fastest route to allow them to be more productive and not be slowed down by having to go back on an area they have already been to.

I feel it will increase productivity as it will allow drivers to take on more delivery's as they can accomplish them in a timelier manner with the aid of this application. I know there is a demand for this specific application as I am creating it for a client who is unable to find anything to fit his required needs on the market. There are other similar applications such as google maps that allows you to do this but my application will be more specific to the client's needs. I will do this by adding in different areas to make it best suited for the company such as allowing the drivers a login and letting the admin track the driver's locations at any given time. This will allow them to best predict the driver's arrival time for the customer and how long the route will take in total.

#### 6.1.2 Background

The reason I am creating this application is because I know there is a need for it as a family friend has previously approached me with the idea as he was unable to find anything like it in the market. He wanted an application that he could implement within his company that would allow the driver to input addresses and it would then calculate the most efficient route for them to complete all their deliveries that day. As well as fix other problems he encountered within his company.

This would then increase productivity and allow him to analyse the performance of each driver and see who he can give more deliveries to without increasing his route time drastically. Another feature I thought would help is to allow the admin account on the application access to all of the driver's locations, this would benefit as it will allow him to see how far into a route any driver is at any given time.

This application will also aide the company during busier periods as the route calculator can help more inexperienced drivers best navigate their routes that they would not be familiar with. It also helps management during busy times as they can evenly distribute the extra work through all the drivers without majorly increasing the drivers route time by using the application to aide them.

Although there are other applications out there in the market mine will differ as it is specific to the client's needs and will give functions that the other applications don't offer such as the drivers locations for the admin account.

### **6.1.3 Technical Approach**

I firstly will make the map and once the map is fully functioning I will then expand from there. I will then create the address input section which will calculate the distance from one address to the other and find the fastest route to go to each of the houses and make delivers. Following this I will then create the login page for the drivers and create a database to store their information such as username and password. Once all of this is working I will then begin to consider creating the admin account which will show the drivers location to allow him to see how far into their route they are.

I have researched my idea and although there are a few applications that are similar the majority are paid applications, with only a couple that are available for free. The most popular and free application that I found is google maps. But google maps allows you to first select a route and then you can add a destination to your route after first creating. My application will differ from this as you will input all the addresses first and then it will generate the route. Unlike google maps where you input one route first and then add the other stops after.

Another feature my application will have that makes it unique is its login for the administrator that will allow the admin to know the location of any given driver and find out how far into their route they are. Once these features are working I can then consider further expansion of the application.

### **6.1.4 Technical Details**

As we need to use a system that allows our supervisor access to our project work at all times I will use github to make this possible. I will also be using android studios and I wish to implement Android XML and Java. I want to create a mobile application for all android users.

The requirements needed to use this application will be an android smartphone with internet access (Wi-Fi or 4G), and you will need to be a driver or managing level of the company.

### **6.1.5 Evaluation**

I will evaluate my application by performing a number of tests. Such as Integration tests that allow me to ensure all components work well and are compatible working with each other. And unit testing which will validate components of my application to ensure they handle the output as the input correctly. For Functional testing, I will get the client to test the application as he can provide feedback as he runs the delivery company. I will also evaluate the system with the end user by implementing it into the company's local depot and getting all the drivers from that depot to test it with each of their routes for that week. This will give me great feedback as they will then be able to tell me what worked well, where needs improvement and what is missing if anything. These tests will ensure my application works to the best of its ability.



## 6.3 Monthly Journals

### 6.3.1 September Monthly Journal

#### **My Achievements**

This month, I was able to decide what my project would be. I choose to make an Nci mobile application, which will allow students ease of access to important college information and links all in the one place. Once I had finalised my idea I went about researching other applications that were similar or in the same market. I downloaded all of the similar applications to my phone to further exam them. I went through each seeing the different sections each had as well as the functionalities that were included and what was missing. This research helped as it gave me more ideas of what I want to include in my app and what does not work. It also allowed me to see a lot of different layouts and what looks best as well as what seems to work the best. Some of the apps seemed too overloaded with information and other ones seemed to lack any information at all which then made them lack a purpose. I feel the research helped me greatly and I will hopefully be able to make a very useful and functional app by using the feedback I have found from my research. I also have to do a presentation to ensure I can go ahead with my chosen Idea this will take place next week. If I am unfortunate and they deny my proposal of this application I will have to start from scratch and find a new idea and do more research once I've chosen. But hopefully I will be able to go ahead with my current idea as I feel I can do a lot of work on it and there is a demand for this application within the college.

