Jordan Colbert

CS 499

June 5, 2022

Southern New Hampshire University

Milestone Four Narrative

My artifact is an Android application written Java, designed to allow a user to create, update, view, and delete events in a database stored in SQLite. It was created last year in a previous course. I selected this item as my example of database management due to it using a database for information. The database is created upon startup of the app, and the information is gathered from a query written specifically for SQL, passed from the program to it. I believe that it showcases my ability to write for a database and control how the information is viewed and managed.

My improvements involve granting the user a bit more control over the information gathered from the database. Previously, the events were sorted by ID, an automatically generated field that only refers to what order each entry was created in. Instead, I gave the user the option to sort each event by title, either ascending or descending. This was accomplished by using the same code that allowed the user to view the events and putting it in a method that queries the database with an added filter—ascending or descending by the “title” column. Both options have separate methods and buttons. This falls in line with my initially stated plans to give the user a finer control over the database. I was optimistic about how much control I would give the user and suggested that I let them sort in every single column. However, I may end up only going by title and time due to the description column being a bit less important to a user. I doubt anyone would need to sort events by a vague, short description that could be anything.

Developing the improvements was straightforward. As stated, I reused code from the method that allowed a user to view events so that I could rebuild that information after it is sorted on the SQL end. I did, however, run into issues when it came to logic. Because the database and main classes were separate, I had to change a couple of variables to public static for outside references. I also had to think about how I would handle rebuilding the event information. Initially, I just wanted to call the original method that allowed this function, but I had it tied to a button only. Instead, I ended up copying the code into the ascending and descending methods due to issues in clearing the text. I experimented with a “checker” variable that would be 0 through 2 depending on whether the info was at its initial, ascending, or descending phase. This ended up being a bit more complicated, however, so I just decided to reuse the code. To this end, I learned that simplicity can sometimes outweigh redundancy, and decisions such as these truly depend on the situation.