Thongsavik Sirivong

Kevin De La Torre

Dharam Kathiriya

Andy Munoz

Personal Scheduling System

We used the Java Programming language to build the PSS project. Each member of the team was assigned to build different parts of the project. Thongsavik Sirivong was responsible for constructing a DataFile class that required reading tasks from a JSON file and processing them into a format the calendar could use and writing the tasks from the calendar to a JSON file. Moreover, Thongsavik worked on the Report Class which generates the report or the schedules of the tasks for the user. Kevin De La Torre worked on the User class and the Calendar class and helped debug some of the other parts. The User class required the implementation of reading and setting the filename to be used in the program but we ended up not really using it as we went with a more driver-based approach in handling which file we were using. The Calendar Class was in charge of scheduling and managing the tasks as well as interfacing with the Report Class. Dharam Kathiriya was in charge of the superclass Task, and its subclasses Transient Task, Recurring Task, and Anti-Task. The Task Class was tasked with holding the basic information (attributes) and methods of the various types of tasks. These subclasses inherited the values and method of the Task Class. The subclasses validated the inputs given when creating new classes and provided functionality needed on a per task type basis. Andy Munoz was responsible for Driver Class (main method) which required to take user inputs such as the input file, task details, and the type of operations and tell the parts of the PSS system responsible to process them. The team members also helped each other to resolve any issues with their code. As of now the PSS

system is fully functional as a CLI. It can Read tasks from a file, schedule tasks, edit tasks, delete tasks, and generate a schedule of all the tasks in the system so far. Using Object Oriented design really helped us in this project. It made it easy to modularize everything and made it much easier to keep things organized as we tried to keep major functions of the PSS system to their respective classes. It also helped immensely in making it easy for everyone in the group to be able to work on their part separately and put their whole focus on just one aspect of the system and using github combine them all together in the end. Also as a final remark some of us were having trouble compiling it and others weren't. The ones that compiled were using vscode, so in case of errors popping up using vscode is worth a shot.