# SFWR ENG 3A04 Project

## Cabpool

## Application Assessment

### Group 10

| | |
|---|---|
| Amanda Boudreau | 0344130 |
| Jordan Floyd | 1225123 |
| Mario Machado | 1211979 |
| Rakesh Mistry | 1221428 |
| Ali Reda | 1206754 |

December 6, 2014
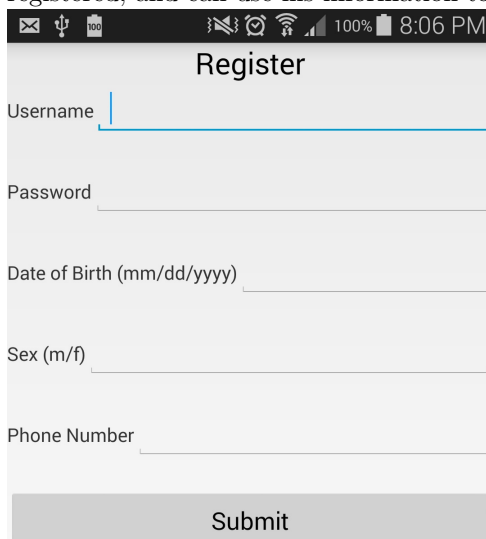
# Contents

# 1    Introduction

This document provides the reader with an assessment of the the Cabpool app. Section 1 provides the user with a guide to the use of the application. Section 2 provides an overview of the outcomes of the testing which is followed by an in depth description of the functionality testing the application underwent. Section 3 concludes with an assessment of the application and recommendations for future work.

# 2    User Guide

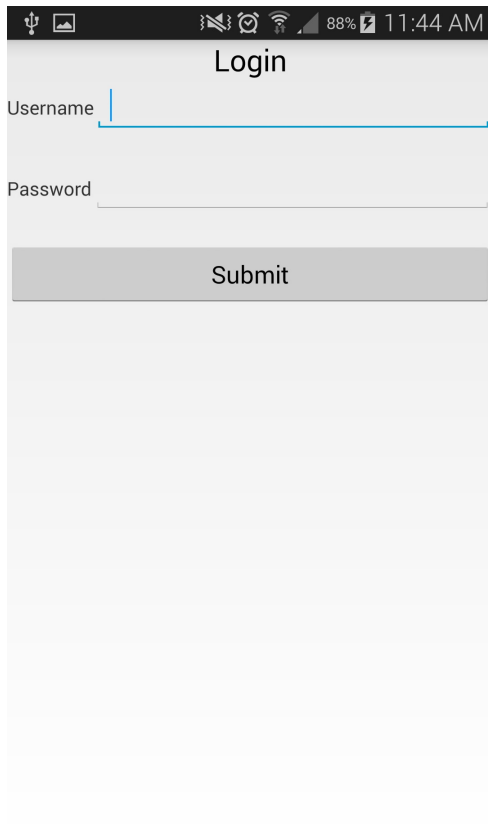This section will be a user guide for the application Cabpool.

## 2.1    How Do I Register

To register on the application Cabpool, enter the application. And you will have the option to "Register". Click the register button. This will prompt a new page where you must enter needed credentials. You will be asked to enter your wanted user name, a password, your date of birth, your sex, and you phone number. Assuming all the inputs are valid, the user must now submit. After the submission, the user is now registered, and can use his information to login in to the system.



## 2.2    How Do I Login

After registration, the user will now be able to login. To do so, simple click the login button on the main menu. This will take the user to the login screen. The user now will input the username and password he registered with. The user must then click submit. Assuming inputs are correct, the user is now logged in.

## 2.3 How Do I Enter the Settings Screen

To enter the settings screen simply go to the main menu and select the "settings" button. This will take you to the settings screen.
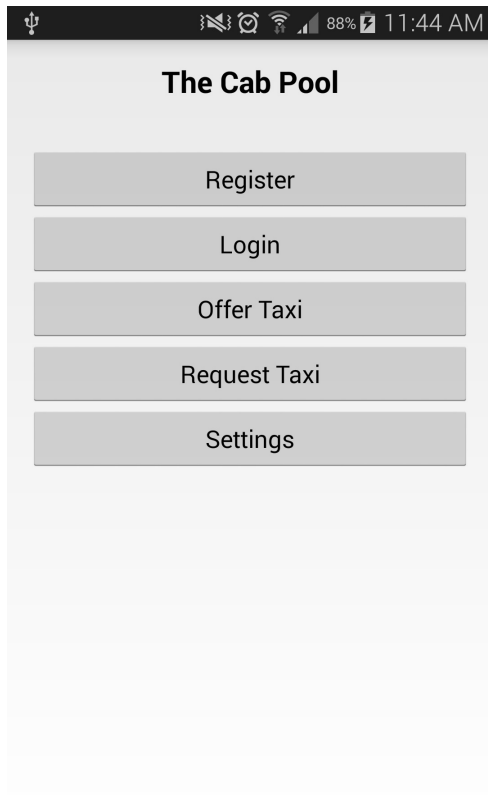
## 2.4 How Do I Add A Favourite Location

If a user wishes to add a favorite location He will first need to be logged in. After this, the user must select the setting button on the main menu. There he will have the option "Favourite Locations". The user must click that. On that page the user will be prompted to enter a location. Assuming the location exists, the user will enter the location and submit his input.

