**Taxi Sharing Application**

Andrew Dodge – 100938015

An Honours Project submitted to

The School of Computer Science in partial fulfillment of

Bachelor of Computer Science Honours with

Software Engineering Stream

Carleton University

April 22nd, 2019

Approved by:

Honours Project Supervisor: Dr. Doron Nussbaum

School of Computer Science Undergraduate Advisor: Edina Storfer

**Abstract**

**Acknowledgments**

**Table of Contents**

Abstract……….……………………………………………………………………………………i

Acknowledgements………………………………………………..………………………………ii

Table of Contents………………………..………………………………….…………………….iii

[Chapter 1: Introduction 1](#_Toc5020153)

Problem Definition……………………………………………………………………………………...1

Motivation……………………………………………………………………………………….……...1

Contributions…………………………………………………………………………………………....2

[Chapter 2: Background 4](#_Toc5020154)

Definitions……………………………………………………………………………………………...4

Development Software………………………………………………………………………….……...4

Previous Work………………………………………………………………………………………....7

[Chapter 3: Main Contribution 8](#_Toc5020155)

[Chapter 4: Implementation 9](#_Toc5020156)

[Chapter 5: Conclusion 10](#_Toc5020157)

[References 11](#_Toc5020158)

[Appendix 12](#_Toc5020159)

Balsamiq Mockups……………………………………………………………………………………...

# **Chapter 1: Introduction**

Taxi sharing, a method of transportation that is a cross between a bus and a taxi. Meaning a taxi will pick up multiple people, either at the same location or multiple along the way, then drop them off at their desired location so long as they are all in the same general area. To give some additional information, taxi sharing is more beneficial than traditional taxis as each patron pays a cheaper fee if the ride is shared with other passenger. Furthermore, the cab driver is paid by each individual passenger meaning that they end the day with a greater income.

*Problem Definition*

The honours project for COMP4905 that was completed this semester was to design and develop a taxi sharing application for an Android device. The focus of the project was on research and design, to determine what applications already existed with this concept and to design an application that functions on multiple different device sizes for a wide range of individuals. A back-end communication system was implemented; however, there was minimal attention paid to it as it was not the focus of the project. This project was done in tandem with another student, Mohamed Gahelrasoul, with an even split in the division of labour. His task was to implement the driver portion of the application as well as to implement the back-end communications. The other half, which will be the focus of this report, was the passenger side of the application. It was decided at the beginning of development that the driver would not have a large variety of options associated with them, this will be further discussed later in the report. This was the reasoning behind the division of labour.

*Motivation*

The motivation behind this project was to develop a taxi sharing application that would form the base of a future project. This app is to be further expanded and evolve into an autonomous vehicle taxi sharing application. Where driverless vehicles will be accessible to passengers to take them to and from their destinations for a fee. This motivation was the reasoning behind making the driver side minimal, as it will eventually be eliminated entirely. Further motivation for this project stems from a desire for knowledge off app development and design. To aid in creating a solid foundation of skills to be taken into future career paths. To continue, as stated above the purpose of this project was to create an application that will then be evolved in the future. The need for this to be studied stems from how technology is constantly advancing, especially with the discussion of self driving cars becoming more prevalent in today’s society. To have an autonomous vehicle that could transport small groups of people throughout cities and to their destinations would be the next logical step once the technology becomes realized. This concept would also aid alleviate the number of vehicles that would be on the roads, since every person that shares a ride with another would be one less vehicle out that day. Aiding in reducing the amount of pollution emitted in cities with this system in place. It is believed that while the problem being addressed by this project is not of the utmost importance to the world as a whole currently, it does not make it any less important to research and begin implementation. Especially due to the fact that there are a number of taxi sharing apps in circulation already, it is hoped that the future evolution of this project will set it above the apps already being used.

