09 Breaking Bread

Elena Ingraham
Friedrich Amouzou
Hussain Quadri
Phi Trang

Vision

Our vision is to practice object oriented programming by designing a shop that utilizes a lot of the principles and patterns that we have learned in class.

Project description

Breaking Bread is an online bakery store with a catalogue of bakery utensils, merchandise, and customizable baked goods that are shipped directly to your home.

Summary

In the past two weeks, we have adjusted our class diagrams based on the critique given. We also have created most of our classes and the necessary functions within them. Some of our classes have completed functions and some just have the skeleton. We have also considered creating a GUI with Java Swing and ways to incorporate our code into a GUI (seems tough). We have also edited our activity diagrams and centralized our use cases.

Breakdown of work

Elena Ingraham

I would first like to mention the graph of commits on github does not accurately depict the work that I have done thus far since I created my own branch. The graphs do not visualize commits made to branches other than master or merge commits. Which is really silly since it is better practice to write to a specific branch and then commit to master once you are ready. Anyways, thus far I have:

- Helped establish the Use Cases used in the Activity and Sequence diagrams
- Made the Activity Diagram in Part 2
- Made the Skeleton code for the Cart class
- Wrote outlines of the getter and setter methods within the Cart class

Friedrich Amouzou

- Made use-case diagram and documents
- Made class diagrams for both part 2 and 3
- Refactored and Implemented the deal system and wrote the Deal, FixedDeal, PercentDeal, and DateRange classes
- Implemented most of the non-custom Product class

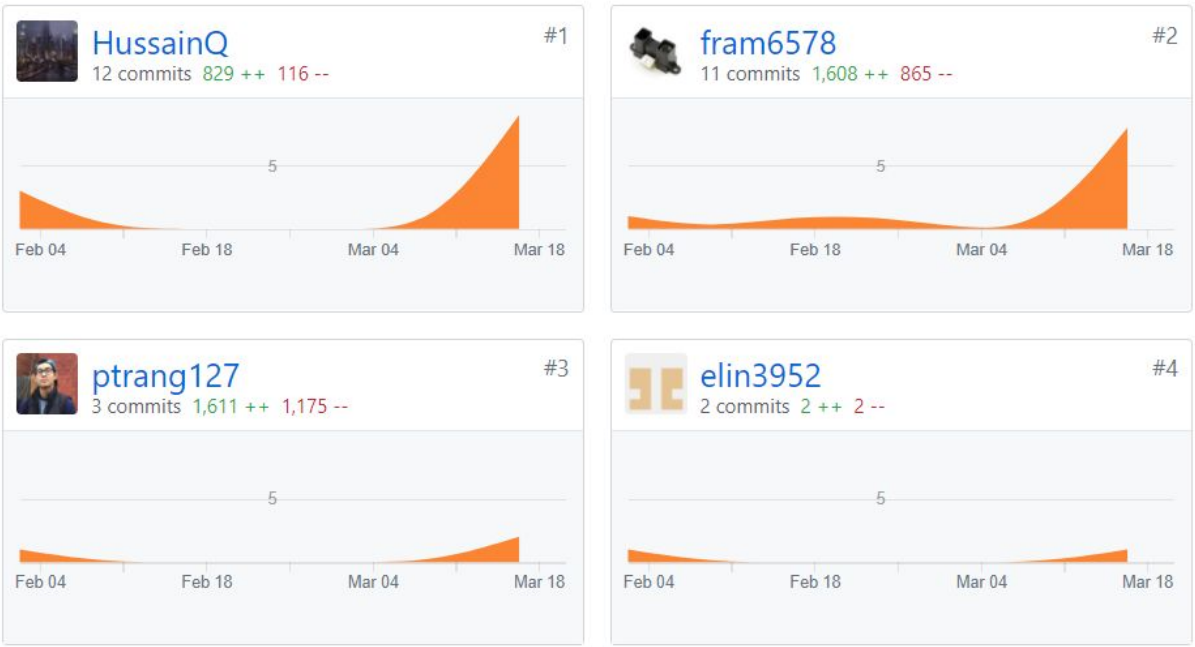
Hussain Quadri

- Created the sequence diagrams for part 2.
- Created, refactored the User class, as well as Admin and Customer subclasses.

Phi Trang

- Created Inventory skeleton class
- Requirements analysis
- Overambitious mock UI
- Java Swing research

GitHub Graph



Estimate Remaining Effort

A considerable amount of work needs to be done in order to have implemented the design in our class diagram. The most difficult task will be to implement our UI system and our database system since it requires getting accustomed to entirely new libraries. Beyond just implementing the existing diagram and a lot of refactoring needs to be done in order to optimize our code, specifically by following design patterns and standard conventions.

