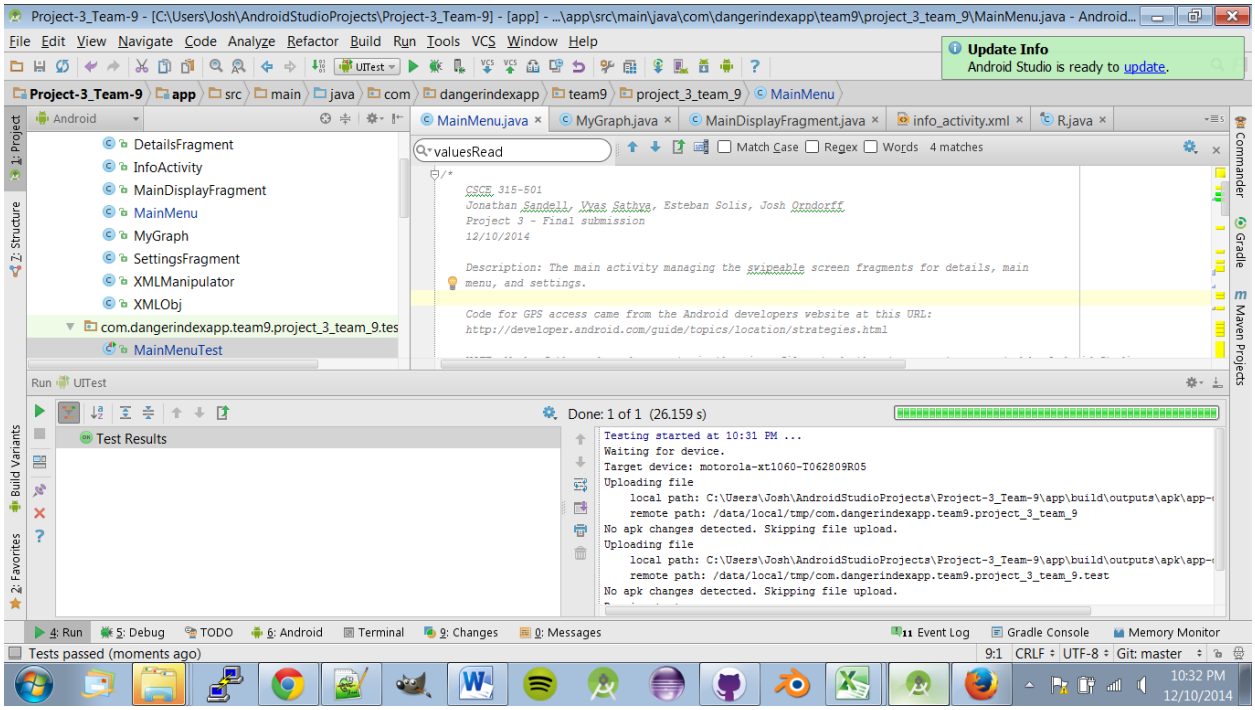
CSCE 315-501 Team 9
Project 3 – Android App
Jonathan Sandell, Vyas Sathya, Esteban Solis, Josh Orndorff
General Report (including Test-Driven Development)
12/10/2014

