

**Grocery Delivery Service**

**Report 3**

**Blake Jagielski**

## Table of Contents

|  |       |
|--|-------|
| • Problem Statement.....                                 | 4-5   |
| • Glossary of Terms.....                                 | 5-6   |
| • Functional Requirements.....                           | 6     |
| • Nonfunctional Requirements.....                        | 7     |
| • User-Interface Requirements.....                       | 7     |
| • Stakeholders.....                                      | 8     |
| • Actors & Goals.....                                    | 8     |
| • Use Case Casual Description.....                       | 8-9   |
| • Use Case Diagram.....                                  | 9     |
| • Traceability Matrix.....                               | 9-10  |
| • Fully Dressed Descriptions.....                        | 10    |
| • System Sequence Diagram.....                           | 11    |
| • Preliminary Designs.....                               | 11-13 |
| • User Effort Estimation.....                            | 13-14 |
| • Domain Model.....                                      | 14    |
| • System Operations Contract.....                        | 15    |
| • Project Size Estimation.....                           | 15-16 |
| • Interaction Diagram.....                               | 16-17 |
| • Project Coordination & Progress Report.....            | 17    |
| o Use Cases Implemented.....                             | 17    |
| o Functional Cases.....                                  | 17    |
| o Cases Being Tackled.....                               | 17    |
| o Other Project Activities.....                          | 17    |
| • Class Diagram, Data Types, & Operation Signatures..... | 18    |
| • Traceability Matrix.....                               | 18    |
| • System Architecture & System Design.....               | 19    |
| • Architectural Styles.....                              | 19    |
| • Identifying Subsystems.....                            | 19    |
| • Persistent Data Storage.....                           | 19    |
| • Global Control Flow.....                               | 19-20 |
| • Execution Order.....                                   | 19    |
| • Time Dependency.....                                   | 20    |
| • Hardware Requirements.....                             | 20    |

|   |       |
|---|-------|
| • User Interface Design & Implementation..... | 20    |
| • Design of Tests.....                        | 21    |
| • History of work.....                        | 21-22 |
| • References.....                             | 22    |
| • Summary of Changes.....                     | 22    |

**Problem Statement:** Problems constantly arise in everyone's lives that make them unable to do the everyday tasks needed to survive. Throughout the recent years, technological advancements have made it possible for these tasks to get completed by an outside party. The grocery delivery service applications have been very beneficial to those who are unable to go to the store due to injury or sickness and also to those who really don't like going to the store. As a customer, I prefer to get specific items when going to the store and like certain brands more than I do others. What the current grocery delivery services lack are the ability to specifically select a replacement product if the wanted product is not in stock. Most of the time, if a product is not in stock, the delivery person will substitute for another brand or not buy the item at all. This can be frustrating as the consumer because the one product they did not have may have been something that you needed right away. Even though the delivery person made the substitution, they probably selected the nearest item that looked similar when you could have made the selection yourself on what other brand you wanted. This issue could be easily avoided if the applications would allow for the buyer to see the number of items that are in stock and be able to select a back-up item if the item that they originally wanted is out of stock or low in stock. That way if the delivery person gets to the store and the selected item is not in stock, they know automatically what to substitute it with instead

of picking a random brand. The final issue that doesn't occur as often is delivering to the wrong address. With COVID, a lot of delivery services have been dropping the items on the porch without directly having contact with the person they are delivering to. The issue with this is they do not 100 percent know that the items are being delivered to the right house.

### **Glossary of Terms**

- # In Stock: This will display the number in stock of each item.

**Only 1 left in stock  
(more on the way).**

**18 left in stock**

- Flat-Rate: Every consumer will get charged the same flat-rate delivery fee no matter how big or small the order is.
- Returns: Returning an item will only be accepted if the item is received damaged or the wrong item was delivered.
- Concurrent User: the process of having several users on the site at one time.

Once the site goes live across the nation, it will be extremely important that the site can withstand an unlimited amount of users at one time without the possibility of a website overload or shutdown.

- Special Limitations: these would most likely be found in the terms and conditions of the application and identify things such as specific delivery cut-off times and price changes with on-sale items.
- Authentication: require all users to provide a valid ID and also a picture of the front of the residence where the items are being delivered to always ensure that the items arrive at the proper address.



