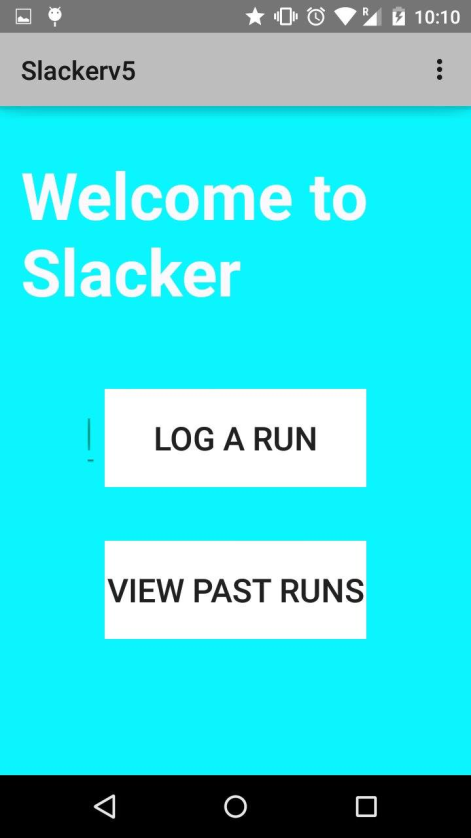
**Criterion C: Development**

I used Android Studio to develop my application. I had 2 classes with code written in Java and 3 XML files written in XML for the display (the database did not have a corresponding XML file). The majority of my project focused on the creation and implementation of the SQLite database. However, other classes and their XML files were made to transition the user in the process of entering and saving data. I attempted to make this application as simple as possible. All sources and resources used to code this project is presented in the appendix.

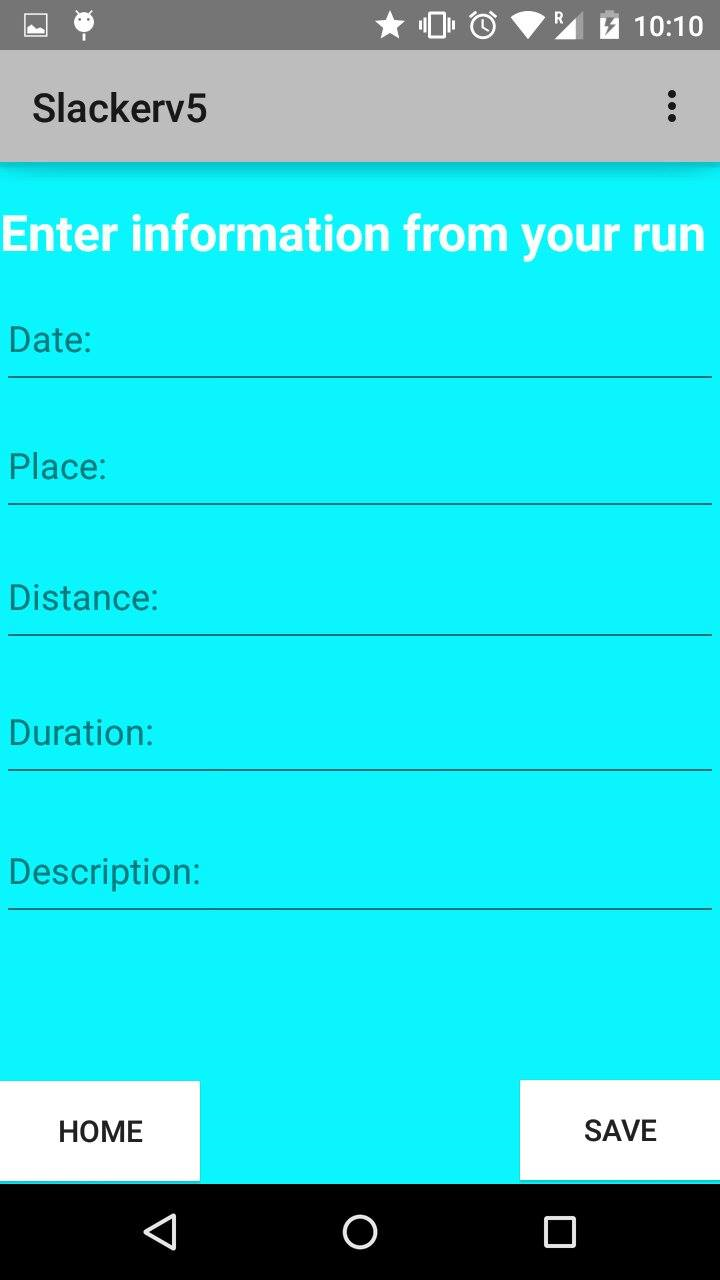
Start Screen

The start screen has a simple 2 button layout. The user is greeted with a Textview welcoming them to the application then followed by 2 buttons. These buttons were given specific IDs in the XML file. A method in the corresponding Java file connected the XML and Java class so once the button is clicked, the method would be called to display the next page/class. Again, the start screen and the rest of the application was made to be as simple as possible, therefore, I only created 2 buttons for the start screen.



