

Grocery Delivery Service

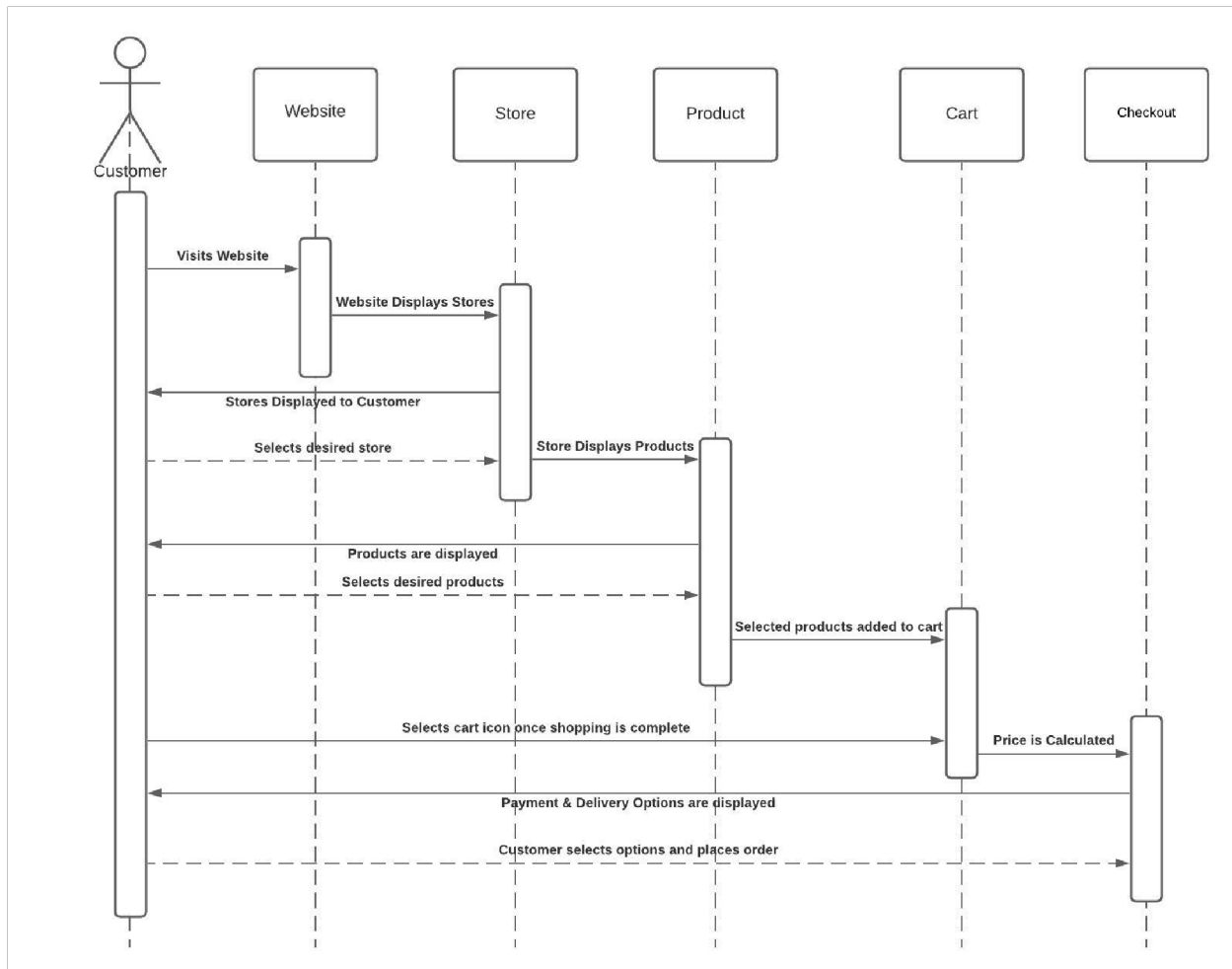
Report 2

Blake Jagielski

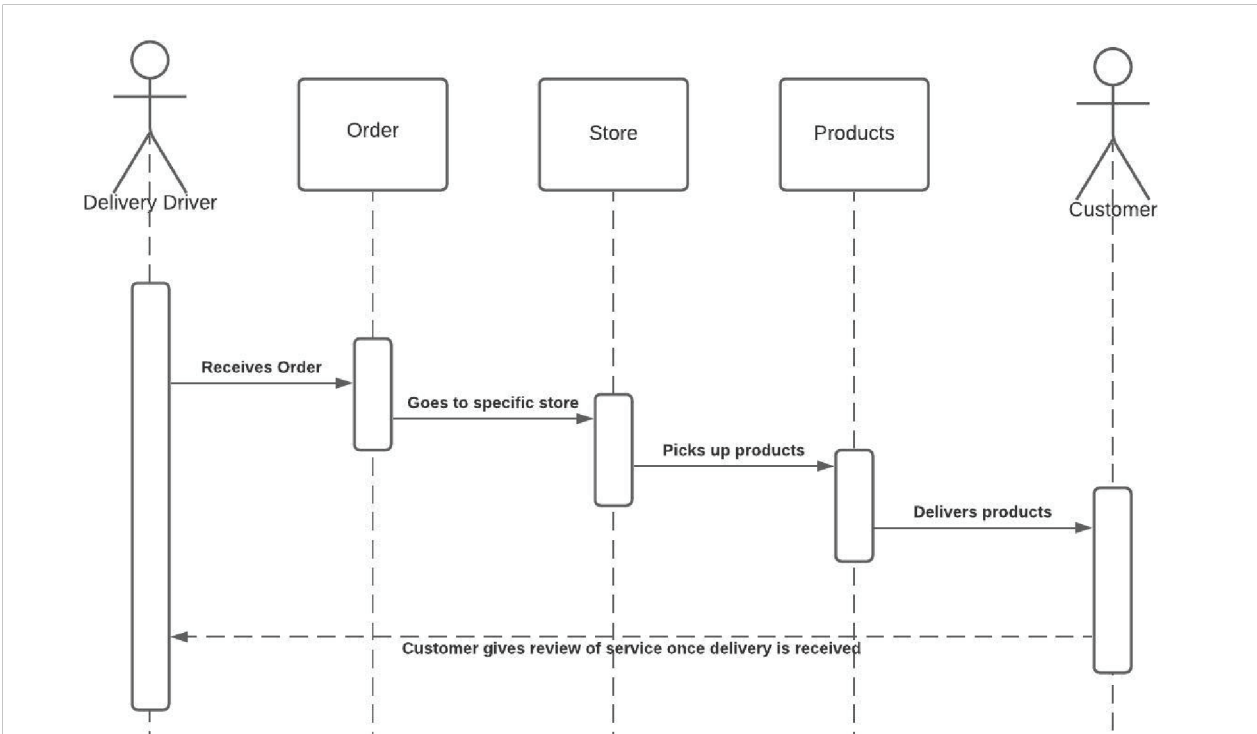
Table of Contents

• Interaction Diagram.....	3-4
• Project Coordination & Progress Report.....	4-5
• Use Cases Implemented.....	4
o Functional Cases.....	4
o Cases Being Tackled.....	4
o Other Project Activities.....	4-5
• Class Diagram, Data Types, & Operation Signatures.....	5
• Traceability Matrix.....	5
• System Architecture & System Design.....	6-7
• Architectural Styles.....	6
• Identifying Subsystems.....	6
• Persistent Data Storage.....	7
• Global Control Flow.....	6-7
• Execution Order.....	6
• Time Dependency.....	7
• Hardware Requirements.....	7
• User Interface Design & Implementation.....	7
• Design of Tests.....	7-8
• Plan of work.....	8
• References.....	8

Interaction Diagram



- For the purpose of this diagram, I used the Open-Closed Principle, meaning that the source code of the classes cannot be changed, but its behavior can be extended. The source code for the products remains the same once a user adds it to the cart, but the product information extends to the class cart. The source code for the cart remains the same as once the cart is cleared it returns back to its original state.