## Functional Requirements

| Requirement       | Priority Weight | Description   |
|-------------------|-----------------|---|
| Stock Check       | High            | Allow users to see the number of items that a store has in stock.                             |
| Item Substitution | High            | Allow users to select a replacement item if the desired item is out of stock or low in stock. |
| Proper Delivery   | Low             | Ensure that the items arrive on time, not damaged, and to the correct address.                |

## Nonfunctional Requirements

| Requirement | Priority Weight | Description |
|-------------|-----------------|-------------|
|             |                 |             |

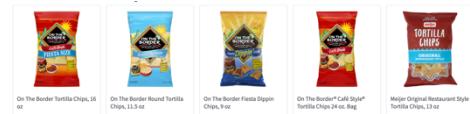
|                 |      |  |
|-----------------|------|--|
| Reliability     | High | Ensuring that the application can sustain a high volume of users at one time.            |
| Usability       | High | Creating a user-friendly application that anyone over the age of 16 will be able to use. |
| Delivery Prompt | High | Customer will be prompted to sign-off on the app once the delivery person has arrived.   |

## User Interface Requirements

| Requirement          | Priority Weight | Description  |
|----------------------|-----------------|--|
| Items in Stock       | High            | Actively display the number of an item in stock.               |
| Item Substitution    | High            | Prompt a list of similar items for substitution                |
| Address Verification | Low             | Take a picture of your residence to ensure of proper delivery. |

18 left in stock

Similar



**Stakeholders:**

- Blake Jagielski (Myself, Project Creator)
- Grocery Stores
  - Meijer
  - Kroger
  - Walmart
  - Etc.
- Grocery Store Customers
- Investors
- Management Employees
- Financial Employees
- Programming Employees

**Actors & Goals:**

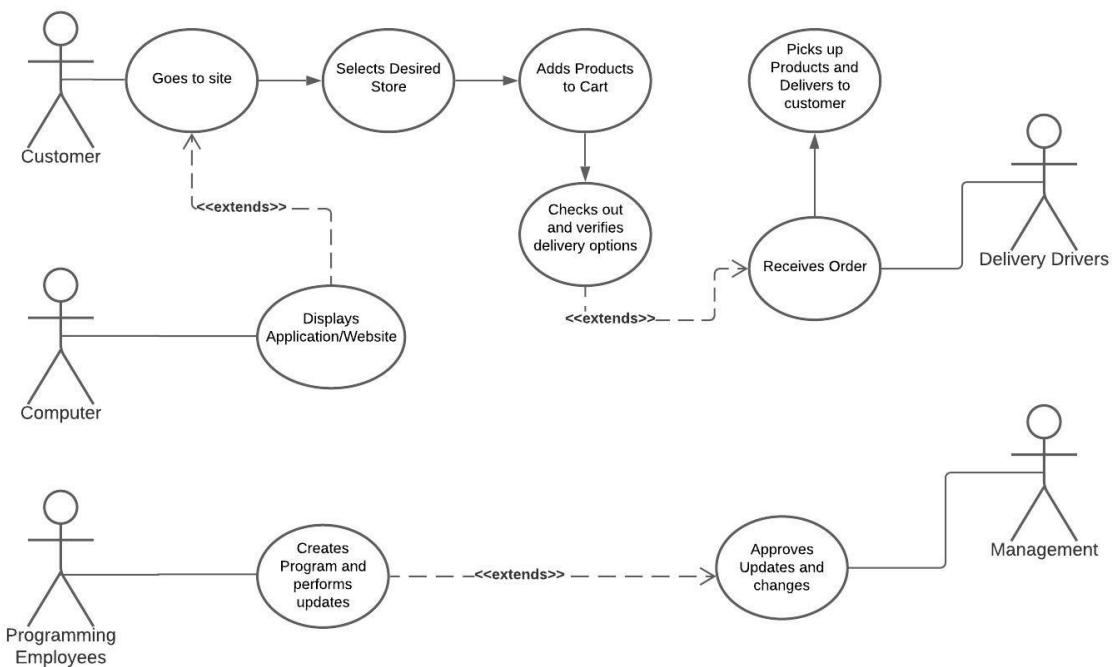
- Customers
  - Will use the application to purchase grocery products from their favorite stores.
  - The goal is for the customer to be able to use the application with ease and without the difficulties and errors found in other delivery applications.
- Programming Employees
  - Will be in charge of any debugging or updates that need to be applied to the application.
  - The goal is for the programming employees to complete the tasks and projects given to them in an efficient and timely manner.
- Computer or Cellular Device
  - These devices can be used to navigate the grocery delivery application and will create a user-friendly experience for all customers.
- Management Employees
  - Control business operations and communicate with store locations about brand deals and product specifications.
- Delivery Drivers
  - Receive delivery notifications through the application and respond by going to the selected store(s) and picking up the desired items.

