

CS 4720 - F17 - Final Project Proposal

Device Name: iPhone 6/7/8

Platform: iOS

Name: Anna Wu

Computing ID: yw6fy

Name: Justine Wen

Computing ID: jmw2vr

App Name: UVA Ride n Go

Project Description:

The goal of UVA Ride n Go is to provide UVA student with s a centralized location to exchange ridesharing and carpooling information. Travelling for the average college student has always been an expensive habit. For the students with cars, the price of gas and automobile maintenance can creep up on them the more they drive. For students without cars, the cost of ubers and taxis are astronomical, leaving them stuck in their college town. This application gives both types of students the opportunity to find ridesharing opportunities that will reduce the costs of travel for everyone.

What we propose to do is create an app that will do the following:

- The system shall allow a student to create a profile using their computing ID;
- The system shall verify that the student is a UVA student by sending an email and asking them to verify through there.
- The system shall allow a student to fill out their ride-sharing information regarding when they want to be picked up, when they need to reach a certain location, and where they need to travel to.
- This system will list all the current driver that are travelling on a certain day and their destination locations.

We plan to incorporate the following features:

- Camera - A student can take a picture of themselves to add to their profile so that drivers and riders can identify each other.
- GPS/ Location-awareness - The app will use the location of the user so that they can track where they are going on a map while the driver is driving. The driver will also be able to use GPS to find a path to their destination.
- Build and consume your own web service using a third-party platform (i.e Firebase) - Firebase was used as our main form of storage and it was also utilized for login purposes in our application.
- Open shared activity / feature - Because our application involves a lot of necessary communication between the rider and driver, we had out application open both the phone and messages application on your phone to make communication easier for the people involved.

- Device Shake - After you go to either a driver or a passenger's profile page, if you shake your phone it will automatically call the person's number on their profile. This will make it easier for drivers to call passengers when they are driving and trying to find passengers for pick up.

Wireframe Description:

Our wireframe shows the three main paths of our application. The first path is the Sign Up path which will take the user through the sign-up and personal information logging portion of our application. All information from the sign up and personal information pages are stored in our Firebase database. The sign in information is also performed using Firebase. The second path is the rider's path for our application. The rider's path includes the rider inputting their ride information and finding the best driver for their given requirements on a tableview display. We chose to use tableview because it was the easiest way to see the most drivers with the limited information that was most pertinent to the rider. It took up an appropriate amount of real estate on the phone while serving the purpose that we needed it to. In this method, potential riders can easily find more interested once they gained interest by clicking on the user. Bombarding them with too much information at once might be overwhelming. After they click the driver of their choice, they can choose to message the driver, which in turn opens up the phone's messenger application. The last path is the driver's path for our application. The driver's main focus is creating a post of their driving time and information so that riders can find them through our application. The driver's input their information through our application and once the data is stored in our application, it can then be pulled later for riders to match with.

Additional Documentation:

Platform Justification - What are the benefits to the platform you chose?

In the case of most UVA students, iOS was the better platform to choose between the two because Apple mobile devices are the more popular mobile device on grounds. Millennial mobile phone users are the population group that percentage-wise uses the most iOS mobile devices in the United States. Because our application caters to the needs of university students and is meant for ease of access for this age group, we chose to create our application on an iOS platform.

Major Features/Screens - Include short descriptions of each (at least 3 of these)

As mentioned above, our application was split into three main paths, each made up of multiple screens. The first path that our application takes is the sign-up path for new users. Users are able to sign up to use the application and their information will be saved on our Firebase database so that they can login later on. The reason users have to use their UVA computing ID is because we made this app to cater only to UVA students and faculty. By forcing users to log in to their UVA accounts to verify their email address, we are essentially performing a check to ensure all the users on our application are UVA related. The second path that our application has is the drivers' path. This portion of our application is focused on registering the information that potential drivers have regarding their travel plans and other specific details. The information that is submitted by the drivers is passed into our Firebase database and is later fetched on the rider's path of our application so that riders can find an appropriate driver. The last path

and major feature of our application is the rider's side. The riders side is the path in which users can search up matching drivers for their travel requests. This part of the application also saves the rider's requests on our Firebase database. More importantly this path fetches the information from our database to present our riders with enough information to make a decision about who they want as a driver for their next trip.

