**Enhancement Three Narrative**

Sarah Deleppo

Southern New Hampshire University

CS-499

Professor Brooke

July 26, 2021

**Database Narrative**

The artifact developed for the Database category of the Computer Science program is based on providing a database layer to an existing project from class CS-320. In said class, we developed a Contact Manager to allow for users to store, update, and manage their contacts VIA Java code. Previously, these contacts and their attributes were stored within a HashMap at runtime rather than a database which exists outside of when the application is currently running. For a full stack application, this was largely impractical as user’s data should be stored regardless of whether the application is currently running. To solve this issue, I had to adjust existing design choices, and opted to create a database layer for the use of this application using MongoDB. This solution allows users to have their data stored long-term as well as have access to all the data management functionality the MongoRepository provides.

As previously stated in the *ePortfolio Selection and Software Design Document,* I chose to create an all-encompassing, full stack project as it aligns with my career goals of becoming a Full Stack Developer. A key layer of the stack is the database layer, which communicates to the backend layer, which then communicates to the frontend layer for a fully functioning application. This database layer is main goal of development for this category and is the final piece of the stack which allows the entire project to function as intended. I chose to work within MongoDB for the creation and management of this database as I had utilized its functionality in a previous full stack project in CS-340, this allowed me to quickly adapt this tool to meet the needs of the overall project. This prior knowledge aided me in as I was familiar with the MongoRepository interface and the tools that came with it, I was able to create a custom finder method to find contacts by their name. The creation of this database improves the overall project by allowing the storage of data long-term and access to additional data management tools within MongoDB.

The development of this database strayed slightly from the initial development plan laid out within *ePortfolio Selection and Software Design Document.* As stated within said document, I began by organizing each layer of the stack into separate folders and created a MongoDB database accessed within the resources folder of my project. However rather than ending here, I focused on creating a repository package that held the ContactRepository interface which extends MongoRepository. This allowed me to use MongoRepository’s methods including: findAll(), findOne(), save(), delete(), and deleteById(). From this point I knew I wanted to incorporate functionality that allows end-users to search for a contact by their name, to implement a solution to this request I created a custom finder method, findByNameContaining(), within ContactRepository.

While I had prior experience using MongoDB, I had only ever used it with Python applications specifically using Pymongo, never Java based applications. To overcome this challenge, I had to research how to implement a MongoDB database into a Java project by using Spring and the dependency “spring-boot-starter-data-mongodb”. The Spring Initializr was a massive help in the creation of the base project, as I was able to choose exactly which dependencies I wanted my project to work with, all within the Spring Framework. From this point I was able to write my ContactRepository interface and use the many methods it provides for my ContactController class to implement CRUD functionality. This emphasized the lessons I’ve learned through the development of this capstone project, seeing how all the layers of the stack fit together to create one cohesive application for the end-user.