

**PROJECT REPORT**

Warehouse Management

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | PF1126 |
| **Group:** | Vu Minh Long |
| **Instructor** | Nguyen Dinh Cuong |
| **Team members:** | StudentId – Vu Quang Minh  StudentId – Phan Canh Long Vu  StudentId – Nguyen Gia Long |

**Index**

[I. Project introduction 4](#_Toc173402610)

[**1.** **Proposed System** 4](#_Toc173402611)

[**2.** **The scope of the project to be applied** 4](#_Toc173402612)

[**3.** **System Name** 4](#_Toc173402613)

[**4.** **Deployment Environment** 4](#_Toc173402614)

[**5.** **Development Tools** 4](#_Toc173402615)

[**6.** **Customer Requirements (System features)** 4](#_Toc173402616)

[II. Analyze System Requirements 5](#_Toc173402617)

[**1.** **Use Case** 6](#_Toc173402618)

[**1.1** **Login Description** 6](#_Toc173402619)

[**1.2** **Check Login Description** 7](#_Toc173402620)

[**1.3** **Create Supplier Description** 8](#_Toc173402621)

[**1.4** **Update Supplier Description** 9](#_Toc173402622)

[**1.5** **Delete Supplier Description** 10](#_Toc173402623)

[**1.6** **View Supplier Description** 11](#_Toc173402624)

[**1.7** **Add Category Description** 12](#_Toc173402625)

[**1.8** **Update Category Description** 13](#_Toc173402626)

[**1.9** **Delete Category Description** 14](#_Toc173402627)

[**1.10** **View Category Description** 15](#_Toc173402628)

[**1.11** **Add Product Description** 16](#_Toc173402629)

[**1.12** **Update Product Description** 17](#_Toc173402630)

[**1.13** **Delete Product Description** 18](#_Toc173402631)

[**1.14** **View Product Description** 19](#_Toc173402632)

[**1.15** **Add Shipment Description** 20](#_Toc173402633)

[**1.16** **Update Shipment Description** 21](#_Toc173402634)

[**1.17** **Delete Shipment Description** 22](#_Toc173402635)

[**1.18** **View Shipment Description** 23](#_Toc173402636)

[**1.19** **Add Warehouse Description** 24](#_Toc173402637)

[**1.20** **Update Warehouse Description** 25](#_Toc173402638)

[**1.21** **Delete Warehouse** 26](#_Toc173402639)

[**1.22** **View warehouse** 27](#_Toc173402640)

[**1.23** **Add Inventory Audit Description** 28](#_Toc173402641)

[**1.24** **Update Inventory Audit Description** 29](#_Toc173402642)

[**1.25** **Delete Inventory Audit Description** 30](#_Toc173402643)

[**1.26** **View Inventory Audit Description** 31](#_Toc173402644)

[**1.27** **Add User Description** 32](#_Toc173402645)

[**1.28** **Update User Description** 33](#_Toc173402646)

[**1.29** **Delete User Description** 34](#_Toc173402647)

[**1.30** **View User Description** 35](#_Toc173402648)

[**1.31** **Change Password Description** 36](#_Toc173402649)

[**1.32** **View Report Description** 37](#_Toc173402650)

[**2.** **Activity Diagram**: 39](#_Toc173402651)

[**2.1** **Login activity** 39](#_Toc173402652)

[**2.2** **Change password activity** 39](#_Toc173402653)

[**2.3** **View account list activity** 39](#_Toc173402654)

[**2.4** **View supplier list activity** 39](#_Toc173402655)

[**2.5** **View category list activity** 39](#_Toc173402656)

[**2.6** **View product list activity** 39](#_Toc173402657)

[**2.7** **View warehouse stock activity** 39](#_Toc173402658)

[**2.8** **View shipment list activity** 39](#_Toc173402659)

[**2.9** **View inventory audit activity** 40](#_Toc173402660)

[**2.10** **View warehouse list activity** 40](#_Toc173402661)

[**2.11** **Create account activity** 40](#_Toc173402662)

[**2.12** **Create supplier activity** 40](#_Toc173402663)

[**2.13** **Create category activity** 40](#_Toc173402664)

[**2.14** **Create product activity** 40](#_Toc173402665)

[**2.15** **Create inbound shipment activity** 40](#_Toc173402666)

[**2.16** **Create outbound shipment activity** 40](#_Toc173402667)

[**2.17** **Create transfer shipment** 40](#_Toc173402668)

[**2.18** **Create inventory audit activity** 40](#_Toc173402669)

[**2.19** **Create warehouse activity** 40](#_Toc173402670)

[**2.20** **Edit user activity** 40](#_Toc173402671)

[**2.21** **Edit category activity** 40](#_Toc173402672)

[**2.22** **Edit warehouse activity** 40](#_Toc173402673)

[**2.23** **Edit supplier activity** 40](#_Toc173402674)

[**2.24** **Edit shipment activity** 40](#_Toc173402675)

[**2.25** **Edit product activity** 41](#_Toc173402676)

[**2.26** **Edit inventory audit activity** 41](#_Toc173402677)

[**2.27** **Delete user activity** 41](#_Toc173402678)

[**2.28** **Delete product activity** 41](#_Toc173402679)

[**2.29** **Delete category activity** 41](#_Toc173402680)

[**2.30** **Delete warehouse activity** 41](#_Toc173402681)

[**2.31** **Delete shipment activity** 41](#_Toc173402682)

[**2.32** **Delete inventory audit activity** 41](#_Toc173402683)

[**2.33** **Delete supplier activity** 41](#_Toc173402684)

[**2.34** **View report activity** 41](#_Toc173402685)

[III. Design Details 42](#_Toc173402686)

[1. UI Design 42](#_Toc173402687)

[1.1 Login Interface 42](#_Toc173402688)

[1.2 Home Menu Interface 42](#_Toc173402689)

[1.3 User List Interface 43](#_Toc173402690)

[1.4 Change Password Interface 43](#_Toc173402691)

[1.5 Category List Interface 44](#_Toc173402692)

[1.6 Add Category Interface 44](#_Toc173402693)

[1.7 Product List Interface 45](#_Toc173402694)

[1.8 Add-Edit Product Interface 45](#_Toc173402695)

[1.9 Warehouse Stock Interface 46](#_Toc173402696)

[1.10 Supplier List Interface 46](#_Toc173402697)

[1.11 Add Supplier Interface 47](#_Toc173402698)

[1.12 Warehouse List Interface 47](#_Toc173402699)

[1.13 Add Warehouse Interface 48](#_Toc173402700)

[1.14 Shipment List Interface 48](#_Toc173402701)

[1.15 Add-Edit Stock Transfer Interface 49](#_Toc173402702)

[1.16 Add-Edit Outbound Shipment Interface 49](#_Toc173402703)

[1.17 Add-Edit Inbound Shipment Interface 50](#_Toc173402704)

[1.18 Inventory Audit List Interface 50](#_Toc173402705)

[1.19 Add-Edit Inventory Audit Interface 51](#_Toc173402706)

[1.20 Report Export Interface 51](#_Toc173402707)

[2. Code Design (Class Diagram) 53](#_Toc173402708)

[3. Sequence Diagram 55](#_Toc173402709)

[3.1 Login sequence diagram 55](#_Toc173402710)

[3.2 Change password sequence diagram 56](#_Toc173402711)

[3.3 View account sequence diagram 56](#_Toc173402712)

[3.4 56](#_Toc173402713)

[4. Database Design 57](#_Toc173402714)

[IV. Test 66](#_Toc173402715)

[1. Login test 66](#_Toc173402716)

[2. Change password test 71](#_Toc173402717)

[3. Create new user test 76](#_Toc173402718)

[4. Edit user information test 80](#_Toc173402719)

[5. Add product test 83](#_Toc173402720)

[6. Edit product test 87](#_Toc173402721)

[7. Create new category test 90](#_Toc173402722)

[8. Edit category test 93](#_Toc173402723)

[9. Create new warehouse test 96](#_Toc173402724)

[10. Edit warehouse test 99](#_Toc173402725)

[11. Create new shipment test 102](#_Toc173402726)

[12. Edit shipment test 104](#_Toc173402727)

[13. Create new inventory audit test 107](#_Toc173402728)

[14. Edit inventory audit test 109](#_Toc173402729)

[15. Edit inventory audit test 109](#_Toc173402730)

[16. Create new suppliers test 112](#_Toc173402731)

[17. Edit supplier test 115](#_Toc173402732)

[V. Task Assign (to each team member) 119](#_Toc173402733)

[VI. Installation Instructions 120](#_Toc173402734)

[Appendix 121](#_Toc173402735)

[Document format 122](#_Toc173402736)

# **Project introduction**

Software for managing export, import, and finished product inventory with main functions: user management, supplier management, product management, warehouse import/export.

1. **Proposed System**

* User management.
* Product Management.
* Supplier management.
* Warehouse management.
* Inventory management.
* Manage warehouse release and rotation.
* Export report

1. **The scope of the project to be applied**

* **Target Users:** Small and medium business companies.

1. **System Name**

* Warehouse Management System.

1. **Deployment Environment**

* **Supported Platforms:** Window, Mac (Linux?).

1. **Development Tools**

* **Database Management:** MySQL Workbench 8.0.34
* **Diagramming Tool:** Draw.io
* **Code Editor:** Visual Studio Code

1. **Customer Requirements (System features)**

* User management
* Supplier management
* Warehouse management
* Product management
* Stock management
* Inbound/Outbound/Transfer management
* Inventory Audit

# **Analyze System Requirements**

Warehouse management software integrates many important functions to help optimize goods and warehouse management. The goods management function allows adding, editing, deleting goods and checking inventory details. Warehouse management supports adding, editing, and deleting warehouses, along with import and export management functions including management of import and export activities and periodic inventory audit.

Inventory management helps perform periodic or unscheduled inventory counts and inventory adjustments when necessary. Partner management includes information about suppliers. The user management and authorization system allows account management and authorization to use system functions. Finally, the reporting and statistics function provides detailed reports on inventory status, import and export of goods to support business decision making.

1. **Use Case**
   1. **Login Description**

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | UC01 |
| Description | Used to access the program and determine usage rights depending on account level. |
| Actor | User |
| Organizational Benefits | Security: Ensure that only authorized users can use system functions  Customization: Allows customization of each user's permissions. |
| Triggers | The user enters the name and password then selects login on the software interface |
| Preconditions | The user already has an account in the system  The system is active and can access the database |
| Postconditions | If successful, the user will be redirected to the system home screen with authenticated access.  If it fails, the user will receive an error message and can try entering again. |
| Main Course | 1. User enters username and password.  2. The system authenticates login information.  3. If the information is valid, the system logs the user in and sends them to the main screen.  4. Users now have access to functions and data based on their account. |
| Alternate Courses | AC1: Wrong login information: User entered incorrect login name and password  1. Display error message  2. Display the login screen again  3. Request re-entry |
| Exceptions | System error: Connection error or database problem prevents successful login.   1. 1. Display “Database or connection error”. |

* 1. **Check Login Description**

|  |  |
| --- | --- |
| Use Case Name | Check Login Information |
| Use Case ID | UC06 |
| Description | This use case describes the process of checking the validity of user login information into the system. |
| Actor | System |
| Organizational Benefits | Ensure only users with valid login credentials are allowed to access the system, protecting data security. |
| Triggers | The system automatically checks login information when users access the system. |
| Preconditions | The user provides login information (username and password). |
| Postconditions | The system confirms the validity of the login information and allows users to access the system or deny access. |
| Main Course | 1. User provides username and password to log in to the system.  2. The system checks whether the username and password are correct.  3. If the login information is valid, the system allows the user to access the system and generate tokens.  4. If the login information is invalid, the system denies access and displays an error message to the user. |
| Alternate Courses | AC1: If the password is incorrect, the system displays an error message and requires the user to re-enter. |
| Exceptions | EX1: If the username does not exist in the database, the system displays an error message and asks the user to recheck the login information.  EX2: If the system encounters an error while processing login information, the system displays an error message and asks the user to try again. |

* 1. **Create Supplier Description**

|  |  |
| --- | --- |
| Use Case Name | Create supplier |
| Use Case ID | UC02 |
| Description | This use case describes the process of adding a new supplier to the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Helps manage and store supplier information. |
| Triggers | The user needs to add a new provider. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | The new supplier is added to the supplier list in the system. |
| Main Course | 1. User selects the option to add a new supplier.  2. The system displays the supplier information form.  3. User enters supplier details (name, address, contact information, etc.).  4. User submits form.  5. The system validates the entered data.  6. The system saves supplier information into the database.  7. The system confirms the successful addition of a new supplier to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the input data is invalid (for example, contact information is not in the correct format), the system displays an error message and asks the user to correct the information. |

* 1. **Update Supplier Description**

|  |  |
| --- | --- |
| Use Case Name | Update supplier |
| Use Case ID | UC02 |
| Description | This use case describes the process of editing an existing supplier's information in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure supplier information is updated accurately, helping to maintain consistent and reliable supplier data. |
| Triggers | User needs to edit existing supplier information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Supplier information is updated in the system. |
| Main Course | 1. The user selects the supplier's information that needs to be edited  2. The system displays the selected information if it is allowed to be edited  3. User edits information  4. The user confirms the information that needs to be corrected  5. The system validates the corrected data  6. The system updates supplier information in the database.  7. The system confirms the successful vendor update to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, contact information is not in the correct format), the system displays an error message and asks the user to correct the information.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Supplier Description**

|  |  |
| --- | --- |
| Use Case Name | Delete supplier |
| Use Case ID | UC04 |
| Description | This use case describes the process of deleting a supplier's information from the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Make sure the supplier list in the system is always accurate and does not contain unnecessary information. |
| Triggers | The user decides to delete the existing provider. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Supplier information is deleted in the system. |
| Main Course | 1. User selects the supplier he wants to delete.  2. User confirms deletion of provider.  3. The system checks relevant conditions when deleting a supplier.  4. The system deletes supplier information from the database.  5. The system confirms the successful deletion of the supplier to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the supplier is already linked to the order, the system displays an error message and does not allow deletion of the supplier.  EX2: If the system encounters an error while deleting data from the database, the system displays an error message and asks the user to try again. |

* 1. **View Supplier Description**

|  |  |
| --- | --- |
| Use Case Name | View Supplier |
| Use Case ID | UC05 |
| Description | This use case describes the process of viewing detailed information of a supplier in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed information about suppliers, helping them make decisions related to procurement and supplier management effectively. |
| Triggers | Users need to view detailed information of a supplier. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | The supplier's details are displayed to the user. |
| Main Course | 1. User selects the option to view the supplier list.  2. The system displays the list of suppliers.  3. Users select or search for suppliers to view.  4. Users see detailed information and can take other actions (like editing or deleting). |
| Alternate Courses | AC1: If users do not find the desired supplier, they can search or filter the supplier list by various criteria (name, supplier code, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add Category Description**

|  |  |
| --- | --- |
| Use Case Name | Add category |
| Use Case ID | UC06 |
| Description | This use case describes the process of adding a new category to the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor |
| Organizational Benefits | Ensure catalog information is recorded and maintained accurately, helping to manage product catalogs effectively. |
| Triggers | User needs to add a new category. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New categories added to the system. |
| Main Course | 1. User selects the option to add a new category.  2. The system displays sample directory information.  3. User enters category details (name, description)  4. User confirms.  5. The system checks related conditions when adding categories.  6. The system saves directory information to the database.  7. The system confirms the successful addition of the new category to the user. |
| Alternate Courses | AC1: If the user chooses back, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the input data is invalid (for example, the category name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Update Category Description**

|  |  |
| --- | --- |
| Use Case Name | Update category |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing information about an existing category in the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Ensure catalog information is updated accurately, helping to maintain consistent and reliable catalog data. |
| Triggers | User needs to edit existing category information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Category information is updated in the system. |
| Main Course | 1. The user selects the category he wants to edit.  2. The system displays information about the selected category.  3. User edits necessary details.  4. User confirms.  5. The system checks the corrected data.  6. The system updates directory information in the database.  7. The system confirms the successful updating of the catalog to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, the category name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Category Description**

|  |  |
| --- | --- |
| Use Case Name | Delete category |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting a category's information from the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Make sure the list of categories in the system is always accurate and does not contain unnecessary information. |
| Triggers | User needs to delete an existing category. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Category information is deleted from the system. |
| Main Course | 1. The user selects the category they want to delete.  2. The system requests confirmation of deletion.  3. User confirms deletion of category.  4. The system checks the constraints related to the category.  5. The system deletes directory information from the database.  6. The system confirms the successful deletion of the directory for the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without deleting data. |
| Exceptions | EX1: If the category is linked to other products or data, the system displays an error message and does not allow deleting the category.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **View Category Description**

|  |  |
| --- | --- |
| Use Case Name | View category |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of a category in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed information about categories, helping them manage products and categories effectively. |
| Triggers | Users need to view detailed information of a category. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Detailed information of the category is displayed to the user. |
| Main Course | 1. User selects the option to view the category list.  2. The system displays the category list.  3. User selects or searches for the category to view.  4. The system displays detailed information of the selected category (name, description, etc.).  5. Users see detailed information and can take other actions (like edit or delete). |
| Alternate Courses | AC1: If users do not find the desired category, they can search or filter the category list by various criteria (name, category code, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add Product Description**

|  |  |
| --- | --- |
| Use Case Name | Add Product |
| Use Case ID | UC06 |
| Description | This use case describes the process of adding a new product to the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure product information is recorded and maintained accurately, helping to manage products effectively. |
| Triggers | The user needs to add a new product. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New products added to the system. |
| Main Course | 1. User selects the option to add a new product.  2. The system displays product information samples.  3. User enters product details (name, description, price, category, etc.).  4. User confirms.  5. The system validates the entered data.  6. The system saves product information into the database.  7. The system confirms the successful addition of the new product to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the input data is invalid (for example, the product name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Update Product Description**

|  |  |
| --- | --- |
| Use Case Name | Update Product |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing information about an existing product in the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Ensure product information is updated accurately, helping to maintain consistent and reliable product data. |
| Triggers | The user decides to edit the existing product information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Product information is updated in the system. |
| Main Course | 1. User selects the product he wants to edit.  2. The system displays the current product information form.  3. User edits necessary details.  4. User confirms.  5. The system validates the corrected data.  6. The system updates product information in the database.  7. The system confirms the successful product update to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, the product name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Product Description**

|  |  |
| --- | --- |
| Use Case Name | Delete Product |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting a product's information from the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Make sure the product list in the system is always accurate and does not contain unnecessary information. |
| Triggers | User needs to delete an existing product. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Product information is deleted in the system. |
| Main Course | 1. User selects the product he wants to delete.  2. The system requests confirmation of deletion.  3. User confirms product deletion.  4. The system checks constraints related to the product.  5. The system deletes product information from the database.  6. The system confirms the successful deletion of the product to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without deleting data. |
| Exceptions | EX1: If the product has been used in any order, the system displays an error message and does not allow deletion of the product.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **View Product Description**

|  |  |
| --- | --- |
| Use Case Name | View Product |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of a product in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed product information, helping them manage products and categories effectively. |
| Triggers | Users need to see detailed information of a product. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Detailed product information is displayed to the user. |
| Main Course | 1. User selects option to view product list.  2. The system displays the product list.  3. User selects the product to view.  4. The system displays detailed information of the selected product (name, description, price, category, etc.).  5. Users see detailed information and can take other actions (like edit or delete). |
| Alternate Courses | AC1: If users do not find the desired product, they can search or filter the product list by various criteria (name, product code, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add Shipment Description**

|  |  |
| --- | --- |
| Use Case Name | Add shipment |
| Use Case ID | UC06 |
| Description | This use case describes the process of adding a new order to the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure order information is recorded and maintained accurately, helping to manage shipments effectively. |
| Triggers | The user decides to add a new order. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New order added to the system. |
| Main Course | 1. User selects the option to add a new order.  2. The system displays the order information form.  3. User enters order details (order code, description, date, etc.).  4. User confirms.  5. The system validates the entered data.  6. The system saves shipment information into the database.  7. The system confirms the successful addition of a new shipment to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the entered data is invalid (for example, the order code already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Update Shipment Description**

|  |  |
| --- | --- |
| Use Case Name | Update shipment |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing information about an existing order in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure order information is updated accurately, helping maintain order data |
| Triggers | The user decides to correct the incomplete order information. |
| Preconditions | The user must log in to the system with appropriate permissions.  The order is in the system and in processing status. |
| Postconditions | Order information is updated in the system. |
| Main Course | 1. The user selects the order he wants to edit.  2. The system displays current order information.  3. User edits necessary details.  4. User confirms.  5. The system validates the corrected data.  6. The system updates order information in the database.  7. The system confirms the successful order update for the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, the order code already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Shipment Description**

|  |  |
| --- | --- |
| Use Case Name | Delete shipment |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting an order from the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Ensure the shipment list in the system is always accurate and does not contain unnecessary information. |
| Triggers | User needs to delete an existing order. |
| Preconditions | The user must log in to the system with appropriate permissions.  The order is in the system and in processing status. |
| Postconditions | The order is deleted from the system. |
| Main Course | 1. User selects the order he wants to delete.  2. The system requests confirmation of deletion.  3. User confirms deletion of order.  4. The system checks the constraints related to the order.  5. The system deletes order information from the database.  6. The system confirms the successful order deletion for the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without deleting data. |
| Exceptions | EX1: If the order has been completed, the system displays an error message and does not allow deleting the shipment.  EX2: If the system encounters an error while saving data to the database, the system displays an error message and asks the user to try again later. |

