**SDA Project Report: Clothing Store**

|  |  |  |
| --- | --- | --- |
| Group Member Roll Number | Group Member Section | Group Member Name |
| 21K-3225 | BCS-5K | Usman Rasheed |
| 21K-4924 | BCS-5K | Muneeb Ali |

# Proposal:

The project aims to replicate the experience of browsing and shopping on a clothing store website, like Outfitters or Khaadi. Users can create an account, log in, select clothing categories, choose specific items, indicate preferences such as size, color, and quantity, add chosen items to their cart, input shipping details for delivery, and ultimately view a receipt summarizing their selections.

The goal is to simulate a seamless shopping experience, mirroring the process individuals undergo when purchasing clothing online. This involves user registration, personalized item selection, cart management, and checkout with shipping information, ending with a receipt being generated.  
  
  
**The functions that the project would include would be:**

A **Registration page** for the user

A **Login page** for the user

A Search Bar to search for clothes by their name

* A **Homepage** to view all other pages
* A page for each **Category** available (Men, Women, Kids, Winter) that Displays the clothes
* Clothes should come in Different **Color Variations**
* A **Product Detail Page**, where you select the details of the clothe you decide to buy
* A **Cart** option where you can view what you’ve selected, You should be able to **Delete from Cart and Proceed to Checkout**
* A **Checkout Form** for inputting your user details
* A **Receipt** option to view your order details
* An **Owner Registration, Login, and Homepage.**
* Owner can **View User Detail, Receipt Details And Clothing Details**
* Owner can **Insert Clothes**
* Owner Can Delete Clothes

Below are the following Diagrams/Scenario’s:

* Fully Dressed Use Case Scenario
* Use Case Diagram
* Class Diagram
* Activity Diagram
* State Diagram
* Sequence Diagram
* Collaboration Diagram
* Non Functional Requirements  
    
    
  **FULLY DRESSED USE CASE FOR USER**

**Use case name:** “purchasing online clothes”.

**Primary Actor:** User

**Secondary Actor:** system and owner

**Preconditions:**   
-The user has a registered account on the online clothing store.  
-the user is logged into the system.

**Post-conditions:**  
-The user’s shopping car is empty.  
-The user’s account reflects the recent purchase.  
-The owner’s inventory is updated with the sold items.

**Triggers:**  
-The user initiates the purchase process by selecting the checkout option in the shopping cart.

**Main Success Scenarios:**  
-The user logs into the online clothing store.  
-The user browses through the available clothing categories and selects desired items.  
-The user adds selected items to the shopping cart.  
-The user reviews the shopping cart, makes modifications if necessary, and proceeds to checkout.  
-The user selects a payment method and provides necessary payment information.  
-The system processes the payment securely and generates a confirmation for the user.  
-The owner’s inventory is updated to reflect the purchased items.  
-The user receives an order confirmation with shipping details.

**Extensions:**  
-Invalid Login: If the user provides incorrect login credentials, an error message is displayed, and user is prompted to re-enter valid information.  
-Payment Failures: If the payment processing fails, the system give error and guide the user.

**Exception:**  
- Clothing website displays appropriate error messages if the user submits incorrect  
data, such as a wrong phone number or insufficient payment information.  
- The user goes back to the home screen if they decide to stop the shopping  
process at any screen.

**Quality:**  
-Security: The system must ensure the security of user data, especially during personal information.  
-Usability: The online store should provide a user-friendly interface for easy navigation.  
  
**FULLY DRESSES USE CASE FOR OWNER**

**Use Case Name:** Manage clothing store

**Primary Actor:** Owner

**Secondary Actor:**-System: Aims to provide the owner with tools to manage inventory effectively.

**Preconditions:**  
-The owner has a valid account and is logged into the system.  
-The owner has access rights to manage inventory.