## 2.5 How Do I Add A Friend To Friends List

If a user wishes to add a friend to his friends list, he will first need to be logged in. After this, the user must select the setting button on the main menu. There he will have the option "Friends List". The user must click that. On that page the user will be prompted to enter a username. Assuming the username exists, the user will enter the username, and the user will be sent a friends Request. If a user wants to remove a friend, he enters the username and selects "Remove Friends" Button.
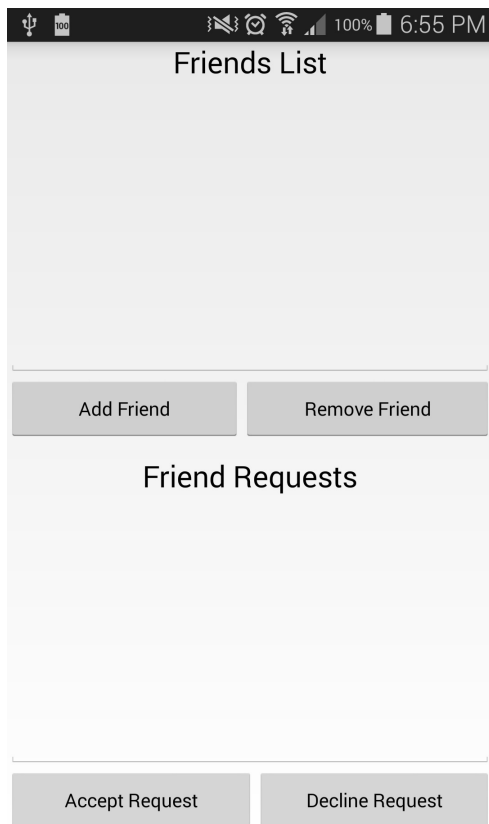
## 2.6 How Do I Accept A Friend Request

To accept a friend request, enter the Friends List screen on the settings screen. There you will have the option to add or remove a user to a friends list, or to accept existing incoming requests. The user has the choice to decline or accept the user that is requesting to be his friend.
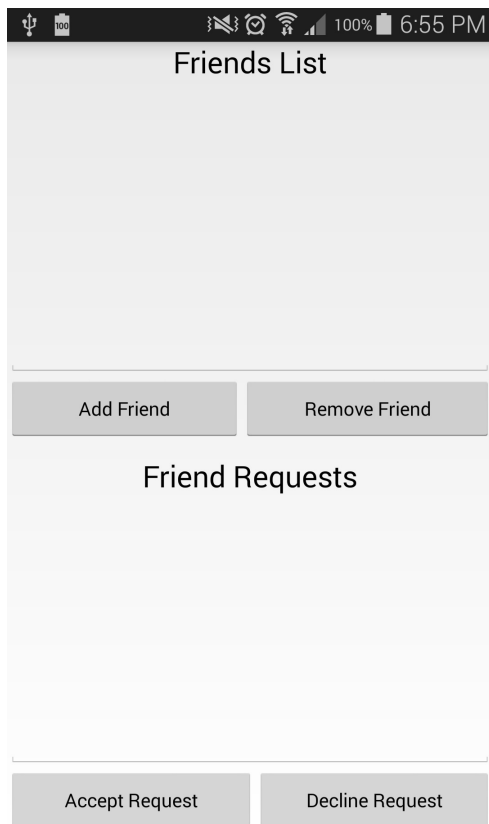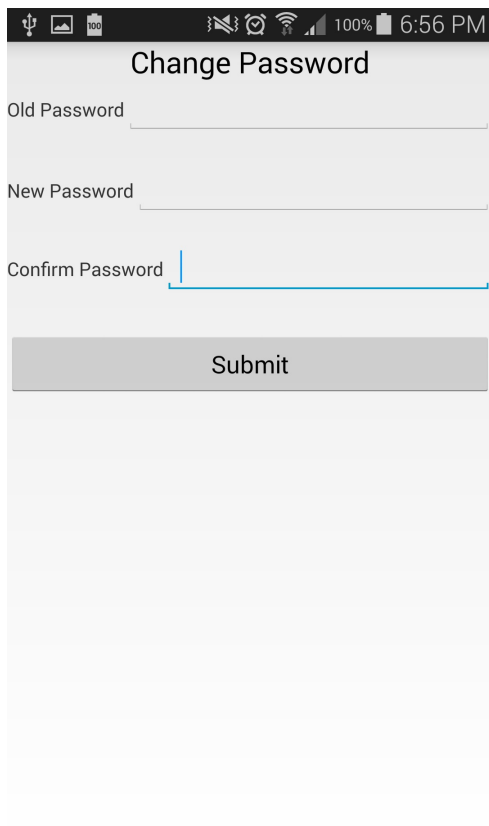
## 2.7 How Do I Add or Remove A User onto Blacklist

If a user wishes to add or remove a user to the blacklist, he will first need to be logged in. After this, the user must select the setting button on the main menu. There he will have the option "Blacklist". The user must click that. On that page the user will be prompted to enter a username. Assuming the username exists, the user will enter the username, and the user Will be blacklisted after submission. If the user wants to remove a user from his blacklist, He can select a blacklist user and remove from blacklist.
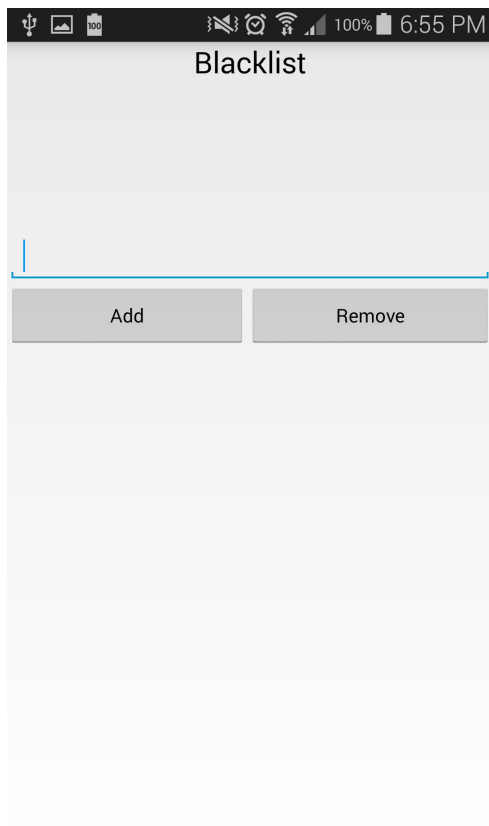
## 2.8   How Do I Change My Password

If a user wishes to change his password, He must select the Settings Screen, followed by the change password screen. There he will be prompted to enter his current password, and then his new password. The user must submit the changes for them to take effect.

