The challenges I faced while doing this project was where to implement and use adapter design pattern. So, I research and solved this problem by using the adapter to select the size of a product and then using it inside the factory class to create an item with the specified size. I also got stuck at the point where I didn't understand how to add libraries within a code so that I can test the code with JUnit, whenever I run the code, it shows me an unexpected error in which it was written "JAVA packages doesn't exist" so I looked for the problem on the google and then I added the libraries from the IntelliJ in that way I fixed the problem which I faced.

When I try to decouple the classes to understand in a better way, I was not able to do that because I merged the patterns, and then all patterns got mixed so I separated all modules by searching for the solution on google how to decouple the modules in Java then I implement the logic to the patterns. As it helps to call the classes directly, re-use the code, and doesn't require to interact with all classes because we can have different modules with packages by doing decoupling.

I added the size of the item to add more categories in a POS but when I run the code and I gave the input as “Large” it throws an exception which was unexcepted and the code was failed I was shocked as the code stopped in between at that time I closely look into each and every class where I got wrong and by checking tens of times, I got the solution for it. It happened because I didn’t declare the size of the product in the Testing class which was already declared in the Driver class. So, I fix the bug by adding size () to the Testing class and then finally the code runs smoothly, without any exception and errors.

I was also having difficulties with the testing part, when I test the code with JUnit. I encountered with an error where “GroceryItemFactory” didn’t pass the test and rest of the two (adapter and singleton patterns) passed. It was so challenging for me because testing the code and not knowing the actual problem is quite difficult for me. It took me 2 days to check the error in the program because I used Assertion so that I can make assumptions for testing if the values will be equal and null but assert throws an error because it getting fail due to incorrect value which I figured it out and then I made some changes to the program. At last, I was able to passed all the test cases with zero (0) errors.

While doing this project, I learned many things and faced some challenges which was I am not aware of before. I enjoyed while writing up the code and I learned how to use design pattern in a practical way. This subject and project taught me everything about Oops concept and design pattern and now I am able to use the design patterns.