**Milestone Two: Narrative**

**Enhancement One: Software Design and Engineering**

Amanda Kerr

CS-499: Computer Science Capstone

Professor Kalinowski

March 23, 2025

The selected artifact originated from the CS 360 course, titled Mobile Architecture & Programming. This artifact served as the final project for the course, which required students to create a fully functional Android application that fulfilled specific criteria. The application required secure user access, a database to manage objects, and adherence to UX/UI standards. It was designed to serve as a repository of events, displayed in a straightforward list format, and provide users with notifications about upcoming events.

I selected this artifact for several reasons. The initial version of the app featured a poor structure, relying on a simple activity-based approach. By shifting to the more standard and organized MVVM architecture, I was able to explore a different structural design and assess what elements needed to transition from one format to another. Additionally, I initially aimed to incorporate a calendar display in my first app submission, but I was unable to achieve it, which left me feeling disappointed at the time. Lastly, the original project required SMS notifications, which are now considered outdated. I decided that modernizing the notification system to a push or alert format would further test my design capabilities. These were three necessary changes that this artifact needed, which I felt met the requirements for this enhancement category. Unfortunately, the app sends a push notification only once daily. I intend to implement the new feature that allows for a preferred notification time at final submission.

Originally, I planned to meet course outcomes 1, 2, 4, and 5 with this enhancement. Below follows how each outcome is or is not being met:

* Outcome 1: MET
  + My enhancements are currently hosted in a public repository, which enables a collaborative environment. My app runs successfully in the emulator, enabling prototype-style testing by non-developers.
  + <https://github.com/amarkerr/CS499Capstone/tree/Enhancement1>
* Outcome 2: PARTIALLY MET
  + My intention for this outcome was to achieve clean code along with clear comments. The code is indeed clean; however, the comments are quite sparse. As I implement more enhancements, I plan to dedicate time to adding the necessary comments..
* Outcome 4: MET
  + Incorporating the calendar and push notifications updates the app to modern standards. I needed several workarounds to ensure the calendar met my UX/UI requirements. The basic Android calendar only allows for highlighting the selected date. In the app, I utilize a Maven artifact in my dependencies to access the Material Calendar View. This enabled me to create classes and functions that add visual effects to the calendar for the current date and dates with events..
* Outcome 5: MET
  + While this app originally had a secure login, users could bypass it by creating an account with empty credentials. This loophole allowed anyone to log in at the login screen simply by pressing the button, giving access to the phone owner’s events. I made a minor modification that now requires users to provide credentials during both registration and login.

Despite my personal challenges, I felt quite confident as I approached these changes. I divided the updates into three clear categories: calendar, notifications, and structure. I began with the original Android calendar, incorporating various UX/UI features that I envisioned. This process led me to adopt a different calendar, which I found very satisfying. Next, I addressed the push notifications while retaining a structure akin to the SMS notification. To ensure I could meet the late deadline, I chose not to alter the notification timeframe, as this would have necessitated changes to the Event Edit and Event Add Java and XML files. Subsequently, I re-evaluated the structure, extracting database and user login functionalities into separate Java class files. This adjustment enabled me to transition smoothly to the MVVM architecture that I had aimed for.