Objective:
Implement the software. Review the implementation and draw UML class diagram based on the actual implementation. Reverse engineer code to UML. Collaborate using GitHub.

Scenario:
Your team has been asked to implement the OSC prototype. The objective is not design conformance but to evaluate and refine the design based on the needed implementation. See Assignment 2 for further details.

Requirements:
-

- The user can login with a username and plain-text password (no hashing / encryption required)
  registration not required, login function to obtain and verify uname / pw from existing db or file
- The system will list the name, description, and price of current items from available inventory (to include items from 4 available categories) - household items (e.g.,aluminium foil, toothpaste), books (e.g., SW Design Book), toys (e.g., Spiderman figure, doll), and small electronics (e.g., camera, mobile phone)
- The user can choose items (name and quantity) to build a shopping cart (cart) that is0 associated with their username
- The cart will contain items, quantities, prices, and a calculated total for all items
- The user can remove items from the cart
- The user can proceed to purchase the items
- The user will provide a shipping address (required for each purchase - can be obtained from user database or entered at this stage)
- The user will provide a credit card number (required 10 digit special OSC purchasing card)
- The user must confirm the purchase
- The purchase (all items in cart, prices, total, credit card, address, username) is stored between uses (e.g., order history)
- The user can view past purchases
- The inventory of items is updated and redisplayed to the user after a completed purchase (minus the items purchased)
- The user can log out of the system

Names, NetIDs, and GitHub Usernames:

