**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 12](#_Toc5020155)

[Chapter 4: Implementation 13](#_Toc5020156)

[Chapter 5: Conclusion 14](#_Toc5020157)

[References 15](#_Toc5020158)

[Appendix 16](#_Toc5020159)

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

# **Chapter 1: Introduction**

Taxi sharing is a modern method of transportation that establishes a cross between bus and taxi travel. The central idea of which being a taxi that can pick up multiple people, either at the same location or multiple locations along the way, and then drop them off at their desired location (preferably 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 passengers. 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, which was completed this semester, was to design and develop a taxi sharing system for the Android device. The taxi sharing system consists of three components: a passenger application, a taxi driven application and a back-end. The focus of the project was on research and design, to determine what applications already existed within 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 division of labour. His task was to implement the driver application of the system as well as to implement the back-end communications. The other half of the project, which is the primary focus of this report, is the passenger application component of the system. It was decided at the beginning of development that the driver would not have a large variety of options associated with them, which will be further discussed later in the report. This was the critical reasoning behind the division of labour.

*Motivation*

The motivation behind this project was to develop a taxi sharing system which would then form the base for a future project. This system is to be further expanded 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 an intrinsic desire to expand on knowledge of application development and design and contribute to this development in a meaningful way. This will 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 is to create an application in the present that will then be evolved in a future project. The importance of studying this technology stems from the certitude of continuous advancements in modern technology. Especially with the introduction of self-driving cars having become 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 to alleviate the number of vehicles on the roads, since every person who decides to share a ride with someone would ensure one less vehicle out on the road. This is crucial in modern society, where pollution from transportation has become significant and pervasive. Reducing the number of vehicles on the road would aid in decreasing the amount of pollution emitted in cities that possess this system. The problem being addressed by this project may not appear to be globally important to the present population. However, it does not make it any less important to research and begin implementation now. Especially due to the fact that there are a number of taxi sharing applications in circulation already. There is hope that the future evolution of this project will set it above the standards of current applications.

**HE GAVE YOU NOTES FOR ^ PARAGRAPH THAT I CANNOT READ..**

*Contributions*

The work completed during this project was a full implementation of a taxi sharing system. **HE SAYS SOMETHING HERE?** The project consisted of two separate applications that share a back-end database and communication system. The passenger application was the focus of this project, and consisted of two components, research and implementation. The research focused on how taxi sharing works and how to implement a working system. The ultimate goal of the project was to determine what information was required in order to match passengers with taxis. The implementation aspect focused on the design of the application itself. Firstly, to get it to be visually appealing when using it and secondly, to ensure it can function smoothly on Android devices. The manner in which the design was implemented makes it so that the application will scale based on which Android device the application is installed on. This is especially important when it comes to the text visible on screen. The design allows for the application’s text to be able to mirror the user’s text-size preference. If a user has their text default set to “large” then the application will use this decision on load. With regards to the application itself, it is fully functional, and able to request a ride in order to 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 follows a basic layout. Chapter 2 will be discussed, followed by the presentation of background information. This background information will contain material such as the kind of research that went into development, as well as other applications that were used as inspiration and research tools. Chapter 3 follows the background discussion and 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.

**YOU MAY HAVE TO EDIT THIS PARAGRAPH AND TALK SPECIFICALLY ABOUT EACH CHAPTER, SINCE I DON’T KNOW THAT INFO ^**

# **Chapter 2: Background**

This chapter reviews the systems that attempt to provide….. **bloody hell I have no idea what this dude is tryna say wut.** 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. It will begin with a basic definition of the purpose of the application, followed by a description of the software that was 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*

Taxi sharing is defined as a service that combines the simplicity of taking a taxi incorporated 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, and these additional passengers will be picked up along the route. This is opposed to a taxi, which involves the typical privacy that standard taxis provide. Furthermore, taxi sharing monetarily incentivises riders to share the ride with strangers, due to the collective fee that contributes to reduced individual fees. With respect to this, since each passenger must pay a fee to use the service, this means the driver would yield 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. An important aspect of Balsamiq 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 application flow. For example, 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. Development also becomes greatly simplified with the addition of examples to help base each stage of design on. All Balsamiq Mockups for this project are visible within the Appendix section at the bottom of this report, as well as a description of 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), which 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. This is the visual method of creating 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 the activity needs in order 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 application 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 application 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*

