Rodriques Perry

CS-499

Milestone 2 Narrative

17 November 2024

**The artifact used for Software Engineering and Design is from CS-340, completed on 23 June 2024. The goal of this project is an interactive database for rescue animal centers. The project is written in Python/Dash and is designed to interact with MongoDB, through the use of PyMongo, to work as a backend and frontend structure to Create, Read, Update, and Delete (CRUD) animals from an animal shelter.**

This artifact was selected because it clearly demonstrates a significant translation of a Python-based project with MongoDB and Dash into a more complex, modern full-stack web application using **Spring Boot** for the backend and **React** for the frontend. This transformation shows a strong understanding of software development principles and technologies, as well as the ability to improve the project by refactoring it into a more scalable and maintainable solution.

By implementing **RESTful endpoints** in Spring Boot, I moved from a script-based approach to a more structured service layer. This is a key improvement in software design, showing the ability to design **scalable and modular systems**.

In the original artifact, the dashboard was implemented using **Dash**, a Python framework for building analytical web applications. The translated project replaces Dash with **React**, which is a more flexible and widely used tool for creating interactive UIs. By breaking down the UI into **React components**, I leveraged a modular approach to front-end development. This approach not only makes the code more maintainable but also enhances my understanding of **component-based architecture**, which is a key skill in modern web development. The React components interact with the Spring Boot backend via **API calls**, demonstrating the ability to integrate a frontend with a backend using **RESTful services** and **asynchronous data fetching through axios.**

**There are a few improvements in software design as well.** By switching to a Spring Boot backend and React frontend, I applied the **separation of concerns** principle, where data access, business logic, and presentation are managed separately. This improves the maintainability of the codebase. Furthermore, the backend now communicates with the frontend via a **REST API**, allowing for better scalability and flexibility. This makes it easier to integrate additional clients or services in the future, adhering to **industry standards** for web service architectures. Lastly, with **React** managing the frontend, user interaction becomes more dynamic. The React frontend, unlike Dash, can provide a richer and faster user experience, with interactive features such as **live filtering**, **sorting**, and the possibility to easily update data on the fly.

The translation moved from a tightly coupled, single-language, single-framework solution (Python/Dash) to a more loosely coupled, multi-tiered, full-stack solution (Spring Boot/React). This allowed for the application to handle larger user bases, with Spring Boot’s built-in scalability features. It also allowed for integration of modern frontend technologies for a more dynamic, responsive user interface. Lastly, it improved maintainability and flexibility by using a separation of concerns approach, where the backend and frontend can evolve independently.

All-in-all, the process to modify the artifact was smooth. I encountered a small problem with my CSS file that I found a way to fix via the internet. Based on the enhancements made to the artifact, I have successfully met the course outcomes as originally planned for Module One by demonstrating effective planning, design choices, and the application of modern tools and techniques. The project is scalable, modular, and follows best practices in full-stack development, ensuring both the frontend and backend are well-structured for future improvements. Additionally, my approach to data encapsulation and RESTful API design further strengthens the overall design.