## 2.9 How Do I Delete My Account

If i user wishes to be removed by the system and delete their account, he can do so in the settings screen. On the screen, the user can select the "Delete Account" button. If selected, this will tell the user that his account has ben deleted
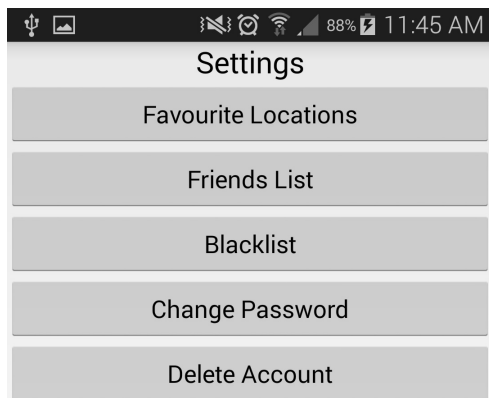
## 2.10 How Do I Offer a Taxi

If a user wishes to offer a taxi, he must select the offer taxi button. There the user will be capable of of offering his current route/location using the QR Code. The user can then submit is offer.

## 2.11 How Do I Request A Taxi

To request a taxi, the user must click the request taxi option in the main menu. The user must load the available offers, and choose one to request. On the other end, the user offering will recieve a confirmation.
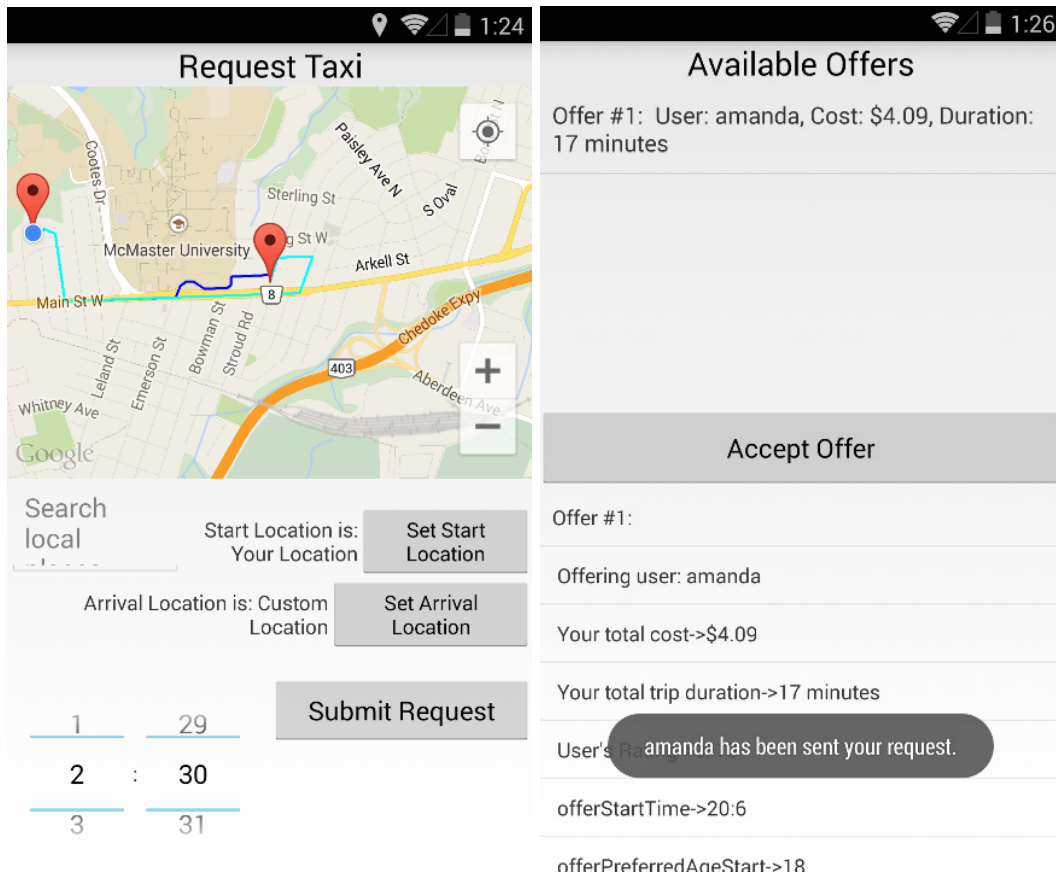
## 2.12   How Do I Accept an incoming request

If a user offered a taxi, and later on another user accepted that offer. The first user will get a message in the application asking him if he would like to accept or deny the incoming request
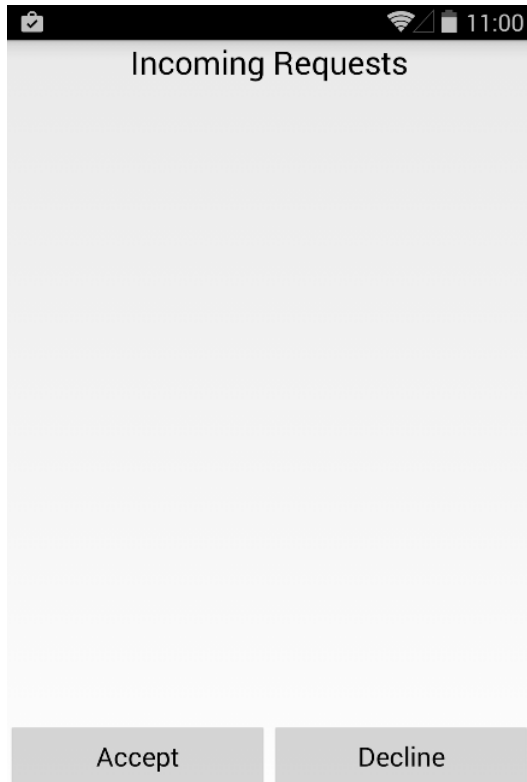
## 2.13 How Do I Scan A QR Code

Assuming the user wants to offer a taxi, and he is on the offer taxi screen. He will need to input a QR Code to show up in the requests on the point of view of the other users. To Do so, the user will have a "Scan Code" Button. The user must click that, and scan a valid QR Code. The user then selects "Submit Code". And after that, his offer will be sent out.