Following the description of software used throughout the project, previous work is reviewed. There are numerous applications **NAME SOME** recognized in the field of taxi sharing. However, only a select few were used as examples or research tools when designing the application. The applications that became a research focus included: Uber, Lyft, Via, and Waymo One. These applications were all used as research tools at one point in development, each for their own reasons. Most of the research was used to determine how taxi sharing works at a fundamental level. For example, how the user profiles are set up, or what components are needed to determine a proper match, as well as what is a reasonable cost for a typical ride using these applications. All applications used have a taxi sharing component to them. While Uber and Lyft are apps in their own regard, they have since implemented a taxi sharing element similar to the goal of this project’s application.

To begin, the most popular modern application will be discussed. Uber currently holds the title as the most popular taxi application on the market. It is available in sixty-five countries as well as six hundred cities worldwide. There are approximately fifteen million trips completed each day using this application. Therefore, it was a prime research tool when designing this project. However, Uber is typically just a taxi service that passengers can use to request a ride for themselves, or for small groups of passengers, which will take them to their requested destination. The application being designed in this project is specifically a taxi sharing system. Meaning, Uber does not entirely fit the description of the project. This being said, during research it was found that Uber has a taxi sharing component available to their application. It is called UberPool, which is their variation of a taxi sharing application. It functions the way one would expect a taxi sharing application to function. A user can open the app and enter their destination and size of their group, (maximum 2 people), before requesting a ride. The system would then match them with a driver and send them to a pick-up location. During this waiting period other users are matched to the car until the vehicle is full. The application will also display the cost of the ride before pick-up occurs as well as the estimated travel time, so that the user knows the amount the ride will cost before getting in the Uber. During the ride the driver will either continuously pick up passengers along a similar route or drive certain passengers straight to their destination if there are no further customers. Once the destination is reached, the appropriate passenger exits the vehicle and is automatically billed via the payment method listed on the application. This ride system greatly inspired the system put in place by the project application. The only element left out was displaying the estimated duration. The research gained from this application, other than the overall system, aided with back-end development. Analyzing Uber’s system aided in determining the amount of luggage a user would be allowed to bring with them. It also aided in determining how to drop off riders, as well as determining how many passengers are allowed per group. It was determined from comparing the differences between Uber and UberPool that while UberPool takes slightly longer to complete a trip, it is always approximately 65% **VERIFY THIS STAT** cheaper than taking a regular Uber.

To continue, Lyft was also used as a research tool for this project. Lyft is a popular North American taxi application. It is currently being utilized in 350 US cities as well as the Canadian cities, Toronto and Ottawa. While Uber dominates the market with sixty-five percent of all application-based taxis, Lyft holds onto a significant thirty-one percent of that market. Therefore, they were an important source of research for this project. Furthermore, while researching the company it was found that Lyft has also introduced their own variation on taxi sharing, called Lyft Line. Lyft Line essentially functions similarly to UberPool, however, the main difference found here was that while Uber will slowly match users with other passengers along the way (picking up and dropping off passengers as the ride progresses), Lyft will attempt to match the group of passengers together at the beginning of the ride and not pick up additional passengers once the drive has begun. However, if a passenger requests a ride while a trip is ongoing, and the pick-up and drop-off is optimal, then the Lyft driver will be instructed to pick them up along the way. This feature was found to be similar to the idea that this project was trying to achieve and thus the project was implemented in a similar fashion. Similarities were also observed between the two systems. For example, there were similarities in how they displayed the approximate cost and duration of the ride before the trip begun. Another example was how the application determined the optimal route based on all the individuals’ desired destinations. This means that while a user may be picked up first they may not necessarily be dropped off first, they will be dropped off in the most optimal order possible. These two apps heavily influenced the back-end system that the project was aiming for, therefore there are distinct similarities with regards to the order actions are carried out during a match. With regards to user interface, the design of this project was not nearly as influenced by these apps as the back-end system was. Both Uber and Lyft have their own distinct way of displaying information to the user. Both applications have a visible road map as the background with their assortment of elements organized on top of the map. This is the only similarity with regards to the user interface. The project’s application once logged into, has a map as the background with elements organized on top of it. However, the design of the application is unique and not based off one of these popular applications.

