1. Use cases
   1. User Story: As a user, I want to change my password and/or full name. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Change Credentials | 1. Open Change Credentials Window |
| 1. Enter new password and or new name and choose confirm | 1. Display success message |

* 1. User Story: As an admin, I want to add a new user into the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Add User | 1. Open Add User Window |
| 1. Enter user information and choose add | 1. Display success message |

* 1. User Story: As an Admin, I want to update the type/class of a user into the system (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Change User Type | 1. Open Change User Type Window |
| 1. Enter username and new type choose change type | 1. Display success message |

* 1. User Story: As a Manager, I want to add a new product or update a current product in the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Manage Product | 1. Open Manage Product Window |
| 1. Input product ID and choose load | 1. Fill in remaining data |
| 1. Change one or more fields and choose Save | 1. Display Success Message |

* 1. User Story: As a manager, I want to view a summary report of sales from the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose View Sales | 1. Open View Sales Window |

* 1. User Story: As a cashier, I want to add a new customer or update a current customer in the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Manage Customer | 1. Open Manage Customer Window |
| 1. Input customer ID and choose load | 1. Fill in remaining data |
| 1. Change one or more fields and choose Save | 1. Display Success Message |

* 1. User Story: As a cashier, I want to add a new purchase or update a current purchase in the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Manage Purchase | 1. Open Manage Purchase Window |
| 1. Input purchase ID and choose load | 1. Fill in remaining data |
| 1. Change one or more fields and choose Save | 1. Display Success Message |

* 1. User Story: As a customer, I want to add a new purchase into the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Add Purchase | 1. Open Add Purchase Window |
| 1. Enter purchase information and choose add | 1. Display success message |

* 1. User Story: As a customer, I want to view my purchase history into the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose View Purchase History | 1. Open View Purchase History Window |

* 1. User Story:  As a customer, I want to search for product information in the system. (Common Use Case)

|  |  |
| --- | --- |
| Actor Action | System Response |
| 1. Choose Search Product | 1. Open Product Search Window |
| 1. Enter relevant information and choose search | 1. Display Search Results |

1. System Protocol
   1. Login
      1. Client Side
         1. Client inputs username and password into login screen
         2. Client selects ‘login’
         3. Controller processes username and password and sends message to server
         4. Controller receives message on success of operation
            1. If the user is successfully logged in, the controller navigates client to proper user screen
      2. Server Side
         1. Server receives and processes message
         2. Server query’s database for existence of user
            1. If object exists server asks for user’s password
         3. Server sends message to client side on success of operation
            1. If the password from the database matches the password from the client the operation is a success, otherwise it is a failure
   2. Load Data
      1. Client Side
         1. Client Navigates to particular UI
         2. Client inputs object’s ID
         3. Client Selects the ‘load’ option
         4. Controller processes ID and sends message to server
         5. Controller receives message on success of operation
            1. If the object is successfully retrieved from server, the controller outputs the object’s data to UI
      2. Server Side
         1. Server receives and processes message
         2. Server query’s database for object’s existence
            1. If object exists server asks for remainder of object’s data
         3. Server sends message to client side on success of operation
            1. Also sends object’s data to client side if the object was found
   3. Save Data
      1. Client Side
         1. Client Navigates to particular UI
         2. Client inputs object’s data
         3. Client Selects the ‘save’ option
         4. Controller processes data and sends message to server
         5. Controller receives message on success of operation
      2. Server Side
         1. Server receives and processes message
         2. Server query’s database for existence of object with same ID
            1. If said object exists server asks database to delete the object and replace it with a new version of the object
         3. Server sends message to client side on success of operation