

Aqsa Shabbir

Fashion Ecommerce Website

## **table of contents**

- Page 3-8: customer problem statements and system requirements
- Page 9-15: functional requirement specification
- Page 16-17: system sequence diagram
- Page 18-20: activity diagram
- Page 21-23: user interface specification
- Page 24: project plan

## **Customer Problem Statement**

### **Problem Statement:**

As a fashion enthusiast and avid shopper, I recognize the increasing demand for online shopping experiences that are both visually appealing and user-friendly. However, many existing fashion e-commerce websites fail to provide customers with the engaging and seamless experience they expect. Customers often encounter difficulties in navigating through cluttered interfaces, finding relevant products, and completing the checkout process efficiently. Also, customers seek personalized recommendations and engaging content to enhance their shopping journey. There is a critical need for a well-designed and intuitive fashion e-commerce website that not only showcases our products effectively but also enhances the overall shopping experience for customers.

### **Glossary of Terms:**

**Fashion E-Commerce Website:** An online platform for selling fashion products, including clothing, accessories, and footwear, to customers worldwide.

**Product Catalog:** A comprehensive collection of fashion items available for purchase on the website.

**Shopping Cart:** A virtual cart where customers can add selected items for purchase before proceeding to checkout.

**Checkout Process:** The process through which customers finalize their purchases by providing shipping and payment information.

**Responsive Design:** Design principles aimed at ensuring optimal viewing and interaction experiences across various devices and screen sizes.

## **System Requirements**

### **Functional Requirements:**

No.	Priority Weight	Description
1	High	The website should have a visually appealing and intuitive interface.
2	High	Customers should be able to browse and filter products efficiently.

3	High	Product pages should include descriptions and high-quality images.
4	High	The shopping cart functionality should allow customers to add/remove items and adjust quantities.
5	High	The checkout process should be streamlined and user-friendly.
6	Low	The website should provide personalized recommendations based on customer preferences.
7	Medium	Customers should have access to engaging content such as fashion tips and trend guides.

#### Nonfunctional Requirements:

- Functionality: The website should be fully functional across different web browsers and devices.
- Usability: The website should be intuitive and easy to navigate, with clear calls to action.
- Reliability: The website should be stable and responsive, with minimal downtime.
- Performance: The website should load quickly and handle high volumes of traffic without slowdowns.
- Supportability: The website should be easy to maintain and update with new products and content.

#### User Interface Requirements:

Visually Appealing Interface: The website should have a modern and visually appealing design that reflects the brand's aesthetic.

Efficient Navigation: The navigation menu should be clear and easy to use, allowing customers to browse products by category.

Product Showcase: Product images should be large and high-quality, with the option to zoom in for closer inspection.

Seamless Checkout Process: The checkout process should be divided into clear steps, with progress indicators to guide customers through each stage.

Responsive Design: The website should be responsive and adapt seamlessly to different screen sizes and devices.

**Engaging Content:** The website should feature engaging content such as fashion articles, style guides, and customer testimonials to enhance the shopping experience.

## Plan of Work

- ~~Week 1-2: Define project scope and gather requirements.~~
- ~~Week 3-4: Set up development environment and create wireframes/mockups.~~
- Week 5-6: Develop frontend structure and styling using HTML/CSS. **(done some of this)**
- Week 7-8: Implement interactive features using JavaScript.
- Week 9-10: Populate product catalog and refine design elements.
- Week 11-12: Conduct comprehensive testing and optimization.
- Week 13-14: Finalize documentation and prepare for deployment.
- Week 15: Deploy website and conduct final presentation/demo.

I've set up my development environment using Visual Studio Code and am gathering imagery for the website. I am brainstorming how I would like each component of the website to look as well. I'm also configuring a GitHub repository for version control and researching unfamiliar aspects like GitHub integration.

## Header w/ Img

Get in Touch

Address

Info

Contact Us  
form

Google  
MAPS

Contact

## Header w/ Img

remove	Image	Product	Price	Qty	Submit
--------	-------	---------	-------	-----	--------