**Post-conditions:**-The inventory database is updated with new products,.

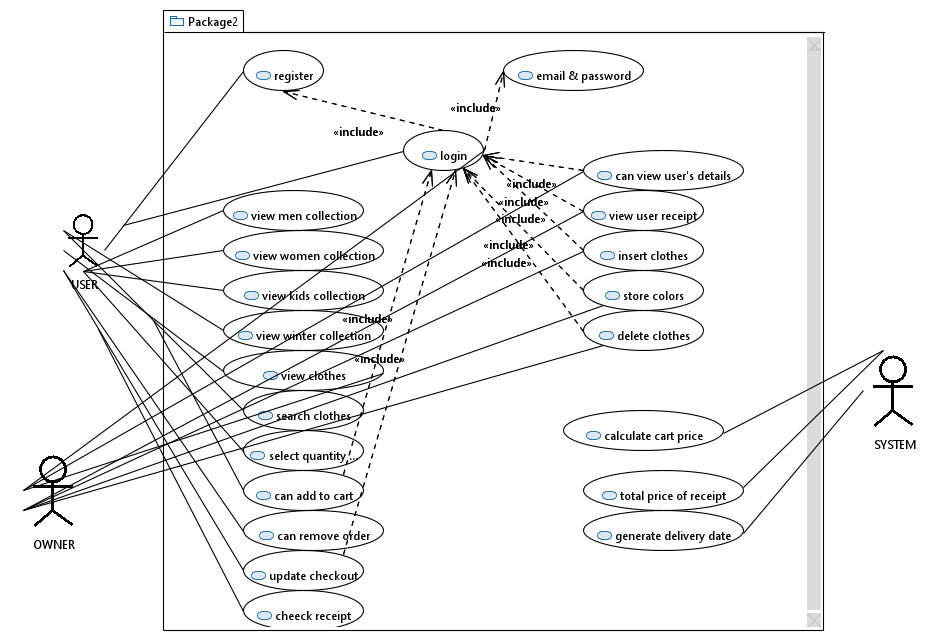
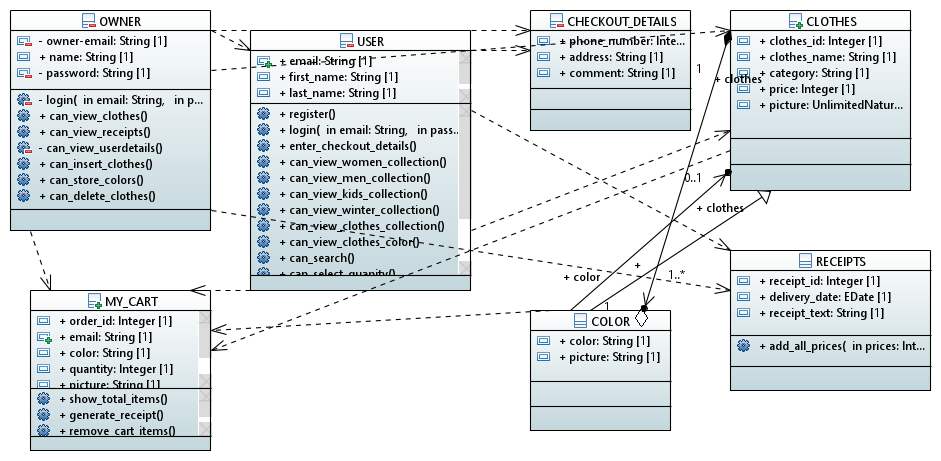
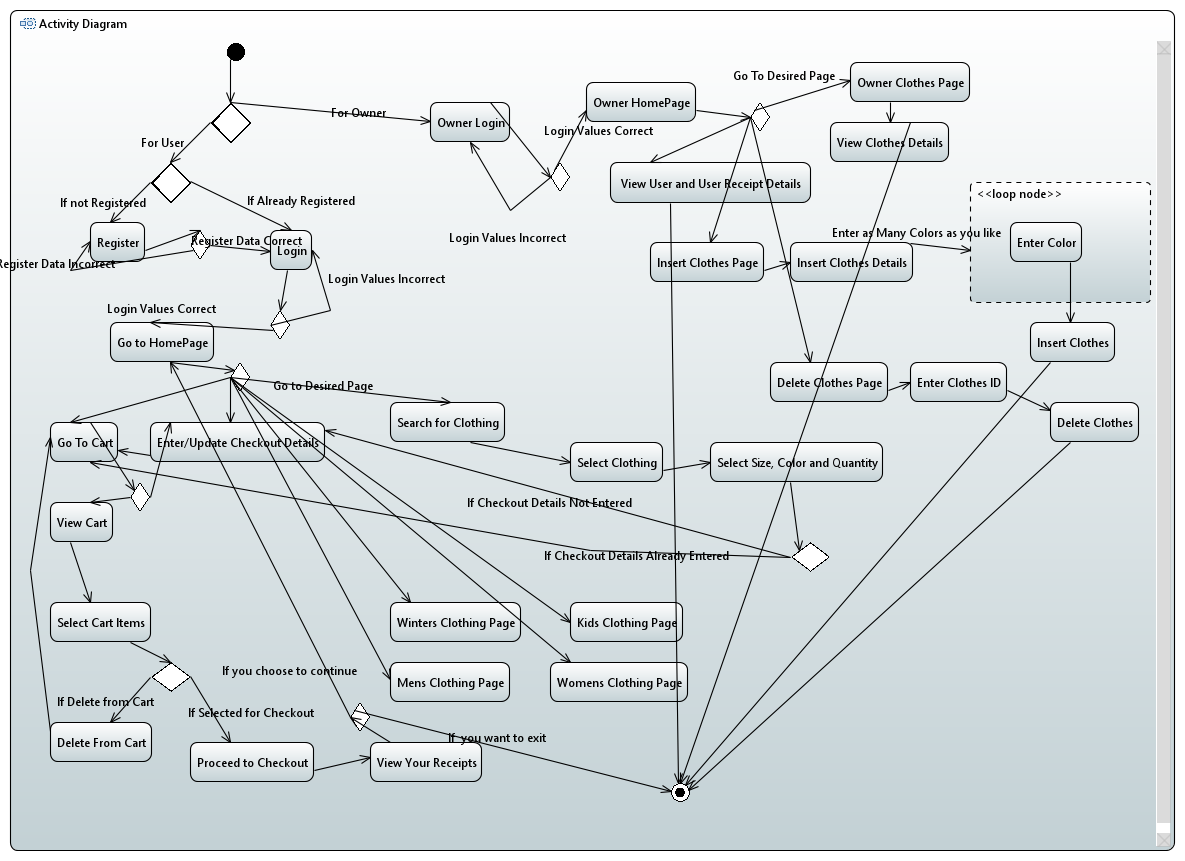
**-**Inventory changes accurately reflect the recent changes made by the owner.

