

Project 2: Summary

The main idea of the project was to employ semaphores to be able to achieve mutual exclusion and coordinate with the other objects in the system concurrently to produce the desired result. In this project we have 4 main entities: Customer, Information Desk, Announcer, and the Agent. This is what it takes to run the simulation of the DMV in our simple scenario. My program begins in the main class which is called Project2.java, it is simply used to create a DMV and then call the runDMV method. In the DMV class I initialize all my semaphores, threads, and any additional data structures required for the project. After this I begin all my threads for all the entities. In this project we have 20 customer threads that are initially created, along with one Announcer, information desk, and 2 Agents. Customers are created and they enter the DMV. They signal the information desk that they have entered the building and need permission to move to the waiting area. Customer is then given a customer number by the information desk and is directed to go to the waiting area. Announcer waits for customers to enter the waiting area. Announcer calls the customer in the waiting area by the number they were assigned. Customer moves to the agent line once the announcer calls their number. Announcer makes sure a maximum of 4 customers are always in the agent Line. Agent waits for the customer to enter the agent line. Agent gets ready and receives an agent number, each agent then takes on a customer in the agent line. Agents then ask customers to take the eye test, and take a photo with the agent. Once the agent is done taking the test, the customer gets their license and departs. Announcer calls the next customer to join the agent line. Agent gets ready for the next customer. We then join the customer thread by their ID.

The main problems I encountered during this project was dealing with allowing one customer at a time to do any of the tasks. At the start of the project, I was unable to properly use mutexes to enforce a critical section, however, once I thought my logic out by creating my pseudo code the implementation became easier. The next biggest problem I had to deal with was an infinite loop that did not allow me to end the simulation. The problem was that my customer and agent queues did not properly poll causing my simulation to keep running. I solved this easily by just properly using the poll method.

I learned a lot during this project, dealing with multiple threads was not an easy task to properly organize them. I have previously never done programming that concurrently ran a lot of threads, so this was a great way for me to understand and learn how to deal with them. Semaphores were a big mystery to me at the start however with this project I believe I have a great understanding of them and how to utilize them properly to get the desired output.