Apply coupon

Cart total

Checkout

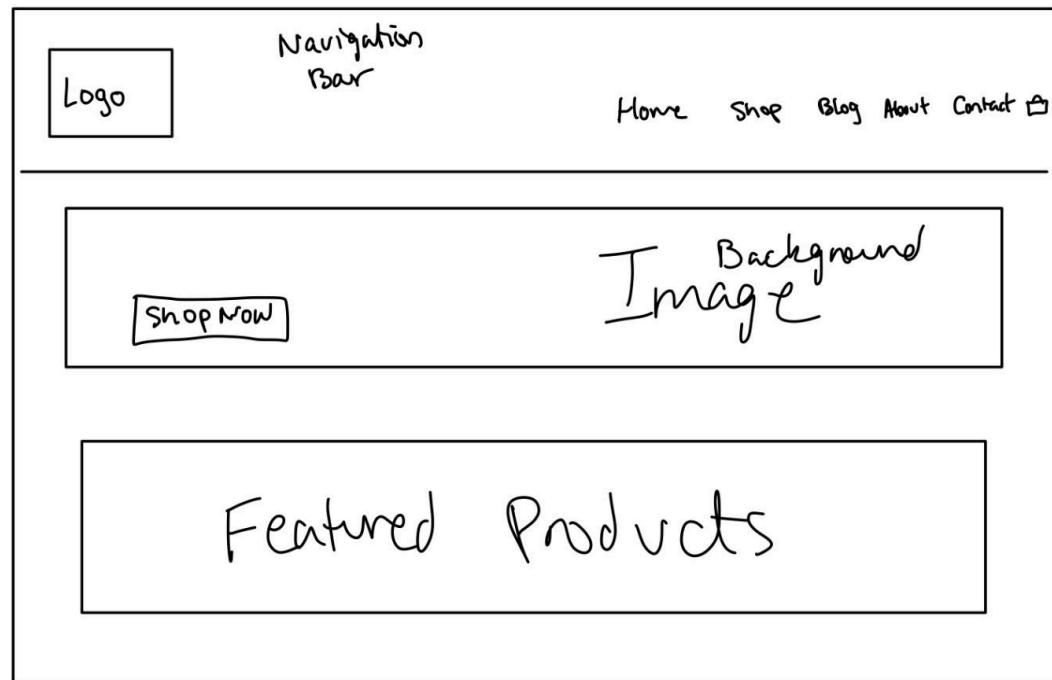
Blog

Blog	Header	img
Blog img	Description	<a href="#">Continue</a>
Blog img	Description	<a href="#">Continue</a>

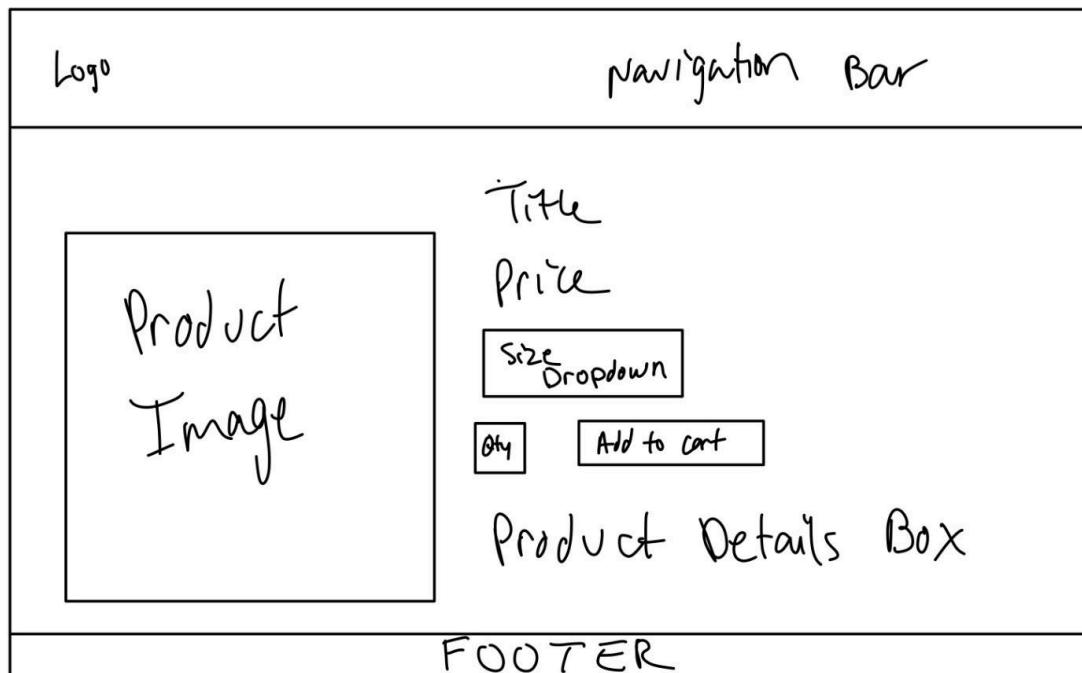
About

Header w/ img	
	Who We Are?