Moving away from the popular taxi sharing options, another application that influenced the project will be discussed. Via is a rather unknown alternative to Uber and Lyft. It currently makes up one percent of taxi sharing application services and is still only available in three cities: Chicago, Washington D.C., and New York City. but this was not a deterrent to using it as a research aspect. Via is not technically a taxi sharing application, but rather a shuttle service. Users with the application will see specific pick-up and drop-off locations and will be able to ride the shuttle for a small fee. This is similar to a city bus but without the uncertainty that the bus will never arrive. Furthermore, since the application is still relatively small, there is a large amount of customer support available to users, as the company does not have to deal with millions of requests daily. The main aspect that was taken from this service and integrated into the project was they way this application deals with pick-up. All passengers arrive at the same location and board the shuttle (or taxi, in terms of the project), at the same time. The goal of the project was to achieve this functionality; not to have to make multiple pickup stops along the route. Therefore, this element was integrated during development. To continue, since Via is closer to a shuttle service than a taxi service, it typically has the same day-to-day customers. It is a popular method for students to get to school or employees to get to work everyday. For this reason, Via attempts to develop a connection between riders and drivers, as the same driver will typically transport the same passengers each day. Via attempts to create a more enjoyable commute for all individuals involved, instead of the commute just having to be a necessary component of someone’s day. Finally, Via offers incentives to users on top of the transportation. This feature is an aspect lacking from the applications of the competitors. The incentives within their service has to do with the cost of each ride. As drivers transport more individuals over a consistent time period they begin to earn a higher income than when they first started in the employ. Furthermore, as a passenger there is a similar system in place. The more a user rides with Via the smaller the regular fare becomes. This became a noticeable feature that was under consideration to be applied to the project.

To conclude this section, there is one additional application that must be discussed. As this project’s motivation is for it to evolve into an autonomous taxi service it would be unjust to ignore the autonomous taxi service currently in development. Waymo One began as Google’s self-driving car project in 2009 and has since evolved into a full autonomous taxi service. The application is currently only in deployment in the Metro Phoenix area of Phoenix, Arizona with hopes that it will be rolled out further in the future. The application works similarly to other taxi applications where a user enters their pick-up and drop-off locations before requesting a ride. The application will then display the given fee for the ride before the user confirms their ride, in order to make sure the user is comfortable paying the given amount. The vehicle will then arrive where specified and carry passengers to their requested destination. All of this is accomplished without a driver in the vehicle. Passengers of this service always know what to expect since every ride is identical. The vehicles are always the same and carry the same branding, and if anything were deemed unsatisfactory during the ride or if users needed to get in contact with support, there is a built-in help system on the application. Users simply need to open the support tab on the application, and they will be able to discuss the issues with their ride to a live representative. While no features were implemented or inspired by aspects of this application it was still an interesting element found during research that deserved mentioning given the scope of this project.

To end this section, it can be concluded that there are numerous options available when it comes to designing the application for this project. A number of which had prominent influence during the design of the application, and a number of which did not have any influence whatsoever. However, regardless of the influence by outside applications, the project which will be discussed in the following two sections is entirely unique. As is the case with most elements in the world, each new age of technology will always incorporate influences from the past. However, while the back-end and flow of the application may be reminiscent of others, the design and feel of this application will be solely its own.