#### **My Plan to follow**

Once I have received approval of my project I plan to begin writing my project proposal documentation. In this I will do further research and narrow down the requirements I will need in my app as well as what sections and functionalities. I will decide the approach I will take when creating my app as well as set out a plan to follow when creating it. I will also research what way I will implement my idea such as what software and language. I will also show how my idea and app will be different to what has been made before and how it will help Nci students.

## 6.3.2 October Monthly Journal

### **My Achievements**

This month, I did my project proposal pitch I was approved with the condition to include the recommendation's that were advised to me. Upon further reflection, I realised I would not be able to implement the added technical needs that were required for my idea to be approved because of this I decided to go with a different Idea. Now my chosen project is a delivery application to assist drivers find the fastest route to several different addresses to make their assigned delivery route faster and more efficient. Since choosing this Idea I have researched several applications to ensure the one I create is unique and suited to the client I am making it for. I have then written my project proposal for my idea and came up with a name for the application it will be called drop solutions. I have created a Gantt chart to allow me keep on track with the creation of my application as well as all the required documentation. I have also begun writing my project requirements as well as finding where to begin my project to create my project prototype. I have also been assigned my supervisor and have organised a meeting with him for guidance overall on the project and find out where he feels needs improvement, the meeting is on Tuesday 1st November.

### **My Plan to follow**

I must complete and upload my project requirements as well as begin coding my application. I also plan to schedule follow up meeting with my supervisor for continued guidance on the project. I plan to stick with my Gantt chart that I have created to ensure I keep on track with the continuous progress of my overall project and ensure I meet all required deadlines on time.

### 6.3.3 November Monthly Journal

#### **My Achievements**

This month, I have been working on my project requirements documents as well as beginning the initial stages of code for the project. I had to rearrange my meeting with Dominic due to work, I met Dominic the following Friday for our first meeting. We discussed overall the project and my plans for beginning. We then spoke about the requirements as I had the majority of the documents already completed I was able to ask very specific questions in the certain areas I was unsure of, such as class diagram's and a fourth and fifth functional requirement. Dominic had several suggestions for more functionality which I took on board and adapted to my project and then included in my requirements. We then discussed what API's were need for my project and the github requirement, he suggested using Google's API and getting the key as this can be a bit tricky to setup. He also suggested to get github connected and working with android studios to avoid conflict with it later in the project. After my meeting the following few days I spent on sourcing the key from Google for the maps as well as requesting a student github account as it allows your code to be stored privately instead of public. Once I had the key from Google and received the permissions from github I began connecting android studios to github once this was successful I created a general map page and inputted the key. I then coded to add a specific location as the marker on the map and have a certain zoom that the map would automatically load to once launched. I then created a login page for the application which will tell the user when they have inputted the incorrect password and if the password is incorrect. I have yet to setup a database because of this I have just created two generic usernames and passwords that are hard coded into the application for the time being. I will delete theses once the database of the usernames and passwords is created and added. I have now completed my requirement documentation and have sent it on to Dominic as he said he will review it and give feedback. I am currently staying on top of all the work that I must do and hope it stays like this.

#### **My Plan to follow**

Once I receive feedback on my requirements I will make the recommended changes and then upload. I then plan to continue coding and have a good basis of the project to present a functional prototype for the midpoint presentation in December. I also plan to ensure my project requirements will also adapt and change with my application to ensure they include the new functionalities I may include or different choices that have been made or added or taken away. After my requirements upload I will have no more documentation until just before the mid-point after uploading the report for the mid-point. I plan to focus just on the coding and the prototype. I have discussed with Dominic about what to show as the prototype I have chosen to either show the map and some of its features or the login page and some of its features. I will decide when I have done more work on each and have a better understanding of which one will give a better idea of the application and how much functionality each requires to be somewhat completed. I am also working on connecting a number of pages to be able to show somewhat of a skeleton of my final project for my prototype. This is what I will be working on from now until the mid-point.

