**Design Document for Project 2**

Grant Robertson – 20 November, 2019

**User Story 1:** “As a user, I want to add a new product or update a current product in the system.”

Use Case 1: Successfully adding a new product.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Manage Products Button from the main screen.  Main Screen: | 2. Manage Product Screen appears for user to enter information about their product.  Manage Product Screen: |
| 3. User inputs information and then selects the save product.  Manage Product Screen with Information: | 4. The Manage Product UI sends the new product to the server to be saved. On the server side, the new product is added to the database. The server succeeds, and sends back a confirmation message.  Confirmation Message: |
| 5. User clicks okay button, and then adds another product (return to step 3) or loads a current product, or exits the manage product window. | 6. User is finished managing products, and returns to Main Screen. |

Use Case 2: User attempts to add a product with a price not recognizable as a double.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Add New Product Button from the main screen.  Main Screen: | 2. Manage Product Screen appears for user to enter information about their product.  Manage Product Screen: |
| 3. User inputs information (entering a Product price that cannot be parsed into a double) and then selects the save product button.  Manage Product Screen with Information: | 4. Before product is sent to the server to be saved, the error is detected and the product is rejected. The user is notified.  Error Page: |
| 5. User clicks okay button, is returned to the Manage Product window, and then may modify the price and other product information, load an existing product, or exit the Manage Product Window. | 6. User exited Manage Product Window and returned to Main Screen. |

**User Story 2:** “As a user, I want to add a new customer or update a current customer in the system.”

Use Case 1: User successfully adds customer to the system.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Manage Customers Button from the main screen.  Main Screen: | 2. Manage Customer Screen appears for user to enter information about the new customer.  Manage Customer Screen: |
| 3. User inputs Information and then selects the add customer button.  Manage Customer Screen with Information: | 4. The Manage Customer UI sends the new customer to the server to be saved. On the server side, the new customer is added to the database. The server succeeds, and sends back a confirmation message.  Confirmation Page: |
| 5. User clicks okay button, and then adds another customer (return to step 3), loads an existing customer, or closes the add customer window by hitting the x. | 6. User is finished managing customers, and returns to the Main Screen. |

Use Case 2: User attempts to load a nonexistent customer.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Manage Customer Button from the main screen.  Main Screen: | 2. Manage Customer Screen appears for user to enter information about the new customer.  Manage Customer Screen: |
| 3. User inputs information a Customer ID that is not associated with a customer within the database.  Add Customer Screen with Information: | 4. The ID is sent to the server to retrieve the customer. No customer is found, so the server sends back an indication of failure. Then the system displays an error page indicating the given Customer associated with the ID does not exist.  Error Page: |
| 5. User clicks okay button, is returned to the Manage Customer window, and then may modify the Customer ID to an existing customer, create a customer with that ID, or press the x button and exit the window and return to the main screen. | 6. User is finished managing customers, and returns to the Main Screen. |

**User Story 3:** “As a user, I want to add a new purchase or update a current purchase in the system.”

Use Case 1: User successfully records purchase.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Manage Purchases Button from the main screen.  Main Screen: | 2. Manage Purchases Screen appears with current date and time for user to enter information about their product.  Manage Purchase Screen: |
| 3. User creates a unique Purchase ID, inputs the relevant Product and Customer IDs, and inputs the quantity of product purchased.  Manage Purchase Screen with Information: | 4. System automatically populates the Customer Name, Product Name, Individual Product Price, Cost, Tax, and Total Cost through messages to the server and subsequent queries to SQL.  Add Purchase Screen with populated information: |
| 5. User clicks save button. | 6. System displays success page.  Confirmation Page: |
| 9. User clicks okay button, and then adds another purchase (return to step 3), loads an existing one, or closes the manage purchase window by hitting the x button. | 10. Return User to the Main Screen. |

Use Case 2: User attempts to use an invalid Customer ID to complete purchase.

|  |  |
| --- | --- |
| **Actor** | **System** |
| 1. User Selects the Manage Purchases Button from the main screen.  Main Screen: | 2. Manage Purchase Screen appears with current date and time for user to enter information about their product.  Manage Purchase Screen: |
| 3. User creates a unique Purchase ID, and enters an invalid Customer ID.  Add Purchase Screen with Information: | 4. System automatically generates a warning of an invalid Customer ID.  Customer ID Warning |
| 5. If user attempts to continue without changing the Customer ID, they will finish adding information to the page. | 6. If attempting to add a purchase with a bad customer ID, an error message will be generated. No message will be sent to the server.  Error Page: |
| 7. User is returned to the add purchase screen, and can either edit the information to use a correct Customer ID (user then follows step 3 of Use Case 1), load another purchase, or click the x button and return to the main screen. | 10. Return User to the Main Screen. |