*Contributions*

Finally, the work completed during this project was a full implementation of a taxi sharing system. The project consisted of two separate apps that share a back-end database and communication system. The passenger application was the focus of this project, the work completed for the system designed began as a large amount of research focusing on how taxi sharing works and what goes into creating a working system. It was the goal to determine what fields are necessary when designing an application similar to this, as the user fields would be necessary in creating the matching component of the application. Other than research, the main focus of the project was in the design of the application itself. Firstly, to get it to be visually appealing when using it. Secondly, and most importantly, to get it to function smoothly on Android devices. The manner in which the design was implemented makes it so the app will scale based on which Android device the app is installed on. This is especially important when it comes to the text visible on screen. The design allows for the apps text to mirror the users text size preference, if a user has their text default to large then the app will use this decision on load. With regards to the app itself, it is fully functional, it is possible to request a ride and be matched with a driver as well as other passengers. This information is displayed on the screens for all users to inspect. More detail will be given on the interactions and abilities of the users further into the report. The driver side application implemented by Mr. Gahelrasoul as mentioned above will be further expanded upon in his report, as well as the communication system.

The report will follow a basic layout, following this chapter the background will be discussed at length. Such as, what research went into development as well as the other apps that were used as either inspiration or research tools. The two chapters that follow the background discussion will further discuss the main goal of the project, delving into an in-depth discussion of the problem attempting to be implemented and the objectives of the application itself. As well as, how the project was implemented and carried out. This section will discuss in detail the process of the applications development such as how the application was completed from start to finish, as well as the testing which was conducted during development. Finishing with the results of the testing and how the findings were used to finalize the application. The final chapter will be a conclusion that will briefly summarize the information that was started throughout the report.

# **Chapter 2: Background**

As was discussed above this project was to design a taxi sharing application using Android Studio that would function across a number of Android devices. This chapter will define the needed information that is required to understand the concepts of the project. Beginning with a basic definition of the purpose of the application and following into a description of the software used. Finally, this section will discuss the applications similar to the one designed in the project and how they were either used as research material or inspiration towards the completed project.

*Definitions*

To begin, taxi sharing, as stated above, is a service that combines the simplicity of taking a taxi with the passenger sharing ability of a bus. The main difference between this service and a regular cab is that a passenger will be matched with strangers who have similar destinations, or additional passengers will be picked up along the route. Instead of the typical privacy that a taxi would bring to the ride. Furthermore, taxi sharing incentivises sharing the ride with strangers as for each additional passenger the fee for the ride decreases per person. With respect to this, since each passenger must pay a fee to use the service this means the driver yields a greater income if there are a higher number of passengers. With this in mind, this was the goal of the project, to design an application that would achieve full taxi sharing capabilities. From matching similar users together, to displaying their trip on the screen, to completing the ride and displaying the proper information to the user post trip.

*Development Software*

Before discussing the components of the application and how the system came together it is important to touch upon the program used to design the project. The application was designed using a program called Balsamiq Mockups. This program is a prototyping tool where a user can create low-fidelity prototypes, or digital sketches, of the project being designed. This gives the developer the ability to produce ideas, facilitate discussion, and expand upon understanding before any code is written. The important aspect of Balsamic that sets it apart from other prototyping tools is the ability to link the digital sketches together as if it were a real application. With this feature it is possible to mimic what certain buttons in the application will do. Such as, if clicking a button opens the user’s profile then it is possible to mimic that interaction in the mockup. Once the entire design has been implemented in Balsamiq it becomes quite simple to modify elements and view how interactions function before the development begins. As well as development becomes greatly simplified as there is an example to base each stage of design on. All Balsamiq Mockups for this project are visible within the Appendix section at the bottom of the report, as well as how the images are linked together.