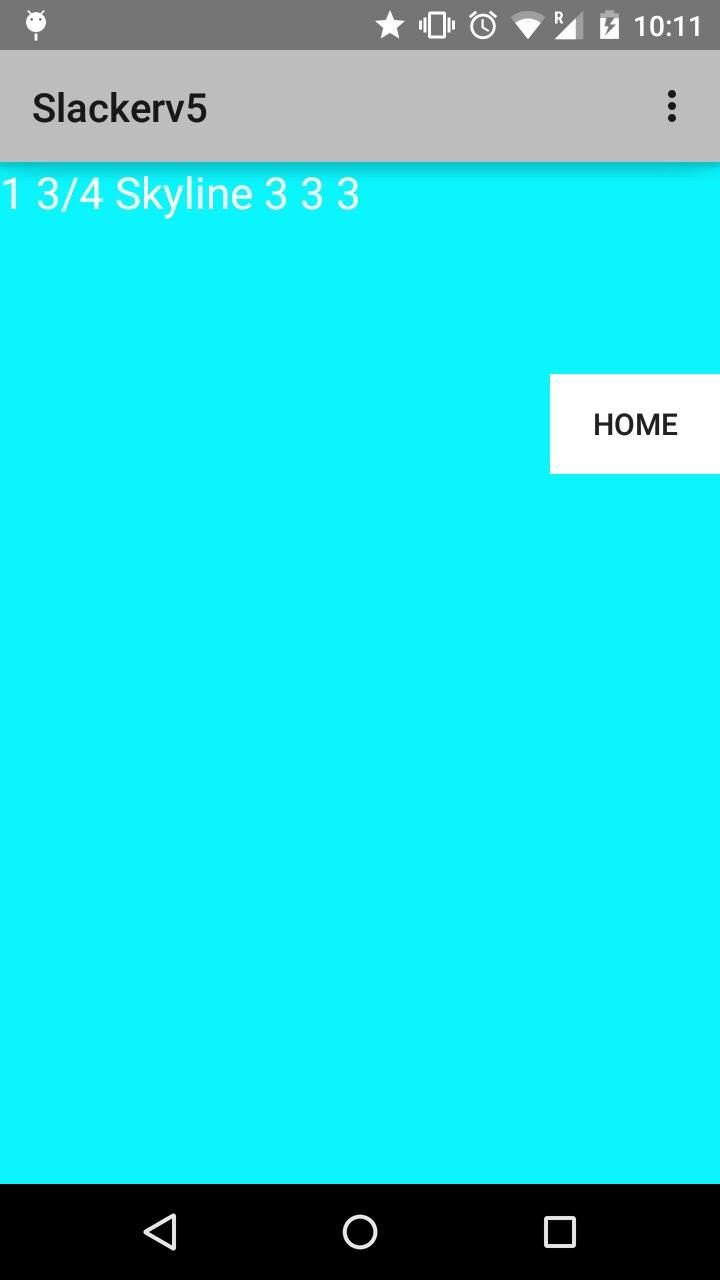
Log a Run

The log a run xml page was used to collect user data. There are five editTexts in which users enter data from their run. On the bottom left, there is a home button which allows the user to display the startscreen when clicked. Also, on the bottom, there is a save button. What the save button does is that it compiles all the user data entries into a string and puts it into the database. After the save button is pressed and the functionabilty is run, there is a toast that displays a message that notifies the user that the data was successfully saved.



Previous Runs Page

On this page, the data from the previous log a run page is displayed. This is easier said than done. Once the data is uploaded to the database, the previous runs page runs an method that breaks up the string, and displays the broken up data from the user which was imputed in its original form (discussed later). This XML page simply displays the data in a list form for the user. A sample log is shown below for a run at Skyline, on March 4, and with a distance of 4 miles. Again, a home button is shown on the bottom right for ease of use for the user in switching between pages.



Database and Algorithm

The majority of my application is based on the utilization of the database. I used a SQLite database because according to my advisor, a SQL database is too large and inefficient to handle such minuscule data values while in my own research, a shared preference method of saving data was to small and inefficient to multiple data values. This database is a virtual table with columns with titles of UID (an integer representation of each data entry), date, place, duration, distance, and description. I made the UID increase by one each time it was accessed so the user’s data would be entered in a new row. After the database was created, a method called insertData() was called so that the date, distance, description, duration, and place would all compile into one string. All this is done when the user presses the “save” button on the log a run page.

To upload and display the data, another method was created and called to parse through the string of data into textfields. This method is called getAllData() and incorporates a while loop and runs until the cursor runs through the entire string. This algorithm parses through the data, which is in an array form, and separates the date, place, distance, duration, and description from each other to add spaces in-between each value so that it is visually appealing to the user. Once the formatting is correct, it is then put back into a string and into a textview using the setText() method.