* 1. **View Shipment Description**

|  |  |
| --- | --- |
| Use Case Name | View shipment |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of an order in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed information about orders, helping them manage orders effectively. |
| Triggers | Users need to view detailed information of an order. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Detailed information of the order is displayed to the user. |
| Main Course | 1. User selects the option to view the order list.  2. The system displays the order list.  3. User selects the order to view.  4. The system displays detailed information of the selected order (code, description, date, etc.).  5. Users see detailed information and can take other actions (like edit or delete). |
| Alternate Courses | AC1: If users cannot find the desired order, they can search or filter the shipment list by different criteria (code, date, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add Warehouse Description**

|  |  |
| --- | --- |
| Use Case Name | Add warehouse |
| Use Case ID | UC06 |
| Description | This use case describes the process of adding a new warehouse to the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure warehouse information is recorded and maintained accurately, helping to manage warehouse effectively. |
| Triggers | User needs to add a new warehouse. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New warehouse added to the system. |
| Main Course | 1. User selects the option to add a new warehouse.  2. The system displays sample warehouse information.  3. User enters warehouse details (warehouse name, address, description, etc.).  4. User confirms.  5. The system validates the entered data.  6. The system saves warehouse information into the database.  7. The system confirms the successful addition of new warehouse to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the input data is invalid (for example, the warehouse name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error when saving data to the database, the system displays an error message and asks the user to try again. |

* 1. **Update Warehouse Description**

|  |  |
| --- | --- |
| Use Case Name | Update warehouse |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing information about an existing warehouse in the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Ensure warehouse information is updated accurately, helping to maintain consistent and reliable warehouse data. |
| Triggers | User needs to edit existing warehouse information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Warehouse information is updated in the system. |
| Main Course | 1. User selects the warehouse he wants to edit.  2. The system displays selected warehouse information.  3. User edits necessary details.  4. User confirms.  5. The system validates the corrected data.  6. The system updates warehouse information in the database.  7. The system confirms the successful inventory update to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1:If the modified data is invalid (for example, the warehouse name already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Warehouse**

|  |  |
| --- | --- |
| Use Case Name | Delete warehouse |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting warehouse information from the system. |
| Actor | Admin, Director, Warehouse Manager. |
| Organizational Benefits | Make sure the inventory list in the system is always accurate and does not contain unnecessary information. |
| Triggers | The user needs to delete an existing inventory. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Warehouse information is deleted from the system. |
| Main Course | 1. User selects the warehouse he wants to delete.  2. The system requests confirmation of deletion.  3. User confirms deletion of inventory.  4. The system checks warehouse-related conditions.  5. The system deletes warehouse information from the database.  6. The system confirms the successful deletion of inventory to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without deleting data. |
| Exceptions | EX1: If the warehouse has products, the system displays an error message and does not allow deleting the warehouse.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again later. |

* 1. **View warehouse**

|  |  |
| --- | --- |
| Use Case Name | View warehouse |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of a warehouse in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed information about the warehouse, helping them manage the warehouse effectively. |
| Triggers | The user needs to view detailed information about a warehouse. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Detailed information of the warehouse is displayed to the user. |
| Main Course | 1. User selects the option to view the inventory list.  2. The system displays a list of warehouses with detailed information about the warehouse (warehouse name, address, description, etc.).  3. Users see detailed information and can take other actions (like editing or deleting). |
| Alternate Courses | AC1: If users cannot find the desired warehouse, they can search or filter the warehouse list by various criteria (warehouse name, address, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add Inventory Audit Description**

|  |  |
| --- | --- |
| Use Case Name | Add inventory audit |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting warehouse information from the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor. |
| Organizational Benefits | Ensure inventory information is recorded and maintained accurately, helping to manage inventory effectively. |
| Triggers | The user needs to add a new inventory report. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New inventory report added to the system. |
| Main Course | 1. User selects the option to add a new inventory report.  2. The system displays the inventory report information form.  3. User enters inventory report details (report code, description, inventory date, etc.).  4. User confirms.  5. The system validates the entered data.  6. The system saves inventory report information to the database.  7. The system confirms the successful addition of the new inventory report to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the input data is invalid (for example, a report code already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again. |

* 1. **Update Inventory Audit Description**

|  |  |
| --- | --- |
| Use Case Name | Update inventory audit |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing information on an existing inventory report in the system. |
| Actor | Admin, Director, Warehouse Manager |
| Organizational Benefits | Ensure inventory information is updated accurately, helping to maintain consistent and reliable inventory data. |
| Triggers | The user needs to edit the existing inventory report information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Inventory report information is updated in the system. |
| Main Course | 1. The user selects the inventory report he wants to edit.  2. The system displays the selected inventory report information form.  3. User edits necessary details.  4. User confirms.  5. The system validates the corrected data.  6. The system updates inventory report information in the database.  7. The system confirms the successful updating of the inventory report to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, a report code already exists), the system displays an error message and requests the user to correct the information.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again. |

* 1. **Delete Inventory Audit Description**

|  |  |
| --- | --- |
| Use Case Name | Delete inventory audit |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting information from an inventory report from the system. |
| Actor | Admin, Director, Warehouse Manager |
| Organizational Benefits | Make sure the list of inventory reports in the system is always accurate and does not contain unnecessary information. |
| Triggers | The user needs to delete an existing inventory report. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | Inventory report information is deleted from the system. |
| Main Course | 1. The user selects the inventory report he wants to delete.  2. The system requests confirmation of deletion.  3. User confirms deletion of inventory report.  4. The system checks constraints related to inventory reports.  5. The system deletes inventory report information from the database.  6. The system confirms the successful deletion of the inventory report to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without deleting data. |
| Exceptions | EX1: If the inventory report is in completed status, the system displays an error message and does not allow deletion of the inventory report.  EX2: If the system encounters an error while updating data to the database, the system displays an error message and asks the user to try again. |

* 1. **View Inventory Audit Description**

|  |  |
| --- | --- |
| Use Case Name | View inventory audit |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of an inventory report in the system. |
| Actor | Admin, Director, Warehouse Manager |
| Organizational Benefits | Allows users to view detailed information about inventory reports, helping them manage inventory effectively. |
| Triggers | Users need to view detailed information of an inventory report. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | The detailed information of the inventory report is displayed to the user. |
| Main Course | 1. User selects the option to view the list of inventory reports.  2. The system displays a list of inventory reports.  3. The user selects the inventory report to view.  4. The system displays detailed information of the selected inventory report (report code, description, inventory date, etc.).  5. Users see detailed information and can take other actions (like edit or delete). |
| Alternate Courses | AC1: If users do not find the desired inventory report, they can search or filter the list of inventory reports by various criteria (report code, inventory date, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Add User Description**

|  |  |
| --- | --- |
| Use Case Name | Add user |
| Use Case ID | UC06 |
| Description | This use case describes the process of adding a new user to the system. |
| Actor | Admin. |
| Organizational Benefits | Ensure user information is recorded and maintained accurately, helping to manage users effectively. |
| Triggers | The user needs to add a new user. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | New user added to the system. |
| Main Course | 1. User selects the option to add a new user.  2. The system displays the user information form.  3. User enters user details (name, email, permissions, password, etc.).  4. User confirms.  5. The system validates the entered data.  6. The system saves user information into the database.  7. The system confirms the successful addition of the new user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the entered data is invalid (for example, the email already exists), the system displays an error message and requests to correct the information.  EX2: If the system encounters an error while saving data to the database, the system displays an error message and asks to try again. |

* 1. **Update User Description**

|  |  |
| --- | --- |
| Use Case Name | Update user |
| Use Case ID | UC06 |
| Description | This use case describes the process of editing an existing user's information in the system. |
| Actor | Admin |
| Organizational Benefits | Ensure user information is updated accurately, helping to maintain consistent and reliable user data. |
| Triggers | The user needs to edit existing user information. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | User information is updated in the system. |
| Main Course | 1. User selects the user he wants to edit.  2. The system displays the selected user information form.  3. User edits necessary details.  4. User confirms.  5. The system validates the corrected data.  6. The system updates user information in the database.  7. The system confirms the successful user update to the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the modified data is invalid (for example, the email already exists), the system displays an error message and requires the user to correct the information.  EX2: If the system encounters an error while saving data to the database, the system displays an error message and asks the administrator to try again. |

* 1. **Delete User Description**

|  |  |
| --- | --- |
| Use Case Name | Delete user |
| Use Case ID | UC06 |
| Description | This use case describes the process of deleting a user's information from the system. |
| Actor | Admin |
| Organizational Benefits | Make sure the list of users in the system is always accurate and does not contain unnecessary information. |
| Triggers | The user decides to delete an existing user. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | User information is deleted from the system. |
| Main Course | 1. User selects the user he wants to delete.  2. The system requests confirmation of deletion.  3. User confirms user deletion.  4. The system checks the constraints related to the user.  5. The system deletes user information from the database.  6. The system confirms the successful user deletion for the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the user is already associated with critical activities or data, the system displays an error message and does not allow deletion of the user.  EX2: If the system encounters an error while saving data to the database, the system displays an error message and asks the administrator to try again. |

* 1. **View User Description**

|  |  |
| --- | --- |
| Use Case Name | View User |
| Use Case ID | UC06 |
| Description | This use case describes the process of viewing detailed information of a user in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Allows users to view detailed information about other users, helping them manage and track users effectively. |
| Triggers | The user decides to view a user's details. |
| Preconditions | The user must log in to the system with appropriate permissions. |
| Postconditions | User details are displayed to other users. |
| Main Course | 1. User selects the option to view the list of users.  2. The system displays a list of detailed information of users (name, email, role, etc.).  3. The user sees detailed information and can take other actions (like editing or deleting if permission is available). |
| Alternate Courses | AC1: If the user does not find the desired user, they can search or filter the user list by various criteria (name, email, role, etc.). |
| Exceptions | EX1: If the system encounters an error while retrieving data from the database, the system displays an error message and asks the user to try again. |

* 1. **Change Password Description**

|  |  |
| --- | --- |
| Use Case Name | Change password |
| Use Case ID | UC06 |
| Description | This use case describes the process of users changing their passwords in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Ensures user account security and allows users to manage their own login information. |
| Triggers | Users need to change their password. |
| Preconditions | User must log in to the system. |
| Postconditions | The new password is saved in the system and the user uses the new password the next time he logs in. |
| Main Course | 1. User selects the option to change the password.  2. The system displays the password change form.  3. The user enters the current password, new password, and confirms the new password.  4. User confirms.  5. Current password authentication system.  6. The system verifies the validity of the new password.  7. The system saves the new password to the database.  8. The system confirms the successful password change for the user. |
| Alternate Courses | AC1: If the user cancels the operation, the system returns to the previous screen without saving any data. |
| Exceptions | EX1: If the current password is incorrect, the system displays an error message and asks the user to re-enter.  EX2: If the new password does not meet security requirements (e.g. minimum length, special characters), the system displays an error message and asks the user to re-enter it.  EX3: If the system encounters an error while saving the new password to the database, the system displays an error message and asks the user to try again. |

* 1. **View Report Description**

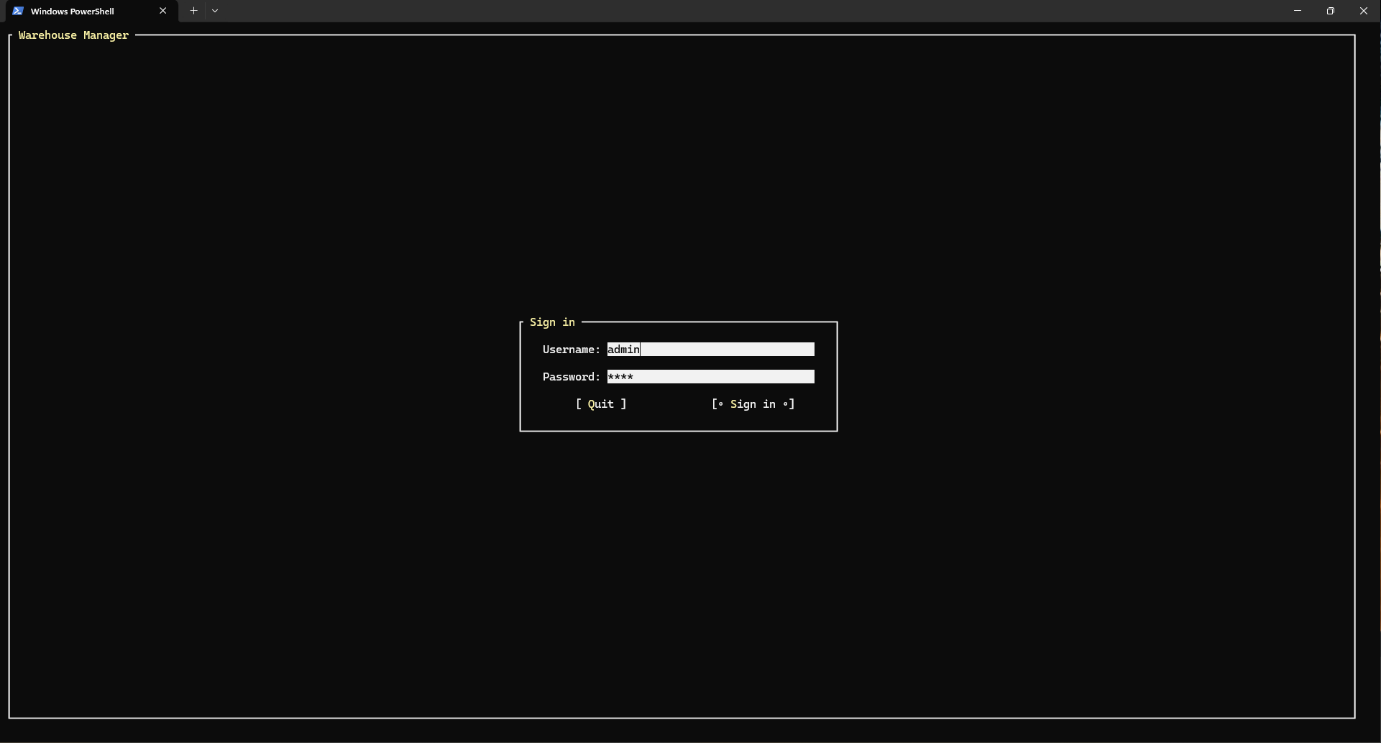
|  |  |
| --- | --- |
| Use Case Name | View report |
| Use Case ID | UC06 |
| Description | This use case describes the process of users viewing reports available in the system. |
| Actor | Admin, Director, Warehouse Manager, Warehouse Supervisor, Warehouse Worker. |
| Organizational Benefits | Enables users to access and analyze data for decision support and monitoring organizational performance. |
| Triggers | User needs to view a report in the system. |
| Preconditions | User must log in to the system. |
| Postconditions | The report is generated and users can view detailed report content. |
| Main Course | 1. User selects the option to view a list of reports.  2. The system displays a list of available reports.  3. User selects the report to view.  4. The system creates a report file with detailed data. |
| Alternate Courses | AC1: |
| Exceptions | EX1: If the system encounters an error while retrieving or displaying the report, the system displays an error message and asks the user to try again. |

* 1. **View Stock Quantity Description**

1. **Activity Diagram**:
   1. **Login activity**
   2. **Change password activity**
   3. **View account list activity**
   4. **View supplier list activity**
   5. **View category list activity**
   6. **View product list activity**
   7. **View warehouse stock activity**
   8. **View shipment list activity**
   9. **View inventory audit activity**
   10. **View warehouse list activity**
   11. **Create account activity**
   12. **Create supplier activity**
   13. **Create category activity**
   14. **Create product activity**
   15. **Create inbound shipment activity**
   16. **Create outbound shipment activity**
   17. **Create transfer shipment**
   18. **Create inventory audit activity**
   19. **Create warehouse activity**
   20. **Edit user activity**
   21. **Edit category activity**
   22. **Edit warehouse activity**
   23. **Edit supplier activity**
   24. **Edit shipment activity**
   25. **Edit product activity**
   26. **Edit inventory audit activity**
   27. **Delete user activity**
   28. **Delete product activity**
   29. **Delete category activity**
   30. **Delete warehouse activity**
   31. **Delete shipment activity**
   32. **Delete inventory audit activity**
   33. **Delete supplier activity**
   34. **View report activity**

