

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
Problem Definition.....	1
Motivation.....	1
Contributions.....	2
Chapter 2: Background	4
Definitions.....	4
Development Software.....	4
Previous Work.....	7
Chapter 3: Main Contribution.....	12
Chapter 4: Implementation	13
Chapter 5: Conclusion.....	14
References.....	15
Appendix.....	16
Balsamiq Mockups.....	TBD

Chapter 1: Introduction

Taxi sharing, a method of transportation that is a cross between a bus and a taxi. Meaning a taxi ~~will~~ ^{can} pick up multiple people, either at the same location or multiple ^{locations} along the way, then drop them off at their desired location ~~so long as they are all in the same general area~~ ^(preferably). 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

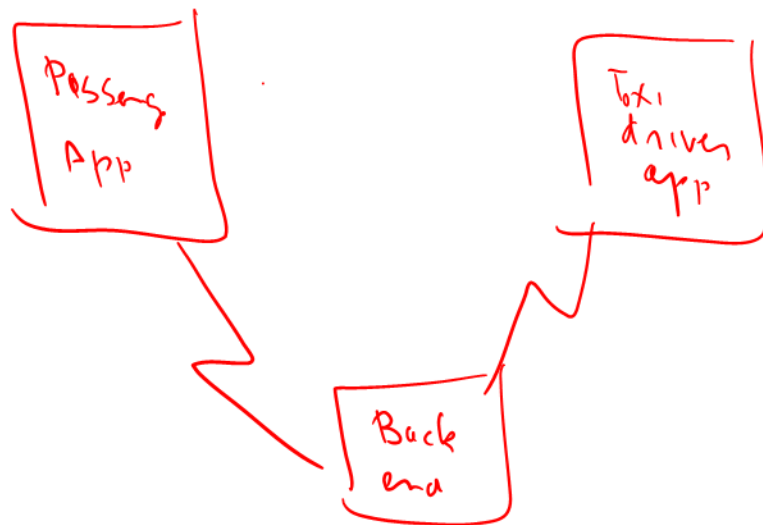
The honours project for COMP4905 ^{which} ~~that~~ was completed this semester, was to design and develop a taxi sharing application for an Android device. The focus of the project was on research and design, to determine what applications already existed with this concept and to design an application 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 ^{app} ~~portion~~ of the ^{system} ~~application~~ as well as to implement the back-end communications. The other half, which ^{is} ~~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.

Handwritten notes: The taxi sharing system consists of three → ①

Motivation

The motivation behind this project was to develop a taxi sharing ^{system} ~~application~~ that would form the base of a future project. This ^{system} ~~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

① → components: a passenger app, a taxi driver app and a back end



2 Organize your ideas

- 2.1 base for future system for autonomous vehicles
- 2.2 System can reduce the number of vehicles on the road
- 2.3 Personal level - learn how to develop apps on mobile devices and to integrate an app into a bigger system

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

Not clear

not clear too verbose

refer to a figure

information is required

in order to match passenger and taxis

The implementation

led to

and consisted of

two

parts

research

and

implementation

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.

is organized as follows Chapter 2 presents information
 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. *Chapter 3* 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.

discuss each chapter separately
 Chapter 2 - - - Chapter 3 - - -

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~~ ^{includes} 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. ^{Am} 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 ^{application flow} ~~what certain buttons in the application will do.~~ ^{For example} 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

Following the description of software used throughout the project, ~~previous work will now be discussed~~. There are numerous applications already in the field of taxi sharing that could be mentioned in this section. 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. Such as, how are the user profiles set up, 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 app will be discussed. Uber currently holds the title as most popular taxi application. It is available in 65 countries as well as in 600 cities worldwide, there are approximately 15 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, users can request a ride for themselves, or for small groups of riders and it will take them to their requested destination. The application being designed in this project is specifically a taxi sharing

application. 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, their variation of a taxi sharing app. It functions as one would expect a taxi sharing app would, a user would 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 a pick-up location, during this waiting period other users are matched to the car until the vehicle is full. The app will also display the cost of the ride before pick-up occurs as well as the estimated travel time, so 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 to 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. Looking at Uber's system it aided in determining the amount of luggage a user would be allowed to bring with them. It aided determining how to drop off riders, also in 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 complete a trip it is always approximately 65% cheaper than taking a regular Uber.