To continue, the application was implemented using the program Android Studio by Google. It is the official integrated development environment, IDE for short, for application development for Android devices. Essentially, it is a program designed specifically to create applications for Android devices, with the ability to both design the user interface and create back-end functionality. Android Studio is based off the IntelliJ IDE and incorporates the shortcuts used within that development tool to aid with programming. There are two main languages a user can program in within Android Studio, users can choose between Java and Kotlin. Both with unique benefits and complications. For this project the language of choice was Java, for ease of implementation and back-end simplicity. With regards to development, Android Studio divides the workload into two distinct parts. The design of the screens, or activities as they are called in Android Studio, is done in an extensible markup language, xml, file. It is possible to add elements, such as text, buttons, and images, by either writing the xml code straight into the source code, or by visually designing the application on the blueprint screen. When using the blueprint screen, it is possible to drag the elements needed onto the given activity, it is the visual method of created each page of the application. Once an element has been added to the activity using this method, Android Studio will automatically add the xml source code to the file. This method of design is reminiscent of the development environment Unity, where visual elements are added to the screen in this manner and the back-end of these elements are programmed afterwards in a separate file. The separate file in this case being a Java file per activity in the application. Each respective Java file contains all the back-end functionality that activity needs to accomplish the tasks assigned to it. Be it a button, referencing user information, or displaying specific information on the screen. Android Studio can also be linked with multiple different resources to aid in the production of the application. For example, Github can be linked to the Android Studio application for ease of backing up the user’s data. Furthermore, since Android Studio uses a Gradle-based system for compilation a user can connect directly to a Github project using a compilation command. This command will direct the compiler to include the source code stored in Github and add said source base to the amalgamation of files that make up the core of the application. This Gradle-based system also allows the user to specify the required permissions and Google Play Services the app will require for installation. Finally, with regards to testing the application using Android Studio, the program has two built in functionalities where a developer can view the work they have completed thus far. Both methods will involve compiling the application and creating an Android Package, APK, that will then be installed on a device. This is where the difference lies, developers can choose to install the application on a virtual Android device, or onto a physical Android device. The virtual machines are built into Android Studio and can be downloaded at any time during development, they are helpful tools as a developer can then test how the application will react on different Android devices, or different versions of the operating system. Furthermore, it can be used to quickly test features of the application as the developer does not have to change environments. The virtual device is stored on the same computer as the development environment. Comparatively, the developer can use a physical Android device to run the application. This is easily done by plugging an Android device into the computer at the time of compilation and selecting it as the desired location. While this method is slower than using a virtual device, it allows the developer to test how the application will function in the real world, especially if the app has network components. With the application on a physical phone the network components can be tested with the user’s data or wireless internet. Android Studio visual examples are listed in the Appendix section of the report.

Speaking of network capabilities this application has an online database and communication system that uses Firebase on the Google Cloud Platform. Firebase is a Backend-as-a-Service, a model for providing web and mobile app developers to link applications to back-end cloud storage. This project’s application uses Firebase for a wide array of features. The primary role that Firebase fulfills is as a database. All user information when an account is created is stored in the Firebase database and is referenced from there when called upon by the back-end code. User authentication is also controlled by Firebase, a user cannot login without a valid account stored on the database. Most importantly Firebase acts as the communication system for the application. Firebase acts a server for the driver and passenger applications to communicate between each other. While the matching algorithm is implemented as back-end code the communication system is done through Firebase. The majority of the back-end was implemented by Mr. Gahelrasoul, further detail will be said on this subject within his report.

*Previous Work*

What app exist already

Uber

Lyft

Taxi sharing

Waymo one (autonomous taxi)

# **Chapter 3: Main Contribution**

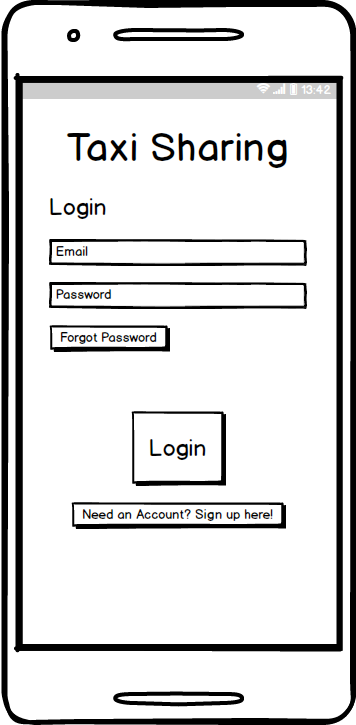
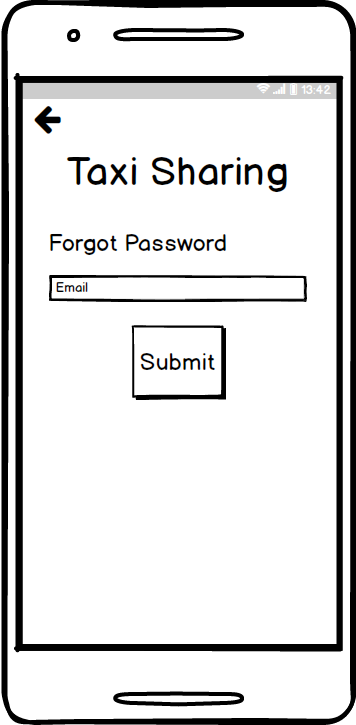
# **Chapter 4: Implementation**