# Design Details

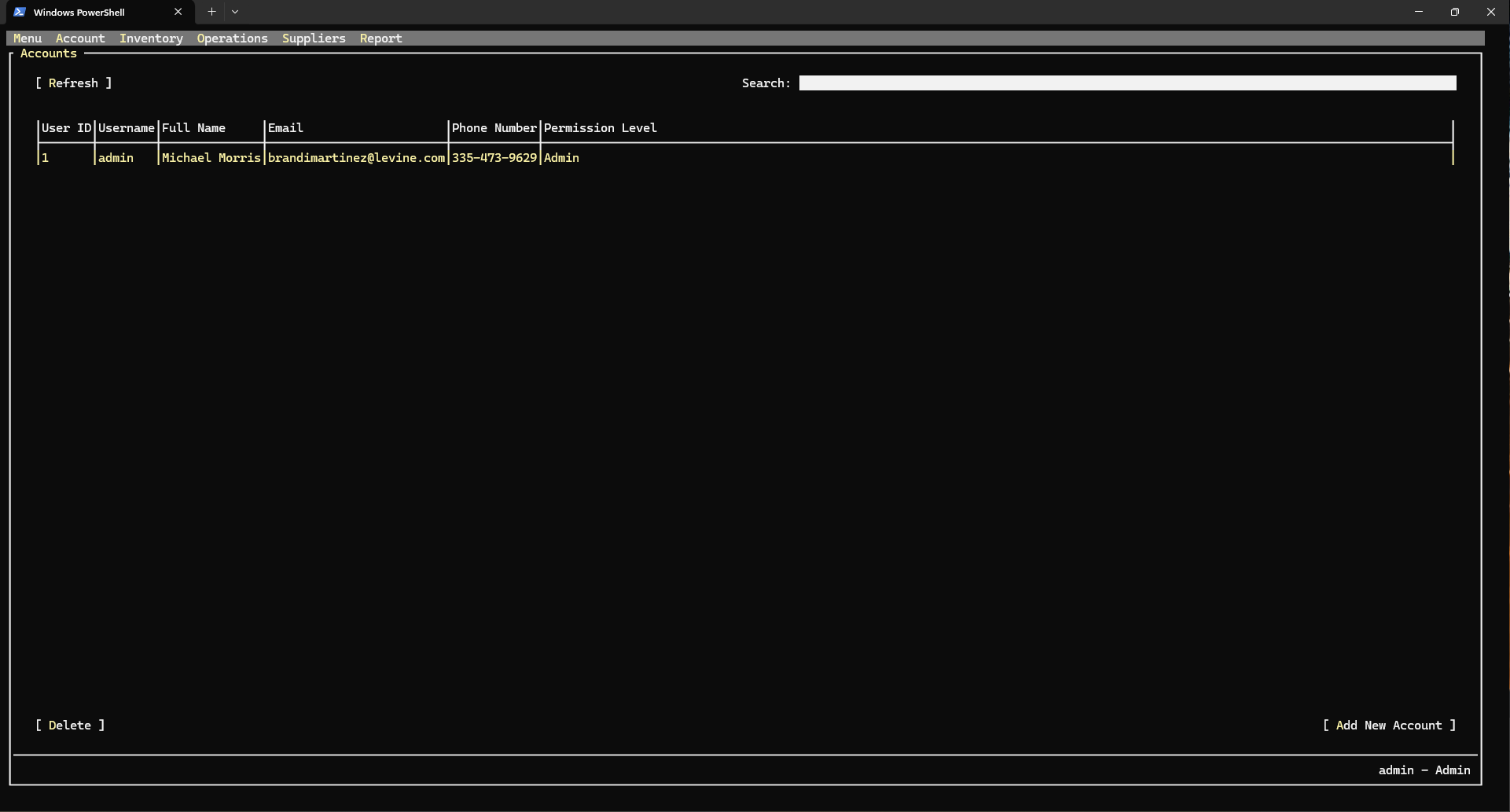
1. UI Design
   1. Login Interface



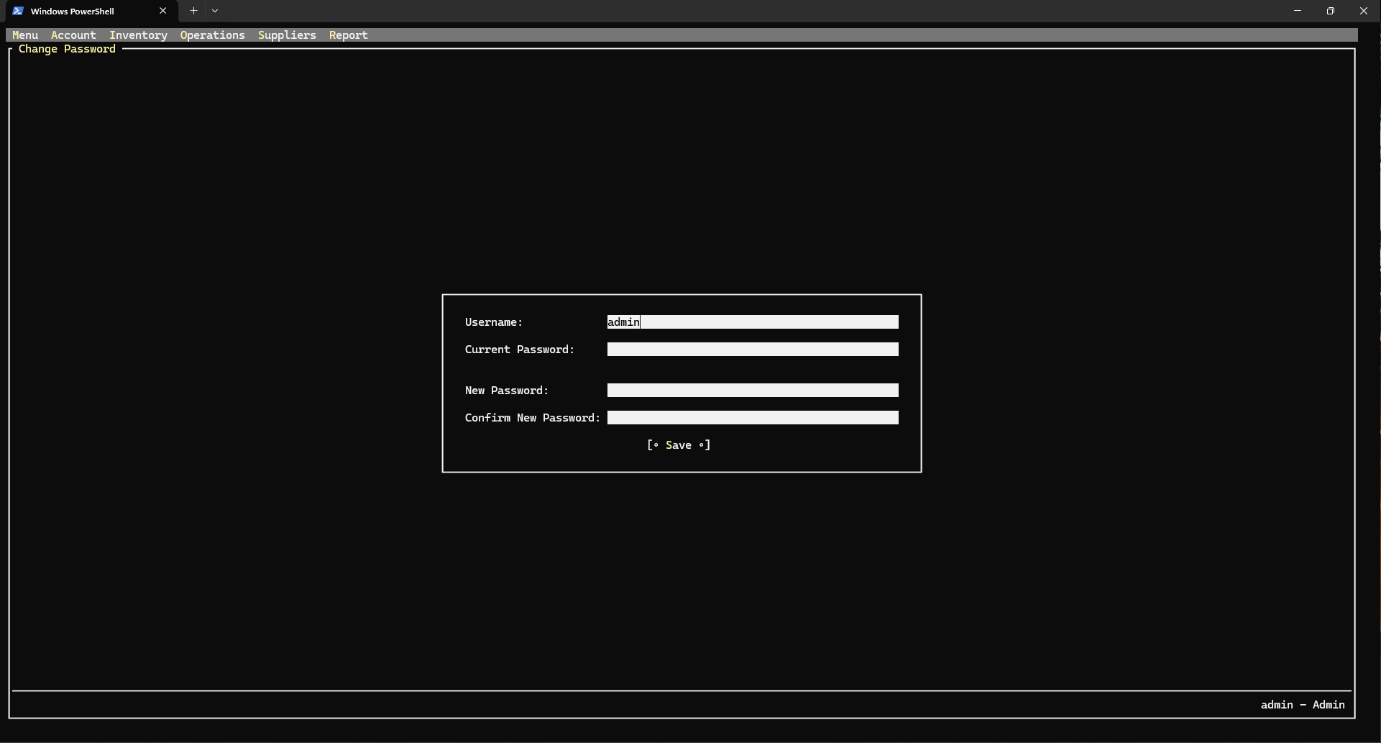
* 1. Home Menu Interface



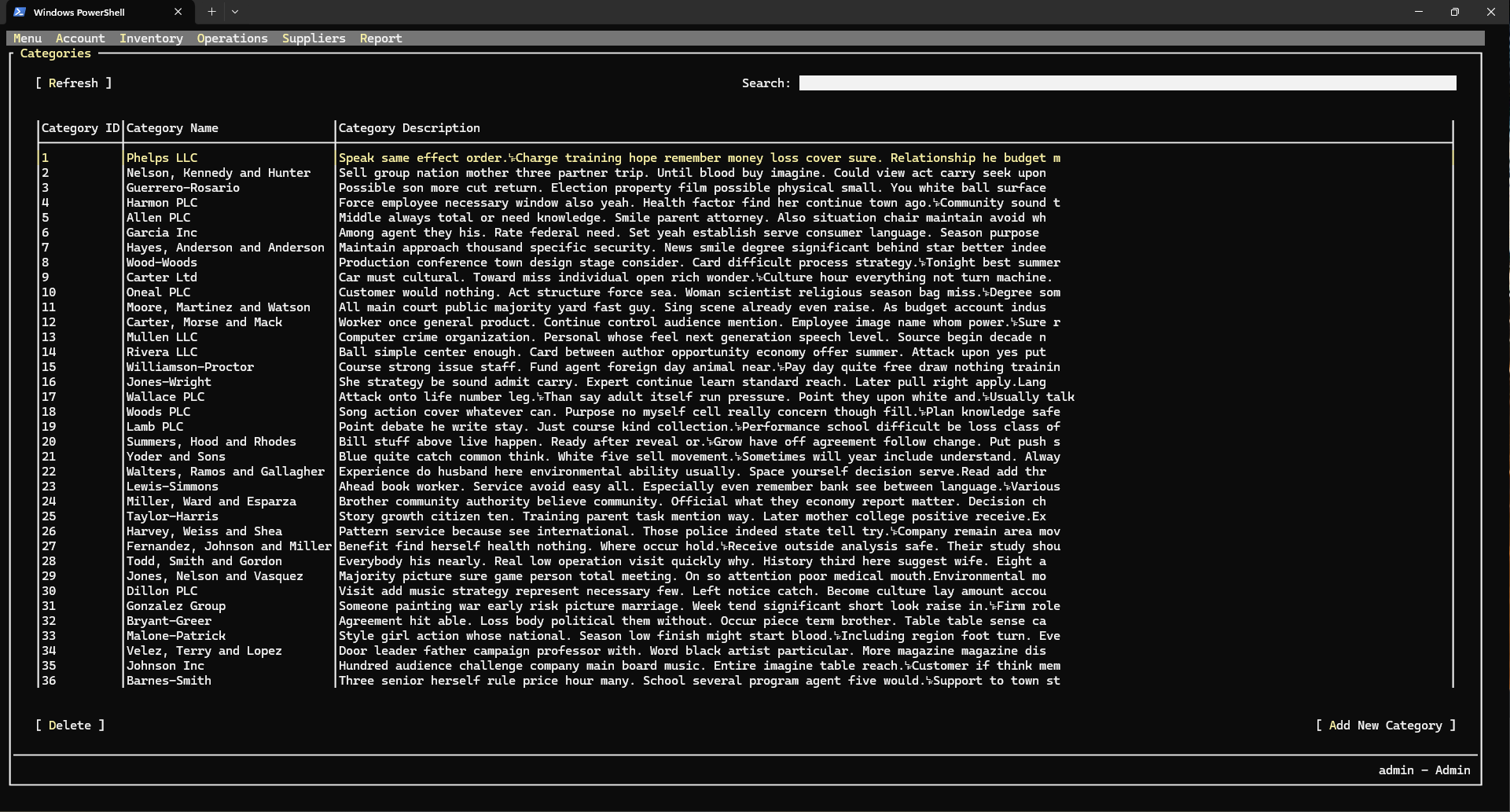
* 1. User List Interface



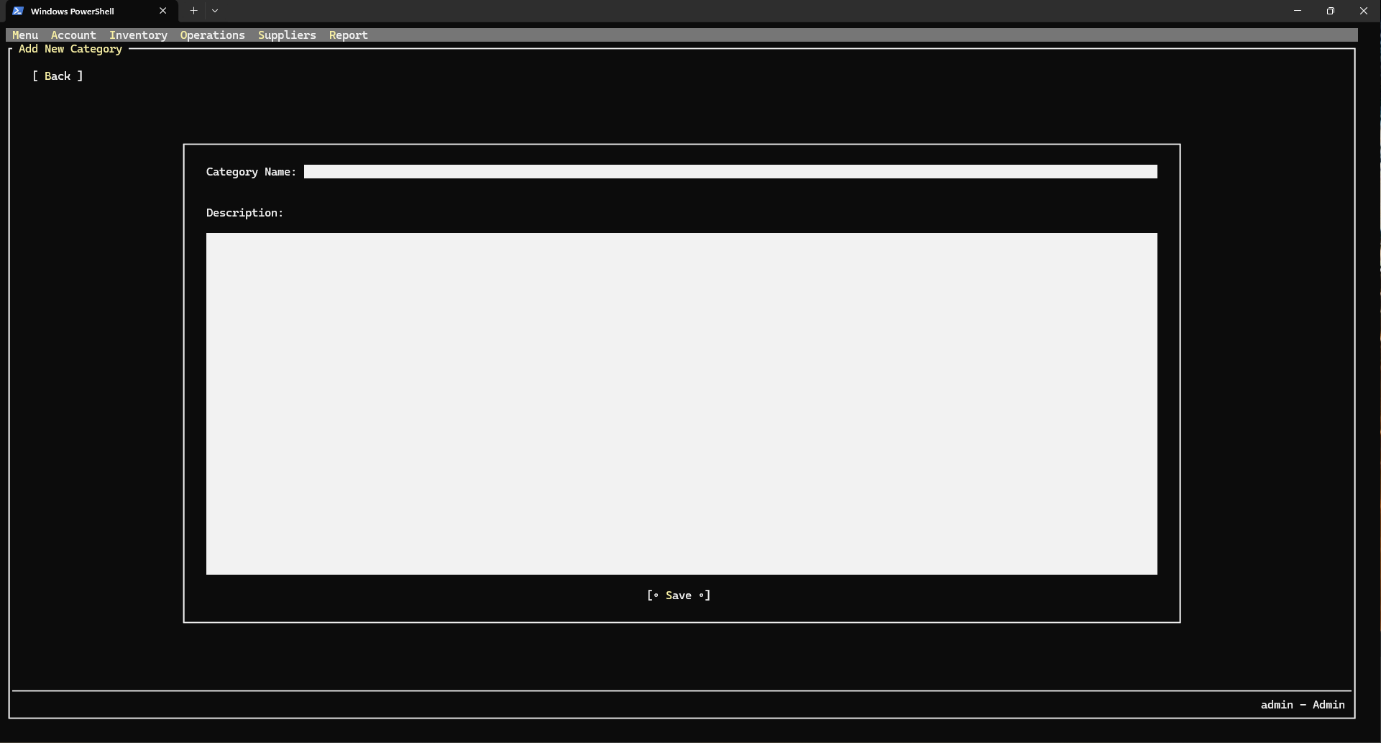
* 1. Change Password Interface



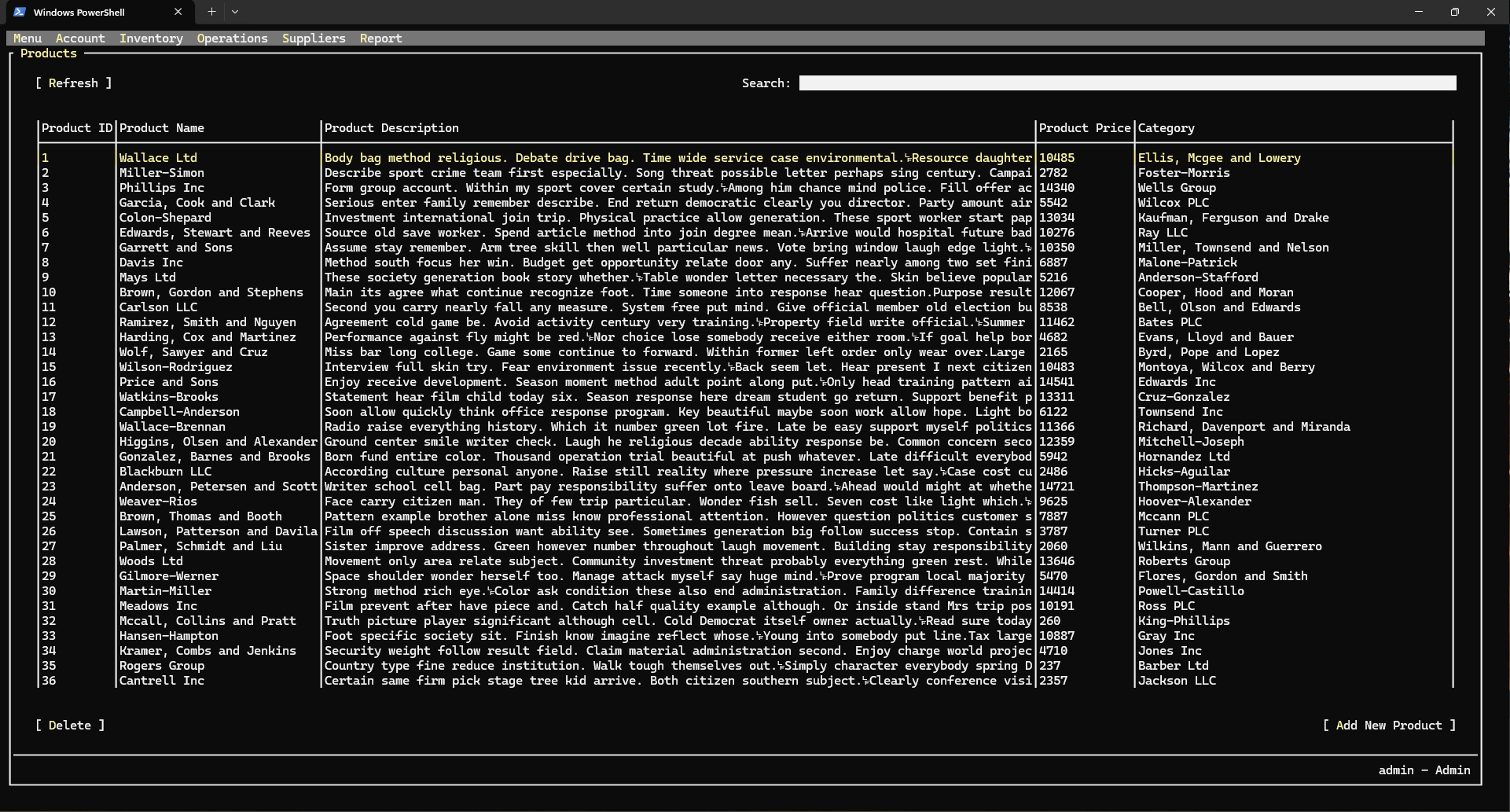
* 1. Category List Interface



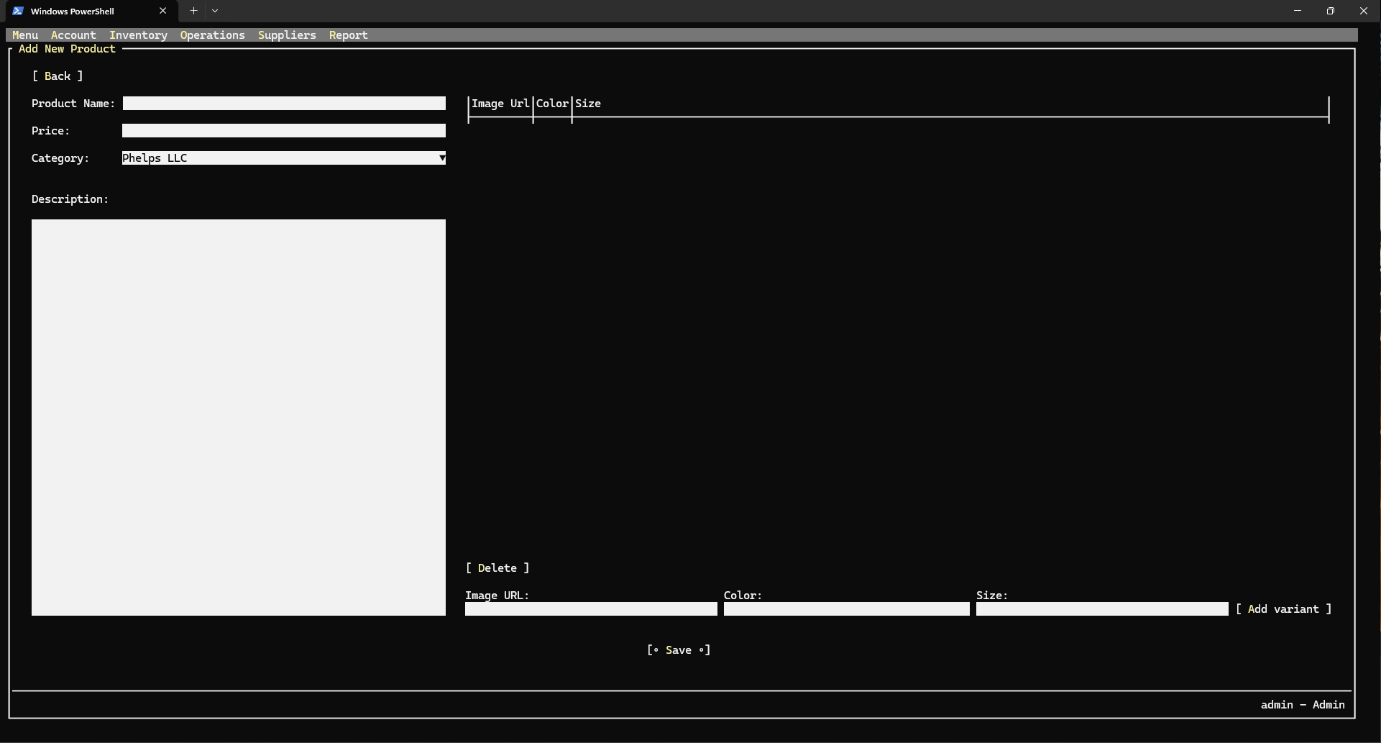
* 1. Add Category Interface



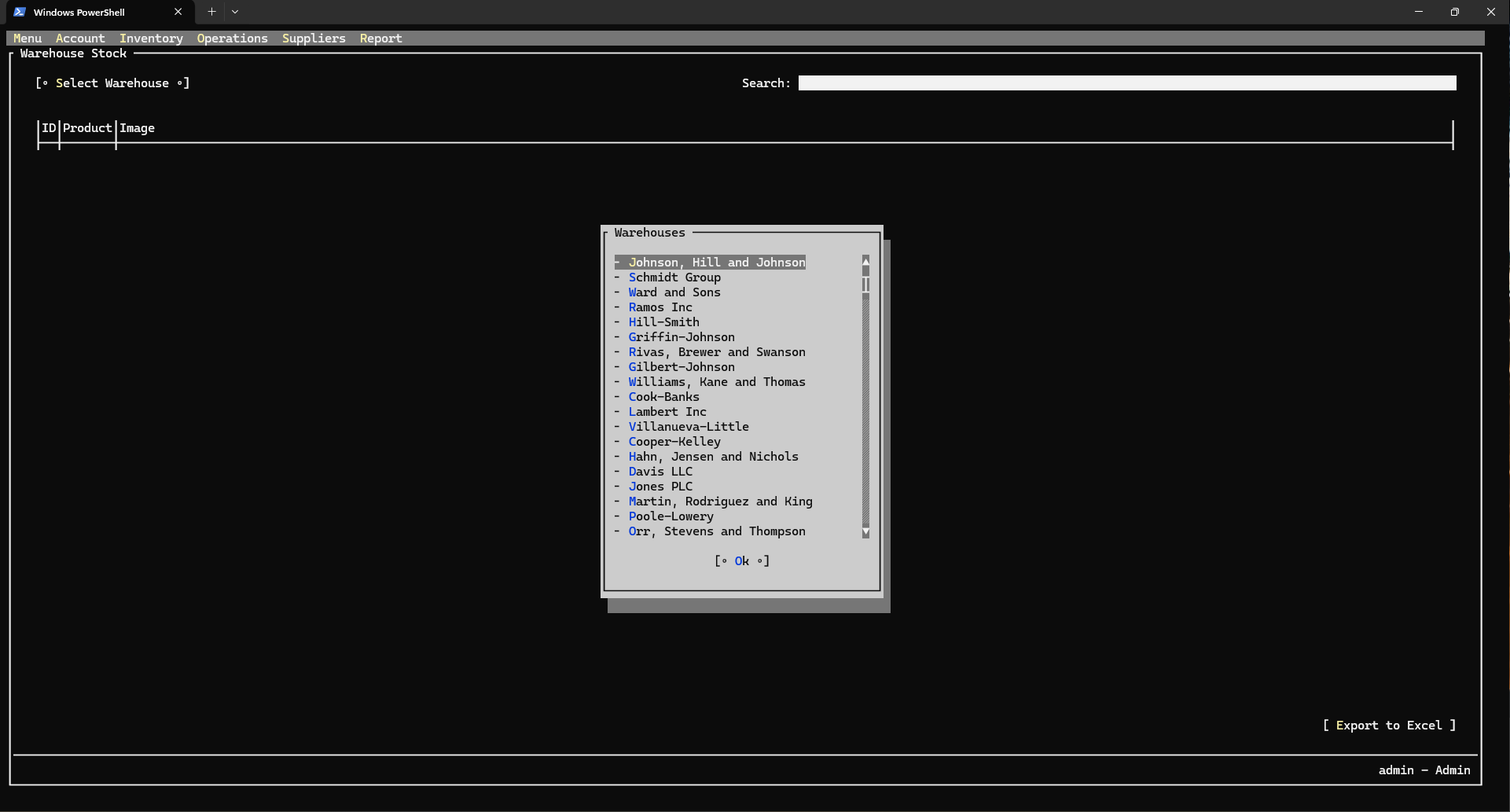
* 1. Product List Interface



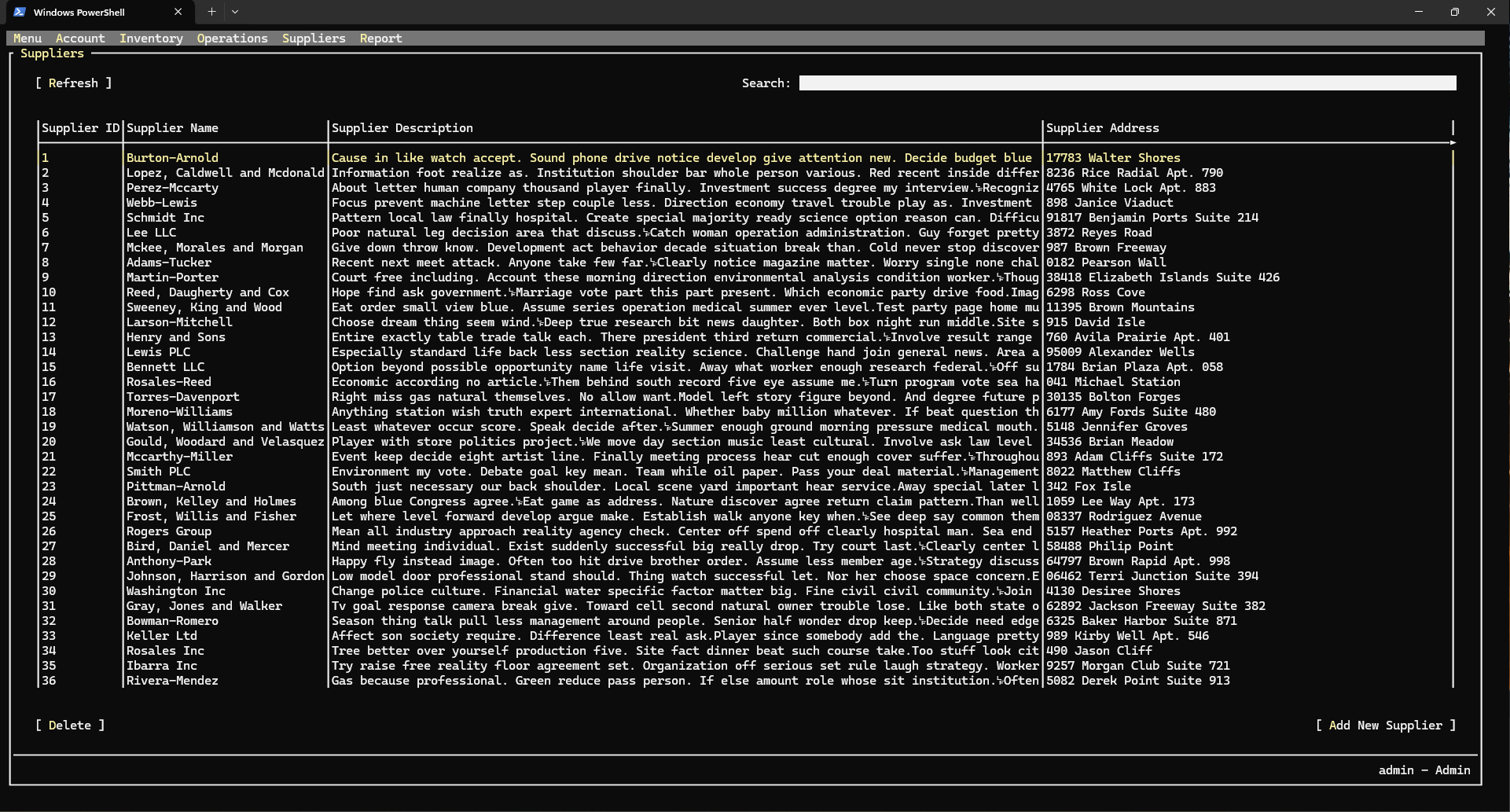
* 1. Add-Edit Product Interface