### 6.3.4 December Monthly Journal

#### My Achievements

This month, I have been focused on the prototype to present for the midpoint presentation. I have tried to add enough functionality to allow the lectures have a good understanding of where I want my project to go and the current stage I am at. I have been having a lot of difficulty at this stage of the project as the directional features have been very difficult. I have researched numerous tutorials and references to try resolve the problems I am facing with implementing them on the map page of the project. I have yet to find a solution but I am looking into the directions API matrix key once that is enabled a large amount of the problems I am having will be rectified. For the midpoint presentation, I found an alternative way to show a very basic map just in case the directions will not be completed by that stage. I have done this by creating markers hard coded into the map page of several different locations and then adding polylines to show the distance from one point to the next.

Although I am not happy with this as I thought I would of completed the directions feature more then I have at this point but due to the problems I encountered this was not feasible. I will continue to work on adding the API matrix directions to the application as it will allow me to calculate route time as well as distance and it will also consider traffic. I am also currently working on the database for where the login information is stored, I'm in very early stages as I have only created the database with some variables saved in each field and I am now focusing on how to connect the two.

I have also made my presentation slides for the midpoint presentation that is on the 20th of December. I have gotten in touch with my supervisor to arrange a meeting to discuss the issues I am currently having and see do they have any advice or tips that would be helpful, I am waiting to hear back. Although I have not made as much progress as I had hoped I feel this was due to the large workload throughout all the other modules we had this year which set me back on some of the prototype work as well as the complications I encountered. Due to all of this my applications is not currently where I would of hoped it would be but I am still happy with the amount of work and functionality I have made possible.

I had my presentation yesterday both my supervisor and the other lecture examining me gave me great feedback and thoughts on how best to proceed going forward. They gave me extra ideas for design purposes and making it easier for the user overall. They both decided that firebase database would be better then the other one I was currently looking into , due to this I will change my database to the fire base system.

### **My Plan to follow**

My plan to follow will be to continue work on the map and address input page to get them connected in a way that whatever addresses are inputted will then be shown as points on a map. I also want to get the directional functionality fixed and this will then allow me to have a much better map overall with all of the functionality it will then have. I also plan to work on the database for the login to connect what I have made with the application, as well as allow an admin to create new information for new users which will then be saved and stored in the same database. I have a lot of work to do but I feel overall I have a very clear plan of what needs to be done and what I want to achieve which makes me feel confident that this will all be made possible.

Due to the changes that were advised at the presentation I will need to do further research to allow me to best implement the required changes to the project. I plan to do this after the exams as I will be taking a break from my project over this time to best prepare and study for my exams. Once they are finished I will then continue the work on my application to fix all the problems I encountered and adding more functionality overall.

### 6.3.5 January Monthly Journal

#### **My Achievements**

This month, once we started back I arranged to meet my supervisor to discuss my feedback for my midpoint presentation to see where I did well and where I did not. Dominic discussed with me the areas I need to make better such as the prototype needed more functionality which I knew and documentation. I discussed the areas that I need his advice with as I am encountering problems with them. They are the google maps API directions and I have now decided to use firebase for the database and authentication process of my application due to the feedback I received from my midpoint presentation.

I have begun work on implementing firebase into my project as well as began the code for the following pages, sign in, sign up and password reset. I hope to finish this and have all the pages fully functional and integrated into the project. I will also work on my documentation to improve the areas I fell down in. I have been in touch with Dominic for some advice on the map page and to arrange another meeting.

Dominic has advised me to consider an alternative to googles direction feature as I am having trouble implementing it into my project. I will do some research and then arrange to meet Dominic to discuss the best option to follow considering all the research and his advice.

#### **My Plan to follow**

My plan to follow I will then begin to look at the map page and try and get the directions and calculate distance time for the route. I will further work on my documentation for the final submission. Once I have the firebase fully working with all the login and register aspects and the maps page I will then begin work on the smaller parts of the project. Such as location of each driver and the overall look and design of the project and any other aspects I need to add or fix. To make the project better overall.

## 6.3.6 February Monthly Journal

### **My Achievements**

This month, I have had meetings with Dominic and been in contact with him through email as well for advice on my project on certain areas. He has been extremely helpful and has given me great advice. I have continued to work on firebase login as well as further considering the maps. I have looked into alternative map API's then Google's. I still have to decide which would be best to go forward with, as others offer more functionality then google.

