CSCI 4448 Final Project
Andrew Gordon, Tommy Hoffman, Dan Zurawski, Conor McGuinness
11/9/2015

3) The primary difference in our class diagram was the implementation of the DatabaseHandler. This was largely due to our lack of discussion on how android handles data persistence. SQLite is native to Android, and there is an implementation of many SQLite functions provided by external libraries. Other differences were the addition of the DB in the class diagram, and the lack of user login.

The class DBHelper extends SQLiteOpenHelper, an class that includes the implementation of several functions native to Android SQLite. With this, we were able to include several functions to manage all the database queries we needed. These were primarily to retrieve a clothing item based on a weather forecast. The DB query inputs parameters based on JSON data from a weather API, and return appropriate Clothing Items. ClothingItem objects were persisted by having each private variable such as temperature or precipitation stored in columns of the database, and instantiating ClothingItem objects from the rows returned from database queries. Our original class diagram simply had a Clothing Controller, which had a function getClothingQuery(id). While this makes sense, there are many SQL related functions that need to be implemented in order to actually make this work. This was achieved by simply extending the SQL helper class.

The final design of the project did not include a user login field or user database. This was because we only ran the project on a single Android device, and this is not where the users database would be located. User verification would be handled on an external server, not on mobile devices, and would be handled using Get or Post requests to see if users existed or needed to be written to Database.