If user does not have the QR Code scanner, the application will prompt the user to download it. User will be redirected to the Play Store where he must download the QR CODE Application
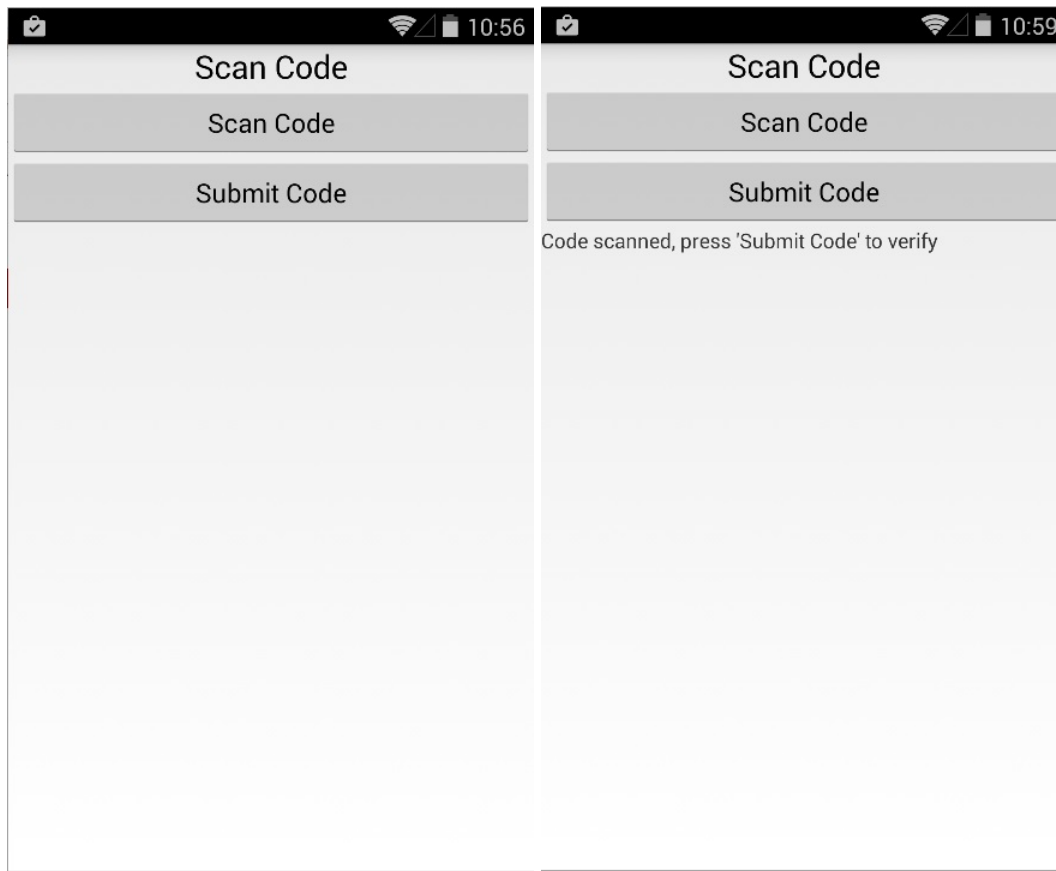
## 2.14 How Do I Cancel Offer

If a user wishes to cancel an offer he made he, we will assume he already has an offer. When a user has an existing offer he will have the option to "Cancel Offer" on the Offers page. User must submit his changes.

## 2.15 How Do I Accept An Offer

To accept an offer someone has made, you must view the search results in the request taxi screen. There you will see "available offers". From the list, you have an option to choose which one you would like. To confirm accpeting the offer, user must select "Accept Offer" Button. From the other users end, he will see a request success.

**Available Offers**

Offer #1:  User: rakesh, Cost: $13.89, Duration: 35 minutes

Offer #2:  User: MissFortune1, Cost: $29.05, Duration: 60 minutes

Offer #3:  User: JohnGoodBody, Cost: $17.76, Duration: 44 minutes

**Accept Offer**

Offer #1:

Offering user:rakesh

Your total cost->$13.89

Your total trip duration->35 minutes

offerStartTime->23:0

offerStartLocation->43.26087833333333 -79.91922500000001

offerCurrentLocation->43.26087833333333 -79.91922500000001

**Your Trip**

Request accepted!

## 2.16   How Do I Rate The Service

After you accept and offer, or your request is accept, and you use the service. You will have an option to "Rate" the service through the "rate" button after the confirmation page.
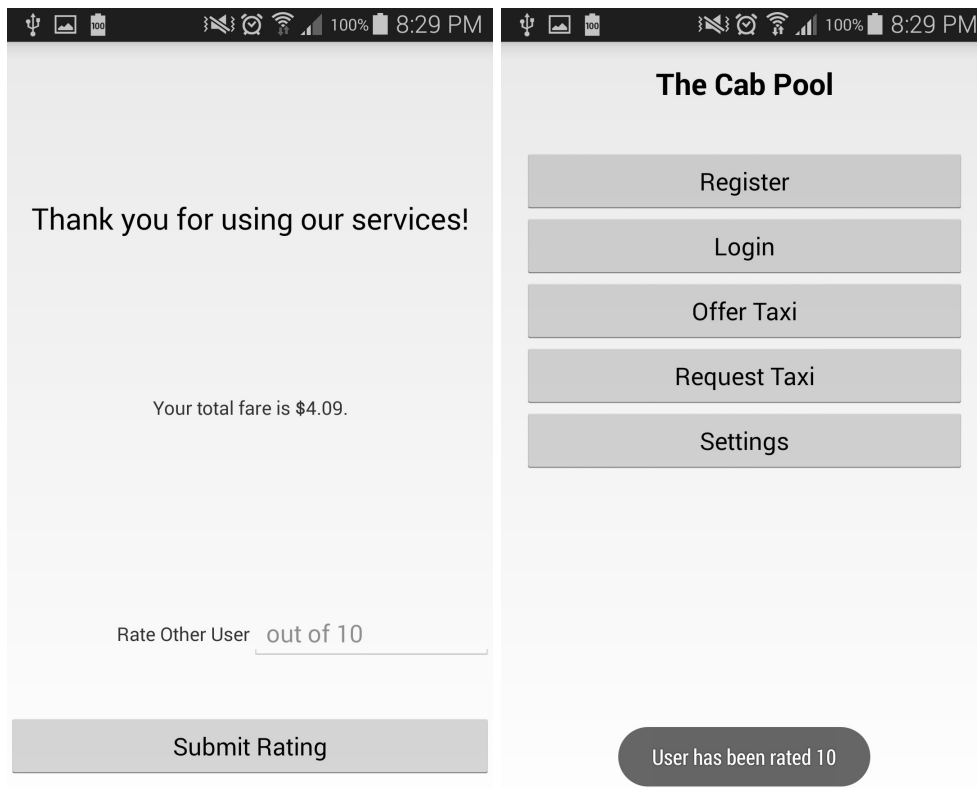
## 3 Testing