Kiran Thapa - kt1397/Diglet7781

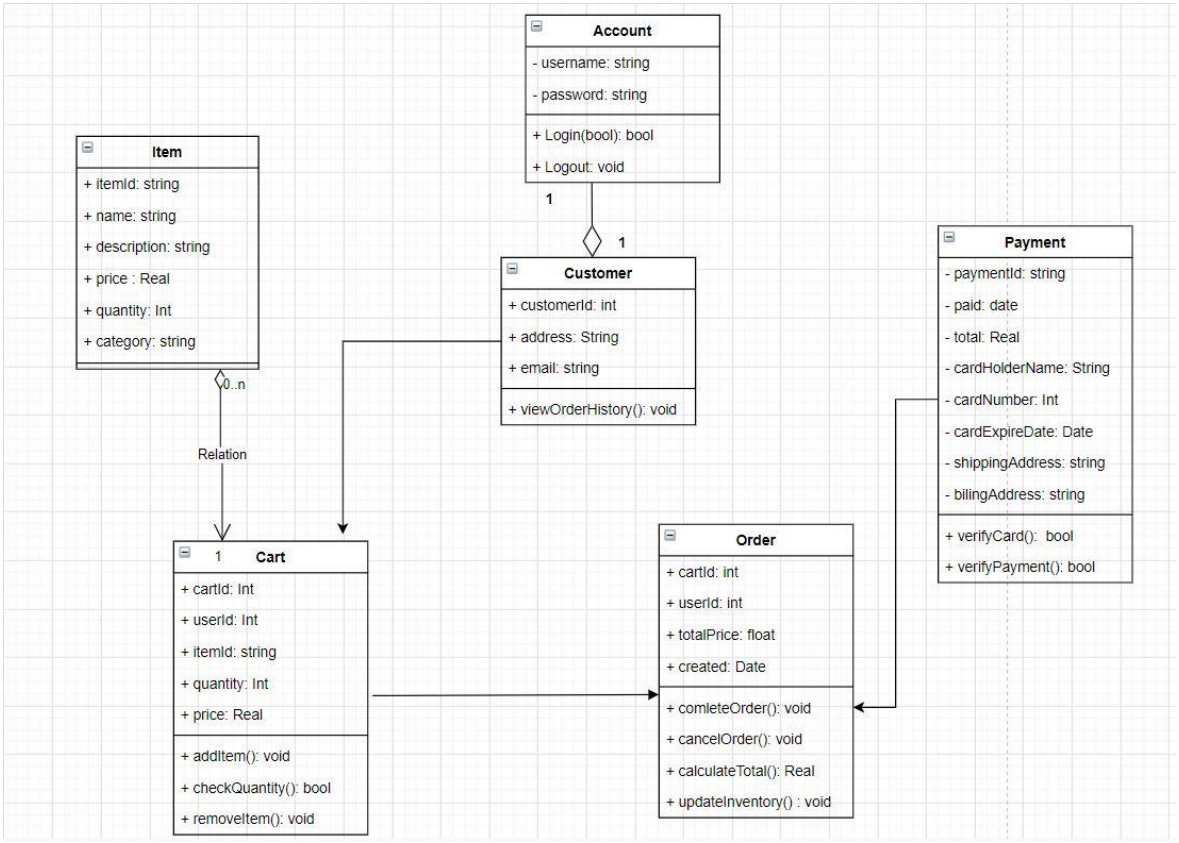
Austin Rowell - arr461/Ausrowe0417

Tom Jackson - tpj24/TomJacks242017

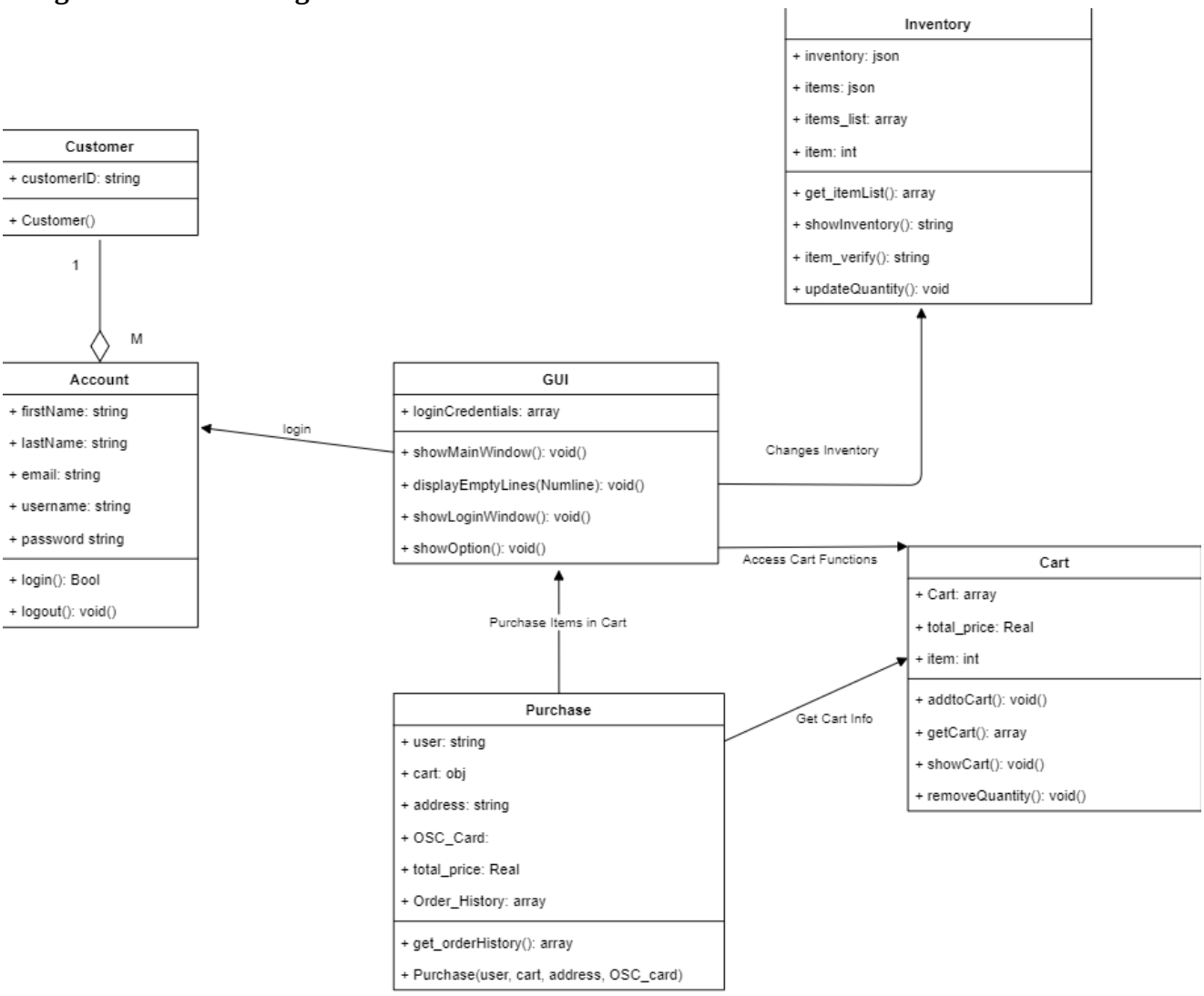
Joshua McQuilling - jm4139/jm4139

GitLab Repository Link (this is the URL):
https://gitlab.com/Ausrowe0417/software_arch_group_hw_3

Assignment 2 Class Diagram:



Assignment 3 Class Diagram:



Similarities and Differences

Discuss similarities and differences between the class diagrams.

Differences

One notable difference is the center of the descriptive diagram has the GUI methods while in the prescriptive, GUI was not even mentioned. Although most methods made it through some were changed and some were removed completely. *Payment* became *Purchase* and *Item* was made into *Inventory*. Along with these two changes, Order was removed due to repetition within the system methods.

Similarities

Overall though, most methods, including *Account, Cart, and Customer,* that were described in the prescriptive diagram made it through to the descriptive version.

Why is the implementation different? What aspects of the implementation did the original prescriptive design not consider? Why were the changes needed?

The thought process needed to create the prescriptive and descriptive diagrams is completely different as well as the implementation behind it. Without the skeleton to work from, writing the code for Assignment 3 was just an idea and then going forth with the code. If we were to use the prescriptive diagram, most of the methods would have been decided already. The code base would be generally the same though the route to get there may have been different. As mentioned above, our prescriptive diagram did not mention the GUI. This was a much needed change because without the GUI, the users who access this system would not have much of a system to work with.

Lessons Learned

What did you learn? How does this assignment compare with other academic assignments or professional experiences?

Two big takeaways from this assignment were one, how to work correctly as a group using the git repository as a guideline, and two, how the design process changes when the code is written before the skeleton of the system. Throughout college so far, and especially during our Software Engineering classes, most labs/assignments were given from the ground up rather than the way it was done in this assignment.

What are the benefits of the upfront design process? Did the upfront design in Assignment 2 help in assigning coding tasks? How?

The biggest benefit that's easily recognizable in this design process is being able to have a generalized direction the system will be developed in. There's a skeleton of a system that all design engineers on the system can reference to and make design decisions based on it. In Assignment 2, creating the class diagram first allowed for us to start the design process early and get a foundation up and running.

How familiar are you with GitHub/GitLab?

Within the group of the four of us, only two of us were comfortable with working with the repository. Unfortunately, using this environment has not been utilized very commonly throughout the four years working in this major. Through the use of the supplied links and other ways of research, all of us were able to figure out the basics at least and get comfortable with the format of GitLab.