# **Chapter 5: Conclusion**

# **References**

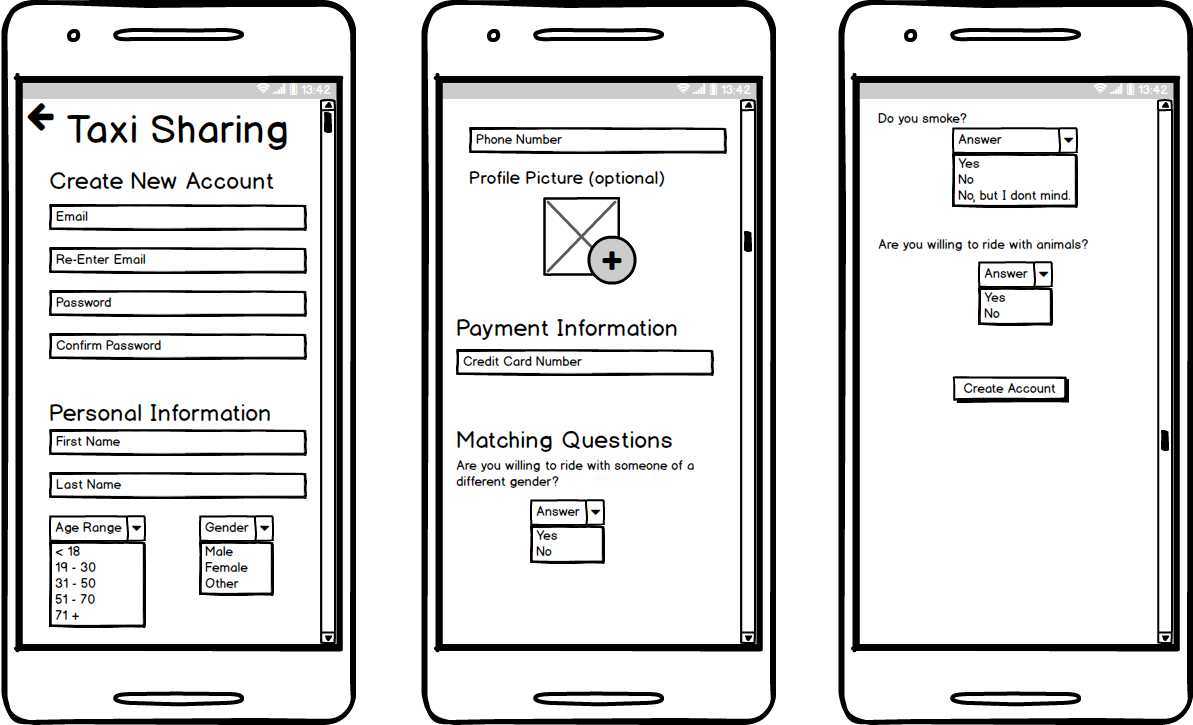
# **Appendix**

|  |  |
| --- | --- |
| **Legend for Mockups** | |
| Symbol | Meaning |
| Red Arrow | Button clicked leads to the screen the arrow is pointing to. |
| Red Star | Buttons with this symbol lead to the Pre-Match Screen. |
| Blue Star | Buttons with this symbol lead to the Login Screen. |
| Red X | Feature was removed during development. |
| Green Star | Button with this symbol lead to Pre-Match Screen 2. |