seems legit

To continue, Lyft was also used as a research tool for this project. Lyft is a popular North American taxi application. It is currently in place in 350 US cities as well as in Toronto, and Ottawa in Canada. While Uber dominates the market with 65% of all application-based taxis, Lyft holds onto a significant 31% 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 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 on going, 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 this project was trying to achieve and thus the project was implemented in similar fashion. Furthermore, similarities arose between the two systems. Such as, displaying the approximate cost and duration of the ride before the trip had begun. As well as the application determining the optimal route based on all 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 apps.

Moving away from the popular taxi sharing options available there was another application that influenced the project. Via is a rather unknown alternative to Uber and Lyft, it currently only makes up one percent of taxi sharing application use as it is currently only available in three cities. Those being, Chicago, Washington D.C., and New York City, but this was not a deterrent as a research point. Via is not technically a taxi sharing app 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 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 to be integrated into the project was the way this application deals with pick up. All passengers will arrive at the same location and will all 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 customers day by day. Users using the service on their way to and from work and rides of the sort. 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 implemented within 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 pickup 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 being accomplished without a driver in the vehicle. Passengers of this service always know what to expect as 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 need to get in contact with support there is a build 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 issue 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.

Therefore, to end this section it can be said that there are numerous different options when it comes to the application being designed by 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 all elements of the world, each new age technology will always have influences from the past. Meaning, 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.

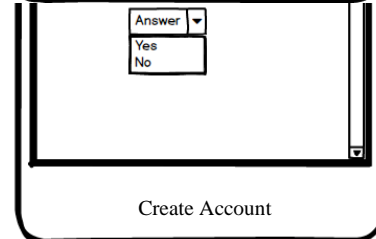
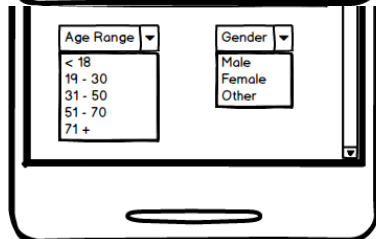
Chapter 3: Main Contribution

Chapter 4: Implementation

Chapter 5: Conclusion

References

Balsamic Mockups



Do you smoke?

Answer

Yes

No

No, but I dont mind.

