**Problem Statement:**

This project will provide the ability for retail stores to manage inventory and order new stock.

**Background:**

Multiple retail stores will be connected to a warehouse’s database. The retail stores will be able to order products currently available for shipment from the warehouse and order the product to restock the shelves.

**Functional Requirements:**

1. Shall provide a login to determine store number and authenticate.

(a). Login name will be store number and passwords will be unique.

(b). System shall provide a forgot password option.

1. The user interface shall provide easily navigation to modify store’s inventory, order new stock and return defective items.

(a).The user interface shall have tabs on top to every page and a logout.

(b). The taps shall be uniform on all pages.

1. The store’s inventory shall be able to be narrowed down in a search and edited as needed.

(a).System shall provide buttons to select filters.

(b).System shall provide a edit button next to each item when pressed the field shall become editable.

1. User shall be able to return items to the warehouse.

(a).System shall provide a page to complete a return forum.

(b).System shall check forum for errors and then store in database for processing.

1. User shall be able to order new stock to retail store.

(a).System shall provide a page to complete a order new stock forum.

(b). System shall check forum for errors and then store in database for processing.

(c). System shall provide a page to check the status of orders.

**Implementation Requirements**

1. Must use Client-Server architecture with front-end accessed from a web browser.

a) Tomcat web server will be used because its architecture is the same for both personal web server and standalone web server.

2. Must be independent of the platform used.

a) A combination of Java Server Pages, Java Servlets and JavaBeans will be used for server logic.

3. Must be easily portable to any database that supports SQL.

4. Must use Model-View-Controller (MVC) architecture to provide loose-coupling between data and view.

5. UI design and implementation must be loosely coupled.

a) Formatting specifications must be separated into a CSS file. In addition, separate source files will be used for the scripting in a web page.

6. System must be designed for easy extension to other types of inventory.

a) Current system will be implemented in two phases with complete testing after each phase.

**Security Requirements**

1. System shall close the session if the user logs off or closes the browser.

2. Database connection must be closed after each query and update.

**Use Cases**

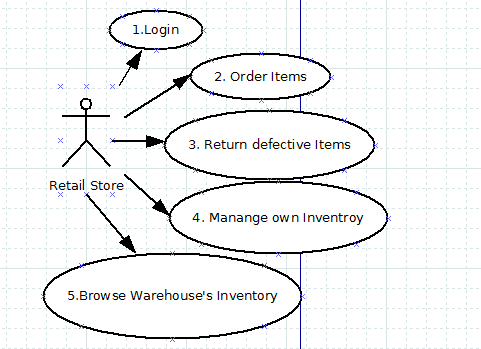
Use Case Search for Inventory

|  |  |
| --- | --- |
| Scenario Name: | Search for Inventory |
| Participating Actors: | Retail Store |
| Entry Condition: | Logged in |
| Flow of events: |  |
| Actor’s Actions:   1. Opens Location Inventory page. 2. Actor narrows down inventory displayed by brand, type and model with drop down boxes. 3. Actor clicks on search 4. Actor sorts by clicking on brand, type, model, most, or least. | System’s Response:   1. Displays Inventory page and all inventories for store in alphabetical order. 2. System displays inventory based on selections. 3. System sorts by putting most relevant on top. |
| Exception Handling: | 1. If no inventory exists for selected parameters system will print “No inventory”. |
| Exit Condition | Inventory displayed |

Use Case Place Orders

|  |  |
| --- | --- |
| Scenario Name: | Place Order |
| Participating Actors: | Retail Store |
| Entry Condition: | Logged in |
| Flow of events: |  |
| Actor’s Actions:   1. Opens place orders page.   3. Actor selects brand, type, model, quantity and optional details.  5. Actor clicks on “place order request”. | System’s Response:   1. Displays order page. 2. System records data plus date/time and store number in database to table “process”. 3. System displays “order sent” popup message. |
| Exception Handling: | 1. If user selection is invalid, system will display error and not process order. 2. If exact same order was placed 8 seconds ago system will inform user before processing. |
| Exit Condition | Product has been ordered. |

**Use Case Diagram**

****