**Spring 2020 CMPE 131**

**Subject: Part II**

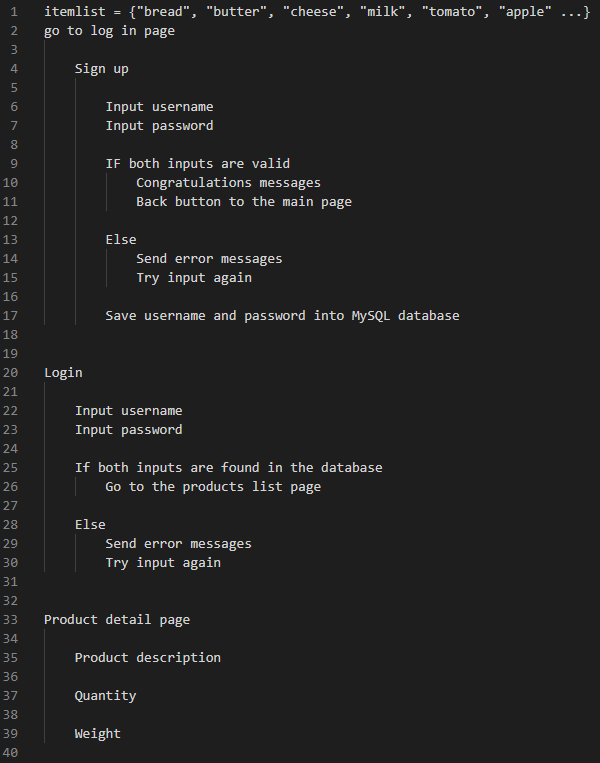
**Date: 3/24/2020**

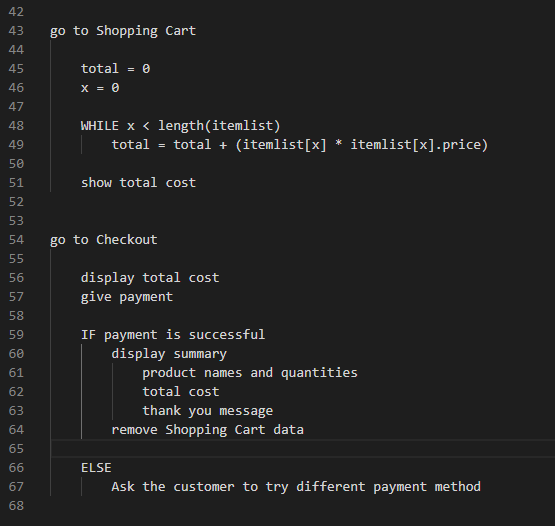
**Group 6:Lesley Becerra, Guido Prado, Makayla Walton, Jooyul Yoon, Snoop, Ben Wu, Yuchao Yan**

**Low-level Design Document:**

Component Design Breakdown from HLD

| Software Component | Component Information |
| --- | --- |
| Header bar | Header Bar is displayed throughout all pages that can be two versions. Store logo direct user to home pag and various buttons directing user to different page  **Not logged in Version**: contains Password and Usernames Entry box, login button  **Logged in Version**: replace Entry box with user’s name and other info, logout button |
| Product Box | Each product has their box containing their information such as Picture, Price, weight, the amount in stock, and button to add and subtract within the box.  -product description can pop-up when cursor hovers over it, or stationary.  -System has to track user’s click on a specific product in the back end for weight and price calculation |
| Product inventory, price and weight validating, and calculating component | -A back end tracking system, where it tracks every time the user clicks “add” or “Minus” button and if adding or subtracting product in cart based on inventory condition.  -Once the product being added, there are two back end calculations:   1. Price of all products being added 2. Weight of all products being added |
| Registration Page | A page full of entry boxes requiring users to enter their name, address, phone number, and their desired Username and password.  These entry boxes will connect to a Database which can be used to validate user checkout or log in. |
| Checkout Page | -Output calculated weight and price of all products being selected by the user. Indicating if products can be delivered  -prompt user to enter address and payment selection if user not logged on.  -prompt guess or member user to validate their shipping and payment info |
| Account Database | Using the account database is to conserve and manage the user’s account information: username, password, personal name, personal address and payment information will be protected. |
| Product Inventory Database | Using the Product inventory database to manage name, weight and amount of product. -When users purchase products, the amount of produce in the database will be reduced automatically.  -When the amount of product becomes zero, then the customer can not add any more product into the shopping cart. |

**Pseudocode Logic** 



**High-level Test Plan:**

Our testing strategy will be described through different scenarios in which the customer/user could encounter any type of bug or error in our website. This way we can be prepared for the different possible problems and have them fixed before launching the website. Each scenario will describe the situation, the outcome expectation and have a specific goal. Furthermore we also plan on keeping adding new scenarios as we keep analyzing our website.

**Home page:**

* Scenario 1: The button the user clicked on does not direct the user to the desired page when clicking on the desired button, making it impossible for users to navigate and do their shopping on the website.
* Scenario 2: User can’t log in to their account due to forgetting a password or username.
* Scenario 3: User gets their password or username wrong due to accidental Capitalization

**Register page:**

* Scenario 1: There is a chance where the system allows the user to register with the same password and username, thus we need a testing method within the Register system prompting the user message that their password or username is already used.

**Shopping page:**

* Scenario 1: When clicking on “add to cart” the amount on the inventory doesn’t change, making the quantity of the item in stock “infinite.” This way the user could keep adding the product to the cart even if the shop doesn't have enough in stock.
* Scenario 2: User is able to add product even the product is out of stock in inventory
* Scenario 3: The photo indicated the wrong product as listed.

**Cart Page:**

* Scenario 1: The cart displays an incorrect amount of products per order, not letting the customer edit those numbers in the page, requiring the user to delete their cart and start again.

**Checkout Page:**

* Scenario 1: weight is being calculated incorrectly. Users are offered free delivery when their cart contains less than 20 pound of product. Or not being offered for free shipping when the cart contains over 20 pound of product.
* Scenario 2: User enters payment information that is different from selected payment option
* Scenario 3: User’s shipping information is different from the one listed.

**Testing Strategy**

The scenarios listed above will be tested as the online ordering system is being created. These scenarios are predicted to cause problems for the user, and will be tested as each design component is made. If more issues than predicted are found, those problems will be documented in order to keep track of errors in the code. The goal of troubleshooting each component is to be able to pinpoint where each error arises for the user so that the final product is an easy to use system with little to no errors.

**All the functions of Online Food Store need to be tested**

| **Module Name** | **Applicable roles** | **Description** |
| --- | --- | --- |
| Login | Customer | Customer: A customer can use customers’ account to login Online Food Store |
| Register | Customer | Customer: A customer can register a customer account through this page |
| Add item button | Customer | Customer: A customer can add items to the shopping cart by click add item button |
| Shopping Cart | Customer | Customer: A customer can click the ‘Shopping cart’ button to view the item which they added. |
| Check Out | Customer | Customer: A customer can click ‘Check Out’ button to Check Out page |
| Account Database | Manager | Manager: A system where User’s information is stored and validated when logged in. |
| Inventory database | Manager | Manager: A system that requires constant updates for user to know what is and is not stocked |
| Manager Account | Manager | Manager: The admin account where the manager can monitor the database |