Open questions your group is struggling with:

For phase 2, we mainly aimed to implement design patterns discussed in phase 1. However, during implementing the state design pattern in the Inventory package, Chongjie Sun, who is responsible for the Inventory packages, decided to remove the state design pattern due to code redundance. In phase 1, we wanted to use state design pattern to allow the inventory object to change its behavior when the internal state of that inventory object changes. In other words, when one of the item quantities equals to zero, the state/behavior of the inventory object should change to out-of-stock status. Otherwise, the inventory object should always be in the normal state. However, during the actual implementation, Chongjie Sun found that this functionality could be simply accomplished by adding a new state attribute in the Inventory entity class. So far, even though we haven't combined the state design pattern in our program, we are not sure if this is a correct decision.

The debugging process of reading the CSV file was another question that our group struggled with. Since none of our group members had background knowledge on reading and writing CSV files, group members, who needed to use the CSV file to store data, learnt everything from scratch. Therefore, we faced a lot of challenges during debugging and connecting with GUI. For example, we took plenty of time to resolve the problem that our code could only read the first line of the CSV file. Another problem was that using reader and writer method of CSV file to fail registration of repeated usernames could not run properly. The processing of developing CSV files took a long time and was fluctuated; however, we resolved all the errors eventually.

What has worked well so far with your design:

Our code organization has followed the clean Architecture, as well as SOLID. In short, our phase 2 code organization is much better than codes in phase 0 and phase 1. And, we have also created different test cases to test our functions. So far, all the designed functions can be used right now, such as viewing order history etc. Different types of data, such as order information and user information, now can be stored in local files to accomplish data persistency. Moreover, implementing the extensions did not cause any serious errors that would hurt the skeleton of the program from phase 1. The GUI and the remaining portions from the program that were to be implemented in Phase 2 were done. All new classes and/or methods that were added made the code more efficient and cleaner.

A summary of what each group member has been working on:

Group Members	Phase 1 Contribution	Phase 2 Contribution	Common Contribution in Phase 2
Meilun (Elva) Shen	Implemented User Package and tests. Revised Specification. Wrote the progress report.	Accomplished data persistency by storing user information in the local csv file. Connecting the User Package to the GUI. Finished tests for User Package.	Communicating and helping each other with the interactions between different packages. Writing documents and preparing the final presentation. Revised CRC model.
Gen (Reagan) Li	Implemented Message Package and tests. Revised the general structure of CRC model.	Connecting the Message Package to the GUI. Finished tests for Message Package.	
Walter Huang	Implemented GUI.	Completed implementing UI panels.	
Chongjie (Kurisu) Sun	Implemented Inventory Package and tests.	Connecting the Inventory Package to the GUI. Finished implementing Inventory Packages and tests.	
Hao Li	Implemented Order Package and tests.	Accomplished data persistency by storing Order information in the local csv file. Connecting the Order Package to the GUI. Finished tests for Order Package.	
Xingru (Jamie) Ren	Implemented Item Package and tests.	Finished tests for Order and Item Packages	
Cheuk In (Michelle) Tam		Accomplished data persistency by storing item information in the local csv file.	