Optional Features - Include specific directions on how to test/demo each feature and declare the exact set that adds up to ~60 pts

For optional features, we include the following features in our application:

- Camera Feature (15 points) - For this feature, UVA members are able to take pictures of upload pictures from their camera roll to save as a profile picture. We included this feature in order to allow users to upload their portrait so that it's easier on both the rider and driver to find each other when it comes time to meet up. In order to use the camera, just click the "Use Camera" button on the personal information page. In order to upload a picture as a profile picture, just click the "Upload Image" button to upload and choose an image from your camera roll.
- Build and consume your own web service (i.e Firebase) (15 points) - In our application, Firebase is heavily used and relied on in order to keep track of all the various riders and drivers that are signing up and inputting information into the application. In order to use Firebase, anytime that you login, signup, or record ride information or personal information on the app, the application is using Firebase.
- Device Shake (10 points) - Device shake was incorporated on our application so that particularly drivers who are driving and cannot stop to text or type in the phone number can just shake their phone in order to call the rider that they are trying to contact. The same goes for riders who are driving to their pick up location and are looking for the driver. This makes communication between the two users easier and safer. In order to use device shake, go to the driver's information page and shake your phone. The application will ask you if you want to call that driver's phone.
- Open shared activity/features (5 points) - In order to make ease of communication easier between the rider and driver, we incorporated two different ways in which the riders and drivers can use your application to redirect to another application on the phone and contact one another. One of the methods is the one mentioned above in Device Shake in which the user can go to the driver's information page and call the user if they shake the device. The second method is our application opens and send messages. On the driver information page, click the button that says "Text Driver" and the application will automatically use messages on your phone to send a pre-populated message to the driver about discussing travel plans.
- GPS/Location-awareness (includes using Google or Apple Maps) (15 points) - This is another heavily used feature of our application that can be found on many of the pages of our application. Because our application is a ride sharing application, GPS comes in handy quite often. In this case, we used GPS to show the riders and driver the exact locations of where they were going and starting. We also used Google Places in order to allow the user to type in common names instead of having to find addresses or GPS coordinate for their destination and starting point. On the last

page we also created a map in which the users can see the fastest route mapped out on Apple maps with two pins marking their start and end location.

Testing Methodologies - What did you do to test the app?

In order to test the application, we had to collect computing IDs from different students and had to have them register so that we could input pre-populated variables to test if all of our features worked. After having students register with their UVA e-mail, we then used their accounts to create dummy drivers for the potential user to sign up for rides. Because our application is heavily based on the use of both drivers and riders inputting information into our application, as of now we only have a couple test cases just because inputting that many dummy accounts would take too much time. A lot of the testing we did happened during the process. In order to figure out what key parts were missing in our application, we had to constantly be testing to see what features we wanted to include. It was also important in highlighting features that we did not need.

Usage - Include any special info we need to run the app (username/passwords, etc.)

One of our features in the application is a shake to call feature for drivers or riders who may be driving to/from drop off or pick up locations and cannot take the time to type out phone numbers. This feature can only be used on a phone, so if you would like to test it please use a phone to try it out. We also only prepopulated a couple dates with drivers. December 11th is the only date pre-populated with multiple drivers available. If you don't want to prepopulate yourself, you can test on that date.

Lessons Learned - What did you learn about mobile development through this process?

The most important lesson that we learned throughout the process is that things are harder than they seem. Like the guest speakers from WillowTree said during class, although something may look simple (such as login), there is actually a lot of effort and time that has to go on behind the scenes in order to make certain features available. Some of the features that took the most time and energy to add to our application were Login/Signup, GPS/Maps, and TableView with the incorporation of pulling information from Firebase. We also learned a lot about the importance of not only making a wireframe but also constantly updating it in order to keep track of changes that you made or would like to make. Often times we would begin coding features that we later on realized were not as important or as necessary as we thought at the beginning. We ended up spending a lot of time and effort on these tasks and because of this, we realized the importance of wireframing. On the more technical side of the project we learned a lot about different features of mobile development, especially about login and signup and the use of Firebase and GPS. Lastly, we learned about overloading ourselves. The scope of this project was too much for our team to handle in the given amount of time, but it was a good learning experience. In the future we know more about how long and how much effort it will take to code up different features and we will have a better grasp of just how much we can cover in a given amount of time.