The firebase console is fully linked to my login activity on my application and I am nearly finished with it and it is almost fully functional. Once I have fully worked out the kinks in it will allow the user to login, sign out and reset passwords by email.

I have a large workload with my other modules as well so I am finding it difficult managing all of them. I have to distribute the work evenly between all of them as well as prioritising what needs to be done first such as tests etc. I haven't done as much work as planned but I plan to catch up on this over the next few weeks.

### **My Plan to follow**

My plan to follow will be to continue my work on the application overall as well as doing more work on the documentation due. I also plan to get my photo taken for the showcase as well as go to the seminar to help with documentation. I hope to have the firebase completed before next month and this will then allow me to further focus on fixing the errors I have with my maps. They are such as routing and timing.

### 6.3.7 March Monthly Journal

#### **My Achievements**

This month, I have met Dominic for more advice on my project and he has given me some ideas of what to add or what pages would look better over others. I have fully completed the firebase part of the application and it allows the user to register and login, change their email or password and sign out. It will also send an email to the user if they forgot the password to allow them to securely change it. It is also connected to the console that will allow an admin to login and control users such as delete a user etc. I have continued work on the maps page I have the location button working in the maps so it allows you to simply click it and the map will show where you are currently.

The maps directions features is not working with the map being embedded so I have decided to use the features that google maps has and have that through an intent this is also completed and fully functional within the map. This will allow the user to input addresses directly into google maps and then a turn by turn navigation route will be generated for all the addresses. Also by using this it will reroute the user if they take a wrong turn and calculate the route time. By using this it is fixing a lot of the errors I encountered with the embedded map.

I have also gotten my showcase photos take and completed my profile for the showcase, which will go into the booklet with all the other students. I have also done some parts of the documentation for my project.

#### **My Plan to follow**

My plan to follow will be to add extra map pages that give the user different functionality. Such as drop pins on a map and all the areas that a pin exists it generates a route to all the areas. I may also implement the places google API into an embedded map in the application which will allow users to see petrol stations nearby for example based on their current location.

Once the majority of the code is completed I plan to do all the documentation making the changes to what I have original wrote. This will allow me to have an extremely accurate description of the application overall including all the newest changes I have implemented.

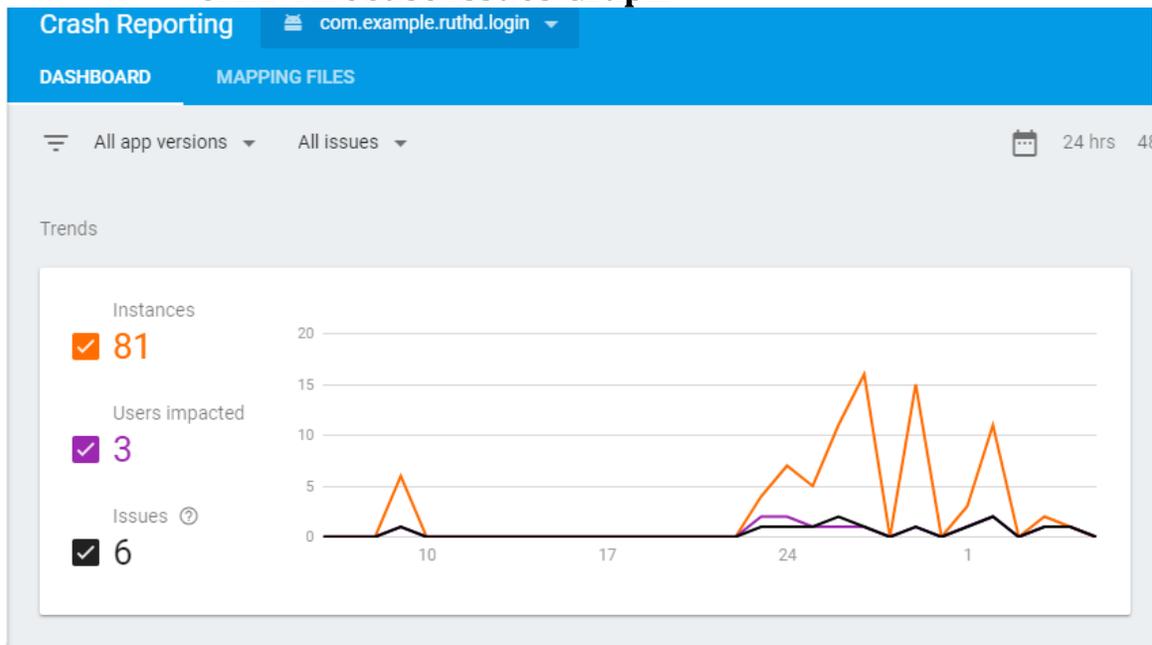
I have a number of functions I am looking into to add to the application but will only add them once the main features such as maps is fully complete as they are only minor functionality's in comparison and will take very little time to implement into the application.