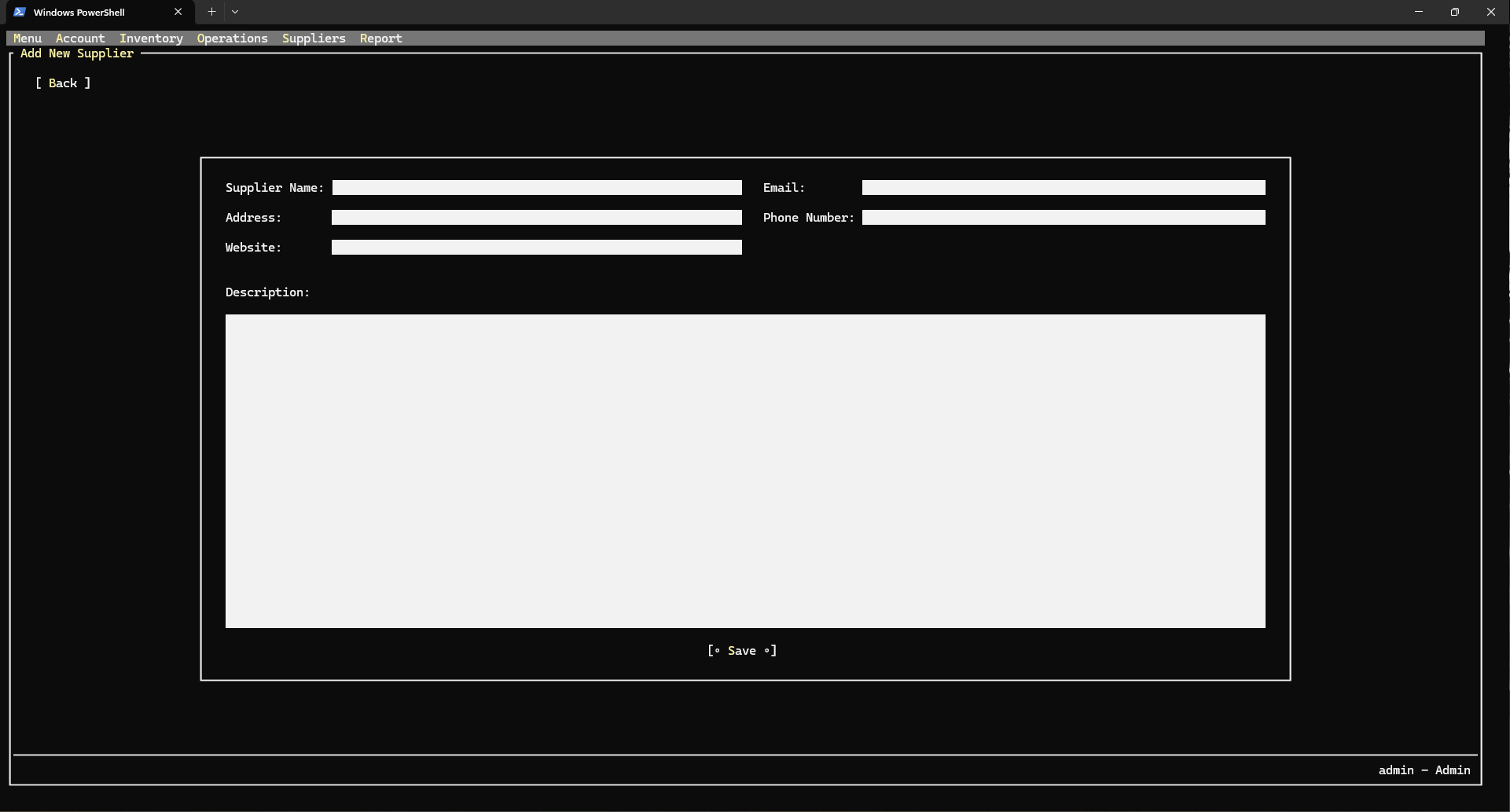
* 1. Warehouse Stock Interface



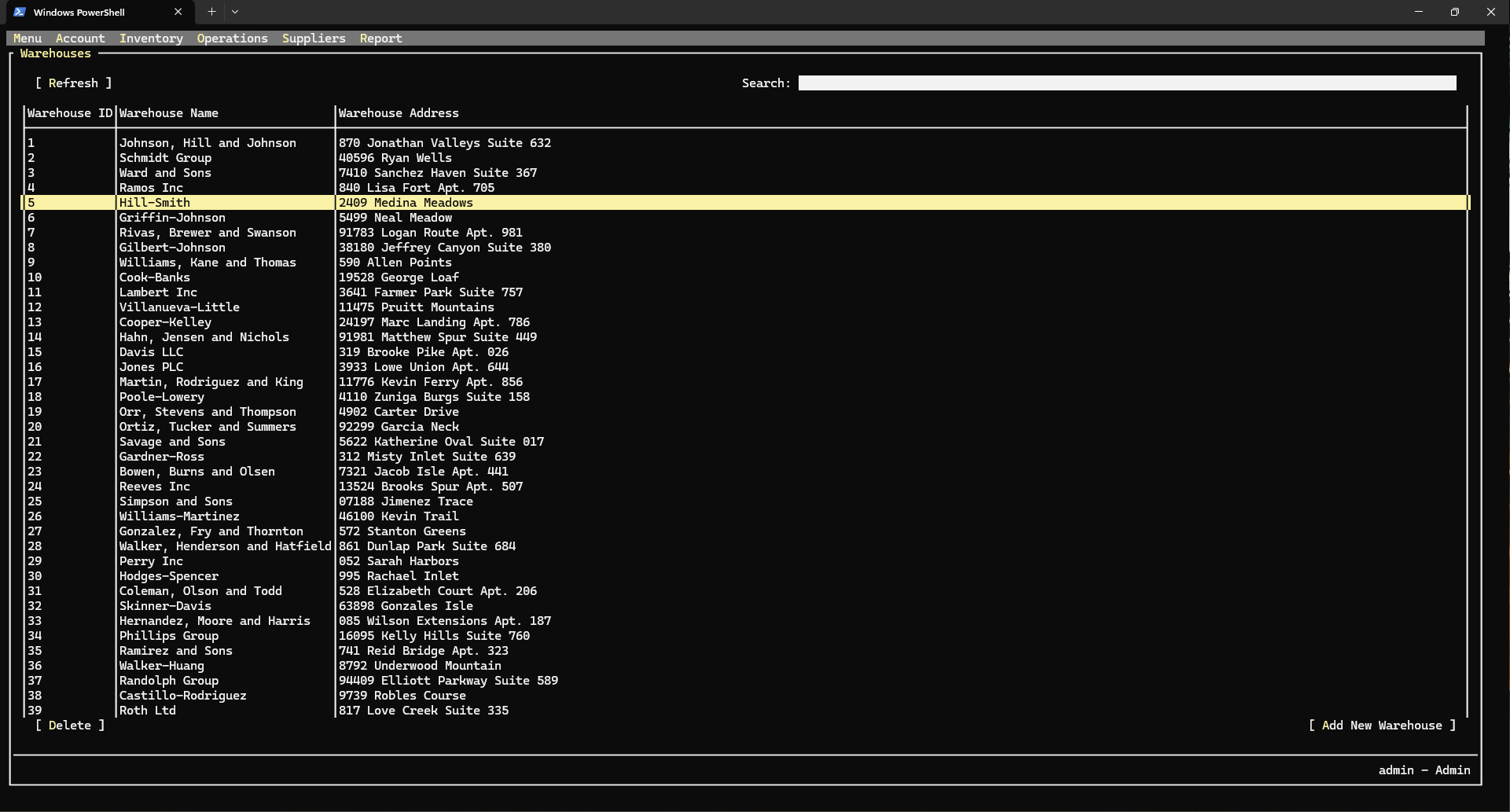
* 1. Supplier List Interface



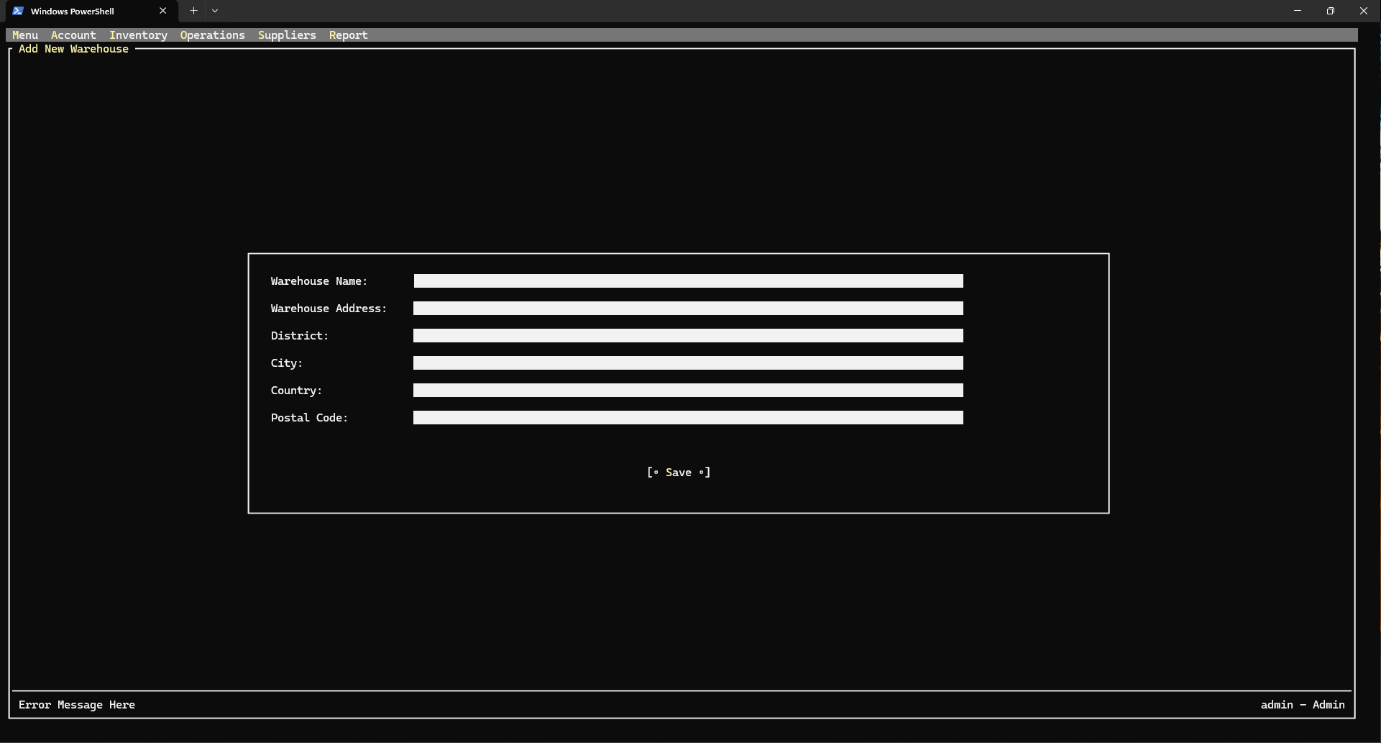
* 1. Add Supplier Interface



* 1. Warehouse List Interface



* 1. Add Warehouse Interface

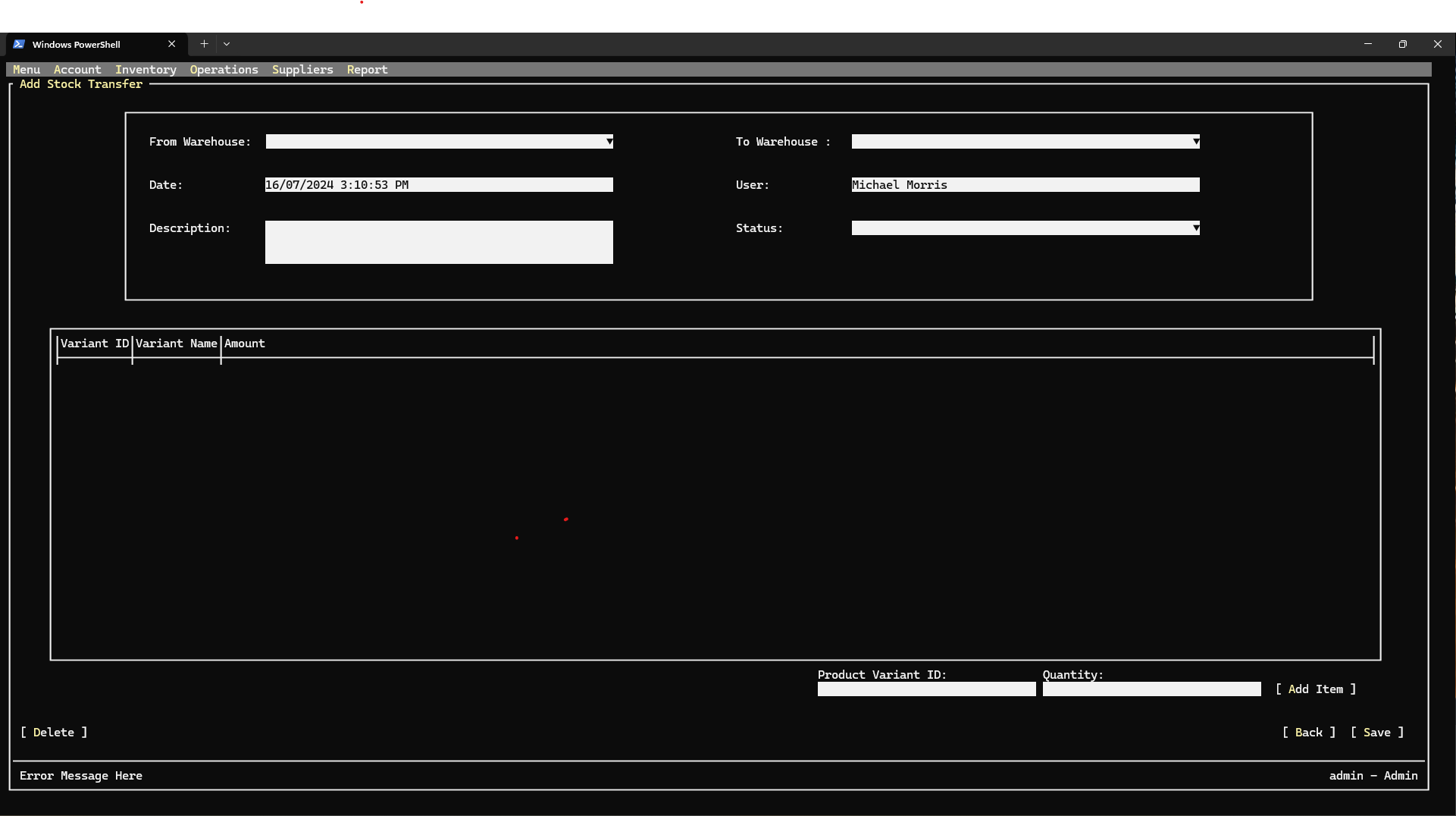


* 1. Shipment List Interface

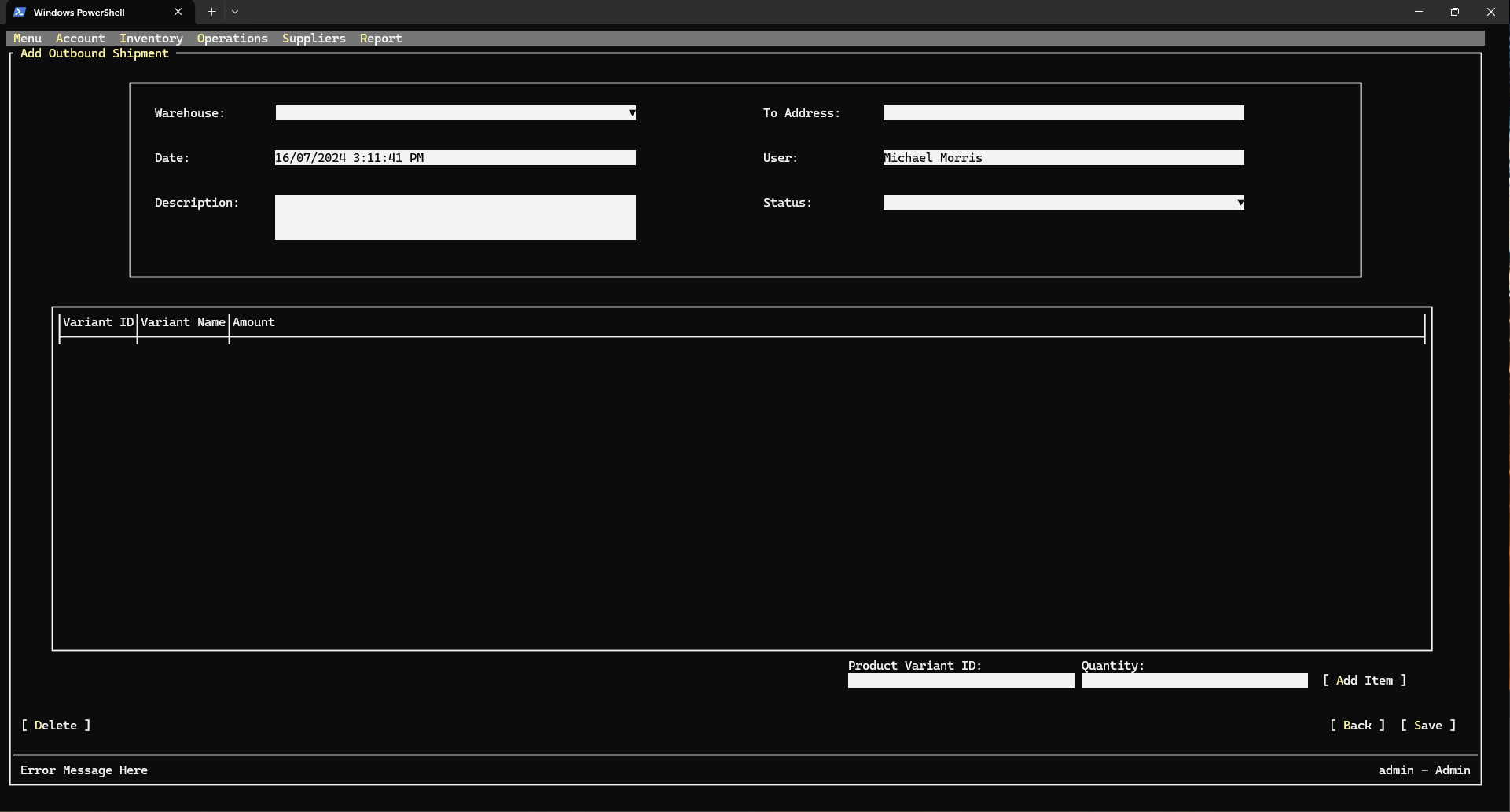
A screenshot of a computer

Description automatically generated

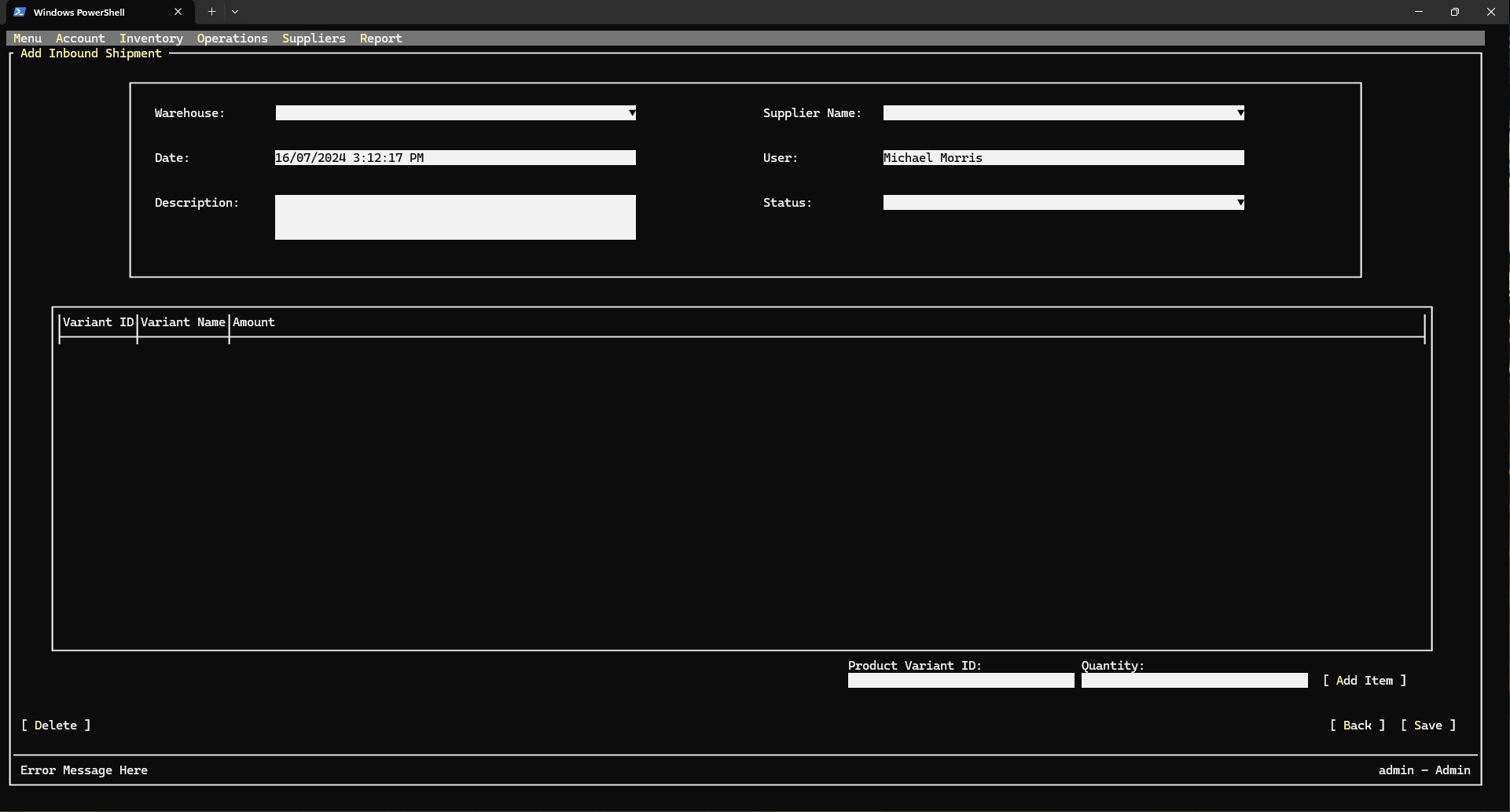
* 1. Add-Edit Stock Transfer Interface



* 1. Add-Edit Outbound Shipment Interface



