Kevin Venalainen - Gavin Servai - Calvin Chan - Michael Sargent

#2453103 -

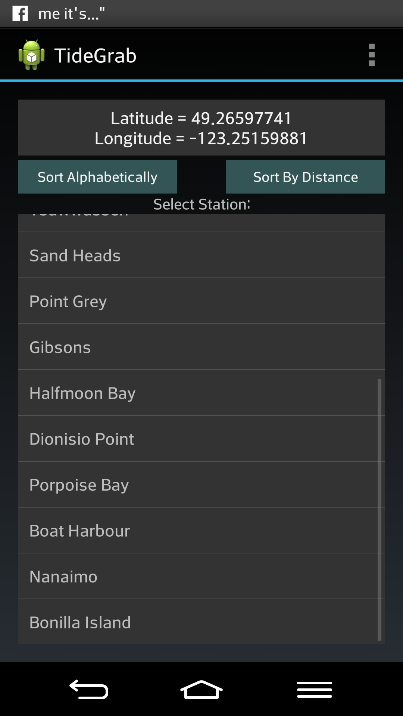
TideGRab

Tide Display for Android

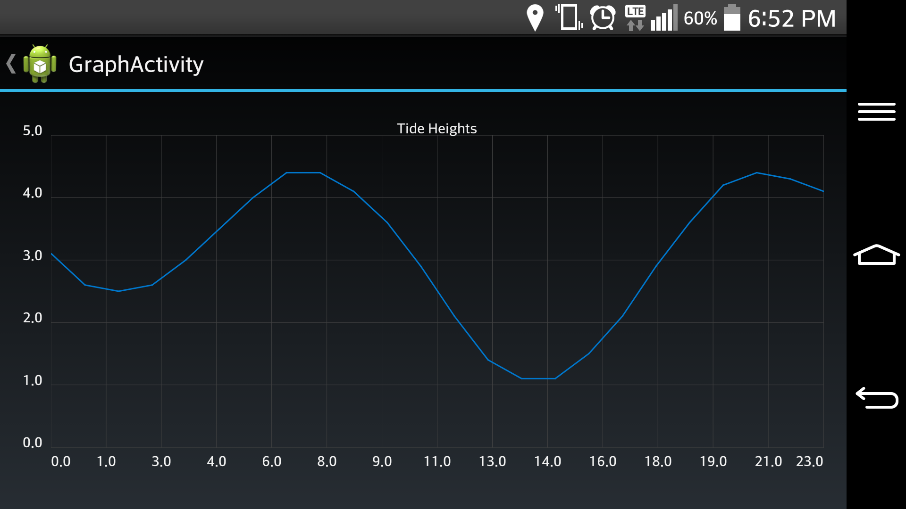
# Code:

All code can be found on the team’s GitHub at: <https://github.com/Kjven/eece310_app>  
  
Summary:

Our project is an Android app which enables the user to easily see the tide heights for stations around their location. When the app is opened the user is greeted with a page displaying their current location and a list of stations. The user can then select a station, sort the list by closest location, or sort the list alphabetically. See figure below for screenshot.



Once the user selects a station, the tide heights are displayed as a sine wave on a liner plot. The user can scroll and zoom this plot to get more accurate info. Tide data for up to 5 days is cached in the event of a bad internet connection. Originally, we planned to display the tide height in a radial plot, however due to licensing concerns we had to change to the linear view. The plot below shows the new graph style that we used.



Deliverables:

Deliverables for this project are simple, just the android app itself. It is local to our PCs and must be run in development mode as we haven’t released to the general market.

# Development Summary:

We used GitHub for source control, with each developer working in their own branch which was then merged with master. Git is difficult to use for new users so we had some issues which resulted in merge conflicts. In future, we need to do more stringent origin fetches and rebases. Towards the end of the project, we all sorted out the Git issues and were able to use it with more success.

Development was completed entirely using Eclipse with the Android SDK plugin. The app is written entirely in java as a native android app. This limited our reach to android only as opposed to android and IOS.

Primarily used waterfall for development. We had clearly defined objectives and requirements which allowed us to expedite the early stages of planning. Work was divided amongst developers and each person developed then integrated their code. Pair programming was also used in the end to help speed integration of the pieces of each person’s work.

Since the primary functions of our app are all graphical, we were unable to find a way to run unit tests. All testing was carried out through the interaction with the app on an actual phone. Each developer had a phone which allowed us to all test concurrently.

Halfway through the project, we were forced to change the graph style from a radial plot to a linear view. This was due to patent concerns and unfortunately took away much of the novelty of our app. To accomplish the new style of graph we were able to pull in an android library called GraphView to handle the graphing.

The app implements caching in order to enable the user to get tide data even when no internet connection is present. When tide data is downloaded, 5 days are parsed and stored in the local storage of the app. Additionally, we have implemented GPS in order to find the closes station. We found his challenging as we are developing across different versions of android (4.0 to 4.4.2).

Care was taken to make the app modular, which was a help in enabling us to integrate the individual components. A UML diagram is included on the next page describing the class structure of the main classes.

