Jeffrey Karl

CS-499

6 October 2024

Milestone Four Databases Narrative

The artifact is a Java-based advising program designed to assist students with academic course planning. Initially, the program stored and retrieved course information using an in-memory ArrayList, allowing users to search for courses by their course ID and display all available courses. However, the program has since been enhanced to use an SQLite database for persistent storage of courses and their prerequisites. This update introduces new functionality, such as the ability to import courses from a CSV file and perform CRUD operations on the courses, making the program more scalable and maintainable.

I selected this artifact for my ePortfolio because it demonstrates essential software development skills, particularly in database integration, collaboration, and real world application development. The transition from an ArrayList to an SQLite database highlights my ability to implement scalable, persistent solutions that can handle larger datasets and complex data relationships, such as multiple course prerequisites. Additionally, the inclusion of CSV import functionality and user-friendly database operations showcases my competence in handling external data sources and designing flexible, maintainable software that can be easily adapted to changing requirements.

In alignment with course outcome one, the development of this artifact required strategies for building a collaborative environment by ensuring that the design could be easily understood and modified by other developers. By decoupling the business logic from the data storage layer, I created a modular system that supports collaboration among diverse audiences, enabling more effective decision making in the context of database driven software. The design decisions, such as separating the business logic from database interaction, were made to allow easier collaboration and future extension by developers who may have varying levels of expertise.

Furthermore, the enhancements also addressed course outcome five by emphasizing a security mindset throughout the development process. The move to SQLite not only improved scalability but also allowed me to expose and mitigate potential security vulnerabilities, such as ensuring that user input is properly validated during CSV imports. I implemented input validation and error handling to reduce the risk of SQL injection and maintain data integrity, especially when working with external data sources. This approach helped to ensure the privacy and security of the stored course data, reinforcing the need to anticipate adversarial exploits in the software architecture.

The enhancements to this artifact align with the course outcomes I planned to meet in Module One. By decoupling the application logic from data storage, I successfully developed maintainable and collaborative software that follows best practices in software design. Additionally, the introduction of SQLite as a persistent storage solution ensures secure and efficient data handling, which was a key objective. I adhered to software design principles such as modularity and separation of concerns, where database interaction and business logic are managed independently, supporting a collaborative development environment.

Throughout the process of enhancing and modifying this artifact, I gained valuable insights into database management and integration with Java applications. I learned how to model complex relationships, such as course prerequisites, in a relational database and handle those relationships programmatically. Additionally, I refined my skills in SQL and database design, which proved essential in ensuring the system's scalability, maintainability, and security. One of the primary challenges I faced was designing the database schema to accommodate multiple prerequisites, which required careful consideration of normalization and data relationships. Another challenge was ensuring error handling and validation during CSV file imports and database operations, as it was critical to maintain data integrity, ensure security, and provide clear feedback to users. Overall, the process enhanced my understanding of database-driven applications and reinforced the importance of thoughtful design and security considerations when making improvements to existing software.