Performance of the application is smooth and seamless on an android phone, there are no actions that take a noticeably long time to register, the application launches quickly and interfaces well with Google Maps. We decided that black-box testing was sufficient for asserting the correct behaviour with respect to the requirements. Due to the nature of our application, testing the code itself (white-box testing) was not deemed necessary. All user interaction with the application is done through a UI and is very restricted. As seen in the following figure, with the exception of innovative features, all requirements were met. The application catches exception cases and displays appropriate error messages, hence we have determined our implementation to be successful. Our application is effective and bug free, having met all base requirements we are confident with additional time innovative features could easily be added to fulfill additional nonfunctional requirements.
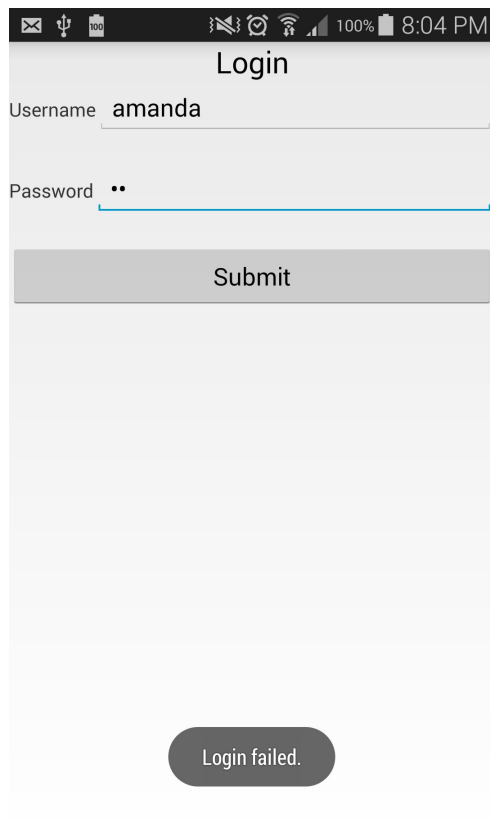
| Requirement | Description | Output | Requirement | Description | Output |
|---|---|---|---|---|---|
| VP 1.1.1 | Display registration | Pass | VP 4.4.1 | Protect Confidential Information | Pass |
| VP 1.1.2 | Request authentication information | Pass | VP 5.1.1 | Request a taxi | Pass |
| VP 1.2.1 | Verify authentication information | Pass | VP 5.1.2 | Request user information | Pass |
| VP 1.2.2 | Error message on invalid information | Pass | VP 5.1.3 | Notify of offers | Pass |
| VP 1.2.3 | Store authentication information | Pass | VP 5.1.4 | Reject match | Fail |
| VP 1.3.1 | Info not intercepted | Pass | VP 5.1.5 | Accept only one taxi | Pass |
| VP 1.4.1 | Protect Confidential Information | Pass | VP 5.2.1 | Match offering and requesting | Pass |
| VP 2.1.1 | Display Login | Pass | VP 5.3.1 | Info not intercepted | Pass |
| VP 2.1.2 | Request Login Credentials | Pass | VP 5.4.1 | Protect Confidential Information | Pass |
| VP 2.2.1 | Verify Login Credentials | Pass | VP 6.1.1 | Arrived and display fare | Pass |
| VP 2.2.2 | Notify User Logged In | Pass | VP 6.1.2 | Rate user | Pass |
| VP 2.3.1 | Info not intercepted | Pass | VP 6.1.3 | Add user to Friends List | Fail |
| VP 2.3.2 | Not access unauthorized | Pass | VP 6.1.4 | Add user to Blacklist | Fail |
| VP 2.4.1 | Protect Confidential Information | Pass | VP 6.2.1 | Calculate Fare and distance | Pass |
| VP 3.1.1 | Select Settings | Pass | VP 6.3.1 | Info not intercepted | Pass |
| VP 3.1.2 | Manage Friends List | Fail | VP 6.4.1 | Protect Confidential Information | Pass |
| VP 3.1.3 | Manage Blacklist | Fail | VP 7.1.1 | Logout | Fail |
| VP 3.1.4 | Manage Favourite Locations | Fail | VP 7.1.2 | Notify logout | Fail |
| VP 3.2.1 | Send Dispatcher Friends List changes | Fail | VP 7.2.1 | Terminate offers/requests | Fail |
| VP 3.2.2 | Send Dispatcher Blacklist changes | Fail | VP 7.2.2 | Notify of logout | Fail |
| VP 3.2.3 | Send Dispatcher Favourite changes | Fail | VP 7.3.1 | Info not intercepted | Pass |
| VP 3.3.1 | Info not intercepted | Pass | VP 7.3.2 | Notify of logout | Fail |
| VP 3.4.1 | Protect Confidential Information | Pass | VP 7.4.1 | Protect Confidential Information | Pass |
| VP 4.1.1 | Scan QR Code | Pass | VP 8.1.1 | Delete account | Pass |
| VP 4.1.2 | Request user information | Pass | VP 8.1.2 | Delete account successful notification | Pass |
| VP 4.1.3 | Notify of Match | Pass | VP 8.2.1 | Confirm delete | Pass |
| VP 4.1.4 | Accept or reject match | Pass | VP 8.2.2 | Notify of deletion | Pass |
| VP 4.2.1 | Match offering and requesting | Pass | VP 8.3.1 | Info not intercepted | Pass |
| VP 4.3.1 | Info not intercepted | Pass | VP 8.4.1 | User info removed from system | Pass |