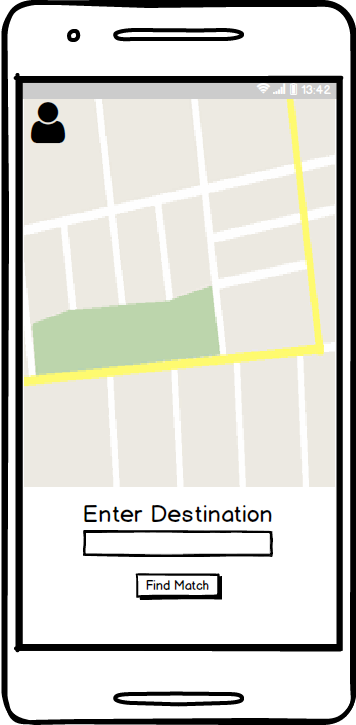
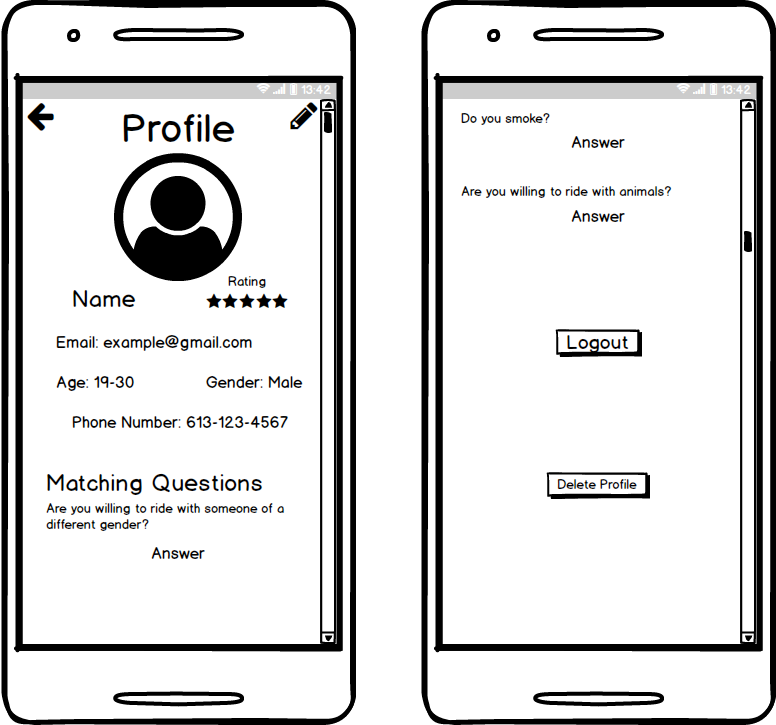
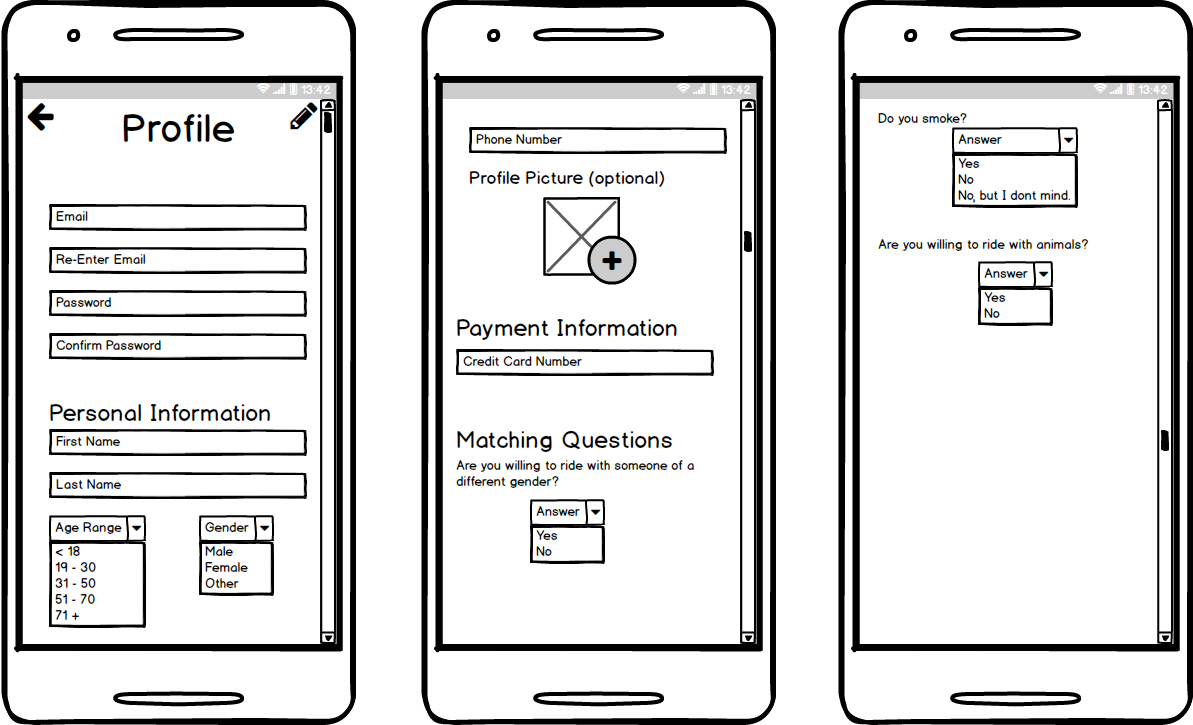
*Balsamic Mockups*