# **Chapter 3: Main Contribution**

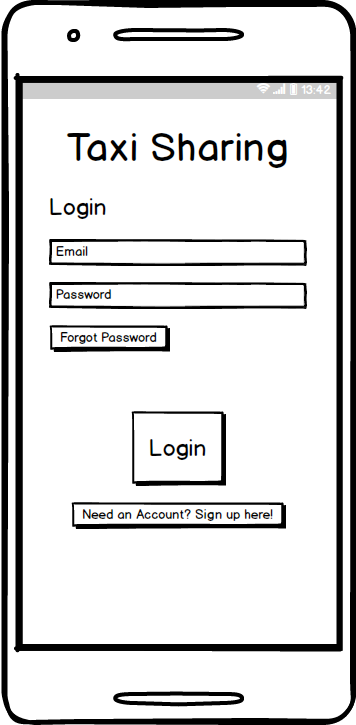
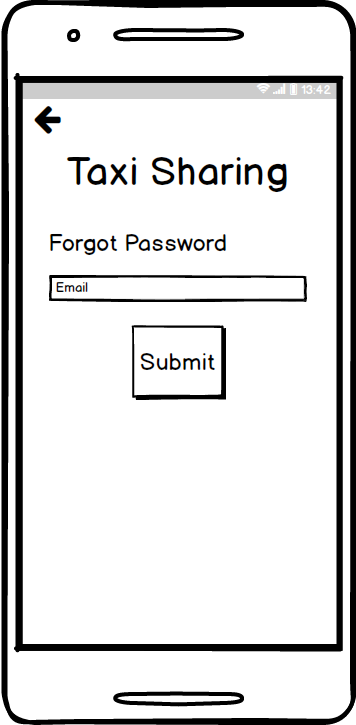
# **Chapter 4: Implementation**

# **Chapter 5: Conclusion**

# **References**

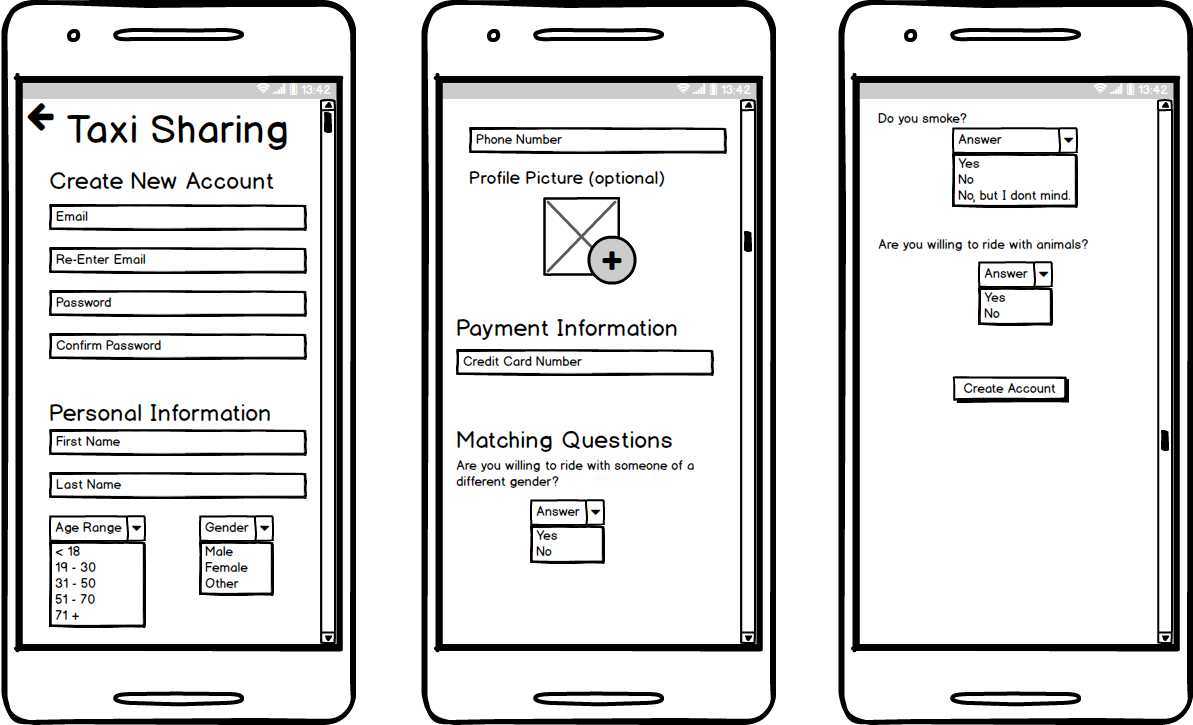
# **Appendix**

|  |  |
| --- | --- |
| **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. |

