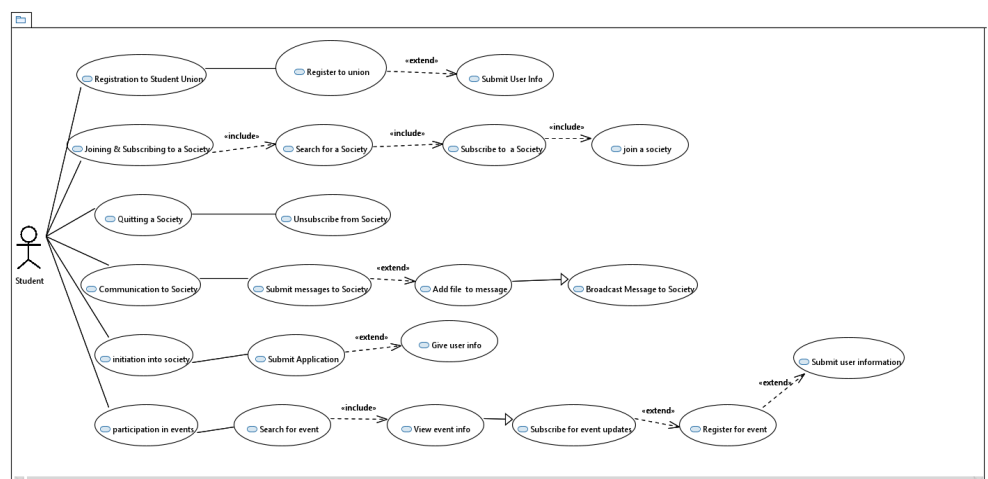
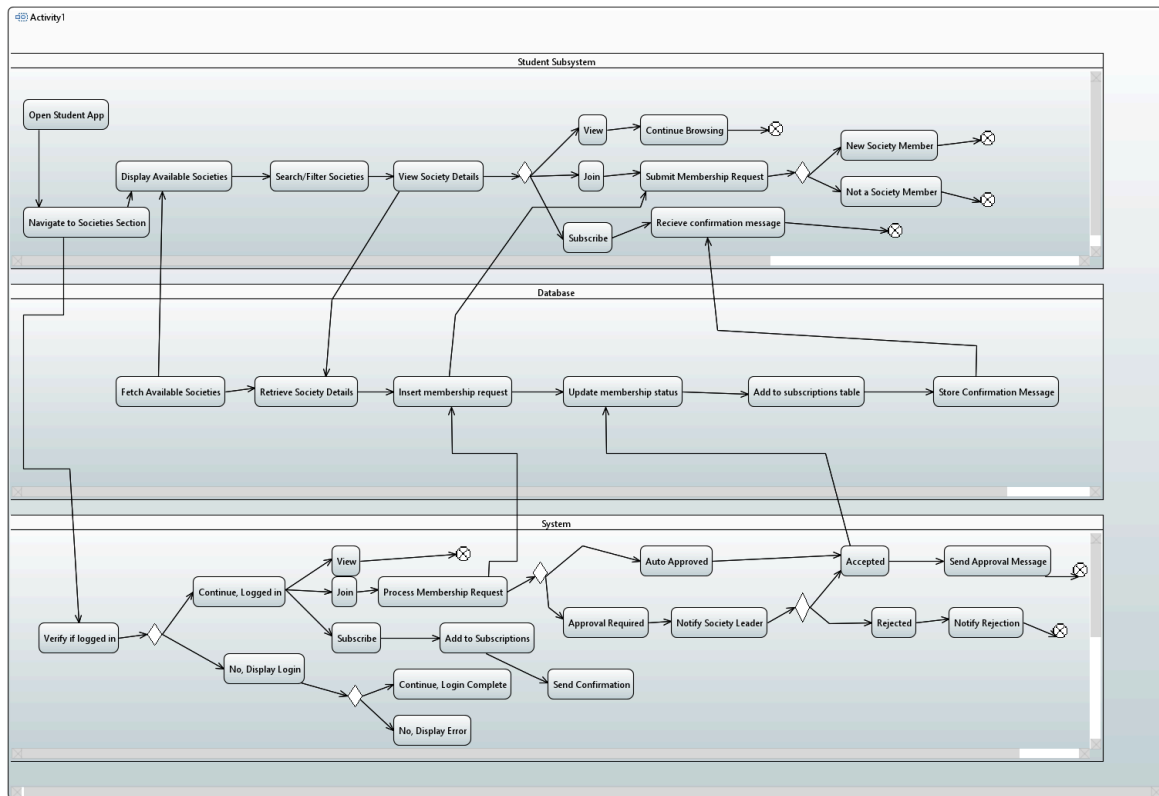


My group took an active approach in completing our project. From week one, we planned out each week, rotating who led the group. In the first week, I took initiative and made the group repository on GitHub and set up the formatting for my group. I hoped to help my group's work come together smoothly and efficiently, so we wouldn't fall behind in future weeks. During the first two weeks, I focused on understanding the requirements for the students' subsection, for which I took the lead in determining who was responsible for each section. Each Wednesday, my group and I met and discussed which task we would take on the following week. I designed my use case diagram early and uploaded it to the group repository, then informed my group members that I had completed my section for the week. I wanted to ensure that my communication was clear so that everyone knew where I was in each phase of the project.