Forgot Password

Login Screen



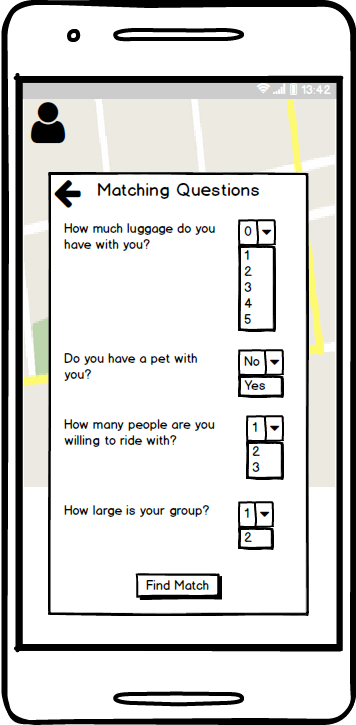
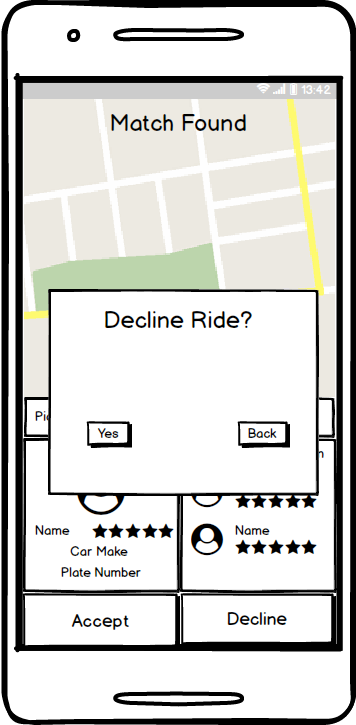
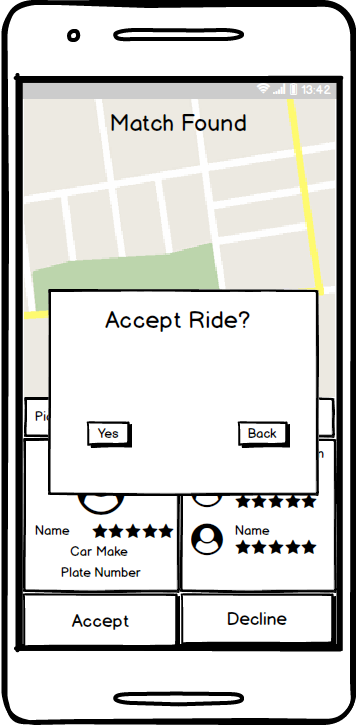
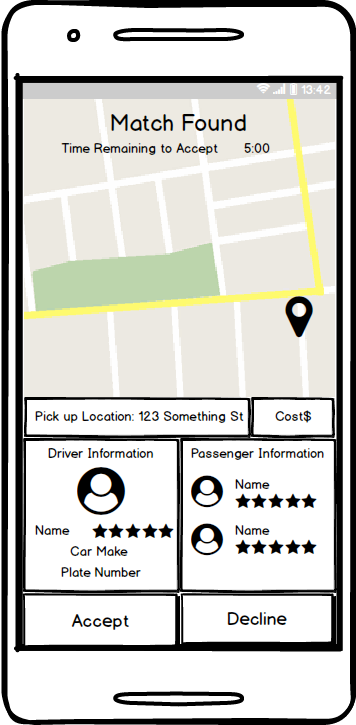
Create Account



