

Paul Romer

CSD 460

Module 12.3 Assignment

July 23, 2025

### Paul's Lessons Learned

I really enjoyed this class. I learned a lot that I missed from the previous classes because of the opportunity to work on a team and on a singular project day to day and week to week.

The first lessons I learned revolved around working with remote team. The project instructions gave us a good start with a few things to consider and figure out at the start, like coordinating schedules. We quickly decided to use Discord as our communication platform. We originally set it up as a group chat, and before the end of the week realized one group chat would get pretty chaotic and created a server to be able to separate conversations by module. This made it easier to stay on top of conversations related to the different assignments. Overall, that worked well. I can see the value in something like a daily/weekly standup over a video or voice call. If this was a real product or even much more involved, it would have helped a lot to have times of more synchronous communication – but wasn't easy with different schedules.

Another lesson for me was the experience working in a GitHub repository with other people. I've uploaded all my assignments to my repo since the start of these courses, but that was direct to main when the project was done since they were atomic assignments. With this project, the first module with real code was almost that. By the next module, I learned the value of protecting main and developing features on feature branches with at least one other approval before submission. One of the teammates didn't have the .gitignore set up right, and it caused a lot of non-project specific and build files being uploaded. The clean-up was easy and there are probably always some setup struggles for any new projects, but it was a lesson learned about best practices when developing with a team.

The Kanban board was a great help in understanding and tracking progress on the project but could have been better. I learned the value of assigning tasks so that it's clear who is working on what, and what stage of development those features are in – that's valuable on large projects.

The last lesson I'll share is around knowing your tools. I don't really like the eclipse IDE. There is so much going on and it doesn't look very nice. I used it for Java development

throughout the previous courses but only did what was strictly necessary – things like git management I did outside of the IDE. That worked because I was using git in a very simple way, and I was working on a new assignment every week. For this project, neither was true anymore and avoidance of Eclipse caught up to me. I had to spend a bunch of extra time figuring out how to set up the workspace and learning what the different perspectives are so that I could run everything in the IDE. I did manage to get a solid setup that worked for this project so that I could work week to week and easily start new just pulling from the remote repo if needed. Once I reached that level of setup, I felt confident that it was organized and polished enough to make it through the project. Going forward into the workforce, I want to make sure to really understand the tools that I use and understand the ‘ethos’ of the software I’m using so that I have a good mental model to be able to work with the tools and work efficiently.