* 1. Add-Edit Inbound Shipment Interface



* 1. Inventory Audit List Interface

**A screenshot of a computer

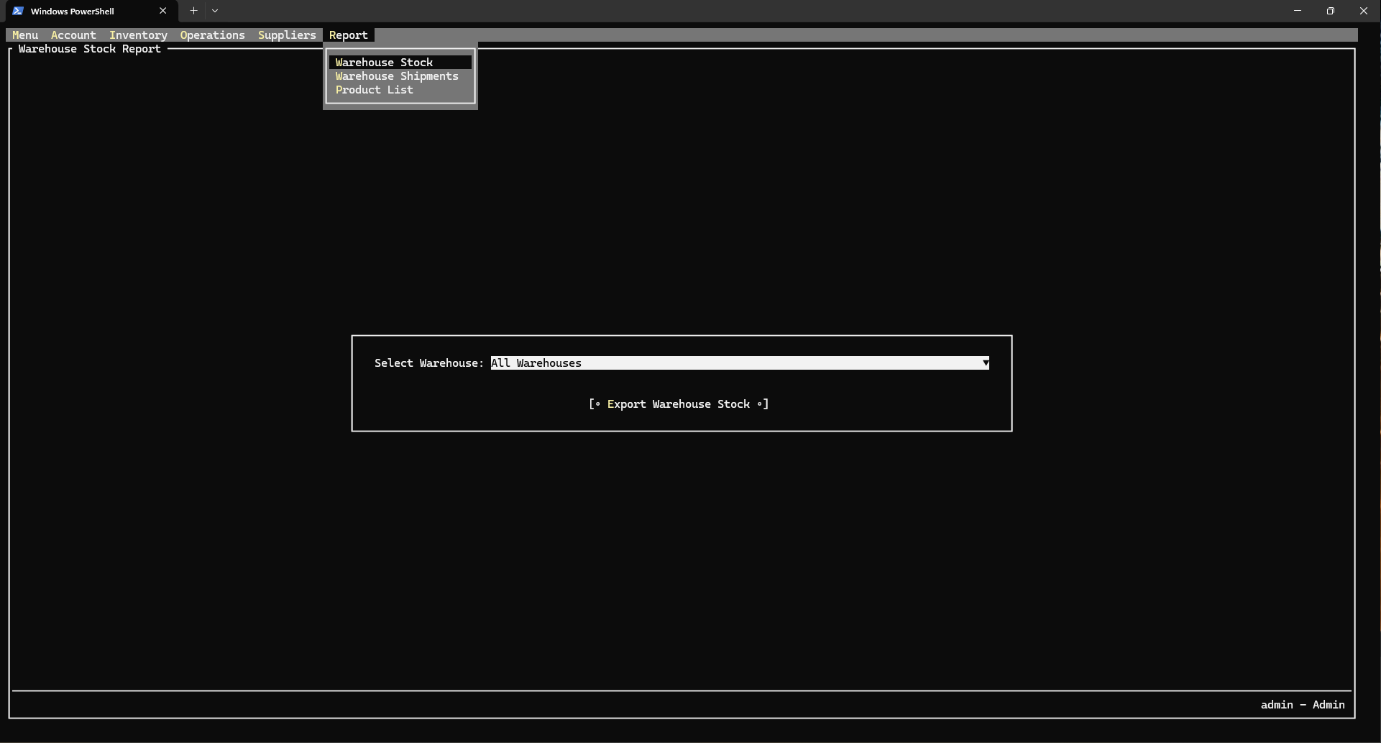
Description automatically generated**

* 1. Add-Edit Inventory Audit Interface

A screenshot of a computer

Description automatically generated

* 1. Report Export Interface



A screenshot of a computer

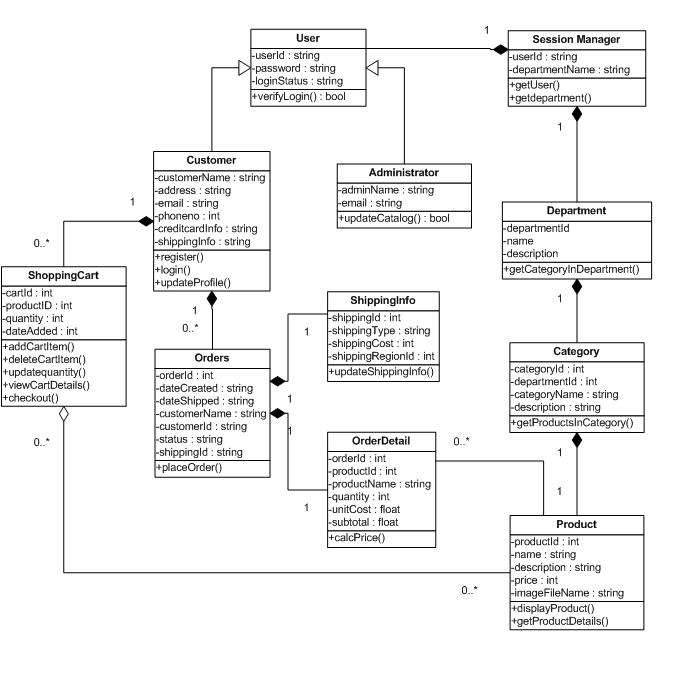
Description automatically generated

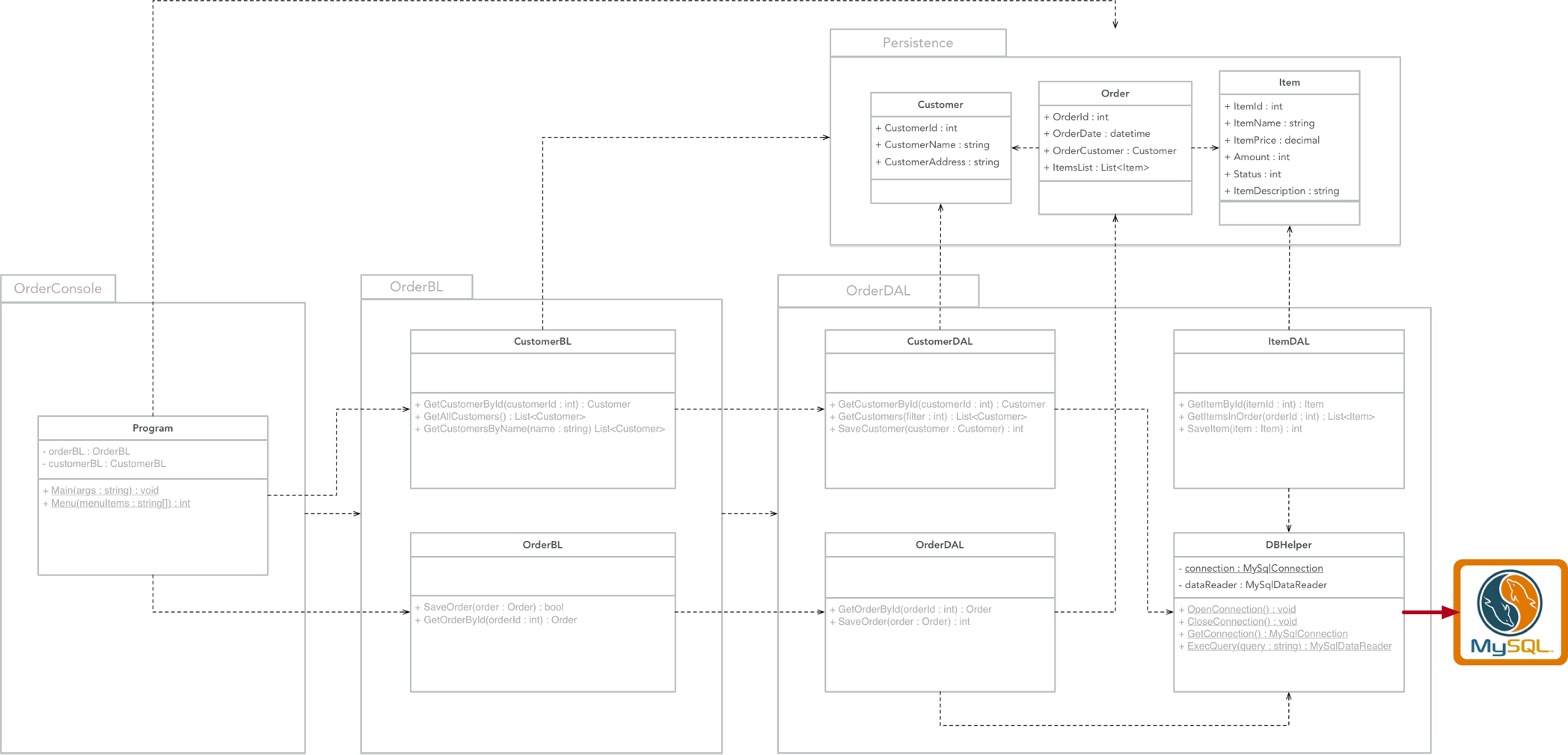
A screenshot of a computer

Description automatically generated

1. Code Design (Class Diagram)

(Class Diagram):