## 6.4 Testing Materials

### 6.4.1 Firebase Issues List

Instances	Users	Versions ?	Issue	Stack trace
49	3	1	java.lang.IllegalStateException AppCompatActivity.java - Line 293 <b>Fatal</b>	android.support.v7.app.AppCompatActivity\$DeclaredOnClickL... android.view.View.performClick (View.java:5669)
16	1	1	java.lang.StackOverflowError MapsActivity.java - Line 26 <b>Fatal</b>	com.example.ruthd.login.MapsActivity.setZoomControlsEnabled (...) com.example.ruthd.login.MapsActivity.setZoomControlsEnabled (...)
9	2	1	java.lang.RuntimeException ActivityThread.java - Line 2646 <b>Fatal</b>	android.app.ActivityThread.performLaunchActivity (ActivityThr...) android.app.ActivityThread.handleLaunchActivity (ActivityThre...
3	1	1	java.lang.NullPointerException RVAdapter.java - Line 59 <b>Fatal</b>	com.example.ruthd.login.RVAdapter.onBindViewHolder (RVViewA...) com.example.ruthd.login.RVAdapter.onBindViewHolder (RVViewA...
2	1	1	java.lang.IllegalStateException AppCompatActivity.java - Line 327 <b>Fatal</b>	android.support.v7.app.AppCompatActivity\$DeclaredOnClickL... android.support.v7.app.AppCompatActivity\$DeclaredOnClickL...
2	1	1	java.lang.RuntimeException ActivityThread.java - Line 2728 <b>Fatal</b>	android.app.ActivityThread.performLaunchActivity (ActivityThr...) android.app.ActivityThread.handleLaunchActivity (ActivityThre...

### 6.4.2 Firebase Issues Graph



## 6.4.3 Firebase Test Lab Results

Test Lab > Drop Solutions

Matrix #475797

Robo test, Just now

Failed	Passed	Skipped	Inconclusive
0	2	0	0

Test execution	Duration	Locale	Orientation	Issues
Pixel, API Level 25	—	English, United States	Portrait	—
Nexus 9, Virtual, API Level 25	—	English, United States	Portrait	—

