Jake Torres

CS 499

10/5/2025

Milestone 4 Narrative

The artifact I selected is my mobile event tracker application from CS 360 (Mobile Architecture and Programming). It was originally created in Android Studio using Java and XML. The app lets a user sign up or log in, choose SMS notification permission, and then view an event page where events can be added, updated, or deleted. The application uses two SQLite databases: one for users and one for events.

I chose this artifact because it was a solid starting point but not fully complete, which made it a good candidate for enhancements. It already showed my ability to design and build working applications, but by improving it, I can also show that I can refactor and restructure code, apply best practices, and prepare an application for modern deployment.

I included this artifact in my ePortfolio because it demonstrates my ability to take an existing working project and expand it into a scalable, full-stack system. Through this enhancement, I showed skills in backend integration, API communication, and database management, which are all essential to modern software development. The milestone focused on connecting the mobile app to a MongoDB database through the web app I built in VS Code. This required adding dependencies and setting up Retrofit and OkHttp so the mobile app could call API endpoints securely and efficiently. I also implemented data transfer objects (UserDto, AuthResp, ErrorDTO) and created repositories for authentication and events. In addition, I revamped key screens in the Android app, including MainActivity (login) and HomeActivity, to improve structure and reliability. On the backend, I added a config class, created an auth.service, and updated the routes in VS Code to return either HTML or JSON depending on whether the request came from the web or the mobile app. This was one of the biggest challenges, as I had to ensure both systems could use the same server while serving different response formats. These improvements showcase my ability to develop cross-platform systems, apply database and network programming skills, and solve complex integration problems.

My plan was to demonstrate mastery of:  
• CO3: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices.  
• CO4: Demonstrate innovative techniques and tools in computing practices for implementing solutions that deliver value.  
I believe I met both outcomes with this enhancement. I designed efficient backend routes and structured data models that reduce redundancy and improve communication between systems. I also used innovative techniques such as Retrofit for asynchronous network calls and OkHttp for improved reliability and debugging support.

Enhancing this artifact taught me how much effort goes into connecting front-end and back-end systems, especially when ensuring that both a mobile app and a web app communicate properly with a shared server. I learned how to configure APIs, manage different data formats (HTML and JSON), and use Retrofit to simplify API calls on Android. One of the biggest challenges was getting the JSON responses to work for the mobile app while maintaining HTML responses for the web app. It took time to restructure my routes and logic in VS Code so the server could detect request types and return the correct response. This experience deepened my understanding of API design, HTTP protocols, and data serialization. Through this process, I gained stronger skills in full-stack development, database integration, and debugging complex systems. I also learned how small architectural choices—like how responses are formatted or how repositories are structured—can significantly affect an application’s performance and maintainability.

As part of this milestone, I added JWT and SALT/HASH to the VS Code web application and changed the Android code to satisfy those requirements. I built a new TokenStore to read the saved JWT and created a MyApp class to ensure the TokenStore exists before API calls. I initially had a problem with the events list not working on mobile due to not setting up the Retrofit client correctly for the JWT, but I resolved it, and I can now add and update events on mobile. I created web controllers to thin out the web routes and added cookies so I can pass userId through token cookies instead of the query. I updated everything to use Passport and had some trouble switching the userId to Passport instead of from the query, but I got it done. Now I am changing the Android Studio app to get userId from the JWT and not the query, and I am also adding delete event.