Project Coordination & Progress Report

Use Cases Implemented

Thus far, there are many use cases that have already been implemented in the project. For example, the computer can display the website along with the customer being able to visit the website. Currently, the customer can only click on the desired stores and view the products and that is as far as I have gotten.

Functional Cases

- Customer goes to site
- Customer selects desired store
- Computer displays website
- Programmer creates application/website

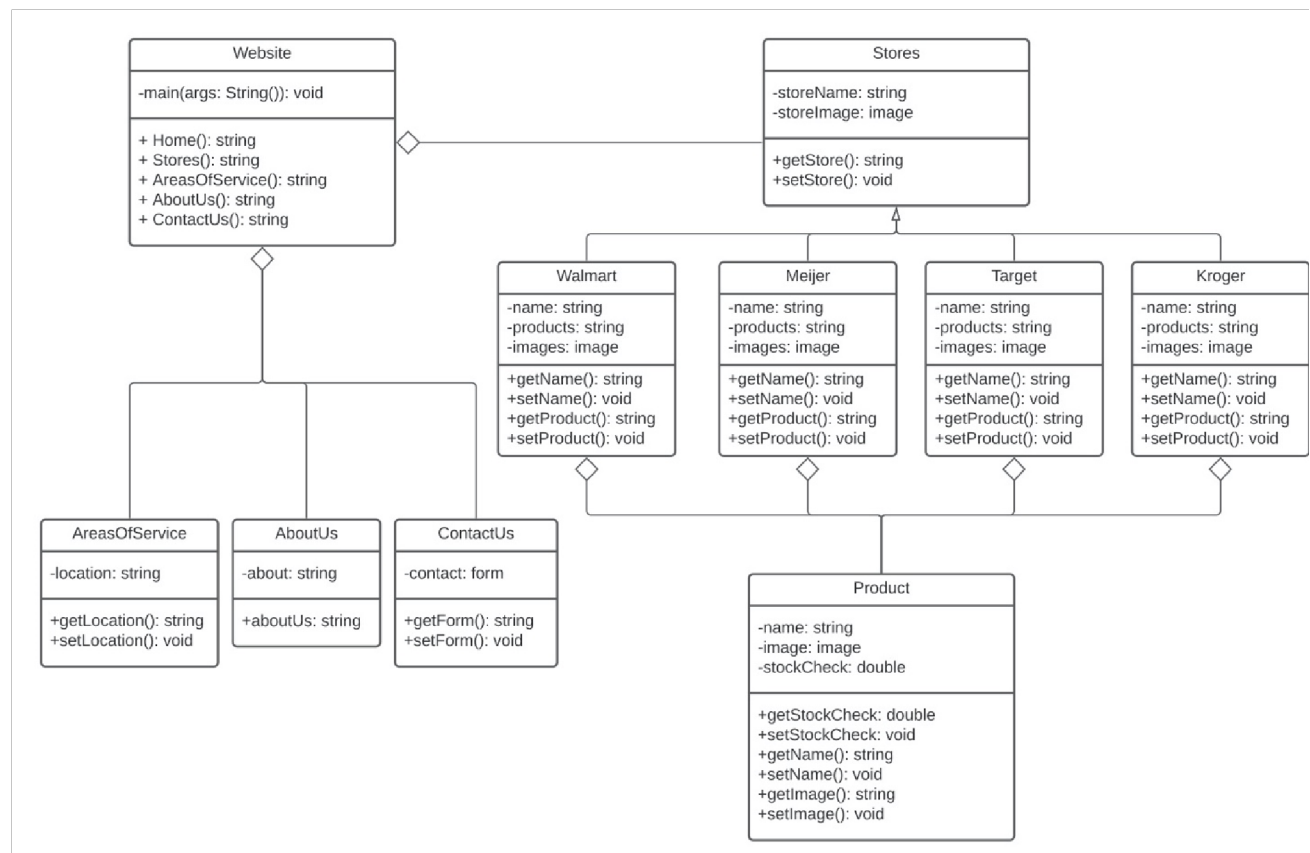
Cases Being Tackled

- Customer adds products to cart
- Customer checks out and verifies delivery options
- Deliver Driver receives order

Other Project Activities

I am currently working on implementing the pop-up menu for an item that is low in stock or out of stock that allows you to select another item of a different brand. I am also working on the checkout procedure for the customer in order to create the best user-friendly experience.