Test Lab > Drop Solutions

Matrix #783022

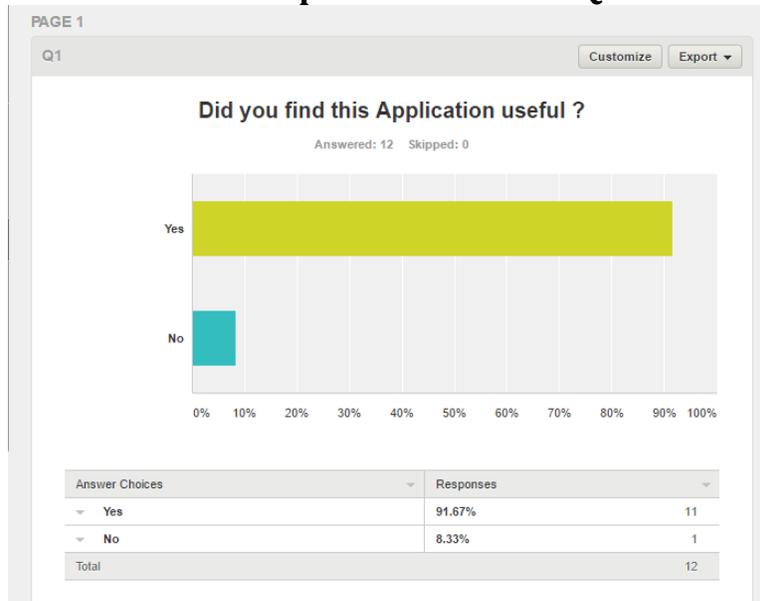
Robo test, Just now

Failed	Passed	Skipped	Inconclusive
0	4	0	0

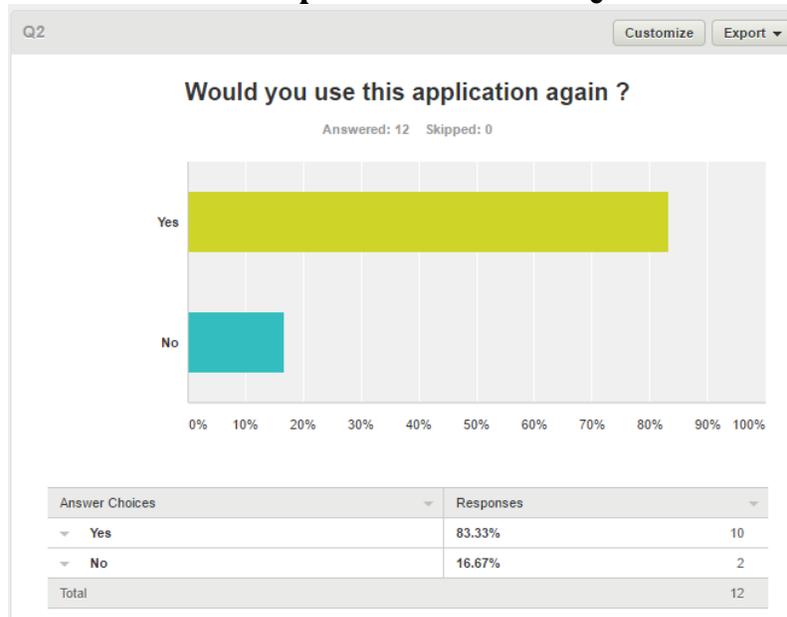
Test execution	Duration	Locale	Orientation	Issues
Low-resolution mdpi phone, Virtual, API Level 25	—	English, United States	Portrait	—
Nexus 6P, Virtual, API Level 25	—	English, United States	Portrait	—
Nexus 6, Virtual, API Level 25	—	English, United States	Portrait	—
Nexus 9, Virtual, API Level 25	—	English, United States	Portrait	—

## 6.4.4 Customer Testing Survey Results

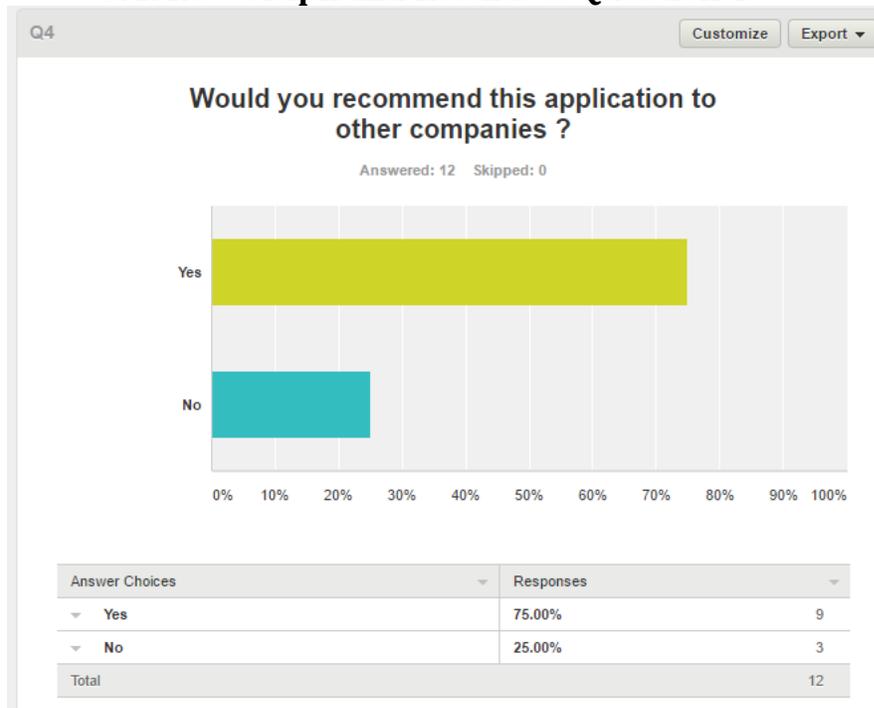
### 6.4.4.1 Graph and Results to Question 1



