My second artifact is the Software Testing, Automation, and Quality Assurance Contact Service.java enhancement from CS 320. This application allowed the scheduling, updating, and deletion of appointments, contacts, and tasks. The program automated this process through Junit testing. This project was worked on throughout November and December of 2023. The artifact is composed of different classes: Contact, Appointment, Task, and tests to run examples of adding and changing contacts and their additional details.

I selected this item because it shows algorithms and data structures through the automation of collecting data and contacts with algorithms such as a HashMap. During the creation of this project, I experimented a lot with the test cases I initialized and made sure every test ran successfully. My skills and abilities were best shown in the Contact classes to find contacts based on their unique ID and test each contact based on deletion, update, or add. I took the most time while running the coverage for every class and making sure it reached over 80%. For my enhancement I replaced the ArrayList with a HashMap for more efficient contact management, therefore it would utilize the unique ID as the main key for lookups. This will allow for quick methods of searching for contacts and faster operations. Every method was checked multiple times to ensure it ran through the desired task and made the changes it was supposed to.

Some of the course outcomes I have met in this Module are evaluating computing solutions to solve problems using algorithmic principles and computer science practices and achieving innovative techniques, skills, and tools for implementing computing solutions that deliver industry-specific goals. Additionally, during the process, I also covered delivering written and visual communications to the specific audience regarding the enhancement.

While modifying the artifact, I learned more about the replacement of an ArrayList with a HashMap. I had replaced the ‘contactList’ with ‘contactMap’, using the unique ID as the key, and would now search up the contact by the ID allowing a faster process through the list. For every deletion or addition of a contact, it is now using the key to identify the proper contact to change. Since ‘getContactList’ was now replaced with ‘getContactMap’, I also had to update the Test class to correspond to the changes made in the ContactService class for proper interaction between both. Some of the changes included accessing the contacts by their ID instead of an index and changing the validation on ‘getContactMap’. To fetch the contact properties as well the HashMap would be using the ID as the key. After making all the changes I ran the tests and passed with no errors, however, I had a new contact failure on my first trial run. A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

After testing the coverage was over 80%, I made a couple of improvements to the Contact Service Class and the test class. The unique ID generation was changed to correct the random ID usage for the user. Testing-wise, Map.keySet().toArray() was not ordered, therefore, it was changed to improve the test reliability by directly getting the contactID from the newly created contact. This helped clear the error and provided a higher coverage percentage. A screenshot of a computer program

AI-generated content may be incorrect. *A screenshot of a computer program

AI-generated content may be incorrect.*

While improving this artifact, I learned more about how the UniqueID functions with the main classes and the changes needed to reflect the HashMap use. As I researched, the enhancements needed for the rest of the classes, there was no need to change any other class to connect with the HashMap as it was already being used correctly. I met all the course outcomes while partially completing the development of a security mindset in designs to expose potential vulnerabilities through the actions of Junit Testing. These skills were demonstrated for all Contact classes, except the Appointment and Task classes. In the future, I will enhance these classes as well to have a fully completed automated project with the implementation of the HashMap.