**Summary:**

Matt:

During this iteration I worked on setting up the basics of the UI for the application, the Parks object itself, and methods of getting the location from the end user.

I implemented the location methods in a class called LocationWrapper by using the Façade pattern. What initially seems like a simple question (“What is your location?”) turns out to be a complicated task on the Android device. Several different libraries needed to be used (Geocoding, Play Services, etc.), so I simplified the interactions with these libraries into two method calls that can be used anywhere in the application and should suffice for all of the location needs of the app in its current state.

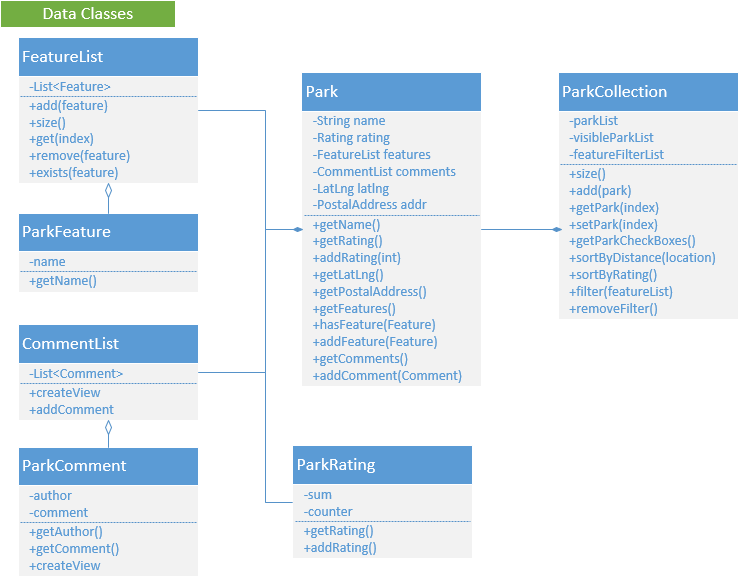
One design pattern I would like to employ (although I don’t know if I have time to accomplish it) is the Factory pattern. In the AddPark, DisplayParkCollection, and DisplayPark activities there is a significant amount of code dedicated to generating the user interface, and a lot of it could be reused. I’m not having a lot of success working with the XML files directly, although this could be my misunderstanding of the tools. I think in general this code could be simplified in a Factory responsible for creating view elements. I foresee that the code used to create views will need to be variable across devices. Encapsulating the view creators in a separate class will provide flexibility for varying device display requirements. I only have a single device to test the code on, so this hasn’t been an urgent need up to this point.

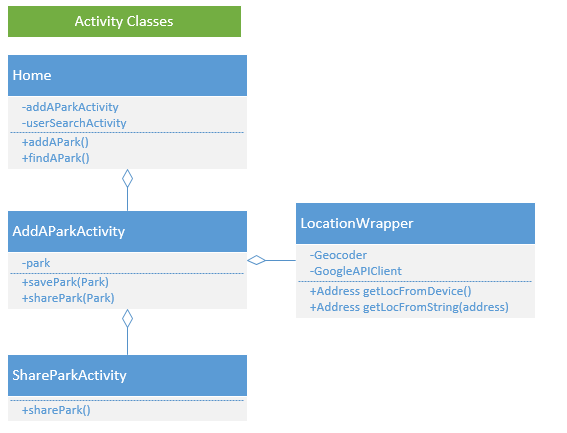
Dheeraj:

During this iteration I worked on setting up the database to add the park details that are collected in Add park activity and retrieve the data upon request from the user in find park activity.

        I have implemented a Database operation java class which gets inputs from the add park activity and the Find park activity. As the front end code has written by Matt I had to change some of the methods to be compatible with my database code. As the database methods are different to change I have followed Adapter pattern, resulting in some private methods acts as adapter between the Matt’s methods and the Database methods.

        Now I am currently working on retrieving data to the display parks on selecting a specific park which builds our prototype.





**Plans for next iteration:**

Tasks:

* Implement FindParks and UpdateParks methods for the SQLite database.
* Implement the DisplayParkCollection and DisplayPark activities.
* Develop sorting and filtering methods for the ParkCollection.

To be completed by end of semester:

We plan to have a working prototype of the application. The application will be able to gather data from a user, store it in a database, then query the database by location and search and sort the results. The application will have social media sharing capabilities.

There are a few things we will not be able to complete this semester in order to finish the product. We will not have a networked database. During planning we did not realize that SQLite was designed for use on a single device, so for now the app just records a personal database rather than a crowdsourced database. We will not have the parks displayed on a Map. The android interface took longer to learn than anticipated. For now the DisplayParkCollection will just show a list of park names and addresses rather than a set of points on a map.