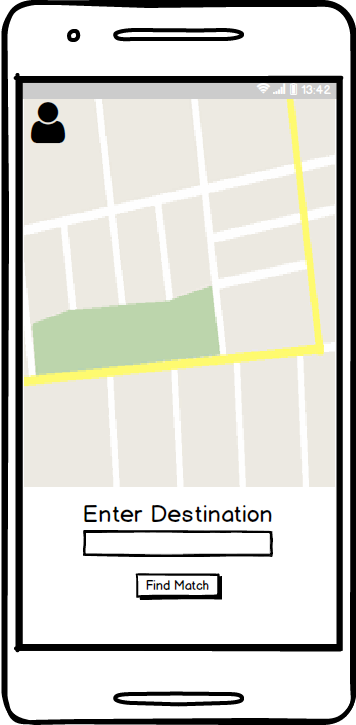
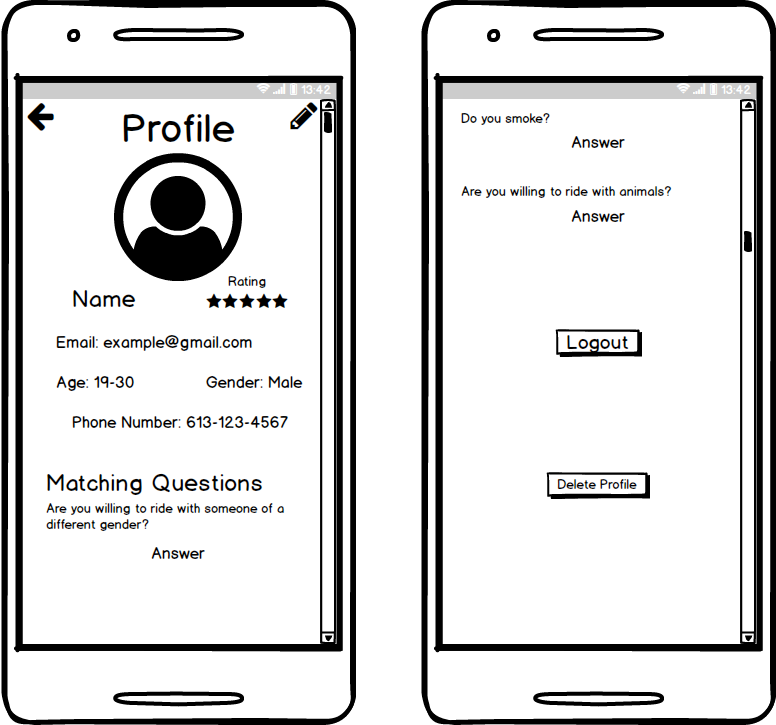
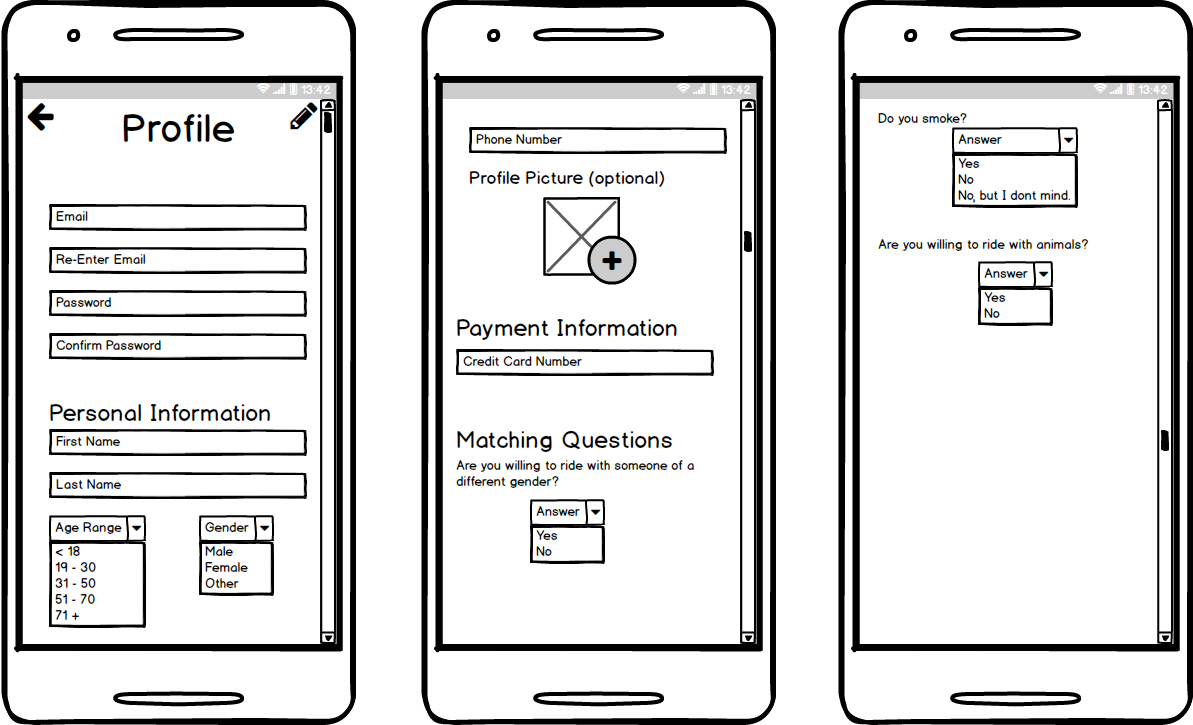
*Balsamic Mockups*