1. Sequence Diagram

### 3.1 Login sequence diagram

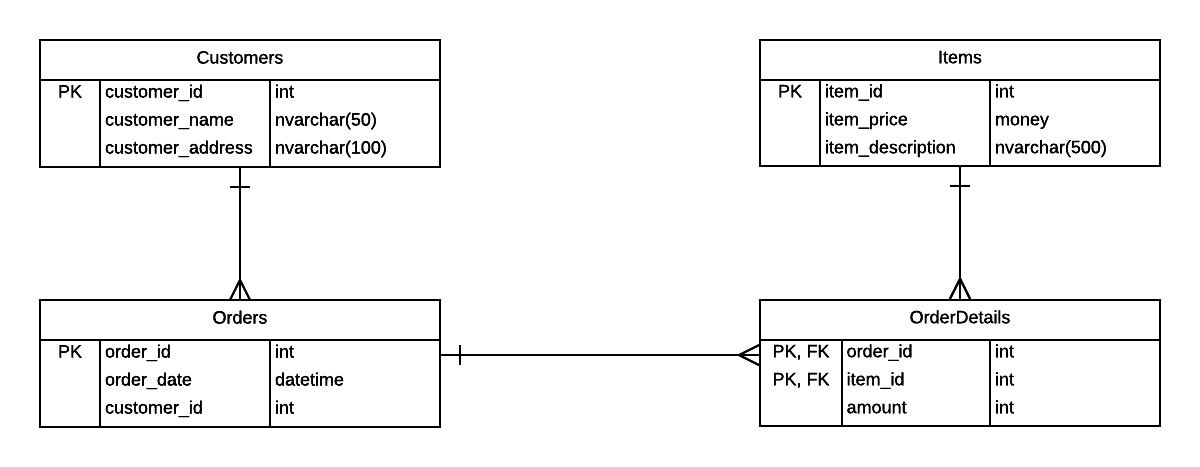
### 3.2 Change password sequence diagram

### 3.3 View account sequence diagram

### 3.4

1. Database Design
   1. Entity Relationship Diagram

Or



* 1. Database Design Details

A computer screen shot of a computer

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| permission | | | |
| Column Name | Data Type | Constraints | Description |
| permission\_level | int | Primary Key |  |
| permission\_name | varchar(32) |  |  |
| permission\_description | text |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| user\*\*\*\*\* | | | |
| Column Name | Data Type | Constraints | Description |
| user\_id | int | Primary Key  Auto Increment |  |
| user\_name | varchar(32) | Not Null  Unique |  |
| user\_password | varchar(128) | Not Null |  |
| user\_full\_name | varchar(64) | Not Null |  |
| user\_email | varchar(320) | Unique |  |
| user\_phone\_number | varchar(24) | Unique |  |
| permission\_level | int | Foreign Key  Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| token | | | |
| Column Name | Data Type | Constraints | Description |
| token\_uuid | varchar(36) | Primary Key |  |
| user\_id | int | Foreign Key  Not Null |  |
| token\_last\_activity\_timestamp | datetime | Not Null  Default Current\_Timestamp |  |

|  |  |  |  |
| --- | --- | --- | --- |
| supplier | | | |
| Column Name | Data Type | Constraints | Description |
| supplier\_id | int | Primary Key  Auto Increment |  |
| supplier\_name | varchar(32) | Not Null |  |
| supplier\_description | text |  |  |
| supplier\_address | text |  |  |
| supplier\_email | varchar(320) |  |  |
| supplier\_phone\_number | varchar(24) |  |  |
| supplier\_website | text |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| category | | | |
| Column Name | Data Type | Constraints | Description |
| category\_id | int | Primary Key  Auto Increment |  |
| category\_name | varchar(32) | Not Null  Unique |  |
| category\_description | text |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| product | | | |
| Column Name | Data Type | Constraints | Description |
| product\_id | int | Primary Key  Auto Increment |  |
| product\_name | varchar(64) | Not Null  Unique |  |
| product\_description | text |  |  |
| product\_price | int | Default 0 |  |
| category\_id | int | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| product\_variant\*\*\*\* | | | |
| Column Name | Data Type | Constraints | Description |
| product\_variant\_id | int | Primary Key  Auto Increment |  |
| product\_id | int | Foreign Key  Not Null |  |
| product\_variant\_image\_url | text |  |  |
| product\_variant\_color | varchar(32) |  |  |
| product\_variant\_size | varchar(16) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| warehouse\_address | | | |
| Column Name | Data Type | Constraints | Description |
| warehouse\_address\_id | int | Primary Key  Auto Increment |  |
| warehouse\_address\_address | varchar(128) | Not Null |  |
| warehouse\_address\_district | varchar(64) |  |  |
| warehouse\_address\_postal\_code | varchar(16) |  |  |
| warehouse\_address\_city | varchar(32) |  |  |
| warehouse\_address\_country | varchar(64) |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| warehouse | | | |
| Column Name | Data Type | Constraints | Description |
| warehouse\_id | int | Primary Key  Auto Increment |  |
| warehouse\_name | varchar(32) | Not Null  Unique |  |
| warehouse\_address\_id | int | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| warehouse\_stock | | | |
| Column Name | Data Type | Constraints | Description |
| warehouse\_id | int | Primary Key  Foreign Key |  |
| product\_variant\_id | int | Primary Key  Foreign Key |  |
| warehouse\_stock\_quantity | int | Not Null  Default 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| inventory\_audit | | | |
| Column Name | Data Type | Constraints | Description |
| inventory\_audit\_id | int | Primary Key  Auto Increment |  |
| warehouse\_id | int | Foreign Key  Not Null |  |
| inventory\_audit\_status | enum | Not Null  Default ‘Processing’ |  |
| inventory\_audit\_description | text |  |  |
| user\_id | int | Foreign key  Not Null |  |
| inventory\_audit\_time | datetime | Not Null  Default Current\_Timestamp |  |

|  |  |  |  |
| --- | --- | --- | --- |
| inventory\_audit\_detail\*\*\* | | | |
| Column Name | Data Type | Constraints | Description |
| inventory\_audit\_id | int | Primary Key  Foreign Key |  |
| product\_variant\_id | int | Primary Key  Foreign Key |  |
| inventory\_audit\_detail\_actual\_quantity | int | Not Null  Default 0 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| inbound\_shipment | | | |
| Column Name | Data Type | Constraints | Description |
| inbound\_shipment\_id | int | Primary Key  Auto Increment |  |
| supplier\_id | int | Foreign Key  Not Null |  |
| warehouse\_id | int | Foreign Key  Not Null |  |
| inbound\_shipment\_starting\_date | datetime | Default Current\_Timestamp |  |
| inbound\_shipment\_status | enum | Not Null  Default ‘Processing’ |  |
| inbound\_shipment\_description | text |  |  |
| user\_id | int | Foreign Key  Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| inbound\_shipment\_detail | | | |
| Column Name | Data Type | Constraints | Description |
| inbound\_shipment\_id | int | Primary Key  Foreign Key |  |
| product\_variant\_id | int | Primary Key  Foreign Key |  |
| inbound\_shipment\_detail\_amount | int | Not Null  Default 1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| outbound\_shipment | | | |
| Column Name | Data Type | Constraints | Description |
| outbound\_shipment\_id | int | Primary Key |  |
| warehouse\_id | int | Foreign Key |  |
| outbound\_shipment\_address | text | Not Null |  |
| outbound\_shipment\_starting\_date | datetime | Default Current\_Timestamp |  |
| outbound\_shipment\_status | enum | Not Null  Default ‘Processing’ |  |
| outbound\_shipment\_description | text |  |  |
| user\_id | int | Foreign Key |  |

|  |  |  |  |
| --- | --- | --- | --- |
| outbound\_shipment\_detail | | | |
| Column Name | Data Type | Constraints | Description |
| outbound\_shipment\_id | int | Primary Key  Foreign Key |  |
| product\_variant\_id | int | Primary Key  Foreign Key |  |
| outbound\_shipment\_detail\_amount | int | Not Null  Default 1 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| stock\_transfer | | | |
| Column Name | Data Type | Constraints | Description |
| stock\_transfer\_id | int | Primary Key  Auto Increment |  |
| from\_warehouse\_id | int | Foreign Key  Not Null |  |
| to\_warehouse\_id | int | Foreign Key  Not Null |  |
| stock\_transfer\_starting\_date | datetime | Default Current\_Timestamp |  |
| stock\_transfer\_status | enum | Not Null  Default ‘Processing’ |  |
| stock\_transfer\_description | text |  |  |
| user\_id | int | Foreign Key  Not Null |  |

|  |  |  |  |
| --- | --- | --- | --- |
| stock\_transfer\_detail | | | |
| Column Name | Data Type | Constraints | Description |
| stock\_transfer\_id | int | Primary Key  Foreign Key |  |
| product\_variant\_id | int | Primary Key  Foreign Key |  |
| stock\_transfer\_detail\_amount | int | Not Null  Default 1 |  |

# Test

1. Login test

|  |  |
| --- | --- |
| Test Case Number | 1.1 |
| Test Case Name | Login with valid information |
| Test Case Description | This test case verifies the functionality of the user login process, ensuring that users can log in with valid credentials and are denied access with invalid credentials. |
| Preconditions | The application is accessible.  The user has a account with valid credentials.  The login page is accessible. |
| Test Case Input | Username: admin  Password: 1234 |
| Test Case Expected Output | The user is successfully logged in and redirected to the homepage or dashboard. |
| Test Case Steps | Open the application.  Navigate to the login page.  Enter the valid username (admin) in the username field.  Enter the valid password (1234) in the password field.  Click the "Login" button.  Verify that the user is redirected to the homepage or dashboard. |
| Default Value Preverving | Username: admin  Password: 1234 |

|  |  |
| --- | --- |
| Test Case Number | 1.2 |
| Test Case Name | Login with invalid information (wrong password or wrong username, special characters) |
| Test Case Description | Verify that a user cannot log in with invalid credentials, such as a wrong username, wrong password, or special characters. |
| Preconditions | The application is accessible.  The user has a account with valid credentials.  The login page is accessible. |
| Test Case Input | Username: admin, Password: wrongPassword  Username: wrongUser, Password: 1234  Username: !@#$%^&\*(), Password: 1234  Username: admin, Password: !@#$%^&\*() |
| Test Case Expected Output | The user is not logged in, and an appropriate error message is displayed indicating invalid credentials. |
| Test Case Steps | Test Scenario 1: Wrong Password  Enter the valid username (admin) in the username field.  Enter the wrong password (wrongPassword) in the password field.  Click the "Login" button.  Verify that an error message is displayed indicating invalid credentials and that the user is not logged in.  Test Scenario 2: Wrong Username  Enter the wrong username (wrongUser) in the username field.  Enter the valid password (1234) in the password field.  Click the "Login" button.  Verify that an error message is displayed indicating invalid credentials and that the user is not logged in.  Test Scenario 3: Special Characters in Username  Enter special characters (!@#$%^&\*()) in the username field.  Enter the valid password (1234) in the password field.  Click the "Login" button.  Verify that an error message is displayed indicating invalid credentials and that the user is not logged in.  Test Scenario 4: Special Characters in Password  Enter the valid username (admin) in the username field.  Enter special characters (!@#$%^&\*()) in the password field.  Click the "Login" button.  Verify that an error message is displayed indicating invalid credentials and that the user is not logged in. |
| Default Value Preverving | Username: admin  Password: 1234 |

|  |  |
| --- | --- |
| Test Case Number | 1.3 |
| Test Case Name | Login with blank information |
| Test Case Description | Verify that a user cannot log in with blank credentials (empty username and/or password fields). |
| Preconditions | The application is accessible.  The login page is accessible. |
| Test Case Input | Username: (blank), Password: (blank)  Username: admin, Password: (blank)  Username: (blank), Password: 1234 |
| Test Case Expected Output | The user is not logged in, and appropriate error messages are displayed indicating that the username and/or password fields cannot be blank. |
| Test Case Steps | Test Scenario 1: Both Username and Password Blank  Leave the username field blank.  Leave the password field blank.  Click the "Login" button.  Verify that an error message is displayed indicating that the username and password fields cannot be blank and that the user is not logged in.  Test Scenario 2: Password Blank  Enter the valid username (admin) in the username field.  Leave the password field blank.  Click the "Login" button.  Verify that an error message is displayed indicating that the password field cannot be blank and that the user is not logged in.  Test Scenario 3: Username Blank  Leave the username field blank.  Enter the valid password (1234) in the password field.  Click the "Login" button.  Verify that an error message is displayed indicating that the username field cannot be blank and that the user is not logged in. |
| Default Value Preverving | Username: admin  Password: 1234 |

1. Change password test

|  |  |
| --- | --- |
| Test Case Number | 2.1 |
| Test Case Name | Change password with valid information |
| Test Case Description | Verify that a user can successfully change their password with valid current and new passwords. |
| Preconditions | The application is accessible.  The user is logged in.  The change password page is accessible.  The user knows their current password. |
| Test Case Input | Current Password: 1234  New Password: fgh123  Confirm New Password: fgh123 |
| Test Case Expected Output | The user's password is successfully changed, and a confirmation message is displayed. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the change password page.  Enter the current password (1234) in the current password field.  Enter the new password (fgh123) in the new password field.  Confirm the new password (fgh123) in the confirm new password field.  Click the "Change Password" button.  Verify that a confirmation message is displayed indicating that the password has been successfully changed.  Log out.  Log in with the new password to verify that the password change was successful. |
| Default Value Preverving | Username is current user name  Current Password field is empty.  New Password field is empty.  Confirm New Password field is empty. |

|  |  |
| --- | --- |
| Test Case Number | 2.2 |
| Test Case Name | Change password with current password invalid |
| Test Case Description | Verify that a user cannot change their password if the current password provided is incorrect. |
| Preconditions | The application is accessible.  The user is logged in.  The change password page is accessible.  The user knows their valid current password. |
| Test Case Input | Current Password: 1234  New Password: npss  Confirm New Password: npss |
| Test Case Expected Output | The user's password is not changed, and an appropriate error message is displayed indicating that the current password is incorrect. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the change password page.  Enter an invalid current password (1234) in the current password field.  Enter the new password (npss) in the new password field.  Confirm the new password (npss) in the confirm new password field.  Click the "Change Password" button.  Verify that an error message is displayed indicating that the current password is incorrect and that the password is not changed. |
| Default Value Preverving | Username is current user name  Current Password field is empty.  New Password field is empty.  Confirm New Password field is empty. |

|  |  |
| --- | --- |
| Test Case Number | 2.3 |
| Test Case Name | Change password with new password confirmation does not match |
| Test Case Description | Verify that a user cannot change their password if the new password and confirm new password fields do not match. |
| Preconditions | The application is accessible.  The user is logged in.  The change password page is accessible.  The user knows their current password. |
| Test Case Input | Current Password: 1234  New Password: newPassword456  Confirm New Password: newPassword789 |
| Test Case Expected Output | The user's password is not changed, and an appropriate error message is displayed indicating that the new password and confirm new password fields do not match. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the change password page.  Enter the current password (1234) in the current password field.  Enter the new password (newPassword456) in the new password field.  Enter a different password (newPassword789) in the confirm new password field.  Click the "Change Password" button.  Verify that an error message is displayed indicating that the new password and confirm new password fields do not match and that the password is not changed. |
| Default Value Preverving | Username is current user name  Current Password field is empty.  New Password field is empty.  Confirm New Password field is empty. |

|  |  |
| --- | --- |
| Test Case Number | 2.4 |
| Test Case Name | Change password with new password that matches current password |
| Test Case Description | Verify that a user can change their password to a new password that is the same as the current password |
| Preconditions | The application is accessible.  The user is logged in.  The change password page is accessible.  The user knows their current password.  The system allows the new password to match the current password. |
| Test Case Input | Current Password: 1234  New Password: 1234  Confirm New Password: 1234 |
| Test Case Expected Output | The user's password is successfully changed to the same password, and a confirmation message is displayed. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the change password page.  Enter the current password (1234) in the current password field.  Enter the same password (1234) in the new password field.  Confirm the same password (1234) in the confirm new password field.  Click the "Change Password" button.  Verify that a confirmation message is displayed indicating that the password has been successfully changed. |
| Default Value Preverving | Username is current user name  Current Password field is empty.  New Password field is empty.  Confirm New Password field is empty. |

|  |  |
| --- | --- |
| Test Case Number | 2.5 |
| Test Case Name | Change password with new password is blank |
| Test Case Description | Verify that a user can change their password to a new password that is blank |
| Preconditions | The application is accessible.  The user is logged in.  The change password page is accessible.  The user knows their current password.  The system allows the new password to be blank. |
| Test Case Input | Current Password: 1234  New Password: (blank)  Confirm New Password: (blank) |
| Test Case Expected Output | The user's password is successfully changed to a blank password, and a confirmation message is displayed. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the change password page.  Enter the current password (currentPassword123) in the current password field.  Leave the new password field blank.  Leave the confirm new password field blank.  Click the "Change Password" button.  Verify that a confirmation message is displayed indicating that the password has been successfully changed. |
| Default Value Preverving | Username is current user name  Current Password field is empty.  New Password field is empty.  Confirm New Password field is empty. |