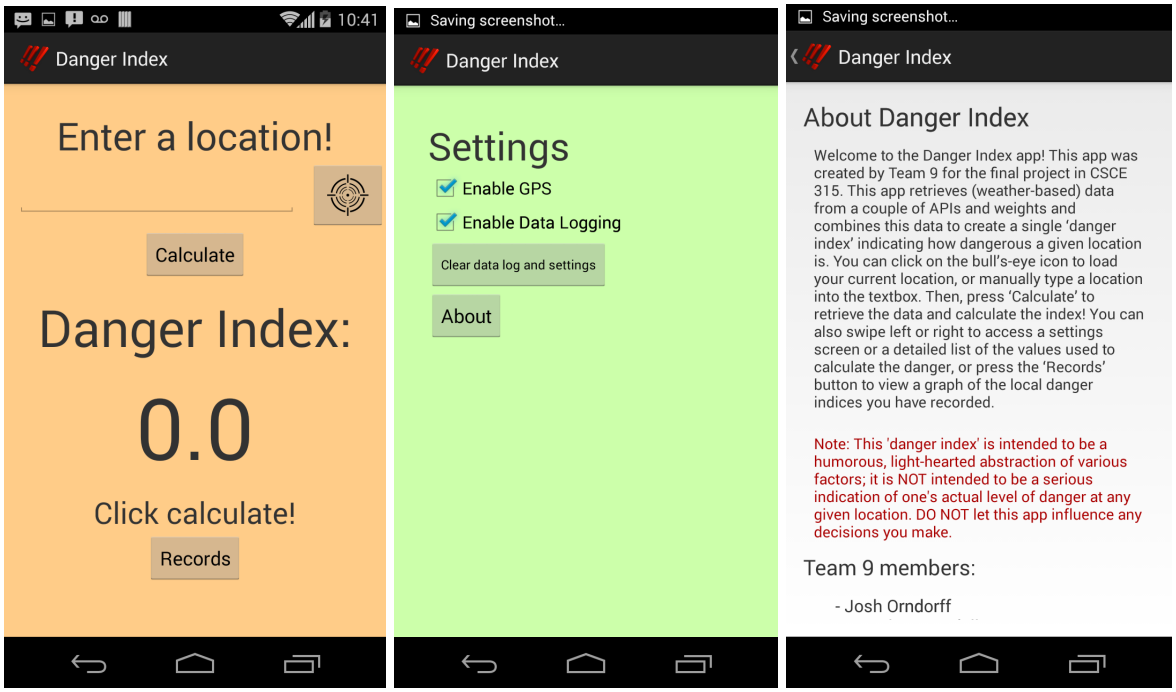
In the completion of our application, we utilized the methodology of test-driven development in order to get practice using professional development techniques. For the most part, we attempted to develop tests before implementing a feature so the feature could then be implemented and satisfy the test, although we made some of the feature tests retroactively (after the feature had been completed). We began with a MainMenuTest class which has a function that tests the various features using assert statements, and built on that test function as the project progressed. As of the last submission, our test function includes tests for features such as GPS coordinate loading from the current location, retrieving data for a manually-entered location, operating on XML objects (including string conversion, saving and reloading), retrieving and reading API data, and changing to a new activity. Since the last core algorithm deadline our changes were mostly minor adjustments, such as changing the background color due to our danger rating, changing to a colored bar graph from a blank line graph, adding the functionality where you can only pull from the current location once, inserting an about activity, and inserting a sound notification. Therefore, we do not have any new tests since the last deliverable. However, a snapshot of the tests being passed is included below:



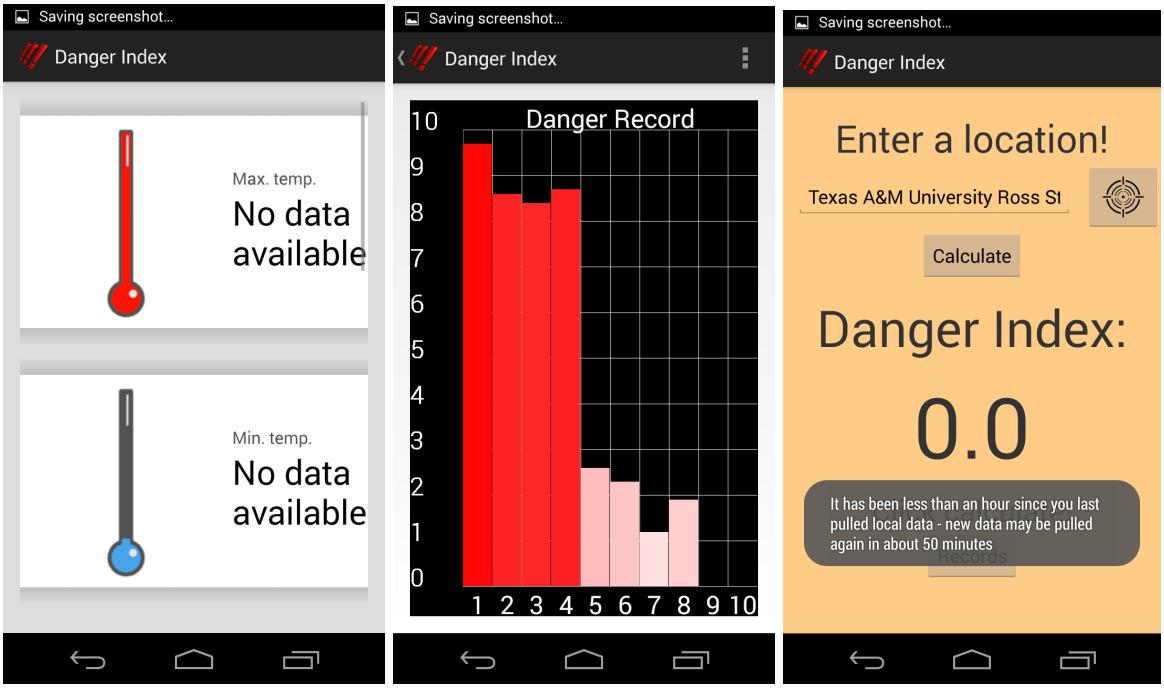
In the development log we did not document the exact times when we did create functions for test-driven development but for most major functionalities (especially those added after the first code deliverable), we tried to initially create tests before creating the full function that implemented