### 6.4.4.2 Graph and Results to Question 2



### 6.4.4.3 Graph and Results to Question 4



### 6.4.4.4 Responses to Question 3

Q3 Export

**What was best about this application and where needs improvement?**

Answered: 12 Skipped: 0

● Responses (12)
☁ Text Analysis
📁 My Categories

Categorize as... Filter by Category Search responses

Showing 12 responses

- the number of functionalities it offers  
 5/2/2017 6:40 PM [View respondent's answers](#)
- i like the list of staff and important numbers  
 5/2/2017 6:39 PM [View respondent's answers](#)
- the important numbers page  
 5/2/2017 6:37 PM [View respondent's answers](#)
- the maps were great, i like the whole application  
 5/2/2017 6:36 PM [View respondent's answers](#)
- i didnt have find it beneficial to me  
 5/2/2017 5:37 PM [View respondent's answers](#)
- improvements in the map routes would be great instead of using google maps  
 5/2/2017 5:36 PM [View respondent's answers](#)
- the management of account section was very good, nothing needed improvement i dont think  
 5/2/2017 5:34 PM [View respondent's answers](#)

### 6.4.4.5 Responses to Question 5

The screenshot shows a survey interface for Question 5. At the top, it says 'Q5' and 'Export'. The question is 'What feature did you find most useful?'. Below the question, it indicates 'Answered: 12 Skipped: 0'. There are three tabs: 'Responses (12)', 'Text Analysis', and 'My Categories'. Below the tabs are filters for 'Categorize as...', 'Filter by Category', and a search bar for 'Search responses'. The main area shows 'Showing 12 responses' with a list of seven entries, each with a text response, a timestamp, and a 'View respondent's answers' link.

Response	Timestamp	Action
i think the staff page is good as it s usefull for new employees	5/4/2017 2:44 PM	<a href="#">View respondent's answers</a>
the contacts page was great	5/4/2017 2:41 PM	<a href="#">View respondent's answers</a>
the group chat was very useful	5/3/2017 12:30 PM	<a href="#">View respondent's answers</a>
the call function	5/3/2017 12:27 PM	<a href="#">View respondent's answers</a>
the group chat	5/2/2017 6:40 PM	<a href="#">View respondent's answers</a>
the list of staff	5/2/2017 6:39 PM	<a href="#">View respondent's answers</a>
the chat and reset password features	5/2/2017 6:37 PM	<a href="#">View respondent's answers</a>

### 6.4.4.6 Responses to Question 6

The screenshot shows a survey interface for Question 6. At the top, it says 'Q6' and 'Export'. The question is 'Is there any changes or functions you would want to add to the application?'. Below the question, it indicates 'Answered: 12 Skipped: 0'. There are three tabs: 'Responses (12)', 'Text Analysis', and 'My Categories'. Below the tabs are filters for 'Categorize as...', 'Filter by Category', and a search bar for 'Search responses'. The main area shows 'Showing 12 responses' with a list of seven entries, each with a text response, a timestamp, and a 'View respondent's answers' link.

Response	Timestamp	Action
i feel it needs to allow more then ten locations for the user to get a route to a one time	5/4/2017 2:44 PM	<a href="#">View respondent's answers</a>
none that i can think of it has alot of functions in it already	5/4/2017 2:41 PM	<a href="#">View respondent's answers</a>
better maps and route generation	5/3/2017 12:30 PM	<a href="#">View respondent's answers</a>
.	5/3/2017 12:27 PM	<a href="#">View respondent's answers</a>
no	5/2/2017 6:40 PM	<a href="#">View respondent's answers</a>
none that i can think of	5/2/2017 6:39 PM	<a href="#">View respondent's answers</a>
maybe sending messages to single users as well	5/2/2017 6:37 PM	<a href="#">View respondent's answers</a>

# 6.4.5 Individual Customer Survey Responses

## Tester 2 Response

#2 **COMPLETE** Edit Delete Export

Collector: Web Link 1 (Web Link)  
Started: Tuesday, May 02, 2017 5:32:52 PM  
Last Modified: Tuesday, May 02, 2017 5:34:20 PM  
Time Spent: 00:01:28