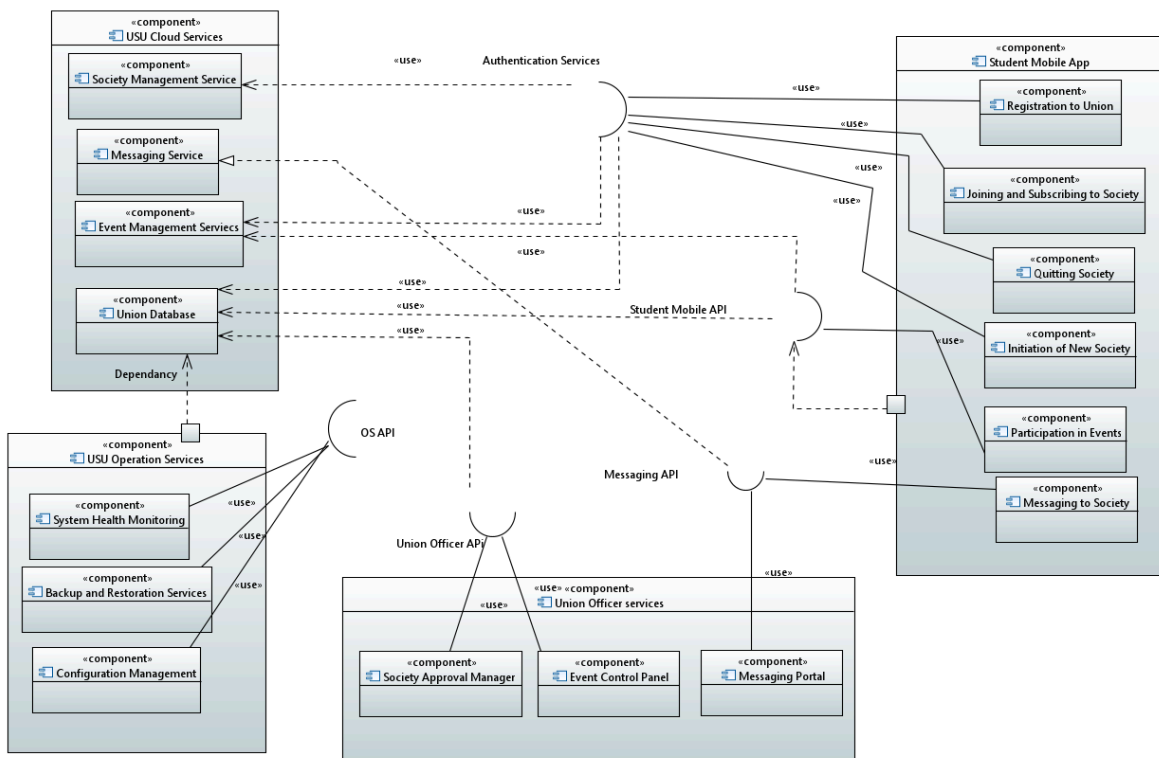
Once task four came around, it was time to combine our component diagrams, and we were falling a bit behind. This was because my group members and I had run into lots of technical difficulties with Papyrus; personally, my program crashed numerous times while working, resulting in my diagrams being corrupted and needing to restart altogether. When working on task 3, I was able to get all my nodes and components to load correctly, other than the final endpoint nodes, which were corrupted and are displayed as circles with x's through them. This was not the only problem; My computers kept crashing with Papyrus, so I had to download Eclipse, then install Papyrus as a plugin to get it to run on my laptop, which too broke when I was working on task 5, and I had to finish my program at the library. This was very frustrating. I kept my group informed each time a crash happened, so they knew that I was at least attempting to get my diagrams done. When working on each diagram, I made sure to upload my project through GitHub so that my changes would be logged. I did have to connect each Papyrus file to GitHub through the command prompt, as the tutorial given by the instructors did not work for me.

In my use case diagram, I focused on showing every action a student can take in the system, from registering with the union to joining societies, messaging, and taking part in events. I tried to make the relationships clear by using *include* and *extend* where tasks depend on each other, so it's easy to see how each part of the system connects.