Class Diagram, Data Types, and Operation Signatures



Traceability Matrix

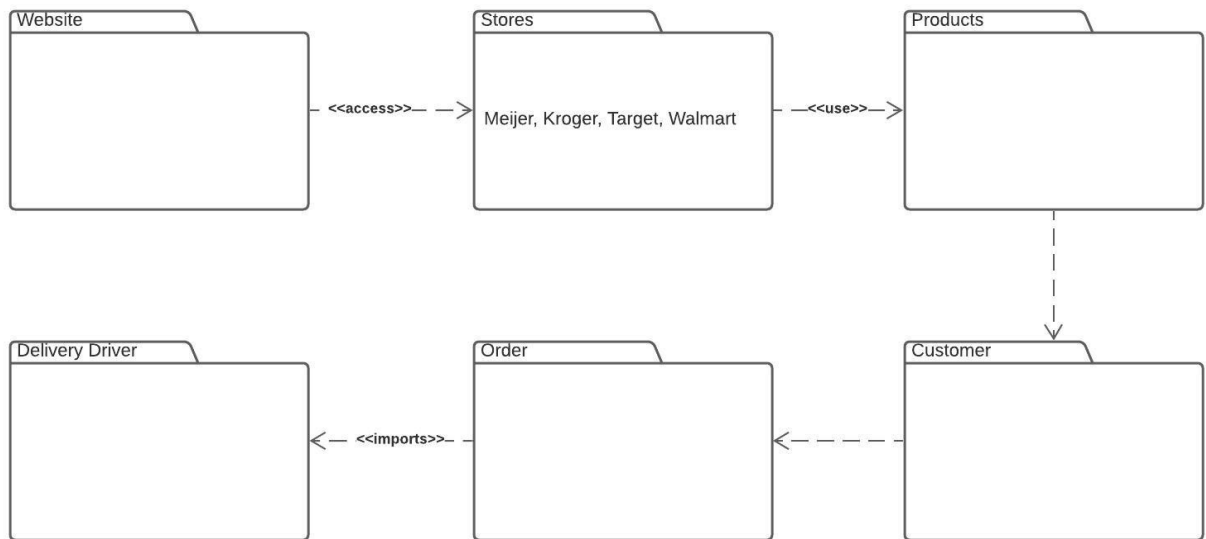
Requirements	PW	Selects Store	Checks out & verifies delivery options	Picks up & deliver products	Adds Products to cart
Items Selected	5	X		X	X
Checkout Process	5		X	X	
Proper Delivery	1		X	X	
Total PW	11	5	6	11	5

System Architecture and System Design

Architectural Styles

- This website will be service-oriented, and customer focused as it will require input from the customer in order to execute an action and complete an order. The overall design of the website is very simplistic to help accommodate a wide range of individuals by making it easier to navigate the site and crowd the web pages with unnecessary information.

Identifying Subsystems



Persistent Data Storage

- The website will only store the customers information if they create an account and choose to store the order in their profile. The only other information that they have the option to save is there address and payment information.

Global Control Flow

Execution Order

The way a user navigates the website is completely up to them. When placing an order, the customer will begin by selecting which store they would like their items to be purchased from, and then from there they can select the items they are wanting to have delivered. The user can choose to navigate the store any way they please through the website and once finished, will complete their purchase through the checkout page and once the order is received and processed, a delivery driver would immediately head to the store to begin picking up the items.

Time Dependency

- The products are the only time dependent items on the website. If you wait on the product page for too long, the number in stock of the item may not be as accurate as when you first visited the page. It is often best to refresh the page to ensure that the wanted product is still in stock or to prevent a glitch in the system thinking that there are some in stock and the product not being available once the order is placed.