Design Changes

Global

- Setters and other void functions return void and raise an exception on error rather than returning a status integer
- Variables with specific implementations, i.e., ArrayList were replaced with the more generic List type in order to utilize the Strategy design pattern

Inventory Class

- Deals are now stored in the inventory class

Deal Class

- The fixed amount off deal and percent off deal were abstracted into subclasses
- The start and end date variables were abstracted into the DateRange class
- Deals now have description strings

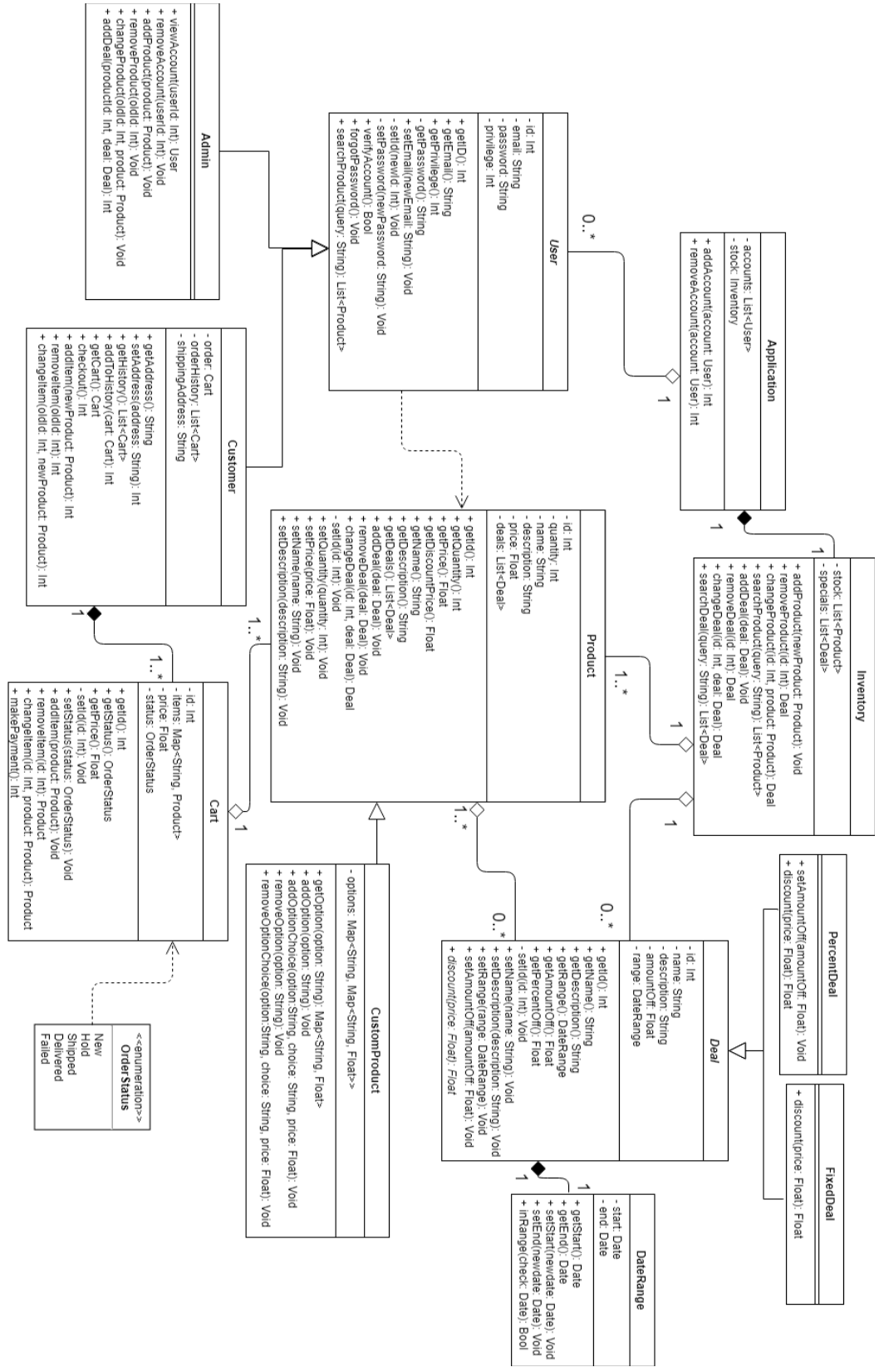
Cart Class

- Items are now stored in a hashmap of their name and product object for ease of storing and printing

Next Iteration

- Fill in our skeleton code with actual functionality.
- Possibly connect our code to a mySQL database to manage our items
- Design a simple GUI that incorporates our classes

Previous class diagram



Completed Class diagram