## 3.1 Register



Upon selecting Register the user is directed to a registration screen. The user must enter the information according to the requested format. As is seen in the above figures, if the user enters information according to the wrong format the appropriate error message is displayed.
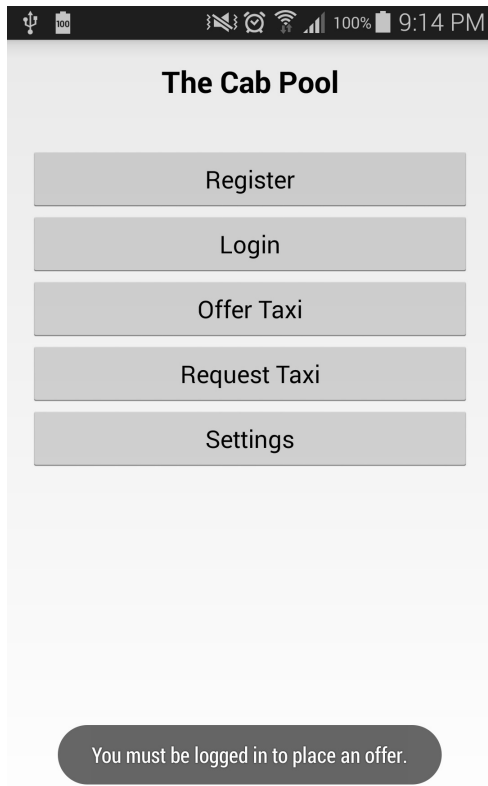
## 3.2   Login Screen



If login is selected, the user is successfully redirected to a login screen at which they must enter their username and password. As seen in the figure above, if an invalid username is entered a corresponding login failure error message is displayed.
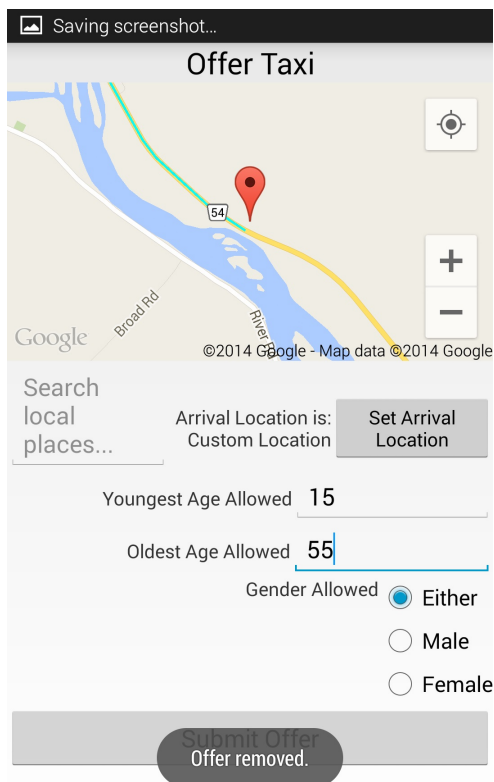
## 3.3  Offer Taxi



When the user selects the offer taxi option, the system confirms that the user is logged in, if they are not an error message directs the user to login. A logged in user will be able to select the offer taxi option and proceed to the offer taxi process which includes scanning the QR code.

## 3.4  Scan QR Code

Within the option to offer a taxi the user must first scan the taxi QR code, to start this process they must select Scan Code. The app will then require them to complete the scanning process. When an invalid code is scanned then an error message is displayed indicating the code is invalid. Following a successful scan the scan code screen indicates the code was scanned and the user must submit it to verify offer. The server will update to indicate the user has entered the taxi associated with the scanned QR code.
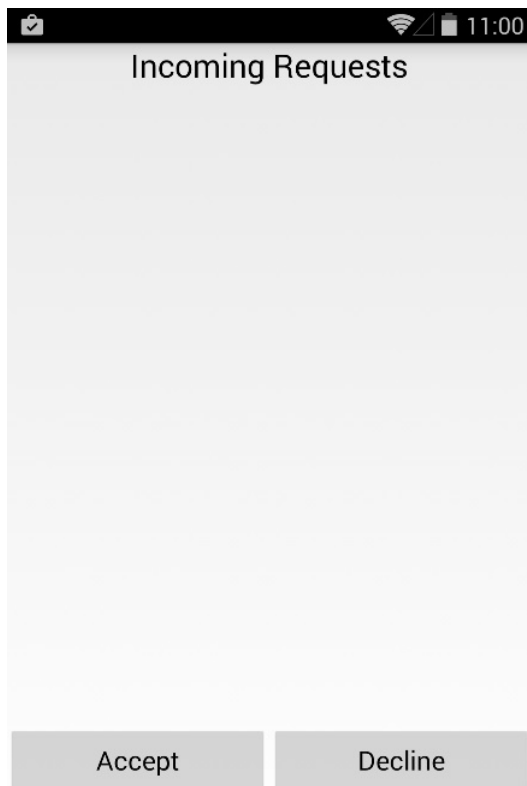
## 3.5    Cancel Offer



The user is able to cancel an offer by returning to the Offer Taxi screen and selecting Remove Offer. Upon clicking Remove Offer the offer is removed from the database. As seen in the above figure, the user is prompted with a message indicated the offer has been removed.