**Use Case Casual Description:**

- Stock Check
  - Customers will be able to see the number in stock of the items that they select to purchase. For example, if an item is out of stock, the customer would be prompted to select a similar item.

- Item Substitution
  - If an item is out of stock or limited in stock, the customer will be prompted to select a similar item. For example, if an item shows that there are only 2 left in stock, the user will see a list of similar items displayed that they will be able to choose from as a back-up item in case the item is out of stock once the delivery driver gets to the store.
- Proper Delivery
  - Customers will be prompted to upload a photo of the front of their residence where the delivery is expected. If their residence is an apartment, they should upload a photo of the apartment door with the number clearly displayed.

### Use Case Diagram:



- In this diagram, the customer visits the website, selects which store they would like to order from, adds the desired products to the cart, and then checks out and verifies the delivery options. The computer is displaying the website to the customer and once the order is placed, the delivery driver will receive notification of a delivery.

### Traceability Matrix:

| Requirement    | Priority Weight | Description                                |
|----------------|-----------------|--|
| Items Selected | High            | User will be able to select items from the |

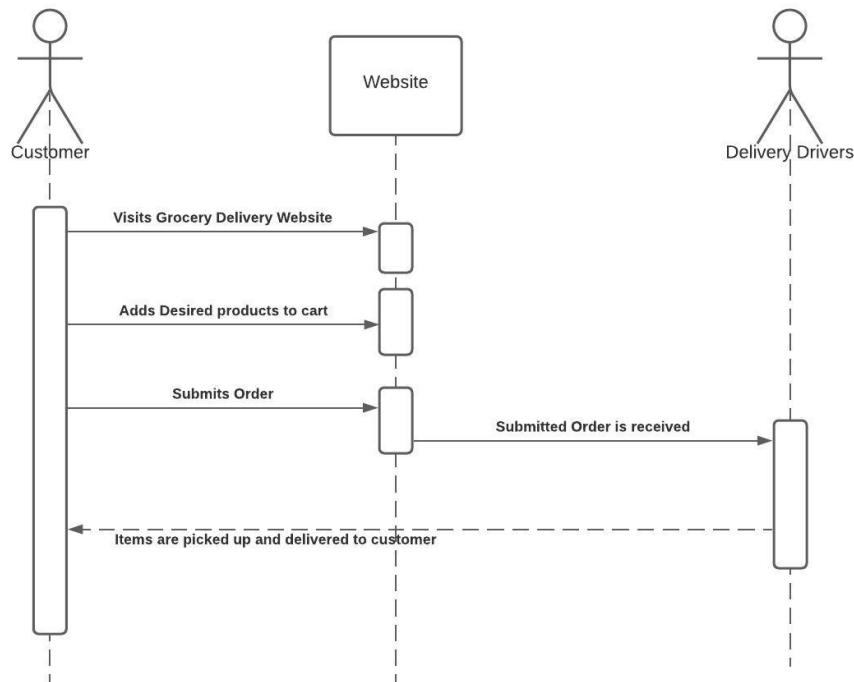
|                  |      |  |
|------------------|------|--|
|                  |      | store's website with ease.   |
| Checkout Process | High | Customers will have a user-friendly experience throughout the ordering process and paying for their items. |
| Proper Delivery  | Low  | Ensure that the items arrive on time, not damaged, and to the correct address.                             |

### Fully-Dressed Description:

- Customer Placing an Order
  - Customer will visit the website on his/her mobile device or computer.
  - Once on the site, user will select the 'Stores' Tab in the navigation bar.
  - The user will then select their preferred store to shop with.
  - The user will then be redirected to that stores page.
  - The user can then select products by categories or use the search bar.
  - Once the user finds a desired product, they will add the product to their cart.
  - Once all of the items have been added to the cart, they will click on the shopping cart icon in the top right-hand corner of the page.
  - The user will click the 'Checkout' Button.
  - This will redirect the user to the checkout page where they will enter address and payment information.
  - Once completed, the user will click 'Submit Order'.
  - Once submitted, the nearest delivery driver will be notified.
  - The delivery driver will go to the store and pick up the requested items.
  - Once finished, the driver will deliver the items to the customers residence and the order is complete.