Forgot Password

Login Screen



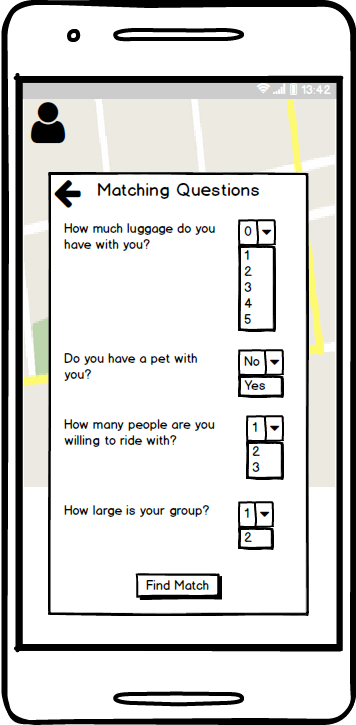
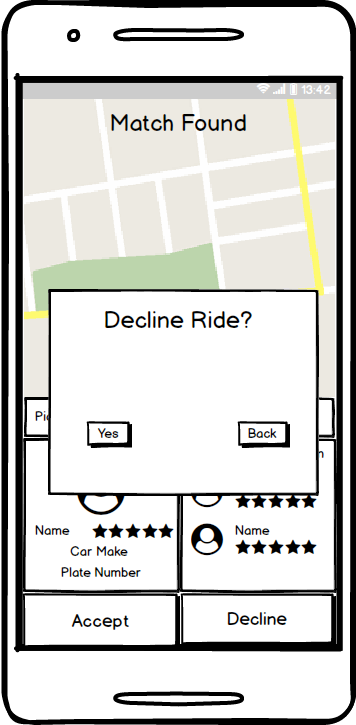
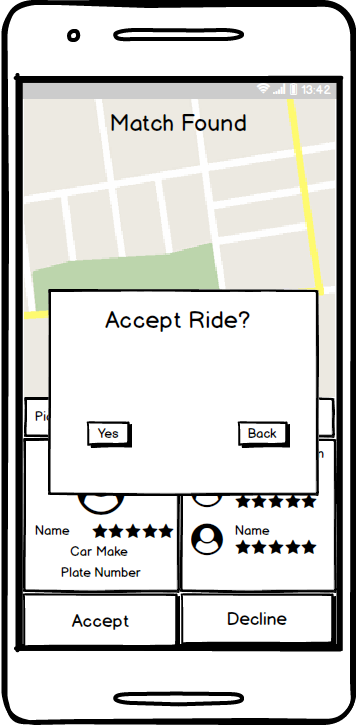
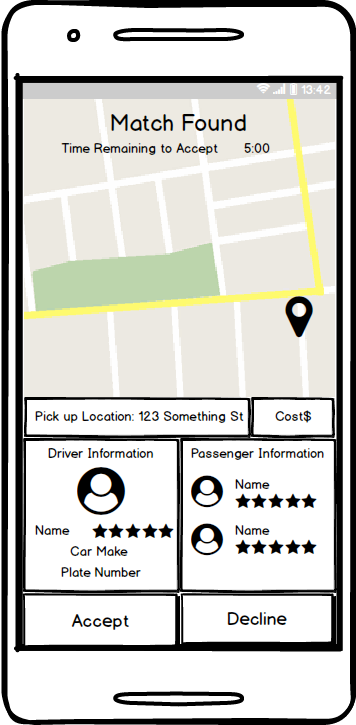
Create Account