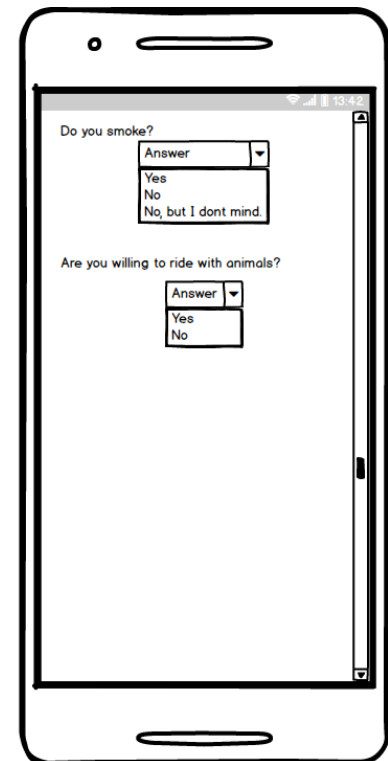
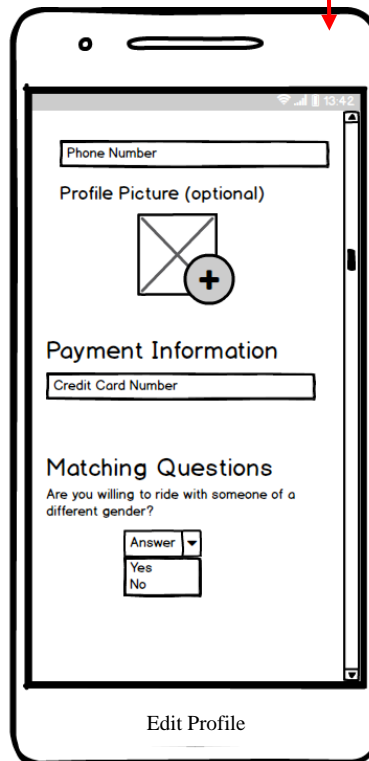
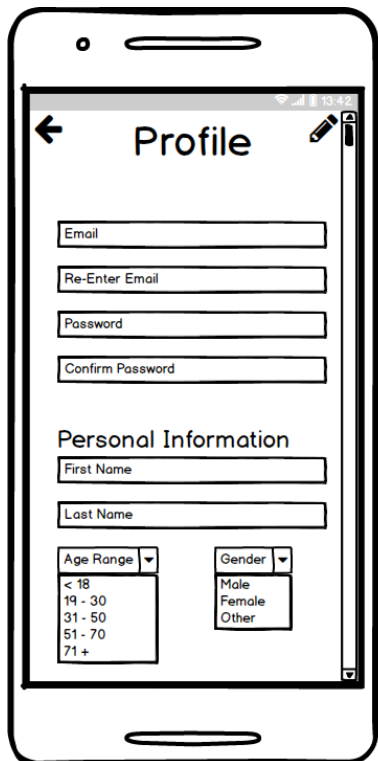
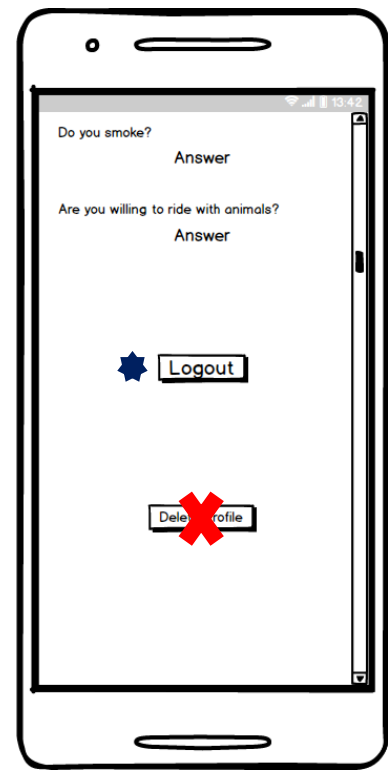
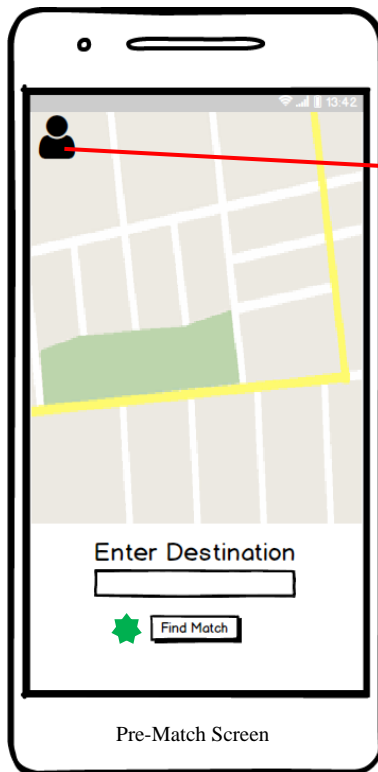
Are you willing to ride with animals?

