CS 499 Milestone Two

Susannah Weatherby

7/20/2025

The artifact used for this enhancement is a set of three java files created on April 28th, 2024 that were used to experiment with automatic testing. These files were AppointmentService.java, ContactService.java, and TaskService.java which each had a class for their respective objects but no main method to access them during runtime. I selected these artifacts because they were incomplete, non-functional programs that I believed I could re-write into a fully functional service as proof of my abilities. To do this, I used the code of each individual file and re-wrote them in python as a singular file with separate class and model and service classes for each object. The appointment service originally did not have a function for updating existing appoints, so an update method was added. To ensure that users did not try to add an appointment that was set for a past date, an error throw was also added to prevent the user from doing so. Finally, I added a looping main method for users to access the create, read, update, and deleted methods for each class.

I believe I have met the course outcomes I planned to mee for this enhancement. However now that I have completed my enhancement, I find myself questing my initial idea for how to integrate the different objects into being children as part of the second enhancement. I believe it might be better to have them still separate, but tasks and/or appointments could require an assigned contact. This might also require contacts having a label for employees and customers or even splitting them into employee contacts and customer contacts, although I worry having an additional class to specify this would clutter the code or make it redundant.

Since the original artifact was only meant for utilizing automatic testing, the process of enhancing the artifact essentially meant rebuilding from a basic framework. The most amount of time spent was on the main method since that initially didn’t exist, so I had to think about how I wanted it to properly connect to the different services of the program. Although the final result is still relatively basic, I believed I learned a bit more about date conversion and adjusting code to make it more user-friendly with the appointment method since the original artifact did not have a function for converting strings into date and time. My first attempt used date and military time to create appointments, but after some consideration I learned how to implement AM/PM into this since I believe most users would attempt to do this anyway or would have a harder time trying to read or write military time.