Hardware Requirements

- For this website, the only necessary requirements are a computer and an internet connection. While the website may work on a mobile device, some items that are displayed on the site may seem disarranged and tasks may be more difficult to complete.

User Interface Design & Implementation

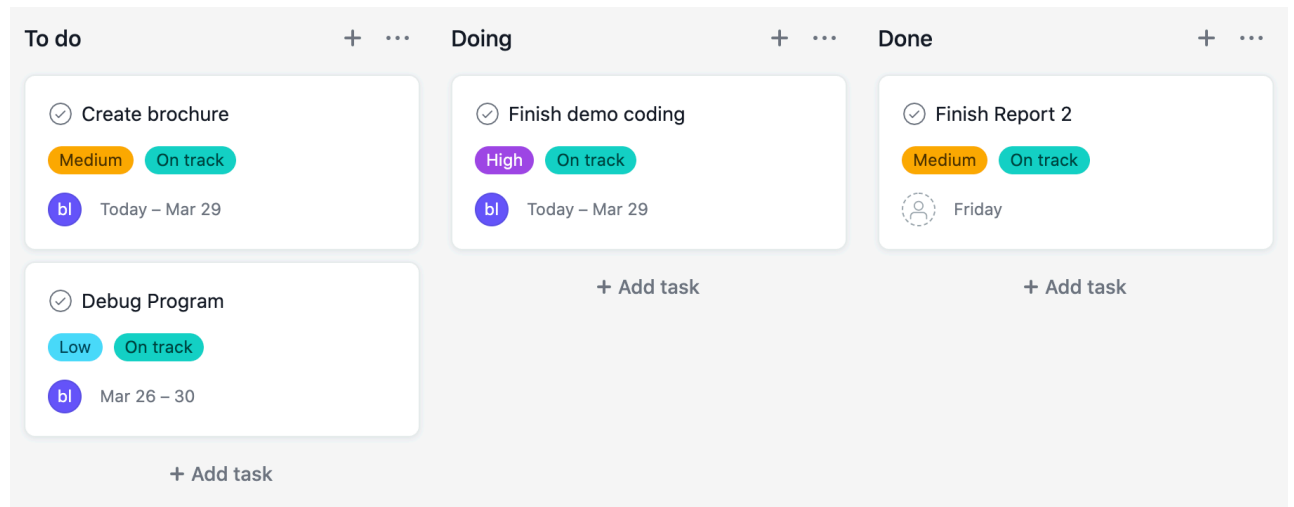
- I did not have to make any changes to my initial screen mockups other than changing a few appearance issues to make the website look more updated and user friendly.

Design of Tests

- For this project, I have implemented a stock check case that notifies the consumer when the item has 3 or less items in stock. This notification prompts a pop-up where the user can select a similar product that is in stock. I am still in the implementation phase of this code and am working on the specifics to have it running for the first project demo. In order to test this, I will visit the website as a customer, and proceed as if I am placing an order, and once I come to an item with less than 3 in stock, I will ensure that it prompts the pop-up.
- I have also begun testing saving the users information to an “account” on the website. Thus far, I have found this rather difficult since I have not created a database for the system, so for the purpose of this project, I will be implementing a “default” user who will already have a picture of their residence uploaded for the delivery driver to see and other information will be present as well.

- With the integration testing strategy, I have been working on the project piece by piece ensuring that every little thing that I add still works with the other blocks of code already present in the program. This approach has made it very simple to find issues that are occurring and address them immediately to avoid any problems once the code is completed.

Plan of Work



References

- Image References have yet to be determined
- Deitel, Harvey M. *Java: How to Program (Late Objects)*. Pearson Education Limited, 2015.
 - This is a reference to a textbook that I am currently using for another class, but I have been looking at some of the concepts from the book for help with the programming.
- Tilley, Scott & Rosenblatt Harry *Systems Analysis and Design, Eleventh Edition*. Cengage Learning, 2017
 - Course Textbook