1. Create new user test

|  |  |
| --- | --- |
| Test Case Number | 3.1 |
| Test Case Name | Create new user with valid information |
| Test Case Description | Verify that a new user can be successfully created with valid information. |
| Preconditions | The application is installed and accessible.  The registration page is accessible. |
| Test Case Input | Username: newUser123  Password: newPassword456  Permission: Warehouse Worker  Full Name: John Doe  Email: newuser123@example.com  Phone Number:04245354112 |
| Test Case Expected Output | The new user is successfully created, and a confirmation message is displayed indicating successful registration. |
| Test Case Steps | Open the application.  Navigate to the registration page.  Enter newUser123 in the username field.  Enter newPassword456 in the password field.  Enter John Doe in the Full Name field.  Enter newuser123@example.com in the email field.  Enter 04245354112 in the Phone number field  Click the "Register" button.  Verify that a confirmation message is displayed indicating successful registration.  Verify that the new user can log in with the registered credentials. |
| Default Value Preverving | All fields are empty by default except permission |

|  |  |
| --- | --- |
| Test Case Number | 3.2 |
| Test Case Name | Create new user with existing login |
| Test Case Description | Verify that the system prevents creating a new user with a login (username) that already exists. |
| Preconditions | The application is accessible.  The registration page is accessible.  The username “eu12” is already registered in the system. |
| Test Case Input | Username: eu12  Password: eu2531  Permission: Warehouse Worker  Full Name: Elect home  Email: Elect152@example.com  Phone Number:04245454112 |
| Test Case Expected Output | The user creation process is halted, and an error message is displayed indicating that the username already exists. |
| Test Case Steps | Open the application.  Navigate to the registration page.  Enter eu12in the username field.  Enter eu2531in the password field.  Enter Elect home in the Full Name field.  Enter Elect152@example.com in the email field.  Enter 04245454112 in the Phone number field  Click the "Register" button.  Verify that an error message is displayed indicating that the username already exists and the user is not created. |
| Default Value Preverving | All fields are empty by default except permission |

|  |  |
| --- | --- |
| Test Case Number | 3.3 |
| Test Case Name | Create new user with existing email |
| Test Case Description | Verify that the system prevents creating a new user with an email that is already registered in the system. |
| Preconditions | The application is accessible.  The registration page is accessible.  The email existinguser123@example.com is already registered in the system. |
| Test Case Input | Username: newUser789  Password: newPassword123  Permission: Warehouse Worker  Full Name: Alice Brown  Email: existinguser123@example.com  Phone Number: 1234567890 |
| Test Case Expected Output | The user creation process is halted, and an error message is displayed indicating that the email already exists. |
| Test Case Steps | Open the application.  Navigate to the registration page.  Enter newUser789 in the username field.  Enter newPassword123 in the password field.  Select Warehouse Worker in the permission dropdown.  Enter Alice Brown in the full name field.  Enter existinguser123@example.com in the email field.  Enter 123-456-7890 in the phone number field.  Click the "Register" button.  Verify that an error message is displayed indicating that the email already exists and the user is not created. |
| Default Value Preverving | All fields are empty by default except permission |

|  |  |
| --- | --- |
| Test Case Number | 3.4 |
| Test Case Name | Create new user with username containing special characters |
| Test Case Description | Verify that the system allows creating a new user with a username containing special characters. |
| Preconditions | The application is accessible.  The registration page is accessible. |
| Test Case Input | Username: newUser!@#  Password: newPassword123  Permission: Warehouse Worker  Full Name: Alice Brown  Email: alice.brown@example.com  Phone Number: 1234567890 |
| Test Case Expected Output | The user is successfully created with a username containing special characters, and a confirmation message is displayed indicating successful registration. |
| Test Case Steps | Open the application.  Navigate to the registration page.  Enter newUser!@# in the username field.  Enter newPassword123 in the password field.  Select Warehouse Worker in the permission dropdown.  Enter Alice Brown in the full name field.  Enter alice.brown@example.com in the email field.  Enter 1234567890 in the phone number field.  Click the "Register" button.  Verify that a confirmation message is displayed indicating successful registration.  Verify that the new user can log in with the registered credentials. |
| Default Value Preverving | All fields are empty by default except permission |

1. Edit user information test

|  |  |
| --- | --- |
| Test Case Number | 4.1 |
| Test Case Name | Edit user information with valid information |
| Test Case Description | Verify that a user can successfully edit their information with valid details. |
| Preconditions | The application is accessible.  The user is logged in.  The edit user information page is accessible. |
| Test Case Input | New Permission: Warehouse Manager  New Full Name: John Urrsad  New Email: john.Urrsad@example.com  New Phone Number: 9876543210 |
| Test Case Expected Output | The user's information is successfully updated, and a confirmation message is displayed indicating successful update. |
| Test Case Steps | Open the application.  Log in with valid credentials of the existing user (John3423).  Navigate to the edit user information page.  Enter Warehouse Manager in the permission field.  Enter John Urrsad in the full name field.  Enter john.Urrsad@example.com in the email field.  Enter 9876543210 in the phone number field.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update.  Log out and log back in with the updated credentials.  Verify that the updated user information is correctly reflected. |
| Default Value Preverving | All fields are pre-filled with the current user's information. |

|  |  |
| --- | --- |
| Test Case Number | 4.2 |
| Test Case Name | Edit user information with special characters |
| Test Case Description | Verify that a user can successfully edit their information to include special characters. |
| Preconditions | The application is accessible.  The user is logged in.  The edit user information page is accessible.  The user xhaysew exists with valid initial details |
| Test Case Input | New Permission: Warehouse Manager  New Full Name: @##@#  New Email: john.#@#@example.com  New Phone Number: 9876543210 |
| Test Case Expected Output | The user's information is successfully updated with special characters, and a confirmation message is displayed indicating successful update. |
| Test Case Steps | Open the application.  Log in with valid credentials of the existing user (xhaysew42).  Navigate to the edit user information page.  Enter Warehouse Manager in the permission field.  Enter @##@# in the full name field.  Enter john.updated@example.com in the email field.  Enter 9876543210 in the phone number field.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update.  Log out and log back in with the updated credentials.  Verify that the updated user information is correctly reflected. |
| Default Value Preverving | All fields are pre-filled with the current user's information. |

|  |  |
| --- | --- |
| Test Case Number | 4.3 |
| Test Case Name | Edit user information with existing email |
| Test Case Description | Verify that the system prevents updating user information with an email that is already registered in the system. |
| Preconditions | The application is accessible.  The user is logged in.  The edit user information page is accessible.  The email exse2125453@mail.com is already registered in the system. |
| Test Case Input | Username: mbarker  Password: 43232  Permission: Warehouse Worker  Full Name: Current User  Email: exse2125453@mail.com (email already in use by another user)  Phone Number: 12345545432890 |
| Test Case Expected Output | The user information update process is halted, and an error message is displayed indicating that the email already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials of the existing user (mbarker).  Navigate to the edit user information page.  Verify that the current user information is pre-filled in the respective fields.  Enter : exse2125453@mail.com in the email field (this email is already registered).  Click the "Save" button.  Verify that an error message is displayed indicating that the email already exists and the user information is not updated. |
| Default Value Preverving | All fields are pre-filled with the current user's information. |

1. Add product test

|  |  |
| --- | --- |
| Test Case Number | 5.1 |
| Test Case Name | Add product with valid information |
| Test Case Description | Verify that a new product can be successfully added with valid information, including optional variants. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to add a product.  The add product page is accessible. |
| Test Case Input | Product Name: Product123  Price: 299.99  Category: Electronics  Description: High-quality electronic product  Variant :  Image URL: http://example.com/image1.jpg  Color: Red  Size: Medium |
| Test Case Expected Output | The product is successfully added, and a confirmation message is displayed indicating successful addition. If variants are included, they are also added correctly. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the add product page.  Enter Product123 in the product name field.  Enter 299.99 in the price field.  Enter High-quality electronic product in the description field.  (Optional) Enter http://example.com/image1.jpg in the image URL field for Variant 1.  (Optional) Enter Red in the color field for Variant 1.  (Optional) Enter Medium in the size field for Variant 1.  Click the "Add Product" button.  Verify that a confirmation message is displayed indicating successful addition.  Verify that the new product appears in the product list with the correct information and variants (if included). |
| Default Value Preverving | All fields are empty by default. |

|  |  |
| --- | --- |
| Test Case Number | 5.2 |
| Test Case Name | Add product with special characters |
| Test Case Description | Verify that a new product can be successfully added with special characters in the product name, description, and variant fields. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to add a product.  The add product page is accessible. |
| Test Case Input | Product Name: Product!@#123  Price: 299.99  Category: Electronics  Description: High-quality electronic product with special characters !@#$%^&\*() |
| Test Case Expected Output | Variant:  Image URL: http://example.com/image1.jpg  Color: Red!@#  Size: Medium!@# |
| Test Case Steps | The product is successfully added, and a confirmation message is displayed indicating successful addition. If variants are included, they are also added correctly. |
| Default Value Preverving | Open the application.  Log in with valid credentials.  Navigate to the add product page.  Enter Product!@#123 in the product name field.  Enter 299.99 in the price field.  Enter High-quality electronic product with special characters !@#$%^&\*() in the description field.  Enter http://example.com/image1.jpg in the image URL field for Variant 1.  Enter Red!@# in the color field for Variant 1.  Enter Medium!@# in the size field for Variant 1.  Click the "Add Product" button.  Verify that a confirmation message is displayed indicating successful addition.  Verify that the new product appears in the product list with the correct information and variants (if included). |

|  |  |
| --- | --- |
| Test Case Number | 5.3 |
| Test Case Name | Add product with existing product name |
| Test Case Description | Verify that the system prevents adding a new product with a name that already exists in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to add a product.  The add product page is accessible.  A product with the name ExistingProduct123 already exists in the database. |
| Test Case Input | Product Name: ExistingProduct123  Price: 299.99  Category: Electronics  Description: Duplicate product test |
| Test Case Expected Output | The product addition process is halted, and an error message is displayed indicating that the product name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the add product page.  Enter ExistingProduct123 in the product name field.  Enter 299.99 in the price field.  Enter Duplicate product test in the description field.  Click the "Add Product" button.  Verify that an error message is displayed indicating that the product name already exists and the product is not added. |
| Default Value Preverving | All fields are empty by default. |

1. Edit product test

|  |  |
| --- | --- |
| Test Case Number | 6.1 |
| Test Case Name | Edit product with valid information |
| Test Case Description | Verify that a product's information can be successfully edited with valid data. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a product.  The product to be edited exists in the database (e.g., Martin inc). |
| Test Case Input | Original Product Name: Martin inc  New Product Name: UpdatedProduct123  New Price: 349.99  New Description: Updated high-quality product |
| Test Case Expected Output | The product information is successfully updated, and a confirmation message is displayed. The product list shows the updated product information. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the product list page.  Search for Martin inc in the product list.  Click the "Edit" button for Martin inc.  Update the product name to UpdatedProduct123.  Update the price to 349.99. |
| Default Value Preverving | Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update.  Verify that the product list shows the updated product information. |

|  |  |
| --- | --- |
| Test Case Number | 6.2 |
| Test Case Name | Edit product with special characters |
| Test Case Description | Verify that the system allows editing a product's information with special characters. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a product.  The product to be edited exists in the database (e.g., Mays ltd). |
| Test Case Input | Original Product Name: Mays ltd  New Product Name: Updated!@#$%^&\*Product123  New Price: 299.99  New Description: Updated product with special characters in the name |
| Test Case Expected Output | The product information is successfully updated, and a confirmation message is displayed. The product list shows the updated product information with special characters. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the product list page.  Search for Mays ltd in the product list.  Click the "Edit" button for Mays ltd  Update the product name to Updated!@#$%^&\*Product123.  Update the price to 299.99.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update.  Verify that the product list shows the updated product information with special characters. |
| Default Value Preverving | All fields are pre-populated with the current product information. |

|  |  |
| --- | --- |
| Test Case Number | 6.3 |
| Test Case Name | Edit product with existing product name |
| Test Case Description | Verify that the system prevents updating a product's name to an existing product name in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a product.  The product to be edited exists in the database (e.g., Davis inc).  Another product with the name Meadows inc already exists in the database. |
| Test Case Input | Original Product Name: Davis inc  New Product Name: Meadows inc  Price: 299.99  Description: Attempt to update product name to an existing name |
| Test Case Expected Output | The product update process is halted, and an error message is displayed indicating that the product name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the product list page.  Search for Davis inc in the product list.  Click the "Edit" button for Davis inc.  Update the product name to Meadows inc  Update the description to Attempt to update product name to an existing name. |
| Default Value Preverving | All fields are pre-populated with the current product information. |

1. Create new category test

|  |  |
| --- | --- |
| Test Case Number | 7.1 |
| Test Case Name | Create category with valid information |
| Test Case Description | Verify that a new category can be successfully created with valid information. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a category. |
| Test Case Input | Category Name: Electronics  Description: Category for electronic products including gadgets and devices. |
| Test Case Expected Output | The new category is successfully created, and a confirmation message is displayed. The category list shows the newly created category. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Click the "Add Category" button.  Enter the category name Electronics.  Enter the description Category for electronic products including gadgets and devices..  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful creation of the category.  Verify that the category list shows the newly created category Electronics. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 7.2 |
| Test Case Name | Create category with special characters |
| Test Case Description | Verify that a new category can be successfully created with special characters in the name and description. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a category. |
| Test Case Input | Category Name: Electronics!@#$%^&\*()  Description: Category for electronic products including gadgets and devices!@#$%^&\*(). |
| Test Case Expected Output | The new category is successfully created, and a confirmation message is displayed. The category list shows the newly created category with special characters. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Click the "Add Category" button.  Enter the category name Electronics!@#$%^&\*().  Enter the description Category for electronic products including gadgets and devices!@#$%^&\*()..  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful creation of the category.  Verify that the category list shows the newly created category Electronics!@#$%^&\*(). |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 7.3 |
| Test Case Name | Create category with existing category name |
| Test Case Description | Verify that the system prevents the creation of a new category with a name that already exists in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a category.  A category with the name ExistingCategory already exists in the database. |
| Test Case Input | Category Name: ExistingCategory  Description: Attempt to create a new category with an existing name. |
| Test Case Expected Output | The category creation process is halted, and an error message is displayed indicating that the category name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Click the "Add Category" button.  Enter the category name ExistingCategory.  Enter the description Attempt to create a new category with an existing name..  Click the "Save" button.  Verify that an error message is displayed indicating that the category name already exists and the category is not created. |
| Default Value Preverving | All fields are empty by default |

1. Edit category test

|  |  |
| --- | --- |
| Test Case Number | 8.1 |
| Test Case Name | Edit category with valid information |
| Test Case Description | Verify that an existing category can be successfully edited with valid information. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a category.  The category to be edited exists in the database (e.g., OldCategoryName). |
| Test Case Input | Original Category Name: Phelps inc  New Category Name: UpdatedCategoryName  New Description: Updated description for the category. |
| Test Case Expected Output | The category information is successfully updated, and a confirmation message is displayed. The category list shows the updated category information. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Search for Phelps inc in the category list.  Click the "Edit" button for Phelps inc.  Update the category name to UpdatedCategoryName.  Update the description to Updated description for the category.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update of the category.  Verify that the category list shows the updated category UpdatedCategoryName. |
| Default Value Preverving | All fields are pre-populated with the current category information. |

|  |  |
| --- | --- |
| Test Case Number | 8.2 |
| Test Case Name | Edit category with special characters |
| Test Case Description | The application is accessible.  The user is logged in with sufficient permissions to edit a category.  The category to be edited exists in the database (e.g., Mullen inc). |
| Preconditions | Original Category Name: Mullen inc  New Category Name: UpdatedCategory!@#$%^&\*()  New Description: Description with special characters!@#$%^&\*(). |
| Test Case Input | The category information is successfully updated, and a confirmation message is displayed. The category list shows the updated category with special characters. |
| Test Case Expected Output | The category information is successfully updated, and a confirmation message is displayed. The category list shows the updated category with special characters. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Search for Mullen inc in the category list.  Click the "Edit" button for Mullen inc.  Update the category name to UpdatedCategory!@#$%^&\*().  Update the description to Description with special characters!@#$%^&\*().  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update of the category.  Verify that the category list shows the updated category UpdatedCategory!@#$%^&\*(). |
| Default Value Preverving | All fields are pre-populated with the current category information. |

|  |  |
| --- | --- |
| Test Case Number | 8.3 |
| Test Case Name | Edit category with existing category name |
| Test Case Description | Verify that the system prevents updating a category with a name that already exists in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a category.  The category to be edited exists in the database (e.g., Lamp pnc).  Another category with the name ExistingCategory already exists in the database. |
| Test Case Input | Original Category Name: Lamp pnc  New Category Name: Dillon inc (an existing category name)  New Description: Attempt to update with an existing category name. |
| Test Case Expected Output | The category update process is halted, and an error message is displayed indicating that the new category name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the category management page.  Search for Lamp pnc in the category list.  Click the "Edit" button for Lamp pnc.  Update the category name to Dillon inc.  Update the description to Attempt to update with an existing category name.  Click the "Save" button.  Verify that an error message is displayed indicating that the category name already exists and the category is not updated. |
| Default Value Preverving | All fields are pre-populated with the current category information. |

1. Create new warehouse test

|  |  |
| --- | --- |
| Test Case Number | 9.1 |
| Test Case Name | Create warehouse with valid information |
| Test Case Description | Verify that a new warehouse can be successfully created with valid information |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a warehouse. |
| Test Case Input | Warehouse Name: Central Depot  Warehouse Address: 456 Depot Lane  District: North District  City: Metropolis  Country: Countryland  Postal Code: 67890 |
| Test Case Expected Output | The new warehouse is successfully created, and a confirmation message is displayed. The warehouse list shows the newly created warehouse. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Click the "Add Warehouse" button.  Enter the warehouse name Central Depot.  Enter the warehouse address 456 Depot Lane.  Enter the district North District.  Enter the city Metropolis.  Enter the country Countryland.  Enter the postal code 67890.  Click the "Save" button. Verify that a confirmation message is displayed indicating successful creation of the warehouse.  Verify that the warehouse list shows the newly created warehouse Main Warehouse. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 9.2 |
| Test Case Name | Create warehouse with special characters |
| Test Case Description | Verify that a new warehouse can be successfully created with special characters |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a warehouse. |
| Test Case Input | Warehouse Name: Warehouse!@#2024  Warehouse Address: 789 Special Blvd, Apt #12  District: West-End!@#  City: Capital-City  Country: Land&%$  Postal Code: 1234!@ |
| Test Case Expected Output | The new warehouse is successfully created, and a confirmation message is displayed. The warehouse list shows the newly created warehouse with all provided details, including special characters. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Click the "Add Warehouse" button.  Enter the warehouse name Warehouse!@#2024.  Enter the warehouse address 789 Special Blvd, Apt #12.  Enter the district West-End!@#.  Enter the city Capital-City.  Enter the country Land&%$.  Enter the postal code 1234!@.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful creation of the warehouse.  Verify that the warehouse list shows the newly created warehouse Warehouse!@#2024 with the correct address, district, city, country, and postal code, including special characters. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 9.3 |
| Test Case Name | Create warehouse with existing warehouse name |
| Test Case Description | Verify that the system prevents the creation of a new warehouse with a name that already exists in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a warehouse.  A warehouse with the name Ramos inc already exists in the database. |
| Test Case Input | Warehouse Name: Ramos  Warehouse Address: 123 Existing St  District: Central District  City: Newtown  Country: Countryland  Postal Code: 54321 |
| Test Case Expected Output | The warehouse creation process is halted, and an error message is displayed indicating that the warehouse name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Click the "Add Warehouse" button.  Enter the warehouse name Ramos.  Enter the warehouse address 123 Existing St.  Enter the district Central District.  Enter the city Newtown.  Enter the country Countryland.  Enter the postal code 54321.  Click the "Save" button.  Verify that an error message is displayed indicating that the warehouse name already exists and the warehouse is not created. |
| Default Value Preverving | All fields are empty by default. |