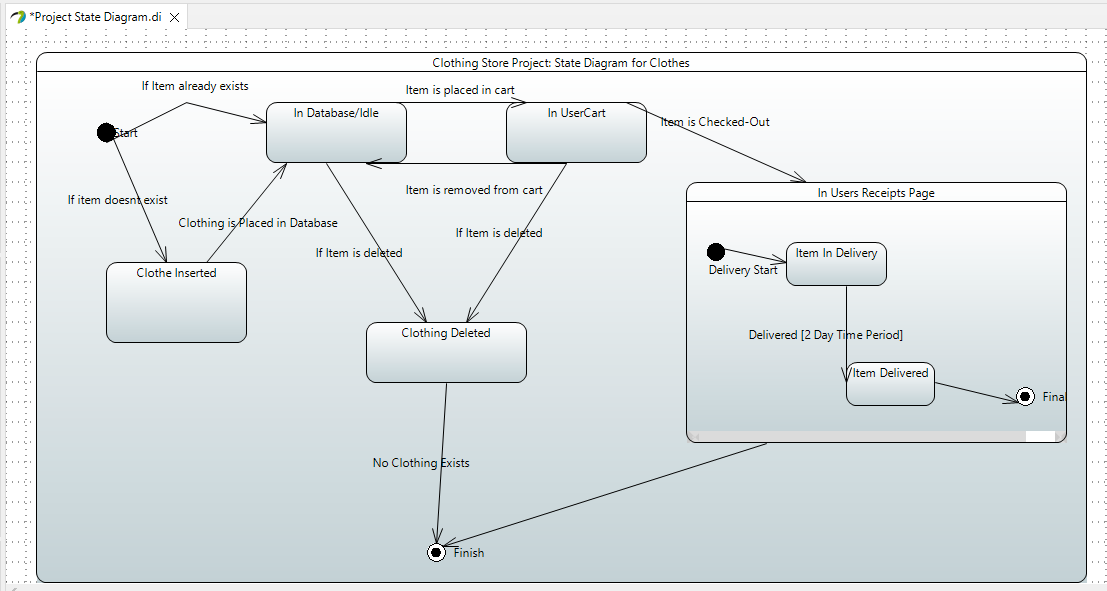
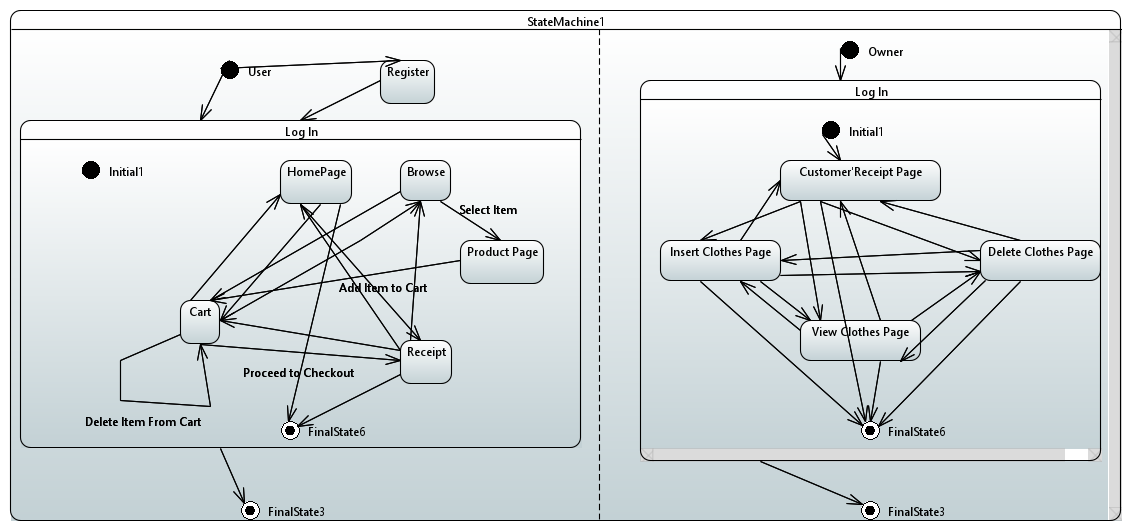
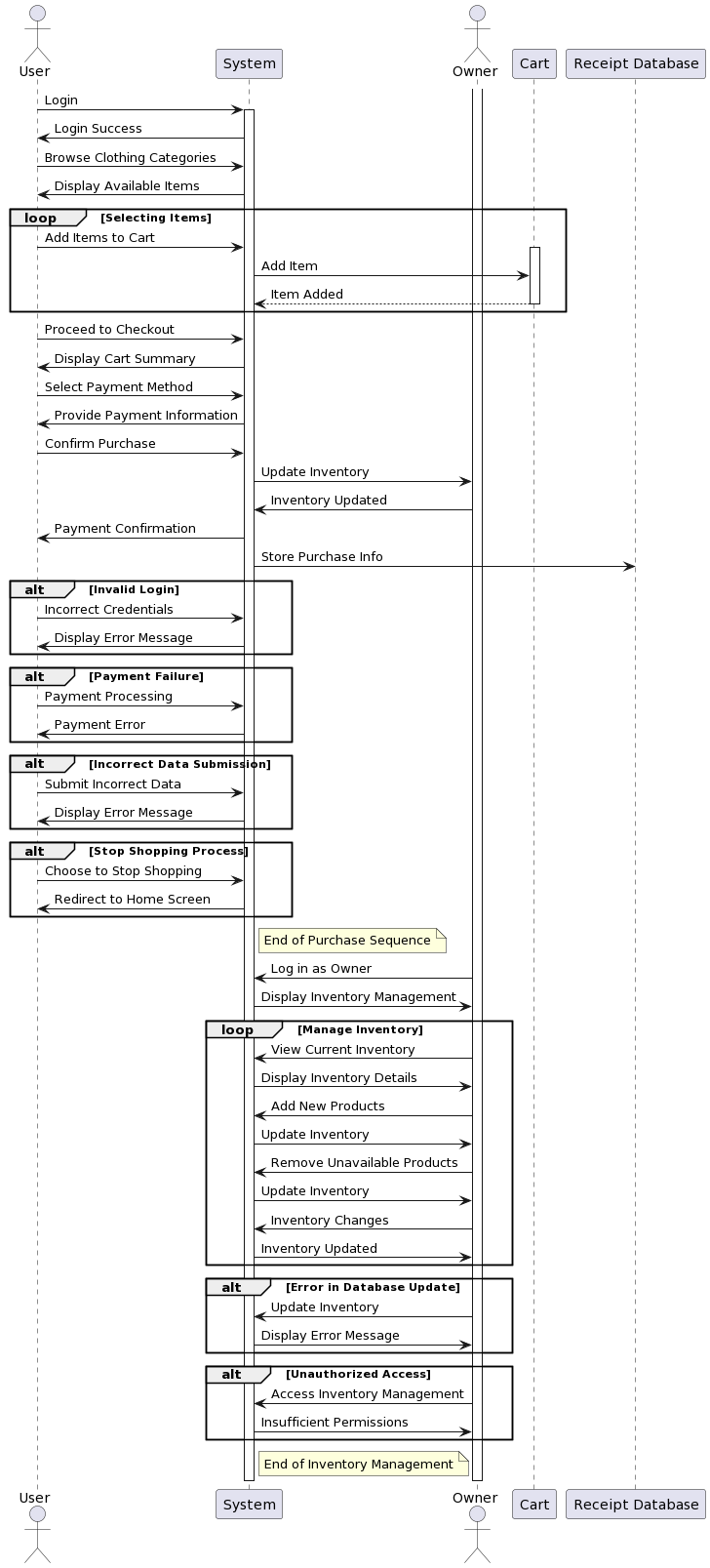
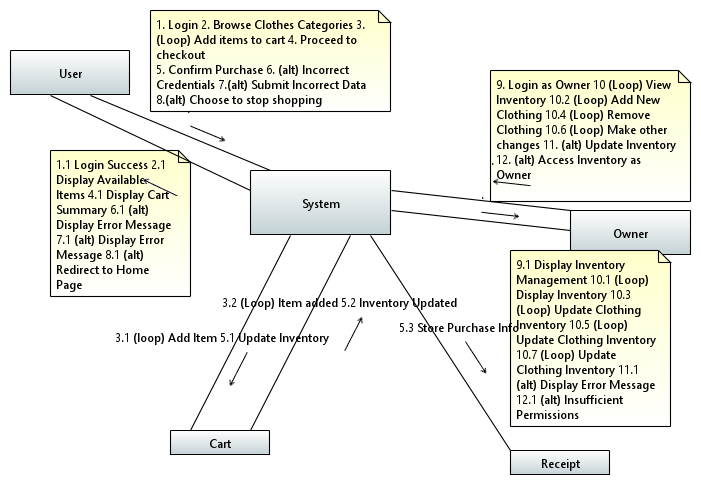
**Triggers:  
-**The owner initiates inventory management tasks by logging into the system and navigating to the inventory management section.

