Sam Borowsky

BUS 403 – Blog 2

Over the last couple weeks, our group has made significant progress by implementing some important features and defining exactly how our app will operate. However, there remain many programming and technical challenges that stand in the way of us releasing a polished and fully functional app. In terms of app development, at the time of the first blog post our group had incorporated login/logout/register functionality and a main storyboard scene that appears when a user successfully logs in. This storyboard scene provides the user with the options, “Get Scooped” (meaning “get a ride”), “Scoop Up” (meaning “give a ride”) and “My Account.” For some time we have been working on implementing the “Get Scooped” feature, which is supposed to present the user with a table of available drivers. Our goal has been to simulate the live version of our app by filling the table with sample driver data. From Professor Fox and a helpful YouTube tutorial I found that creating a table view linked to the “Get Scooped” button is quite easy. To do so, I simply dragged a “Table View Controller” template into my storyboard and connected the “Get Scooped” button and the new table view with a segue. I created a class, “DriversTableViewController,” to manage the table as a whole and another related class, “DriversTableCell,” to format/manage the individual table cells. I’ve used arrays, in the DriversTableViewController.m file to hold the sample data that populates the table, which is the drivers’ names, the makes and models of the drivers’ cars, images of the drivers, the drivers’ individual ratings, the availability status of the drivers, and sample driver distances to the user. The goal is to only select drivers whose availability status is set to yes, and then display these drivers in the table, ordered top to bottom by proximity to the user. However, this has proven to be very challenging and problematic because I am a complete novice when it comes to working with arrays and conditional statements in objective-C. Thus, I am stuck at the point where the app builds a table view that displays all registered drivers, regardless of their sample availability statuses that I have set. Additionally, to polish off this feature of the app, the table must have a fitting title (such as “Available Drivers”) and an arrow in each cell to expand to a driver’s full profile, which will include reviews of the selected driver along with other extra information. This being said, I am encouraged by the fact that we have successfully implemented the concept of translating driver information into a table display that the user will use to secure a ride. I have attached to the end of this blog a photo of the available drivers table in its current state.

With respect to how the app will operate, our group has recently determined how to manage the rider-driver relationship and registration and connect the features of the app with an appealing layout. One major question we deliberated for weeks was how to establish a pricing scheme for rides of various distances. After discussing this extensively with Professor Fox, it is apparent that the best approach is for us to behave as an intermediary party between the riders and drivers, who will make a deal independent of us. In this way, our app will simply locate users who need rides and direct available drivers to the riders. For this service, we will charge drivers some sort of weekly or monthly fee. Other dilemmas we have faced are how to verify that the app’s users are W&L students and whether to allow a user to be both a rider and a driver. The most feasible solution to the first issue seems to be that the final step in the registration process requires the new user to authenticate his or her W&L email address. We prefer this method to trying to verify the registering user’s provided W&L ID number with a database of all W&L ID numbers, which is a possibility we initially considered. Regarding a user’s rider/driver status, we want to permit him or her to be both. One reason for this decision is that we feel it will generally reflect the mentality of our users. In my case, for example, I would definitely want to make some money by driving on Wednesday and Friday nights, because as a member of the tennis team I am limited on those nights by morning workouts on Thursday and Saturday. However, since Sunday is our off day, on Saturday nights I would prefer to go out and have this ride service at my disposal. Additionally, a primary challenge we naturally face in building this app is how to actually connect riders and drivers. We understand that this requires location tracking functionality and the use of databases. In a live version of the app, when a user selects “Get Scooped,” the app must first compute this user’s location. Considering that we already are somewhat familiar with the core location framework, we feel this task should not be very problematic. Next, the app will access a database of available drivers and find these drivers’ distances from the user who requests a ride. This part of the process is particularly difficult, because at this point we hardly have any sense of how users will indicate that they are ready to drive. One method we are exploring is that a user communicates that he or she is an available driver by clicking “Scoop Up,” and then will be displayed in the available drivers table of the user who requests a ride. This means that we need some way for the drivers to indicate that they are finished driving. Furthermore, we must learn how to create and interact with the database such that it always holds the current available drivers. Finally, once a rider selects an available driver, the driver will be notified that he or she is on duty to pick up a customer at a certain address. Our group recognizes that we must work extremely hard to implement all of this functionality in the short span of time we have left. Accordingly, while I will continue to be the main programmer of our app, we will find a reasonable way to share the labor and responsibilities over the next couple weeks.