Sam Borowsky

Blog 3 – Scoop Update

Since last week’s assignment demanded our app’s full architecture implemented in a polished and professional manner, our group has been putting forth a considerable effort to finalize the aesthetics of our app. This process of visual enhancement has taught me a significant amount about app design in general, and specifically which built in XCode objects are ideal to use in particular circumstances. For instance, a couple of Professor Fox’s requirements for last week were to eliminate white backgrounds and upgrade buttons that existed in their default settings. These guidelines were relevant to our group because we had not yet decided on a background scheme for our app, and many of the buttons in our storyboard were disorganized and generic, functioning only to direct the user to the next appropriate view. Accordingly, we experimented with different backgrounds, and ultimately decided that a light gray background is effective for all the pages that are not dynamically generated table views. We like the gray background because it is simple, sleek, and aesthetically pleasing. Implementing the gray background exposed a new problem, however, which was that the white backgrounds of the images we placed on top of the view’s gray background now became visible. Removing the images’ white backgrounds required some playing around with and research into Photoshop. Next, we moved onto refining the home page that appears after a successful login. This page features the Scoop news feed, which is a dynamic table view that displays information about the most recently completed rides. For example, one sample row in this table has the form, “Sam B. got dropped off at The Duchossois Tennis Center 3 mins ago,” and is complete with “Like” and “Comment” options. We want the user to navigate through the rest of the app from the home page, and Professor Fox demonstrated to us how the easiest way to accomplish this is by using a tab bar controller. In addition to allowing the user to easily access each page in the application, the tab bar controller is extremely helpful because it hides the complexity of the segue connections within the storyboard. Although the process of making tab bar icons show up proved to be tedious and frustrating, we now have a professional looking tab bar that indicates to the user all the functionality the app offers. I have attached a picture of the home page to the end of this blog. In terms of aesthetic design, the two tasks that remained were developing the views that correspond to the “Scoop Up” and “Profile” tab bar items. To be clearer, clicking the tab bar item “Scoop Up” directs the user to the driver status/settings page, and clicking the tab bar item “Profile” directs the user to his or her profile for viewing and/or editing. Our approach to managing the available drivers is to add the user to a database of available drivers when he or she selects “Start Scooping,” and then remove the driver from the database when he or she selects “Stop Scooping.” Professor Fox suggested that this binary system, where drivers indicate whether they currently are/are not driving, is satisfactory for the MVP. Finally, we designed the “Profile” page in an organized and visually appealing manner. This page features labels for first name, last name, email, phone, and home address information, a user profile picture, and “Edit,” “Done,” and “Sign Out” buttons to update information, save updated information, and log the user out of the app, respectively. One more notable idea we are pursuing is coordinating color schemes with the various universities that (hopefully) will be using the app. For example, the predominant color of the W&L version of the app is our school’s royal blue. In this sense, our group has made important progress with the app’s layout and aesthetic character. The remaining challenges we face include implementing internal storage for user data and external (MySQL) storage for data pertaining to available drivers, a history of completed rides, etc. At this point, I have already embarked on figuring out how to incorporate internal storage. Professor Fox explained to me that the best way to permanently store user data is through the Info.plist file. Although I think I know exactly how to interact with Info.plist to handle the app’s login and register functionality, the code I have written to do so seems to have broken the app upon launching. Of course, I will try to resolve this problem as soon as possible. Also, I need to learn how to make the labels in the “Profile” page user editable, so that the user can update (and save to Info.plist) his/her personal information. I anticipate that firing up the external storage should not be too problematic, considering that I have experience working with MySQL databases and the external storage system our app requires is pretty basic. Overall, it has been exciting to see how the various components of our app are coming together, and we look forward to having it ready for the App Store within the next couple weeks.