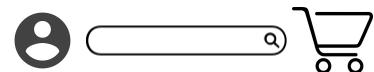
## System Sequence Diagrams:

- Customer placing an order through the website



## Preliminary Design:

# Grocery Delivery Service

[Home](#)[Stores](#)[Areas of Service](#)[About Us](#)[Contact Us](#)

Making Everyday Life Easier One Delivery At A Time



# Grocery Delivery Service

[Home](#)[Stores](#)[Areas of Service](#)[About Us](#)[Contact Us](#)

Click on your desired store to begin shopping!

[Walmart](#)[meijer](#)[Kroger](#)

# Grocery Delivery Service

[Home](#)[Stores](#)[Areas of Service](#)[About Us](#)[Contact Us](#)

We are currently serving the Indianapolis Region and plan on expanding nationwide as we continue to develop our business!



# Grocery Delivery Service

[Home](#)[Stores](#)[Areas of Service](#)[About Us](#)[Contact Us](#)

We want our customers to receive their desired products from their favorite stores from the comfort of their own home. Our service creates a user-friendly experience like no other and provides customers with the guaranteed satisfaction they deserve and results that would make them think they went to the store themselves!





Your Name (required)

Your Email (required)

Subject

Your Message

**Send**

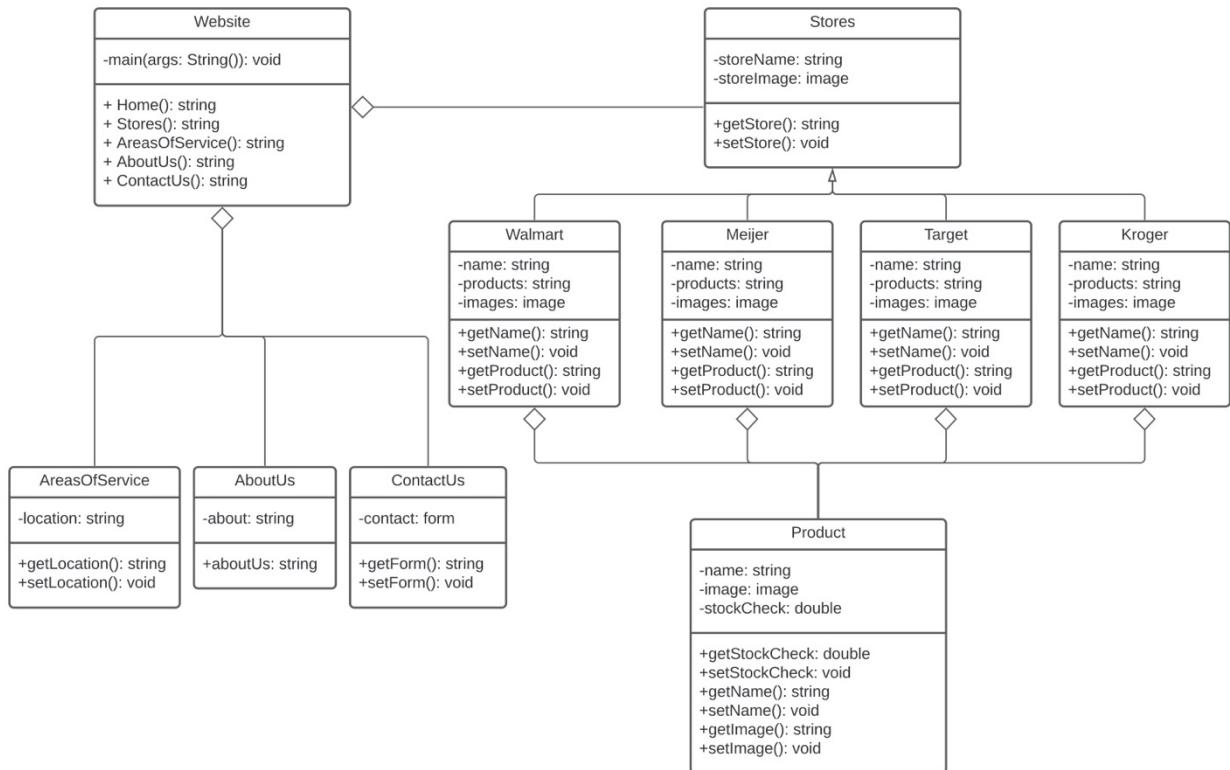


### User Effort Estimation:

- Customer wants to buy a specific product from a specific store
  - Customer enters the website
  - Clicks on the 'Stores' navigation tab
  - Clicks on the desired store they would like to shop
  - Customer can select a certain category or search for their grocery items.
  - Customer selects product
  - Customer adds product to cart
  - Customer clicks on the cart icon
  - Customer clicks on checkout
  - Customer enters payment information (unless saved payment information is available).
  - Customer clicks proceed with payment and delivery options
  - Customer will upload a photo of the front of their residence (unless one is pre-saved to your account.)
  - Transaction is complete
    - Total Mouse Clicks: 10-14 mouse clicks (Only 1 item purchased)
- Customer wants to contact us
  - Customer enters the website
  - Clicks on the 'Contact Us' navigation tab
  - Enters Name, Email, Subject, and their message to us.
  - Clicks send once complete
    - Total Mouse Clicks: 4-7 Mouse Clicks (Dependent on if the 'Tab' Key is used to go through contact form.)

- Customer wants to track their order
  - Customer enters the website
  - Clicks on the ‘My Account’ Icon in the top right corner of the page
  - Selects ‘Track an Order’
  - Selects which order(s) they would like to track
  - Result is displayed
    - Total Mouse Clicks: 4 Mouse Clicks (1 Single Order)

## Domain Model



For the purpose of this project, all 4 of the stores will have the same products displayed when the tab is clicked. A different store image will still be displayed, but the same products will be displayed across each store. The website is the main class in this program and when visited will automatically display the home page. The Stores, Areas of Service, About Us, and Contact Us pages are all separate classes. The products displayed on the store pages will consist of a product name, image, price, and quantity in stock.

## System Operation Contract

### Contract CO1: orderProduct

---

|                          |  |
|--------------------------|--|
| <b>Operation:</b>        | orderProduct()   |
| <b>Cross References:</b> | Use Cases: Add Product to Cart   |
| <b>Preconditions:</b>    | Customer selects desired store   |
| <b>Post Conditions:</b>  | <ul style="list-style-type: none"><li>-An order is created (instance creation).</li><li>-Product was associated with the store (Association Formed)</li><li>-Attributes of the product are now initialized</li></ul> |

### Contract CO2: deliverOrder

---

|                          |  |
|--------------------------|--|
| <b>Operation:</b>        | deliverOrder()   |
| <b>Cross References:</b> | Use Cases: Received Order, Pick up and Deliver Order   |
| <b>Preconditions:</b>    | Customer adds desired products to cart & checks out  |
| <b>Post Conditions:</b>  | <ul style="list-style-type: none"><li>-An order is received.</li><li>-Order is then processed and picked up by delivery driver.</li><li>-Products are delivered to customer.</li></ul> |

