For this project, I worked on testing three parts of a mobile app: the contact service, the task service, and the appointment service. I used something called JUnit to write tests for each part. These tests are supposed to check if everything works the way it should, based on what the app is supposed to do.

To be honest, I was really confused for most of this project. I didn’t fully understand how to write unit tests or how to know if I was doing it right. I felt lost and unsure almost the entire time. I had to do a lot of research just to figure out how to get started. I watched videos, read articles, and looked at examples over and over. Even with all that, I still didn’t feel very confident. It was hard to know if what I was doing was correct.

For the contact service, I tried to make sure contacts couldn’t have names that were too long or phone numbers that were too short. For the task service, I made sure tasks had names and due dates, and that the input made sense. For the appointment service, I checked that the appointment times and information were not missing or wrong. I kept looking back at the project requirements to try and match what was asked.

I don’t think everything I did works perfectly. I don’t even know if all of my tests are actually right. But I really tried my best. I followed what I could understand, and I made sure to test at least the main things. I tried to write tests that would catch mistakes or problems, even if I wasn’t sure I was doing it exactly right.

Reflection

The main kind of testing I used was unit testing. That means I tested one small part of the program at a time, instead of the whole app all together. I didn’t do bigger types of testing, like integration testing (where different parts are tested together) or system testing (where everything is tested at once). I didn’t feel ready for those yet. I was still just trying to figure out how to make simple tests work.

I had to do a lot of learning on my own. I didn’t always know what the test should look like or how to write it the right way. I had to look up how to use JUnit, how to test for problems like bad input, and how to run tests correctly. Most of the time, I would try something, it wouldn’t work, and I would have to change it again and again. It was frustrating, but I learned a lot just by trying over and over.

Even though I still don’t feel very good at testing, I know I learned some things during this project. I learned how hard it is to test your own code. Sometimes, when you write the code yourself, you expect it to work, and you don’t think about how it could break. I had to try to think like someone else using the app and look for the kinds of mistakes they might make.

I know I have a lot to improve. I still don’t feel confident in writing tests, and I don’t feel like I fully understand everything yet. But I want to keep practicing. I know testing is really important, and I want to get better at it. I plan to keep learning, to ask for help when I need it, and to not give up even when it’s hard.

I also learned that it’s important to slow down and focus on doing the work the right way. It might take more time now, but it helps avoid bigger problems later. I want to avoid creating “technical debt,” which means leaving behind bad code or mistakes that someone will have to fix later. I want to build good habits now, so I can be a better developer in the future.