the operation. Some examples of times we refactored our code include adjusting the format for the graph from line graph to bar graph and changing the type and structure of classes used for XML retrieval, eventually settling on the XMLManipulator and XMLObj classes - initially, there was an XMLResult class in addition to XMLObj, but it was deemed unnecessary and its functionality was rolled into XMLObj. Also, another refactoring shift that came early on was the change from saving an XML file in the project ahead of time as a resource to dynamically creating it through the XMLManipulator and XMLObj classes. One example of when Test-Driven development was useful was the test for data retrieval failing after we had made many changes to the retrieval approach - this failure pointed out a logical error in our code which we were then able to fix.

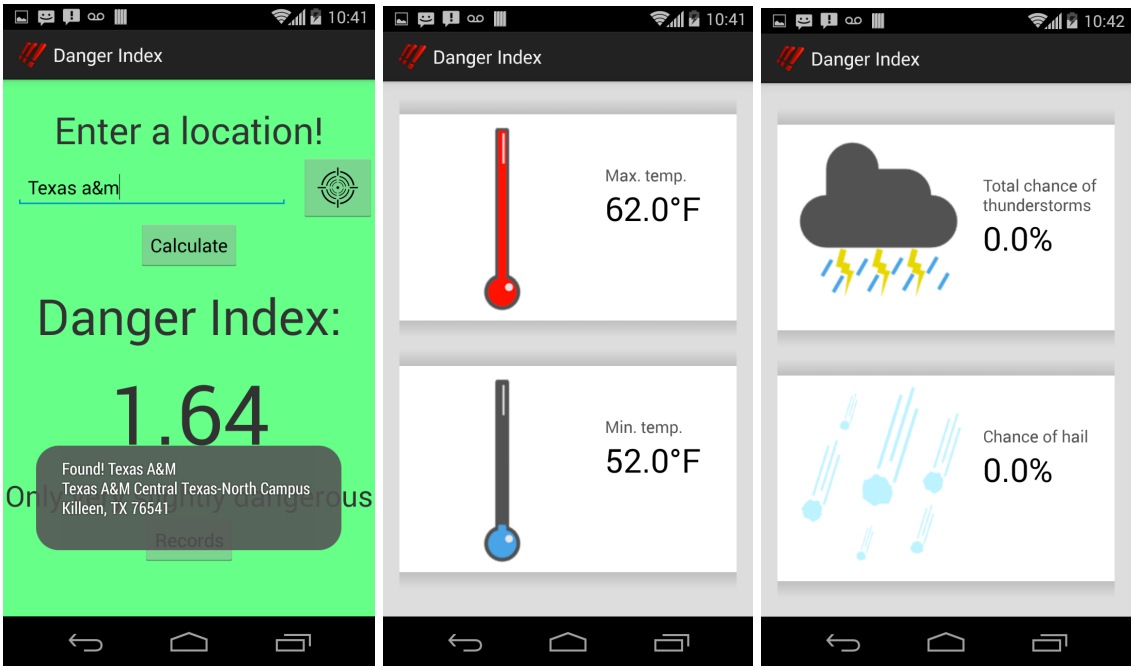
Below are screenshots of the app's final execution:



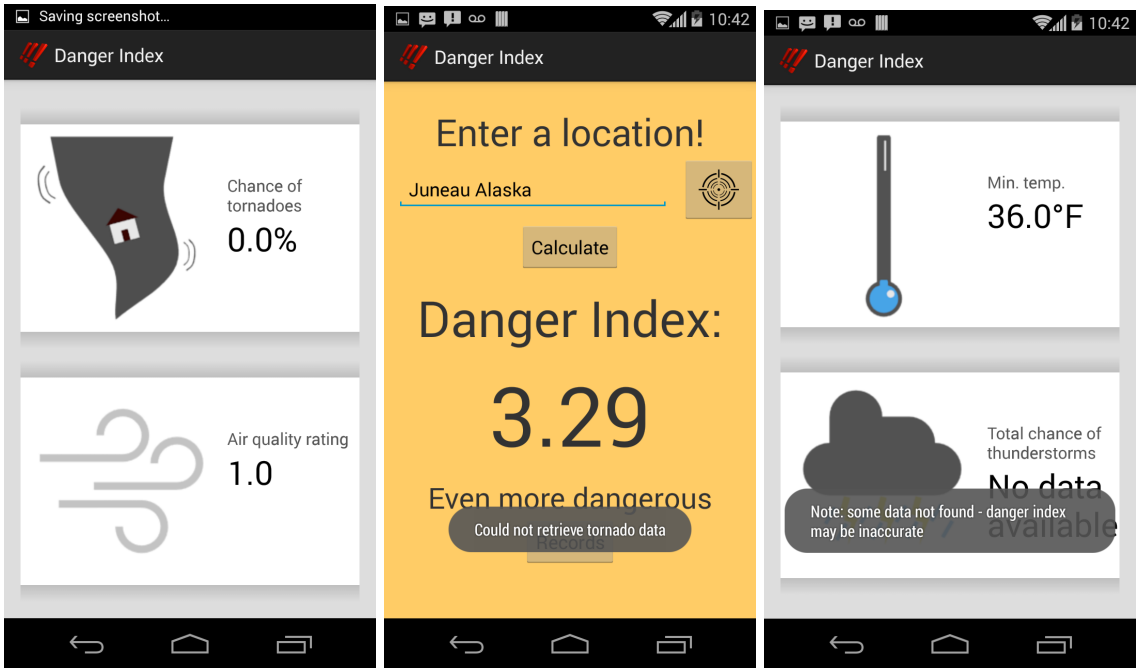
Initial screen on loading, settings view, about section



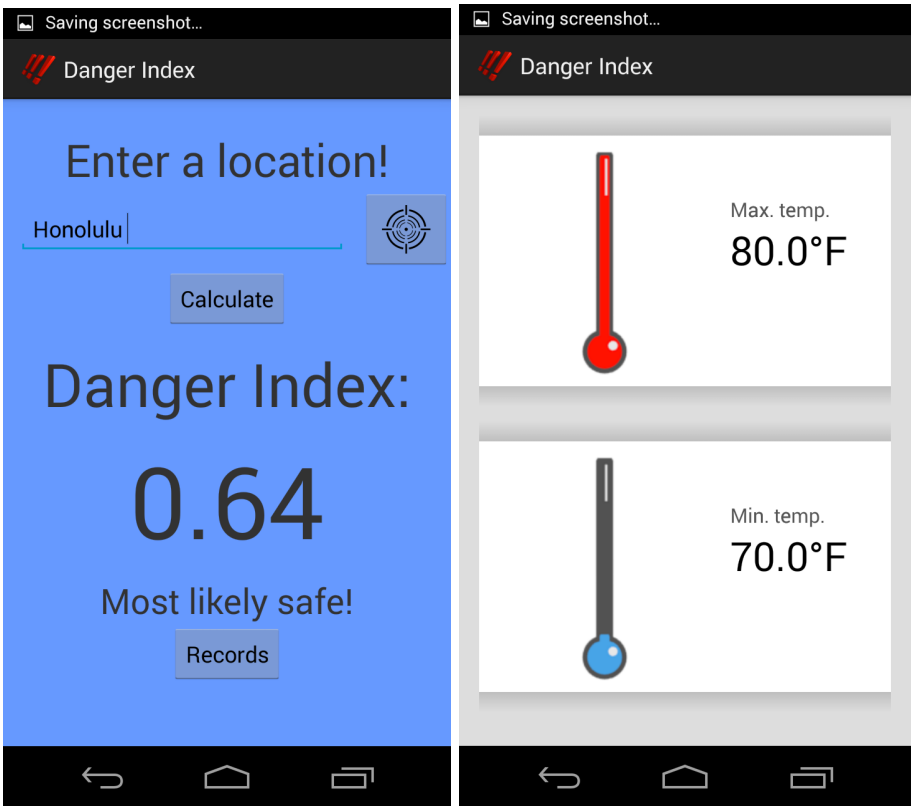
No data yet in details view, records view (with stand-in data), local pulls limited to one per hour



After pulling data for “Texas a&m”, details view, more details view



Even more details view, some data missing for “Juneau Alaska”, details view with missing data



After pulling data for “Honolulu”, great temperatures in Hawaii

Lastly, we were not able to implement XML validation in our app, however we did create a schema and were able to use an external XML validation site to confirm that our schema worked. We paused

the app execution at a breakpoint so we could extract the XML structure for a saved user data file, and then copied the contents of that XML file, along with our XML schema, into http://www.xmlvalidation.com/. After running the test, we received confirmation that our schema matched the data file.