## 3.6   View Incoming Request



As seen in the above diagram, the Incoming Requests screen is empty while the system matches requests with offers. Upon matching users in the database, the app successfully populates the list with matches.
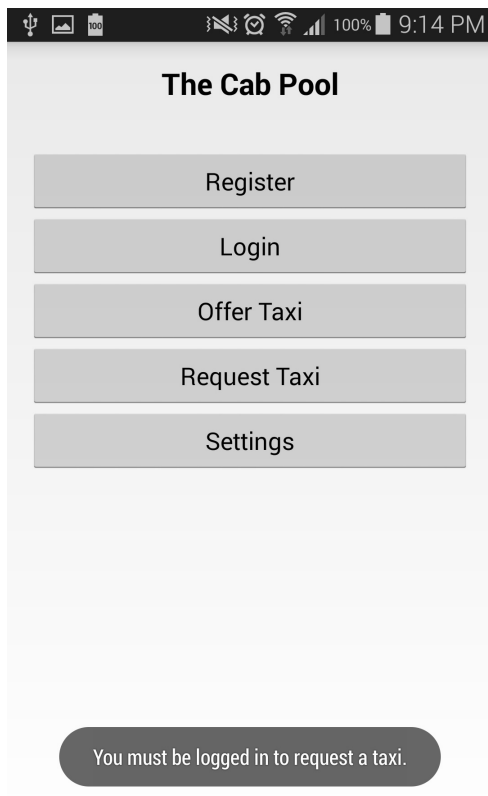
## 3.7   Accept User

The user offering the taxi is able to select and accept a user requesting a taxi.

## 3.8   Decline User

At this point, the user is not able to explicitly decline a user but they are not required to accept their request. Additional requests will populate the incoming requests screen.

## 3.9    Request Taxi



A logged in user is able to select the option to request a taxi. The above diagram is displayed in the event a user selects the Request Taxi option while not logged in.

## 3.10    Cancel Request

At this point, the user is unable to cancel a request, all requests are stored in the database until the request has been completed. The requesting user, however, is able to make additional requests and only accept appropriate offers.

## 3.11    View Search Results

The server successfully communicates with the dispatcher to display search results that match the criteria. Search results displayed include username, user rating, trip fare and duration.

## 3.12 Accept Offer
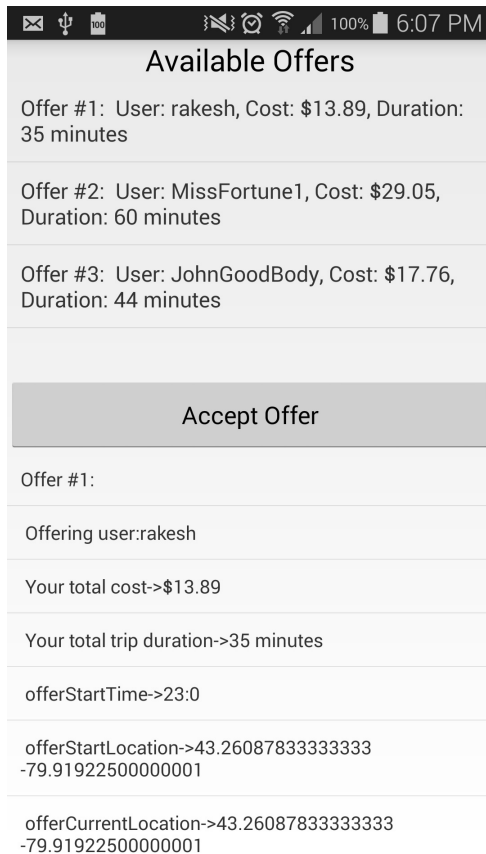


As seen in the above figure, the dispatcher successfully populates the Available Offers screen with offers that match the user search criteria. The user is able to select an offer they would like to accept. This message is sent securely to the dispatcher who allows the offering user to choose whether to accept the requesting user.

## 3.13 Reject Offer

The dispatcher successfully populates the Available Offers screen with offers that match the user search criteria. At this time, the user is not able to explicitly reject an offer, however, in not accepting an offer they have effectively rejected the offer. The lack of offer acceptance is communicated to the dispatcher and the offering user is not able to accept the request of a nonresponsive taxi requester.

## 3.14 Select Settings
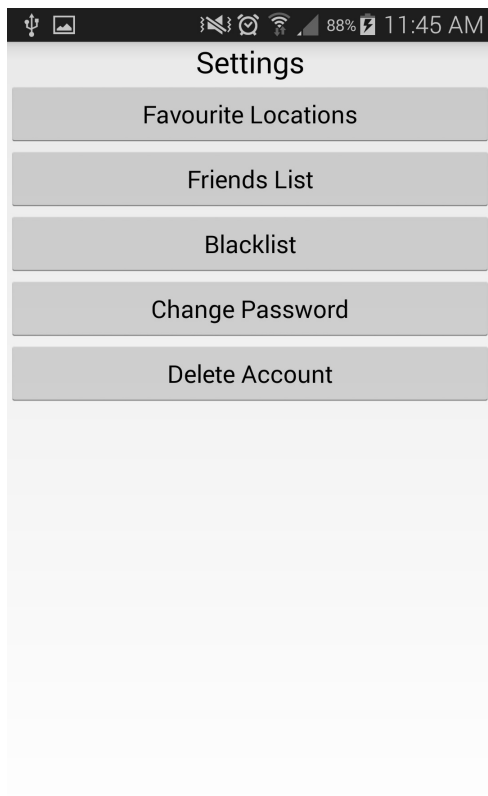


The user is able to select the settings menu. Which displays the appropriate options.

## 3.15 Select Favourite Locations

The user is able to select Favourite Locations. Within the Favourite Locations option there is currently no functionality.

## 3.16 Modify Favourite Locations

At this time there is no modifiability within Favourite Locations.

