Timeline Project: Phase 3

Subgroup: Android

Members: Josh Wright, Daniel Conroy, Andrew Sutton

Github: <https://github.com/joshuawright11/AndroidTimeline>

Prerequisite to run: emulator

**Features:**

The Android Team decided, per the specifications, to have a very simple design and basic functionality. The app is designed to be supplemental to the Desktop app, and is to function as a quick reference and a fast way to add categories and events to existing timelines.

In order to facilitate the speed and accessibility of this design, a very simple approach was taken by which the user first arrives at a login screen, and subsequently accesses their timelines. On this screen the user can easily see their timelines and may select any to view them with the tap of a list item. Or the user can select from the menu the functionality to add a category or add an event. Each of these screens are simple and arranged from top to bottom for easy workflow.

Originally, the display was to be rendered using JavaFX. However, after attempts to implement the display using this code, it was revealed that this would be beyond the scope of this project, and therefore a new, clean display was written for the user to view timelines.

//Insert more features of Display

In creating events and adding categories, the intention was to create a very intuitive and efficient workflow. To this end, it was decided to include “seekbars”, or sliders to determine each of the color components of a category. This enables the user to very easily, graphically, and intuitively adjust the color their category shall possess. Similarly, when adding events, a checkbox near the top is available for the user to select whether the event being created is atomic or duration. Checking or unchecking this box then updates the components visible to the user such that the user will not become distracted by adding more information than needed. Additionally, to choose dates for the events, “DatePickers” were used to provide an intuitive way by which the user can easily select the year, month, and day via a simple and familiar calendar.

**Next Steps:**

Some additional steps that could be taken with this Android app would be making the display more attractive and speeding up performance through multi-threading. Adding the functionality to add timelines would be an acceptable goal, but it might begin to encroach upon the “quick-reference” goal for the app.

**To Run:**

The only means to run the code is to load it into an IDE which possesses an Android Emulator and run the main project. Android has a default Activity it loads, which starts the app.

**Commit History:**

The project was originally divided three ways; Josh was to implement the display, Daniel was to build the activities creating events and categories, and Andrew was to use the database present already in the Desktop application and implement it into the Android setting.

As can be seen in the commit history, database work was not started until late, which limited the functionality of the rest of the application. The result was that some stages of development contingent upon the database were delayed out of necessity until the last week of the project. In the last week, Josh implemented the database.

**Usability Tests:**

Because of a lack of early functionality of the database, there were only very limited usability tests performed. Daniel showed the early stages of the Activities used to create events and categories to ensure that they were easy to understand and use for the novice. In addition, Daniel Gill was used as a source for suggesting the best method for viewing timelines, adding events, and adding categories, having the first present on the main screen and the last two present in a menu. However, the end product has limited functionality per its design and therefore extensive user testing is perhaps less necessary.