Edit Profile

User Profile

Pre-Match Screen

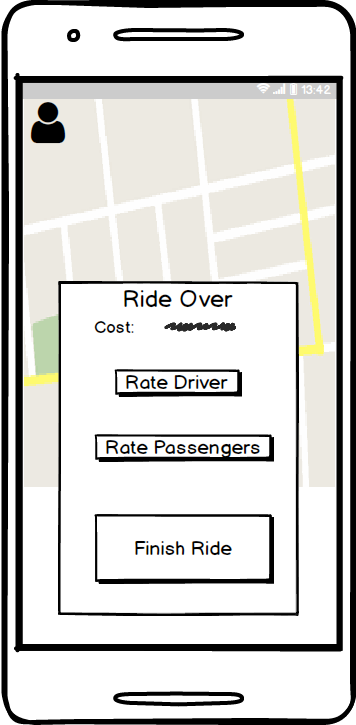
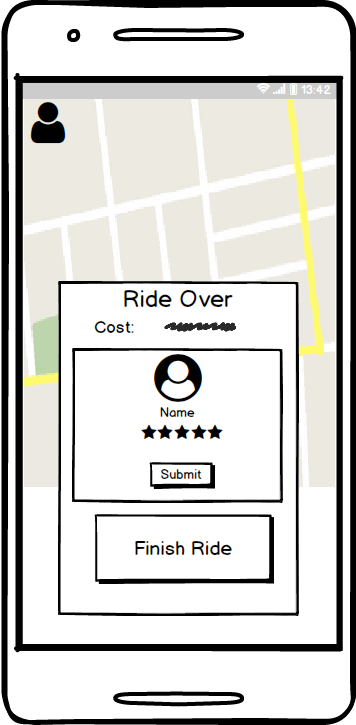
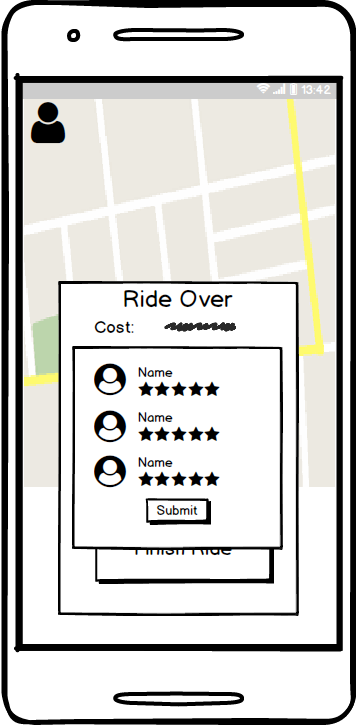


There will be a slight delay in this transition while the application references the database and find the match.

During Ride Screen

Match Found Screen

Pre-Match Screen 2



After the ride has concluded and the passenger has been dropped off the app screen will change to this view.

Rate Passengers Prompt

Rate Driver Prompt

Post-Ride Screen

*Android Studio Design*