Edit Profile

User Profile

Pre-Match Screen

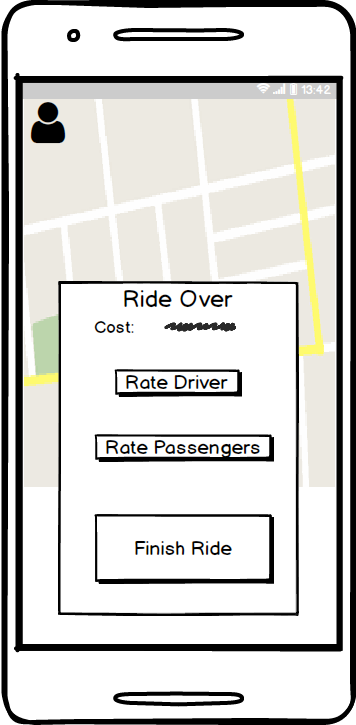
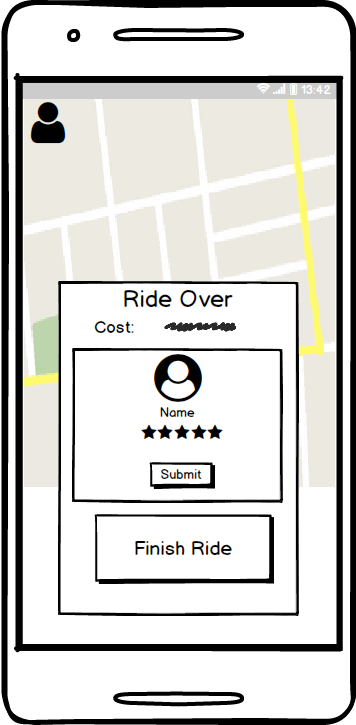
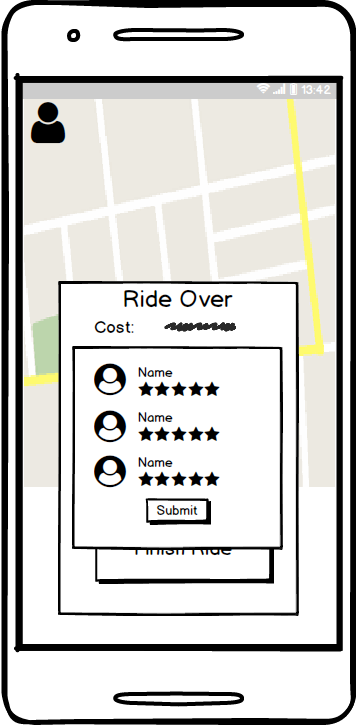


There will be a slight delay in this transition while the application references the database and find the match.

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*