Robert Rhett

Professor: Dr Paul West

Java I with Elementary Data Structures (COIN-325-40)

30 April 2015

Android Port of Login Application

Background: The Login application is designed to be used as a front-end user interface that can be associated with any application that requires connection to a remote database for the purposes of user authentication. In addition to providing login authentication, the Login app provides the user with a self-service method for creating an account, changing the password, and recovering a forgotten password. There is also a check box that allows the user to have the application remember that the login was previously authenticated on a given device such that the username and password will not have to be re-entered each time the app is run on that device.

The biggest change in design between the Java implementation and the Android implementation has to do with presentation management. In Java, the swing toolkit is used to manage the GUI presentation and create the text boxes, buttons, etc that are used to implement the user interface whereas the Android activity vastly simplifies the implementation (at least of this particular project) through the use of activities to define the graphical layout.

The UML diagram created for the java project is still, for the most part, valid. Conceptually, creation of a User object as a container for the information about the current user is necessary as are the Database object with its connection methods and the Account object to provide methods for setting and getting the password. These were re-used, essentially, as-is. On the other hand, in the Android environment, many (if not all) of these objects are probably available as part of built-in toolkits such that manual creation of the objects in code might be unnecessary. Nevertheless, as with most toolkits, they may not provide the exact user interface that one desires so it is beneficial (if not educational) to understand how to create the classes for oneself.

The most difficult part of the transition to the Android environment is connecting to the remote database for the purposes of validating the login credentials. In java this is accomplished quite simply using the mysql/jdbc driver as follows:

Class.forName("com.mysql.jdbc.Driver").newInstance();

conn = DriverManager.getConnection("jdbc:mysql://" + dbname + "?user=" + mUserRO + "&password=" + mPassRO);

Apparently, in Android this process is somewhat more complex. I viewed a couple of videos and read several tutorials on the subject. These utilize JSON parsing and require setting up a webservice (which I have never done before) to access the database through php programming (which I have done before). When I tried implementing this method I discovered that this must have been old material because all of the functions were deprecated. Additional research took me to the newer (only two years old) Remote Database Connector (RDC) which looks like a much simpler method for connecting to the database (although I still could not get it to work). Tutorials on this subject were sparse and still very difficult to understand (at least at my level of knowledge) so I gave up on accessing the remote database.

I would say that approximately 25% of the time I have spent working on the project to port this application to the Android environment has been spent in getting the IDE loaded, updated, and working with all the necessary add-in emulators and such. Another 10% of my time has been spent simply waiting for the code to execute on the emulator (while I have Windows 7 running on a 3Ghz Core2 duo processor, I only have 3Gb of memory. This has been a severe limitation). The major portion of my effort (approximately 50% of my time) has been spent attempting to make a connection to the remote database (at which I was unsuccessful, although I was able to access the database manually, through the browser running on the emulator, so I know it is possible[[1]](#footnote-1)).

The easiest part of the entire process was re-creating the user interface utilizing the Android activities to define the layout, buttons, text boxes, etc. If there is one thing that Android Studio is good for, it is facilitating creation of an attractive user interface.

I am disappointed that I was not able to complete the project as I set out to do but I am satisfied that I have done the best I could given the time and resources at my disposal. There is no question that understanding mobile application development is essential to any successful commercial application implementation and the more tools I am exposed the better. As you may have guessed, this project is not simply an academic exercise for me. I currently have a commercially viable web-based application that has a good deal of mature back-end programming and that is currently in use by several businesses. My goal is to port this to a mobile app and I was really hoping that I this project would provide an opportunity for doing so. Still, the work I have done in this course has been valuable in moving toward this goal and I will continue to push toward completion through the use of the tools and knowledge I have acquired in this course.

1. I am delighted to see that you actually created a user account in my test database using the application I wrote. I only wish I could have provided you the same opportunity on a mobile device. [↑](#footnote-ref-1)