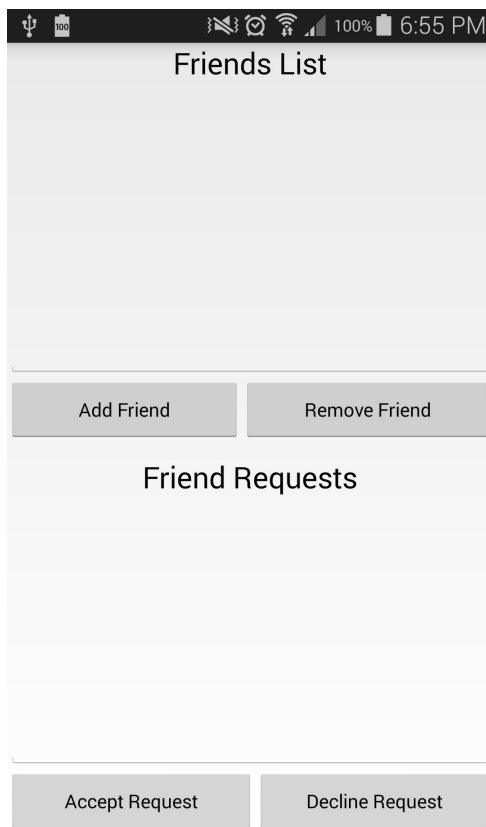## 3.17    Select Blacklist
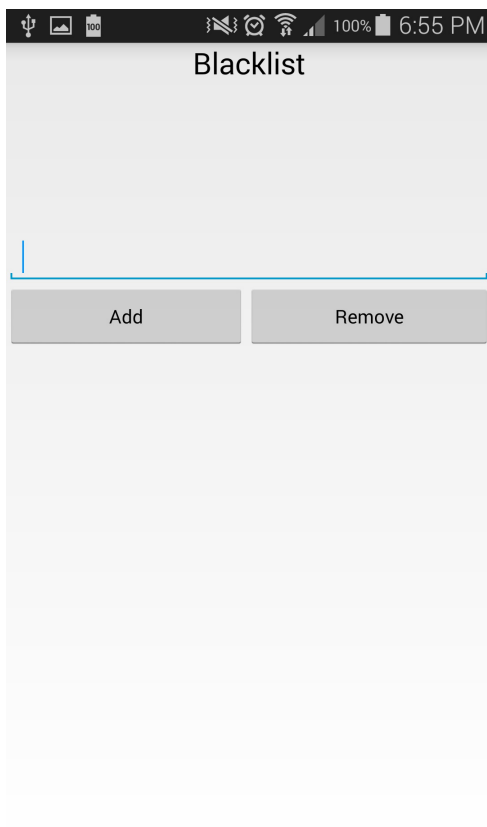


The user is able to select Blacklist . Within the Blacklist option there is currently no functionality.

## 3.18    Modify Blacklist

At this time, the user is unable to modify the Blacklist.

## 3.19   Select Friends List



The application successfully displays the options available within Friends List.

## 3.20   Modify Friends List

There is no functionality within the Friends List at this time.
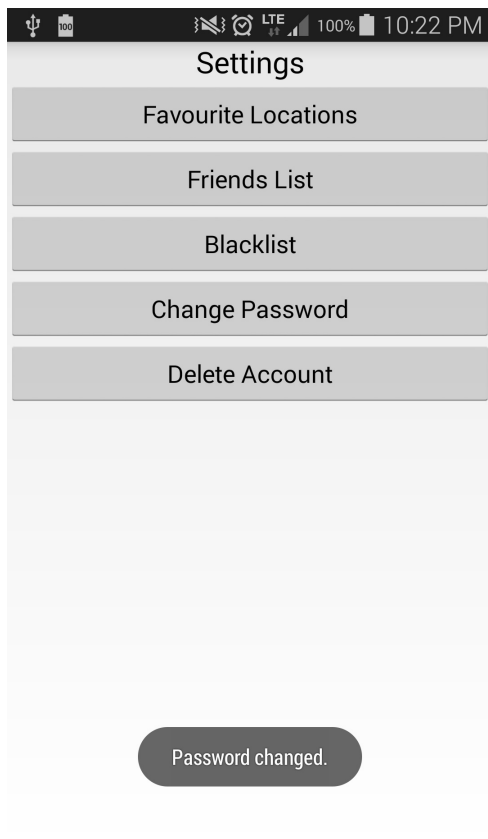
## 3.21   View Friend Request

The user has no functionality within the Friends List.

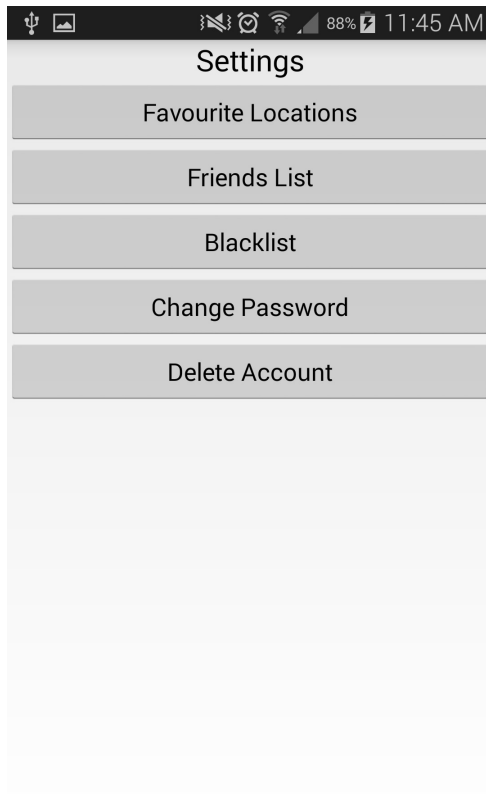## 3.22   Modify Friend Request

At this time, the user is unable to modify Friend Requests.

## 3.23 Change Password



As seen in the above figure, the user is able to change their password.

### 3.24 Deregister



The user is able to delete their account, this removes all user information from the database.

## 4 Future Work

The application runs smoothly on an android phone and interfaces well with Google Maps. Communication with the server is quick and secure. Barcode scanning is quick and effective. Desired user features work well and the application runs as expected. With the exception of innovative features, all requirements were met. Exception cases display appropriate error messages and the user is prompted on how to correct their behavior to achieve desired results. Our application is successful and we are confident it is free of bugs.

At present, the innovative features proposed, including Friends List, Blacklist, and Favourite Locations are not functional. While the options appear in the app selection of these does not provide the desired functionality and this an area requiring future work.

Future Work for this project involves scaling the application to handle a larger user base. Currently the server will not accommodate greater than 30 users, in the future, this limit will be unacceptable and must be increased to reflect the vast number of users anticipated to use this application.

# A  Division of Labour

We worked collaboratively to complete this project. Each group member made an equal contribution to the project. By signing below, we hereby acknowledge that the work was in fact divided equally.

Amanda Boudreau     _____
Jordan Floyd        _____
Mario Machado       _____
Rakesh Mistry       _____
Ali Reda            _____