Answer

Yes

No

Create Account



Pre-Match Screen 2

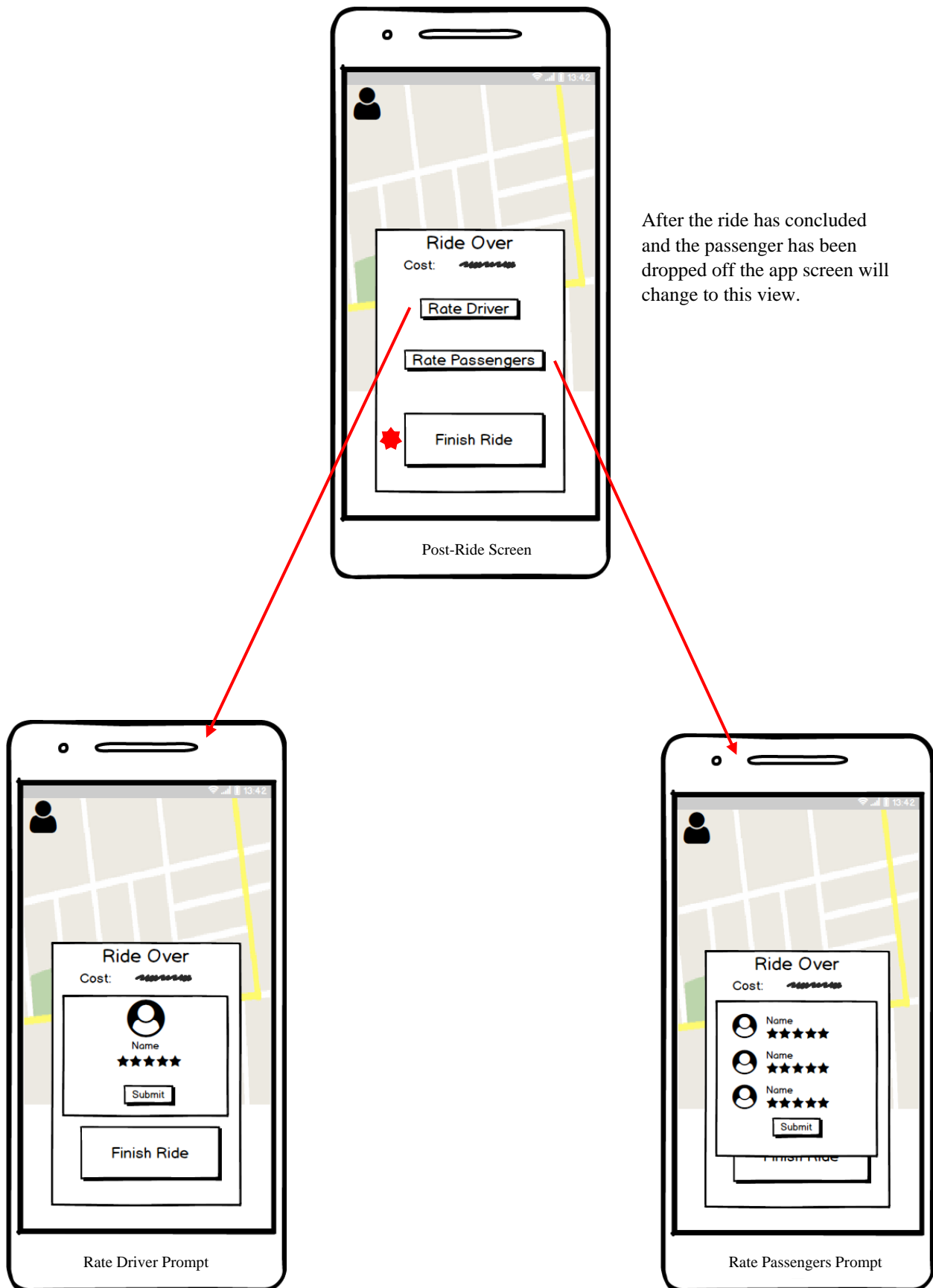
There will be a slight delay in this transition while the application references the database and find the match.

Match Found Screen

During Ride Screen

Accept Ride?

Decline Ride?



Android Studio Design