PAGE 1

Q1: Did you find this Application useful ?  
Yes

Q2: Would you use this application again ?  
Yes

Q3: What was best about this application and where needs improvement?  
the management of account section was very good, nothing needed improvement i dont think

Q4: Would you recommend this application to other companies ?  
Yes

Q5: What feature did you find most useful ?  
i loved the group chat

Q6: Is there any changes or functions you would want to add to the application ?  
I wouldnt make any

## Tester 3 Response

#3 **COMPLETE** [Edit](#) [Delete](#) [Export](#)

Collector: Web Link 1 (Web Link)  
Started: Tuesday, May 02, 2017 5:34:37 PM  
Last Modified: Tuesday, May 02, 2017 5:35:37 PM  
Time Spent: 00:00:59

PAGE 1

Q1: Did you find this Application useful ?

Yes

Q2: Would you use this application again ?

Yes

Q3: What was best about this application and where needs improvement?

improvements in the map routes would be great instead of using google maps

Q4: Would you recommend this application to other companies ?

Yes

Q5: What feature did you find most useful ?

the manage account section

Q6: Is there any changes or functions you would want to add to the application ?

the maps features need improvement in terms of route generation

## Tester 5 Response

#5 **COMPLETE** [Edit](#) [Delete](#) [Export](#)

Collector: Web Link 1 (Web Link)  
Started: Tuesday, May 02, 2017 6:34:09 PM  
Last Modified: Tuesday, May 02, 2017 6:35:36 PM  
Time Spent: 00:01:26

PAGE 1

Q1: Did you find this Application useful ?

Yes

Q2: Would you use this application again ?

Yes

Q3: What was best about this application and where needs improvement?

the maps were great, i like the whole application

Q4: Would you recommend this application to other companies ?

Yes

Q5: What feature did you find most useful ?

the maps

Q6: Is there any changes or functions you would want to add to the application ?

no

## Tester 10 Response

#10 **COMPLETE** [Edit](#) [Delete](#) [Export](#)

 Collector: Web Link 1 (Web Link)  
Started: Wednesday, May 03, 2017 12:28:57 PM  
Last Modified: Wednesday, May 03, 2017 12:30:11 PM  
Time Spent: 00:01:14

PAGE 1

Q1: Did you find this Application useful ?

Yes

Q2: Would you use this application again ?

Yes

Q3: What was best about this application and where needs improvement?

more functionality

Q4: Would you recommend this application to other companies ?

No

Q5: What feature did you find most useful ?

the group chat was very useful

Q6: Is there any changes or functions you would want to add to the application ?

better maps and route generation

## Tester 11 Response

#11 **COMPLETE** [Edit](#) [Delete](#) [Export](#)

 Collector: Web Link 1 (Web Link)  
Started: Thursday, May 04, 2017 2:39:28 PM  
Last Modified: Thursday, May 04, 2017 2:40:51 PM  
Time Spent: 00:01:23

PAGE 1

Q1: Did you find this Application useful ?

Yes

Q2: Would you use this application again ?

Yes

Q3: What was best about this application and where needs improvement?

the group chat page, would be good if you could directly message one user as well

Q4: Would you recommend this application to other companies ?

Yes

Q5: What feature did you find most useful ?

the contacts page was great

Q6: Is there any changes or functions you would want to add to the application ?

none that i can think of it has alot of functions in it already

# Tester 12 Response

#12 **COMPLETE** Edit Delete Export

Collector: Web Link 1 (Web Link)  
Started: Thursday, May 04, 2017 2:41:05 PM  
Last Modified: Thursday, May 04, 2017 2:43:32 PM  
Time Spent: 00:02:27

PAGE 1

Q1: Did you find this Application useful ?  
No

Q2: Would you use this application again ?  
No

Q3: What was best about this application and where needs improvement?  
i didnt like it

Q4: Would you recommend this application to other companies ?  
No

Q5: What feature did you find most useful ?  
i think the staff page is good as it s usefull for new employees

Q6: Is there any changes or functions you would want to add to the application ?  
i feel it needs to allow more then ten locations for the user to get a route to a one time