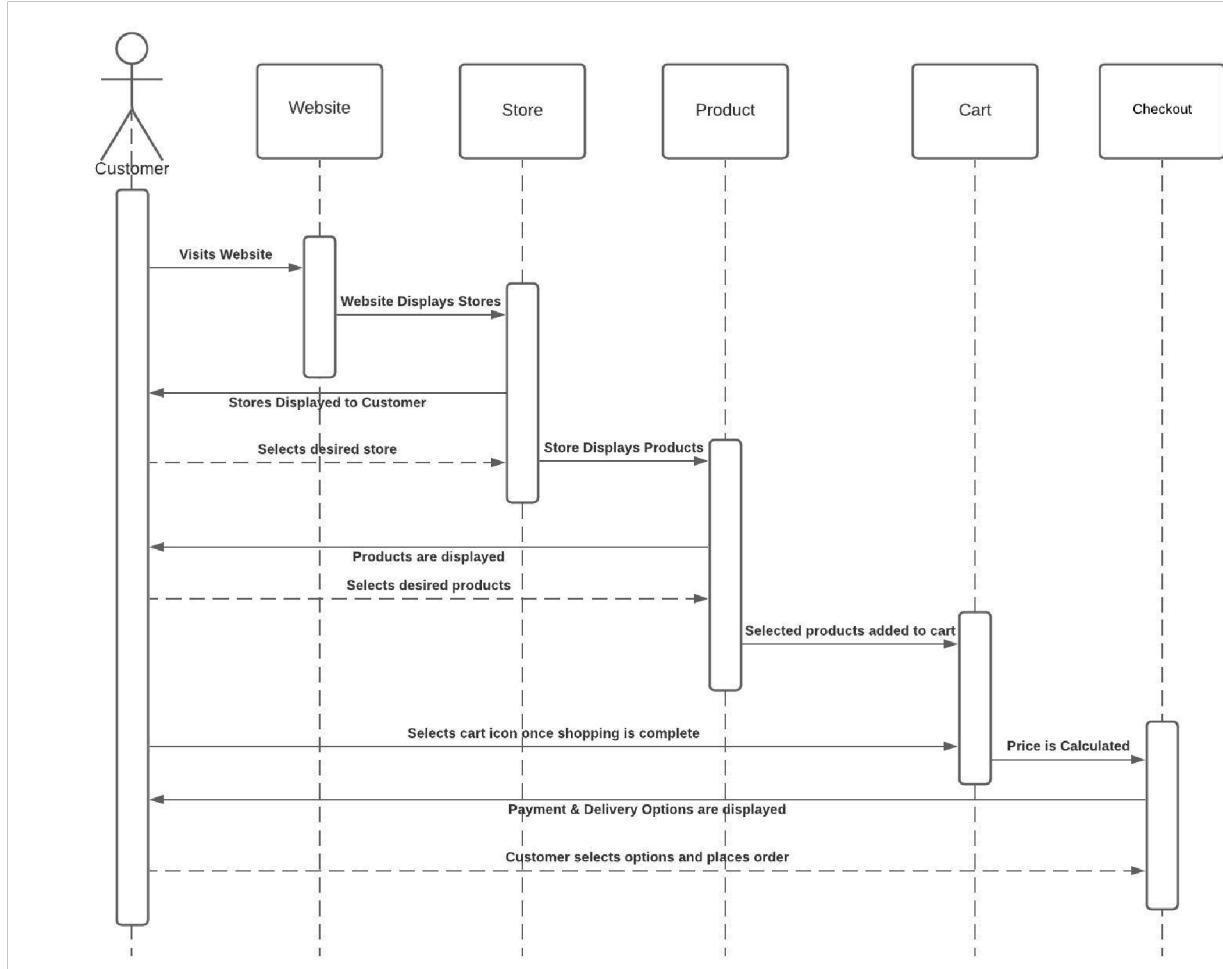
### Project Size Estimation

After evaluating the completed portions of Report 1, I have put together a breakdown of things that the project will include which will better determine the scope and size of the project

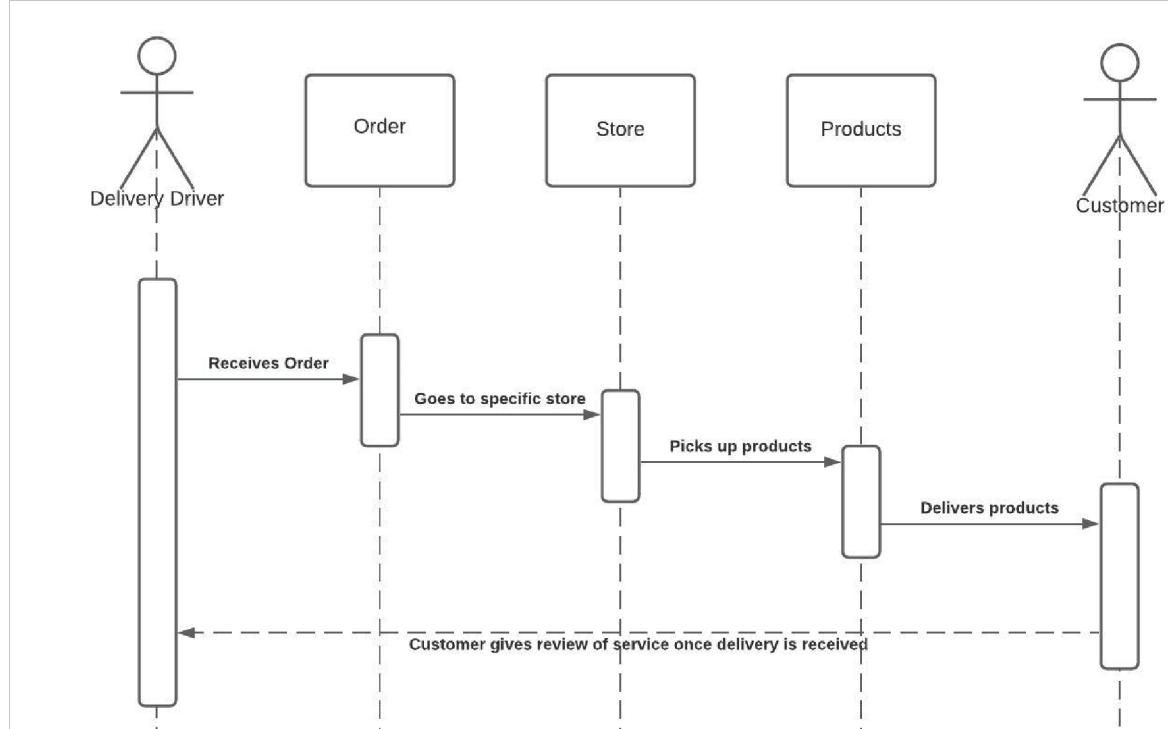
- There are no financial costs associated with this project since it is being completed for a class project.
- I am the only team member involved, so in terms of the project, the entire deliverables must be completed by me within the time period given for when the project is due.
- This project will produce a Grocery Delivery Service website which will contain several classes containing java, javascript, html, and css programming. I am estimating there to be 12-15 pages of source code once completed.