1. Edit warehouse test

|  |  |
| --- | --- |
| Test Case Number | 10.1 |
| Test Case Name | Edit warehouse with valid information |
| Test Case Description | Verify that the system allows editing of a warehouse with valid information and updates the warehouse details successfully. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a warehouse.  A warehouse with the name Pulm pnc already exists in the database. |
| Test Case Input | Original Warehouse Name: Pulm pnc  New Warehouse Name: UpdatedWarehouse  New Warehouse Address: 321 Updated Ave  New District: South District  New City: Newville  New Country: Newland  New Postal Code: 67890 |
| Test Case Expected Output | The warehouse is successfully updated with the new details, and a confirmation message is displayed. The warehouse list shows the updated warehouse with the new information. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Search for Pulm pnc in the warehouse list.  Click the "Edit" button for Pulm pnc.  Update the warehouse name to UpdatedWarehouse.  Update the warehouse address to 321 Updated Ave.  Update the district to South District.  Update the city to Newville.  Update the country to Newland.  Update the postal code to 67890.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update of the warehouse.  Verify that the warehouse list shows the updated |
| Default Value Preverving | All fields are pre-populated with the current warehouse information. |

|  |  |
| --- | --- |
| Test Case Number | 10.2 |
| Test Case Name | Edit warehouse with special characters |
| Test Case Description | Verify that the system allows editing of a warehouse with special characters in the warehouse details and updates the warehouse information successfully. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a warehouse.  A warehouse with the name Cook-banks already exists in the database. |
| Test Case Input | Original Warehouse Name: Cook-banks  New Warehouse Name: Warehouse!@#2024  New Warehouse Address: 123 Special Blvd, Apt #45  New District: East-End!@#  New City: Metro-City  New Country: Land&%$  New Postal Code: 9876!@ |
| Test Case Expected Output | The warehouse is successfully updated with the new details, including special characters, and a confirmation message is displayed. The warehouse list shows the updated warehouse with all provided details, including special characters. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Search for Cook-banks in the warehouse list.  Click the "Edit" button for Cook-banks  Update the warehouse name to Warehouse!@#2024.  Update the warehouse address to 123 Special Blvd, Apt #45.  Update the district to East-End!@#.  Update the city to Metro-City.  Update the country to Land&%$.  Update the postal code to 9876!@.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update of the warehouse.  Verify that the warehouse list shows the updated |
| Default Value Preverving | All fields are pre-populated with the current warehouse information. |

|  |  |
| --- | --- |
| Test Case Number | 10.3 |
| Test Case Name | Edit warehouse with existing warehouse name |
| Test Case Description | Verify that the system prevents updating a warehouse to a name that already exists in the database. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a warehouse.  A warehouse with the name Ward already exists in the database.  A warehouse with the name Sons also exists in the database. |
| Test Case Input | Original Warehouse Name: Ward  New Warehouse Name: Sons (which already exists)  New Warehouse Address: 456 Updated St  New District: North District  New City: Oldtown  New Country: Oldland  New Postal Code: 12345 |
| Test Case Expected Output | The system should prevent the update due to the duplicate warehouse name and display an appropriate error message indicating that the new warehouse name already exists. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the warehouse management page.  Search for Ward in the warehouse list.  Click the "Edit" button for Ward.  Update the warehouse name to Sons.  Update the warehouse address to 456 Updated St.  Update the district to North District.  Update the city to Oldtown.  Update the country to Oldland.  Update the postal code to 12345.  Click the "Save" button.  Verify that an error message is displayed indicating that the warehouse name WarehouseB already exists and the warehouse is not updated.  Verify that the warehouse list still shows WarehouseA with its original details. |
| Default Value Preverving | All fields are pre-populated with the current warehouse information. |

1. Create new shipment test

|  |  |
| --- | --- |
| Test Case Number | 11.1 |
| Test Case Name | Create shipment with valid information |
| Test Case Description | Verify that the system allows the creation of a shipment with valid information and successfully processes the shipment. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a shipment.  Valid data for all required fields is available. |
| Test Case Input | Warehouse: Jounsh  From: Main Storage  To: Vnw  Status: Processing  Username: Current username  Description: Shipment of 100 units of ProductY to CustomerLocation.  Supplier Name: Lpoex |
| Test Case Expected Output | The shipment is successfully created with the provided details, and a confirmation message is displayed. The shipment list shows the new shipment with all the provided details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the shipment management page.  Click the "Create Shipment" button.  Select the warehouse Jounsh from the dropdown.  Select the status as Processing.  Ensure the username is automatically populated with the current username.  Enter the description as Shipment of 100 units of ProductY to CustomerLocation.  Select the supplier name Lpoex from the dropdown.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful creation of the shipment.  Verify that the shipment list shows the new shipment with the correct details. |
| Default Value Preverving | All fields are empty by default except username and date |

|  |  |
| --- | --- |
| Test Case Number | 11.2 |
| Test Case Name | Create shipment with invalid information |
| Test Case Description | Verify that the system correctly handles the creation of a shipment when invalid information is provided and displays appropriate error messages. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a shipment. |
| Test Case Input | Warehouse: InvalidWarehouse (does not exist)  Date:Current date (automatically populated)  Status: UnknownStatus (not a valid status)  Username: Current username (automatically populated)  Description: Shipment of negative quantity units of ProductZ.  Supplier Name: NonExistentSupplier (does not exist) |
| Test Case Expected Output | The system should display error messages for each invalid input, indicating what needs to be corrected. The shipment should not be created. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the shipment management page.  Click the "Create Inbound Shipment" button.  Click the "Save" button.  Verify that the shipment is not created and the user remains on the create shipment page. |
| Default Value Preverving | All fields are empty by default except username and date |

1. Edit shipment test

|  |  |
| --- | --- |
| Test Case Number | 12.1 |
| Test Case Name | Edit shipment with valid information |
| Test Case Description | Verify that the system allows the editing of a shipment with valid information and successfully updates the shipment details. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a shipment.  A shipment with the ID 5 already exists in the database. |
| Test Case Input | Warehouse: Cook\_shiva  Date:Current date (automatically populated)  Status: Completed  Username: Current username (automatically populated )  Description: Updated shipment of 100 units of ProductZ to NewCustomerLocation.  Supplier Name: Sharp |
| Test Case Expected Output | The shipment is successfully updated with the provided details, and a confirmation message is displayed. The shipment list shows the updated shipment with all the provided details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the shipment management page.  Search for the shipment with ID 5.  Click the "Edit" button for the shipment with ID 5.  Select the warehouse Cook-Shiva from the dropdown.  Select the status as Completed.  Verify that the username is automatically populated with the current username.  Enter the description as Updated shipment of 100 units of ProductZ to NewCustomerLocation.  Select the supplier name Sharp from the dropdown.  Click the "Save" button.  Verify that a confirmation message is displayed indicating successful update of the shipment.  Verify that the shipment list shows the updated shipment with the new details. |
| Default Value Preverving | All fields are empty by default except username and date |

|  |  |
| --- | --- |
| Test Case Number | 12.2 |
| Test Case Name | Edit shipment with invalid information |
| Test Case Description | Verify that the system properly handles the editing of a shipment with invalid information and displays appropriate error messages. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit a shipment.  A shipment with the ID 125 already exists in the database. |
| Test Case Input | Warehouse: Blank  Date:Current date  Status: Blank  Username: Current username  Description: Current Description  Supplier Name: Current Supplier Name |
| Test Case Expected Output | The system should validate the input, reject the blank data, and display appropriate error messages indicating that all required fields must be filled. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the shipment management page.  Search for the shipment with ID 5.  Click the "Edit" button for the shipment with ID 5.  Leave the status blank  Click the "Save" button.  Verify that the shipment is not updated and the user remains on the edit shipment page.. |
| Default Value Preverving | All fields are empty by default except username and date |

1. Create new inventory audit test

|  |  |
| --- | --- |
| Test Case Number | 13.1 |
| Test Case Name | Create inventory audit with valid information |
| Test Case Description | Verify that the system allows the creation of an inventory audit with valid information and successfully adds the audit details to the inventory audit list. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create an inventory audit.  A valid warehouse exists in the database. |
| Test Case Input | Warehouse: TestWaer  Date: 2024-08-01 q  Description: Audit of all electronic items.  User: Admn  Status: Processing |
| Test Case Expected Output | The inventory audit is successfully created with the provided details, and a confirmation message is displayed. The audit list shows the new audit with all the provided details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the inventory audit management page.  Click the "Create Inventory Audit" button.  Select the warehouse TestWaer from the dropdown.  Enter the date as 2024-08-01.  Enter the description as Audit of all electronic items.  Verify that the username is automatically populated with the current username.  Select the status as Processing  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful creation of the inventory audit.  Verify that the inventory audit list shows the new audit with the correct details. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 13.2 |
| Test Case Name | Create inventory audit with invalid information |
| Test Case Description | Verify that the system properly handles the creation of an inventory audit with invalid information and displays appropriate error messages. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create an inventory audit. |
| Test Case Input | Warehouse: !@#$%^&\*()  Date: (blank)  Description: (blank)  User: #@#@#  Status: (blank) |
| Test Case Expected Output | The system should validate the input, reject the invalid data, and display appropriate error messages indicating that all required fields must be filled with valid information. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the inventory audit management page.  Click the "Create Inventory Audit" button.  Enter invalid special characters !@#$%^&\*() in the warehouse field.  Leave the date field blank.  Leave the description field blank.  Enter invalid special characters !@#$%^&\*() in the user field  Leave the status field blank.  Click the "Save" button.  Verify that the inventory audit is not created and the user remains on the create inventory audit page. |
| Default Value Preverving | All fields are empty by default |

1. Edit inventory audit test

|  |  |
| --- | --- |
| Test Case Number | 14.1 |
| Test Case Name | Create inventory audit with valid information |
| Test Case Description |  |
| Preconditions |  |
| Test Case Input |  |
| Test Case Expected Output |  |
| Test Case Steps |  |
| Default Value Preverving |  |

|  |  |
| --- | --- |
| Test Case Number | 14.2 |
| Test Case Name | Create inventory audit with invalid information |
| Test Case Description |  |
| Preconditions |  |
| Test Case Input |  |
| Test Case Expected Output |  |
| Test Case Steps |  |
| Default Value Preverving |  |

1. Edit inventory audit test

|  |  |
| --- | --- |
| Test Case Number | 15.1 |
| Test Case Name | Edit inventory audit with valid information |
| Test Case Description | Verify that the system allows editing an existing inventory audit record with valid information. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit inventory audit records.  An existing inventory audit record is available in the system. |
| Test Case Input | Warehouse: Moss  Date: Current date  Description: Updated audit description for accuracy check.  User: Rachel  Status: Processing |
| Test Case Expected Output | The system should successfully update the inventory audit record with the provided information, and a confirmation message should be displayed. The inventory audit list should reflect the updated details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the inventory audit management page.  Locate the existing inventory audit record to be edited.  Click the "Edit" button for the selected inventory audit.  Update the warehouse to Moss.  Update the description to Updated audit description for accuracy check.  Update the status to Processing.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful update of the inventory audit.  Verify that the inventory audit list shows the updated details for the edited record. |
| Default Value Preverving | All fields are empty by default except Warehouse,date,user. |

|  |  |
| --- | --- |
| Test Case Number | 15.2 |
| Test Case Name | Edit inventory audit with invalid information |
| Test Case Description | Verify that the system handles invalid information appropriately when editing an existing inventory audit record. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit inventory audit records.  An existing inventory audit record is available in the system. |
| Test Case Input | Warehouse: Zhang group  Date: Current date  Description: (blank)  User: (blank)  Status: unknown (invalid status) |
| Test Case Expected Output | The system should display error messages for each invalid input and prevent the inventory audit record from being updated. The original details should remain unchanged. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the inventory audit management page.  Locate the existing inventory audit record to be edited.  Click the "Edit" button for the selected inventory audit.  Update the status to blank value.  Click the "Save" button.  Verify that error messages are displayed for each invalid input.  Verify that the inventory audit record is not updated and the original details remain unchanged. |
| Default Value Preverving | All fields are empty by default except Warehouse,date,user. |

1. Create new suppliers test

|  |  |
| --- | --- |
| Test Case Number | 16.1 |
| Test Case Name | Create supplier with valid information |
| Test Case Description | Verify that the system allows the creation of a new supplier with valid information and successfully saves the supplier details. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a new supplier. |
| Test Case Input | Supplier Name: SupplierTesq  Address: 123 Main Street, City, Country  Website: www.suppliert.com  Email: contact@suppliert.com  Phone Number: 1234567890  Description: Leading supplier of electronic components. |
| Test Case Expected Output | The supplier is successfully created with the provided details, and a confirmation message is displayed. The supplier list shows the new supplier with all the provided details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Click the "Create New Supplier" button.  Enter SupplierTesq in the supplier name field.  Enter 123 Main Street, City, Country in the address field.  Enter www. suppliert.com in the website field.  Enter contact@ suppliert.com in the email field.  Enter 1234567890 in the phone number field.  Enter Leading supplier of electronic components. in the description field.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful creation of the supplier.  Verify that the supplier list shows the new supplier with the correct details. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 16.2 |
| Test Case Name | Create supplier with existing supplier name |
| Test Case Description | Verify that the system allows the creation of a new supplier with an existing supplier name |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a new supplier.  A supplier with the name SupplierX already exists in the database. |
| Test Case Input | Supplier Name: SupplierX  Address: 456 Another Street, City, Country  Website: www.newsupplier.com  Email: info@newsupplier.com  Phone Number: 0987654321  Description: New supplier of office supplies. |
| Test Case Expected Output | The system should allow the creation of a new supplier with the same |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Click the "Create New Supplier" button.  Enter SupplierX in the supplier name field (same as an existing supplier).  Enter 456 Another Street, City, Country in the address field.  Enter www.newsupplier.com in the website field.  Enter info@newsupplier.com in the email field.  Enter 0987654321 in the phone number field.  Enter New supplier of office supplies. in the description field.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful creation of the supplier.  Verify that the supplier list shows the new supplier with the correct details. |
| Default Value Preverving | All fields are empty by default |

|  |  |
| --- | --- |
| Test Case Number | 16.3 |
| Test Case Name | Create supplier with special characters |
| Test Case Description | Verify that the system allows the creation of a new supplier with special characters in the supplier name and other fields, and that it handles the data correctly. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to create a new supplier. |
| Test Case Input | Supplier Name: !@#SpecialSupplier$%^  Address: 789 Special Blvd, City, Country  Website: www.specialsupplier.com  Email: info@specialsupplier.com  Phone Number: 9876543210  Description: Supplier with special characters in name. |
| Test Case Expected Output | The system should allow the creation of a supplier with special characters in the fields and handle the data correctly. A confirmation message should be displayed, and the supplier list should show the new supplier with the correct details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Click the "Create New Supplier" button.  Enter !@#SpecialSupplier$%^ in the supplier name field.  Enter 789 Special Blvd, City, Country in the address field.  Enter www.specialsupplier.com in the website field.  Enter info@specialsupplier.com in the email field.  Enter 9876543210 in the phone number field.  Enter Supplier with special characters in name. in the description field.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful creation of the supplier.  Verify that the supplier list shows the new supplier with the correct details. |
| Default Value Preverving | All fields are empty by default |

1. Edit supplier test

|  |  |
| --- | --- |
| Test Case Number | 17.1 |
| Test Case Name | Edit supplier with valid information |
| Test Case Description | Verify that the system allows the editing of an existing supplier with valid information. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit supplier details.  An existing supplier (e.g., SupplierX) is available in the system. |
| Test Case Input | Supplier Name: SupplierY  Address: 456 New Street, New City, New Country  Description: Updated description for SupplierY. |
| Test Case Expected Output | The system should successfully update the supplier details with the provided information, and a confirmation message should be displayed. The supplier list should reflect the updated details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Locate the existing supplier (e.g., SupplierX) in the supplier list.  Click the "Edit" button for the supplier.  Update the supplier name to SupplierY.  Update the address to 456 New Street, New City, New Country.  Update the description to Updated description for SupplierY.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful update of the supplier.  Verify that the supplier list shows the updated details for SupplierY. |
| Default Value Preverving | All fields are pre-populated with the current supplier information. |

|  |  |
| --- | --- |
| Test Case Number | 17.2 |
| Test Case Name | Edit supplier with existing supplier name |
| Test Case Description | Verify that the system allows editing an existing supplier's details. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit supplier details.  At least two suppliers exist in the system, with one of them having the name that will be reused. |
| Test Case Input | Existing Supplier Name: Lewis  New Supplier Name: Lee inc (same as another existing supplier) |
| Test Case Expected Output | The system should allow the update of the supplier’s name to an existing supplier name, either by allowing the duplicate name or handling it in a way specified by the system.A confirmation message should be displayed, and the supplier list should reflect the updated details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Locate the existing supplier with the name Lewis in the supplier list.  Click the "Edit" button for Lewis.  Update the supplier name to Lee inc (same as another existing supplier).  Update the address to 123 Updated Address, New City, New Country.  Update the description to Updated description for Lee inc.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful update of the supplier.  Verify that the supplier list shows the updated details for Lewis, and ensure the system handles the duplicate name as specified. |
| Default Value Preverving | All fields are pre-filled with the current user's information. |

|  |  |
| --- | --- |
| Test Case Number | 17.3 |
| Test Case Name | Edit supplier with special characters |
| Test Case Description | Verify that the system allows editing an existing supplier's details. |
| Preconditions | The application is accessible.  The user is logged in with sufficient permissions to edit supplier details.  An existing supplier record is available in the system. |
| Test Case Input | Existing Supplier Name: Martin  New Supplier Name: Supplier@#%&\*!  Address: 123 Main St., Suite #&\*!@  Description: Updated description with special characters @#%&\*!. |
| Test Case Expected Output | The system should successfully update the supplier details with special characters, and a confirmation message should be displayed. The supplier list should reflect the updated details. |
| Test Case Steps | Open the application.  Log in with valid credentials.  Navigate to the supplier management page.  Locate the existing supplier with the name SupplierA in the supplier list.  Click the "Edit" button for Martin.  Update the supplier name to Supplier@#%&\*!.  Update the address to 123 Main St., Suite #&\*!@.  Update the description to Updated description with special characters @#%&\*!.  Click the "Save" button.  Verify that a confirmation message is displayed indicating the successful update of the supplier.  Verify that the supplier list shows the updated details with special characters. |
| Default Value Preverving | All fields are pre-populated with the current Suppliers information |

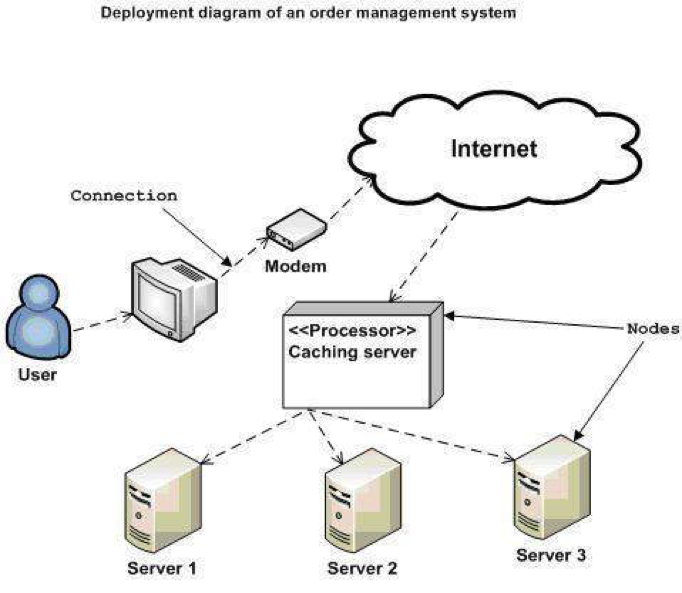
# Task Assign (to each team member)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Group Name | Project Name | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self assessment |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |

# Installation Instructions

< Consistently listing conventions when installing software (Detailed instructions with pictures and notes)>

1. Deployment Diagram



1. Installation steps
   * Database Install
   * Server install
   * Application install

# Appendix

Terms and abbreviations <if available>

<listing terms and abbreviations here>

References <if available >

<List of references here>

Some other issues <if available >

<Results, limitations, experiences, techniques, and other considerations when implementing a project>

# Document format

**Report corver:**

*The cover is printed in blue.*

*With the format as the first page of this document.*

**Paper Size:**

*The report is presented on A4 size paper (210 mm x 297 mm)*

**Top header**

*Left: Logo của VTC Academy*

*Right: Project Name*

*Font: Helvetica Neue (Light) or Arial*

*Font size: 12pt*

**Bottom header**

*Left: Class\_Name –Project\_Name*

*Right: Page\_Number*

*Font: Helvetica Neue (Light) or Arial*

*Font size: 12pt*

**Report Content:**

*Font: Helvetica Neue (Light) or Arial*

*Font size: 12pt*

*Minimum 20 pages*

**Page margin (for A4 size paper)**

*Top: 20 - 25 mm;*

*Bottom: 20 - 25 mm;*

*Left: 30 - 35 mm;*

*Right: 15 - 20 mm;*