**Main Success Scenario:  
-**The owner logs into the online clothing store system.-The owner navigates to the inventory management section.-The owner views the current inventory status.-The owner adds new products to the inventory, specifying details such as product name, category, and quantity. **-**The owner can remove products that are no longer available or have been discontinued.-The system updates the inventory database to reflect the changes made by the owner.

**Extension:**-Error in Database Update: After attempting to add or update products, the system encounters an error in updating the inventory database.-Handling: Display an error message to the owner, indicating that the inventory update failed. Prompt the owner to try the operation again or contact technical support if the issue persists.

**Exceptions:  
-**Unauthorized Access: Exception Scenario: The owner does not have the necessary access rights to manage inventory. **-**Handling: Display an error message indicating insufficient permissions and prompt the owner to contact the system administrator for access.

**Quality:  
-**Usability: The inventory management interface should be user-friendly and intuitive for the owner.-Reliability: The system should reliably update the inventory database without data inconsistencies. **-**Security: Access to inventory management functions should be secure and restricted to authorized owners.  
  
Use Case Diagram:  
****  
  
Class Diagram:  
****  
  
Activity Diagram:  
****  
  
State Diagram:

(For Clothes)  
****  
  
For User and Owner:  
  
  
Sequence Diagram:  
  
Collaboration Diagram:  


Non Functional Requirements:  
  
**1. Security:**

User data encryption during transactions and storage.

Protection against potential threats like SQL injection, cross-site scripting (XSS), etc.

Compliance with industry standards

**2. Performance:**

Response time: Pages should load within 2-3 seconds to ensure a seamless user experience.

Scalability: Ability to handle increased traffic during peak times without significant performance degradation.

System availability: Ensure the website is available at least 99% of the time to avoid service interruptions.

**3. Usability:**

Intuitive user interface: Ensure the website is easy to navigate, with clear categories, search functionality, and a straightforward checkout process.

Responsive design: Website should be accessible and user-friendly across various devices (desktop, mobile, tablet).

**4. Reliability:**

Error handling: Provide descriptive error messages for users and detailed logs for system administrators.

Backup and recovery: Regular backups of the database to prevent data loss and quick recovery in case of system failure.

**5. Scalability:**

Ability to handle increased user loads during promotions or seasonal peaks without degradation in performance.

Easily expandable infrastructure to accommodate growth in product inventory and user base.

**6. Compliance:**

Adherence to legal regulations and data protection laws.

Compliance with industry standards for online transactions, ensuring secure and lawful processing of payments.

**7. Maintainability:**

Modular and documented code for easier maintenance and future updates.

Version control and change management processes to track and implement modifications without disrupting the system's stability.

**8. Interoperability:**

Compatibility with various web browsers (Chrome, Firefox, Safari, etc.) and operating systems (Windows, macOS, Linux).

Integration capabilities with third-party services (payment gateways, shipping providers, etc.) to streamline operations.