*Professional self assessment*  
  
**Program reflection:**  
  
Hello, my name is Wilfredo Mendez III. Throughout my journey in Southern New Hampshire University’s (SNHU) Computer Science program, I have developed a broad range of technical and professional skills that have readied me for a career in the computer science field, including gaining proficiency in multiple programming languages, collaborative teamwork experience, and the ability to design and develop professional, industry-standard projects.

The enhancements showcased in this ePortfolio reflect my growth as a software engineer and readiness for a career in computer science. Each enhancement aligns with the program’s outcomes, showcasing my strengths in key areas such as software engineering, data structures, algorithms and datasets, and database management. My portfolio reflects not only my technical competence but also my ability to communicate effectively with stakeholders and collaborate in team-based environments—skills essential for success in the software industry.

**Professional skills:**

Team Environments:   
Throughout my coursework, I developed strong collaborative skills by working with peers on remote team-based projects, gaining experience using tools such as GitHub to work on projects with peers, Slack for asynchronous communication, and Zoom for real time meetings and development. These experiences prepared me to thrive in distributed work environments, which are becoming increasingly common in the software industry. I developed the ability to share responsibilities, prevent miscommunication, and meet deadlines, all of which are crucial in professional team settings.

Communicating with Stakeholders:  
Effective communication with stakeholders is crucial. I learned to keep this in mind in the CS250: software Development Lifecycle course specifically, I took on roles as different members in a SCRUM environment such as product owner and developer. This project helped me learn how to convey technical concepts in a clear and concise manner, ensuring alignment between team members and stakeholders. With the practice in these environments, I became adept at tailoring my communication to meet different audiences' needs, balancing technical precision with accessibility.

Software engineering and Databases:  
The program gave me hands-on experience with software engineering and database management. I gained experience with using databases such as MongoDB and SQL, and I have become familiar with relational and object oriented database designs and how to work with projects in both fashions. Creating custom UI models for these databases enhanced my front-end development skills across multiple languages. In CS 465: Full Stack Development, I built a full-stack web application, giving me practical experience with both front and back-end systems. These principles are exemplified in the Software and Engineering Enhancement, where I did a complete overhaul on the database setup and UI of the Event Notification App.

Security:  
Security has been a consistent focus throughout the Computer Science program. Throughout my courses I learned to identify and mitigate vulnerabilities related to input handling, memory management, and data validation. In CS305 Software Security, I learned the importance of bounds checking, input sanitization, and proper handling of arrays, pointers, and strings. This knowledge has allowed me to develop software with a security-first mindset, preventing potential exploits like buffer overflows or arbitrary code execution. My capstone project enhancements integrate these principles, showcasing my commitment to building secure and resilient software solutions.

**Artifacts:**

For this portfolio, I have compiled three projects from the program to improve upon three fields: Software Design and Engineering, Algorithms and Data Structure, and Databases. These artifact enhancements aim to showcase my growth as a software developer and my finished projects.

***Software Design and Engineering***

The chosen artifact for this section is the Event Notification App I originally created for the CS360: Mobile Architect & Programming course. This app allowed users to input events, store them, and receive notifications when the event occurred, all secured by a login system. When I first took the course I was initially unfamiliar with mobile development and Java, so when I completed the course the event app had many areas for improvement.

I chose this app as the artifact because it showcases my growth as a developer and highlights my skills in both front and backend developments. The app itself is comprised of four screens; one to manage login and account creation, one to input and edit events, one to show all of the events, and one to allow the user to grant permissions for notifications from the app on the input time and date.

Enhancements done:

-Updated comments throughout  
- Updated UI Design for all screens in app to enhance clarity

-Updated Icons

-Changed SQL database to only accept event details, created second database to securely hold login credentials

- Fixed SMS permission request button

-Enabled notification request to be sent to all devices with API 26 or later (Pre-API26 devices will have it enabled by default.)

-Fixed central recyclerview to show events and store them when the user exits the app

-Updated event screen to automatically load details into text boxes when editing

Reflecting on the enhancements done, I learned a lot about Android Development, SQL databases, and the Java language. I faced a lot of challenges when trying to use android studio because it is not an IDE that I use regularly and testing was difficult, especially when using a virtual phone. I was able to get around these hurdles by reviewing guides and eventually switching to test on the physical phone, and as a result I was able to both add a new IDE to my tool belt and gain experience in testing with physical devices. In conjunction to using a new IDE, I also needed to relearn Java-SQL interactions to properly query SQL databases, and thus have become comfortable with making and working with local SQL databases.

The finished enhancement demonstrates my ability to analyze weaknesses in an existing project and make improvements that align with modern software engineering practices. By tackling both front-end UI elements and back-end database design, I showcased my versatility as a developer. This project now serves as a testament to my technical skills and my ability to improve software iteratively based on feedback and new knowledge.

Course outcomes met:

With this enhancement, I have demonstrated course outcomes 4 and 5:

I have demonstrated an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals. This artifact required using Android Studio, SQLite, and Java to implement a robust event notification app. The improvements focused on refining the user experience, such as updating the UI design, enabling notifications across multiple Android API levels, and managing event data more efficiently with better database handling. These changes showcase my ability to iteratively improve software and adopt innovative practices that align with real-world industry goals.