- As stated in the domain model, every page will have its own likeness except for the product page which will be displayed the same across all of the store pages.

## 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

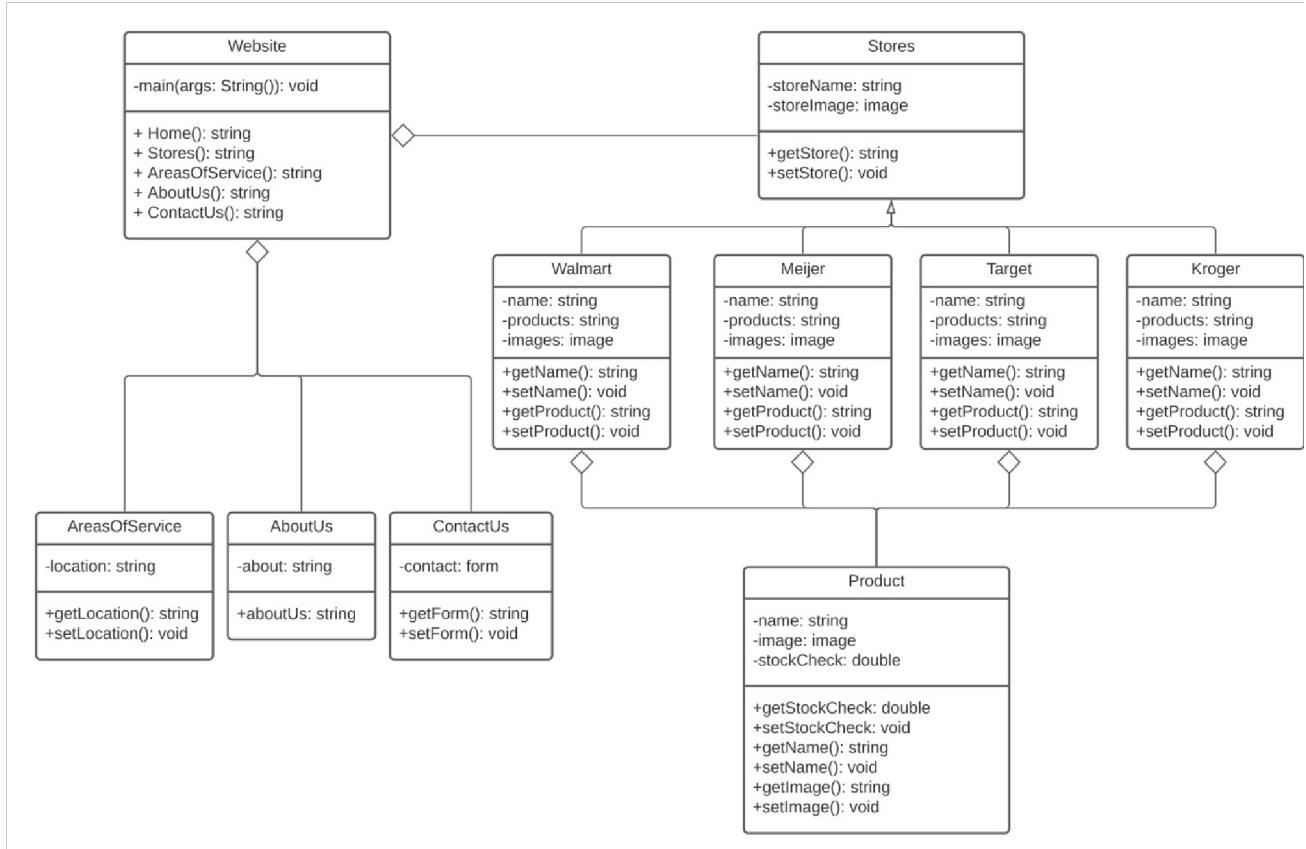
### Cases Being Tackled

- 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



- The home page of the website will display the 5 navigation tabs listed under the website class.
- The class store has 4 subclasses that can each be broken down by their name, products, and images.
- The products class can be broken down by its name, image, and the amount of it in stock.

## 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                           |                       |

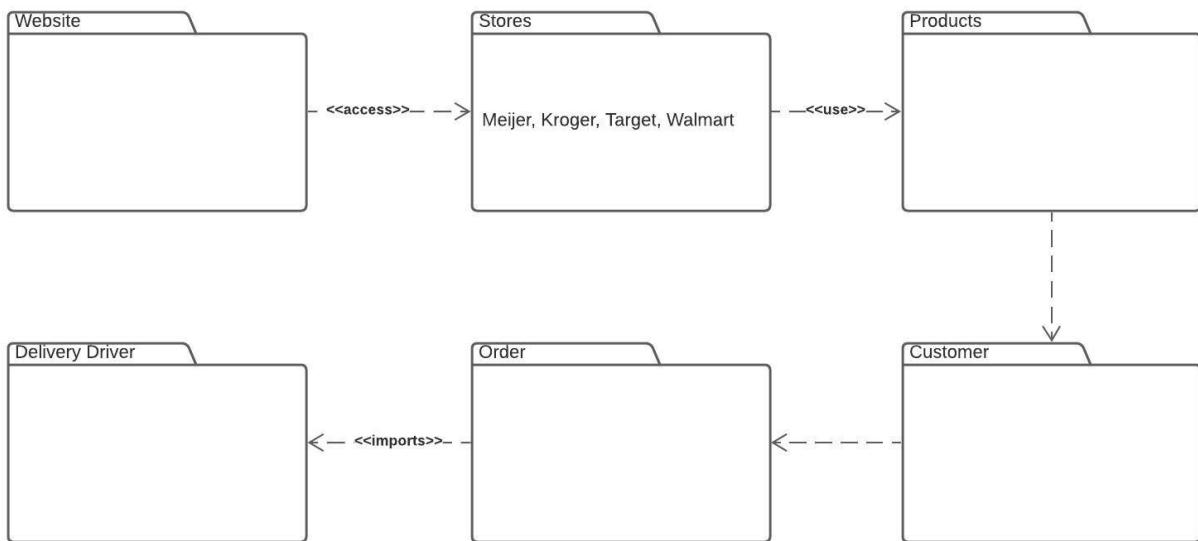
|          |    |   |   |    |   |
|----------|----|---|---|----|---|
| Total PW | 11 | 5 | 6 | 11 | 5 |
|----------|----|---|---|----|---|

## 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.



Below are the stores that you can shop from. Either click on one of the images or make a selection from the drop-down navigation tab labeled "Stores".



 © 2021 Grocery Delivery Service  
Indianapolis, IN 46217  
Phone: (219) 286-9418  
grocerydelivery@gmail.com

- The initial screen mock-ups that I created were made with powerpoint and in order for me to match them exactly it would have taken a lot of CSS coding, which I wanted to stray away from. I believe the look of the new interface gives it a cleaner and more updated look.

## **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.

## **History of Work**

- Began on the HTML & CSS coding once Report 1 was complete.
- Unfortunately, due to unforeseen circumstances, a lot of my goals fell short and my plan to have a finished product by midterm was incomplete.
- Once Demo 1 was submitted, I got to work on catching up on the coding of the website.
- Began combining the two reports and making revisions as needed.
- Completed website over the weekend of May 1<sup>st</sup>.

- Using the next several days to debug and make sure everything is running efficiently and according to my report.

## References

- 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

## SUMMARY OF CHANGES

- Added a brief description to my use case diagram.
- Provided descriptions for the operations and attributes in my class diagram.
- Added a second interaction diagram.
- Included the revised and updated user interface design with a description on why I decided to make the changes from my preliminary design.
- Updated the table of contents after combining the 2 reports.
- Removed repetitive sections.