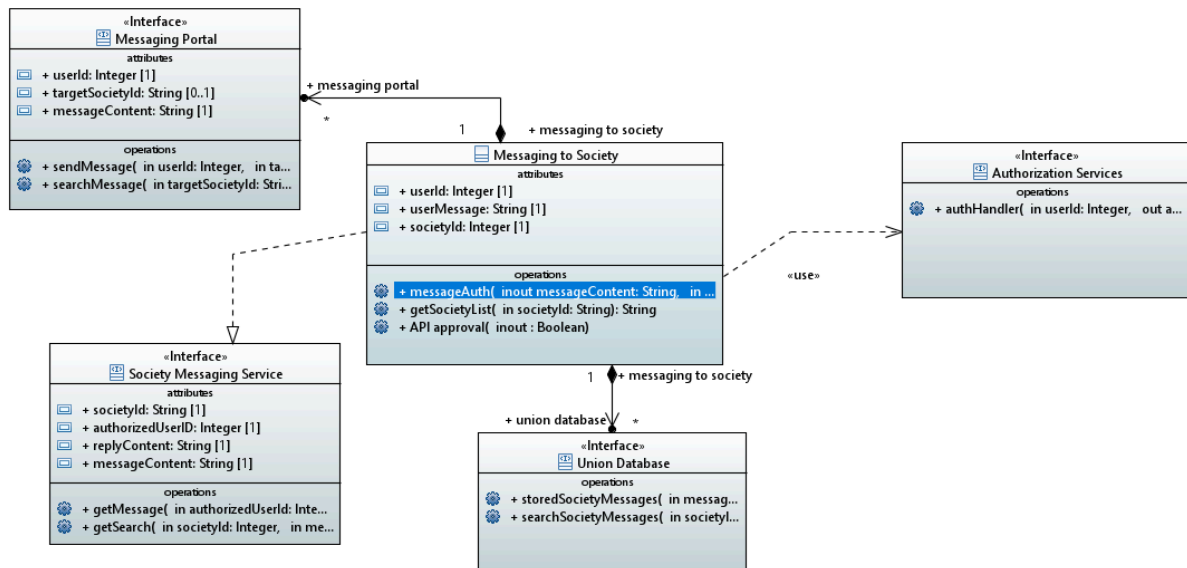




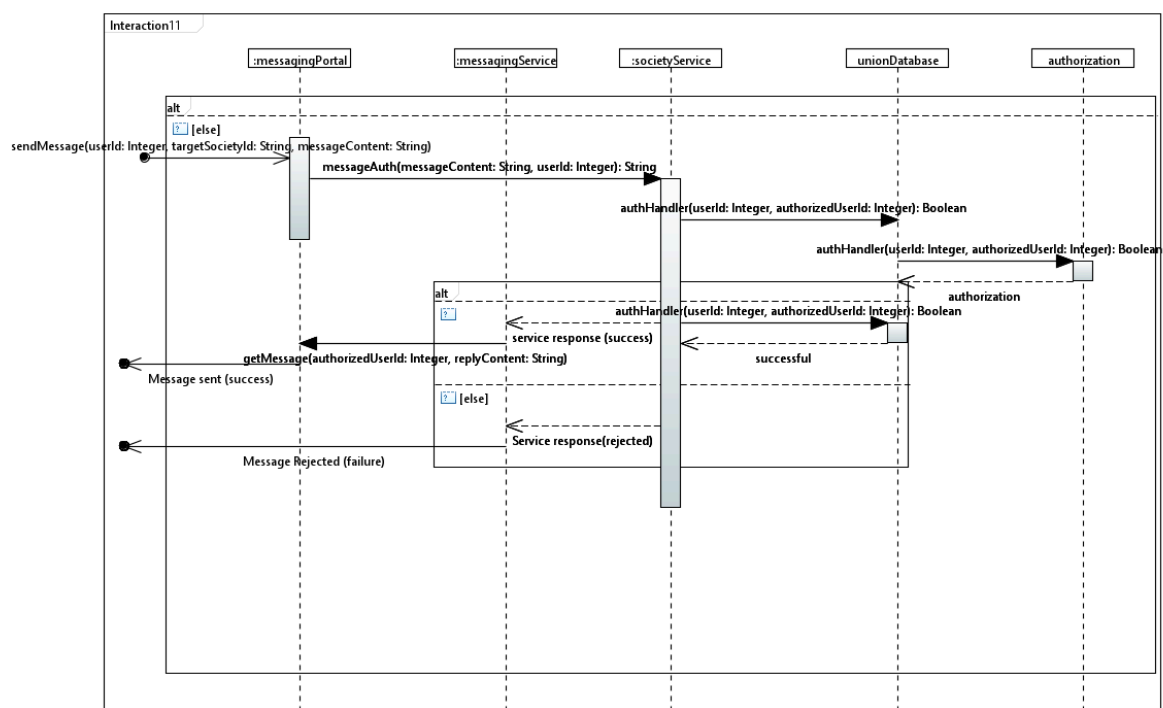
When working on my activity diagram for part of Task 3, I focused on applying swimlanes and ensured that I did not overlook connecting the student app to the system through the database, while also including decision nodes that display the different outcomes of each process.



Moving on to task 4, I found the component diagram to be a bit confusing when it came to connecting interfaces and how to apply use cases to the different components. I used the nodes to connect components that are dependent on each other, and made sure each component was validated and connected to an interface.



On task 5, when doing my class diagram, I made a rough outline of it so I could implement its operations and properties at the same time while designing my sequence diagram. This helped me a lot in applying the correct operations to the student messaging application, as I could visualise what needed to be added to the class diagram.



Overall, my group and I worked well together, and we completed our tasks efficiently, and we were not rushed in submitting our project. We communicated well and often, kept efficient meeting minutes and kept our repositories up to date when finishing tasks. I focused on helping my group stay on track and clearing up any questions they had with a task. As well as making sure I communicated with my group members whenever I needed help or had a question.