I have developed a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources. Security was at the forefront of development in this app, and where a lot of improvement was done to make it more secure. One of the key improvements was separating the login credentials from the event database, preventing unauthorized access to sensitive information. I also ensured there were no hardcoded values (magic numbers) that could leave the system vulnerable to reverse engineering or exploitation. The app now follows best security practices by requesting permissions dynamically, ensuring compliance with Android API requirements, and validating inputs to prevent unauthorized actions or data corruption.

***Algorithms and Data Structure***  
  
This artifact chosen for this section is the Appointment Service Project I originally developed for CS 320: Software Testing, Automation, and Quality Assurance. I chose this artifact to include because it demonstrates my knowledge of data structures, my ability to apply them effectively, and my skills in designing robust test cases to ensure quality software.

The original project comprised four files. The first main file focused on defining the Appointment class, which included attributes such as appointment ID, date, and description, along with corresponding getters, setters, and validation checks. The second main file implemented an ArrayList-based Appointment Service, enabling users to create, read, update, and delete appointments in an arraylist. The remaining two files were test files for the Appointment and AppointmentService classes with built-in tests to validate the logic and boundary conditions of the service.

Enhancements done:  
- Introduced a HashMap data structure to compare to the original ArrayList implementation

-Expanded test cases to compare the efficiency between ArrayList and HashMap implementations, focusing on differences in time complexity and optimization

-Added new tests specifically targeting the HashMap implementation to ensure that the new data structure was thoroughly validated.

-Documented time complexity analysis in comments and reports to highlight the trade-offs between the two data structures for future reference.

-Improved CRUD operations to reflect the optimized HashMap-based design and ensure backward compatibility with the original implementation.

Through this enhancement process, I deepened my understanding of how data structures like ArrayLists and HashMaps impact application performance, especially for services that handle dynamic data. The project allowed me to explore the nuances of time complexity in a practical setting, and I learned how to make informed decisions when choosing data structures based on specific use cases. Implementing a HashMap required careful consideration to maintain the same functionality as the ArrayList version, which challenged my ability to refactor code without introducing new bugs.

This enhancement also strengthened my testing skills. I had to create additional test cases to ensure that both data structures worked as expected, especially under stress conditions, such as large datasets. This experience not only improved my ability to design and execute tests but also emphasized the importance of performance benchmarks in software engineering.

Ultimately, this project demonstrates my ability to analyze, optimize, and validate software systems through a combination of algorithmic knowledge, practical testing, and thoughtful refactoring. It also showcases my capacity to document findings effectively, ensuring that future developers can understand the rationale behind the design choices made.

Program outcomes met:

With this artifact enhancement I have demonstrated course outcomes 1 and 2;

I have employed strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science. The improvements made reflect collaboration strategies in software design, such as writing detailed comments, documenting performance trade-offs, and structuring tests in a way that future developers could easily understand and build upon. By providing thorough documentation of time complexity analyses and benchmarking results, I ensured the artifact could serve as a shared resource for decision-making about data structure selection.

I have designed, developed, and delivered professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts. The communications to future developers in the artifact are are technically sound and adapted to the needs of developers or project stakeholders by explaining why certain data structures were used, the trade-offs between ArrayList and HashMap, and how to replicate tests. The focus on coherent communication highlights my ability to document design choices effectively, ensuring that both technical and non-technical audiences can understand the rationale behind the optimizations.  
  
***Databases:***  
The artifact chosen for this section is the Animal Shelter Dashboard Application that I originally developed for CS 340: Advanced Programming Concepts. I chose this project because it highlights my ability to work with databases, develop full-stack applications, and apply programming concepts across multiple languages and frameworks. I selected this artifact because it reflects my versatility as a developer, showcasing how I can transition between different programming environments and enhance applications based on project needs.

The original artifact was built using Python with the Dash framework, which displayed a preset list of animals from a fictional animal shelter. The data was visualized in three main formats: a pie chart representing animal distributions, a table listing animal details, and an interactive map to show the animal's location.

Enhancements Done:  
  
-Rewrote the entire application using JavaScript, integrating libraries such as React, Plotly.js, and Leaflet.js to maintain the original functionality.

-Separated the frontend and backend logic to follow a modular architecture, ensuring easier maintenance and scalability.

-Exported the dataset into a .txt file, separating it from the code itself.

-Refactored visual elements, such as charts and tables, to improve interactivity and provide a smoother user experience.

-Documented the codebase thoroughly  
  
Rewriting the application from Python to JavaScript was both challenging and rewarding. It required me to learn and apply several new libraries and frameworks, which expanded my knowledge of JavaScript-based development and full-stack web applications. Adapting the code to a new framework emphasized the importance of modularity and separation of concerns, which are critical for maintaining and scaling modern web applications.

Outcomes met:

With this enhancement I have met outcomes 3 and 1;

I have employed strategies for building collaborative environments that enable diverse audiences to support organizational decision-making in the field of computer science. By carefully restructuring the application to follow a modular architecture, I ensured that the frontend and backend components are easier to understand, maintain, and improve. This modularity allows other developers to collaborate more effectively, making it straightforward for them to contribute new features or connect the system to external databases in the future. The clear separation of concerns also ensures that stakeholders can make decisions more effectively, understanding how individual components contribute to the system as a whole.

I have designed and evaluated computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution while managing the trade-offs involved in design choices. My decision to implement React for the frontend, along with Plotly.js and Leaflet.js for charts and maps, reflects a strategic use of modern frameworks that offer greater flexibility and responsiveness.