# Home



# Shop



## **Functional Requirements Specification**

### **Stakeholders:**

- Fashion Retailer (Project Sponsor): Initiates and funds the project, with a vested interest in achieving the objectives of the fashion e-commerce website.
- Customers: End-users who visit the platform to browse and purchase fashion products.
- Website Administrators: Responsible for managing website content, updating product listings, and overseeing customer interactions.
- Developer: Responsible for designing, developing, and maintaining the fashion e-commerce website.
- Marketing Team: Collaborators involved in promoting the website, attracting customers, and increasing sales.

### **Actors and Goals:**

#### **Primary Actor:**

- Customer: To browse, select, and purchase fashion items.

#### **Secondary Actors:**

- Corporate Administrator/Manager: To manage product listings, update website content, and oversee customer interactions.
- System: To facilitate the functionality and operation of the fashion e-commerce website.

### **Use Cases:**

Customer (total: 10)

- Browse products: Explore the variety of fashion items available on the website. (1)
- Search products: Use search functionality to find specific fashion items based on preferences. (1)
- View product details: Access detailed information, including images, descriptions, and prices, for selected fashion items. (1)
- Add to cart: Add desired fashion items to the shopping cart for purchase. (1)
- Checkout: Proceed to checkout and complete the purchase transaction. (2)
- Track order: Track the status of orders, including shipping updates and delivery estimates. (2)
- Contact customer support: Reach out to customer support for assistance with orders, inquiries, or issues. (2)

## Admin (total: 12)

- Add product: Add new fashion items to the website inventory, including images, descriptions, and prices. (2)
- Remove product: Remove fashion items from the website inventory. (2)
- Update product details: Modify product information such as name, description, and price. (2)
- Manage customer orders: View and manage customer orders, including processing, shipping, and cancellations. (2)
- View sales reports: Generate and view reports on sales performance, revenue, and popular items. (2)
- Respond to customer inquiries: Address customer inquiries, complaints, and feedback. (2)

## System (total: 8)

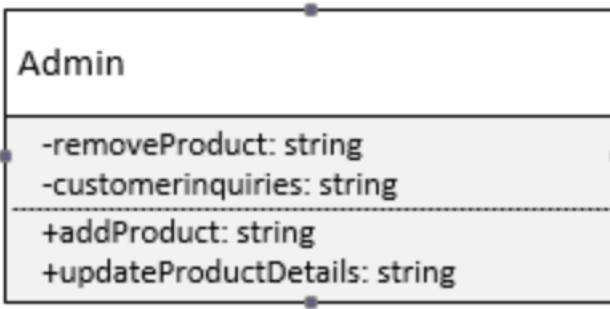
- Manage inventory: Keep track of available fashion items, update stock levels, and notify administrators of low inventory. (2)
- Process payments: Securely process customer payments using various payment methods, including credit/debit cards, PayPal, and other online payment gateways. (2)
- Generate order confirmations: Automatically generate and send order confirmation emails to customers after successful purchases. (1)
- Update website content: Allow administrators to update website content, including images, text, and promotional banners. (1)
- Handle website traffic: Ensure the website can handle high traffic volumes during peak shopping periods without crashing or slowing down. (2)

## Criteria for Success:

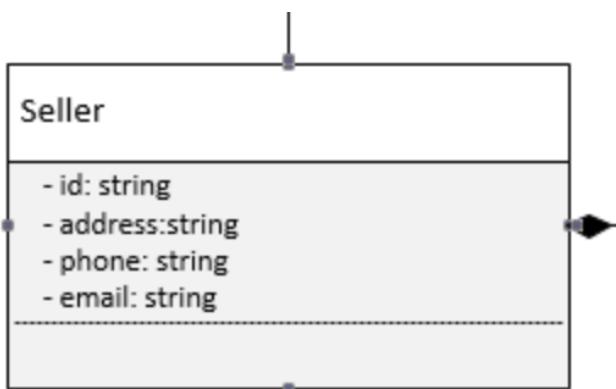
- User-friendly interface allowing easy navigation and browsing of fashion items.
- Secure payment processing system to protect customer information and prevent fraud.
- Efficient inventory management to ensure accurate stock levels and timely restocking of popular items.
- Responsive website design that provides a seamless shopping experience across different devices and screen sizes.
- Effective customer support system to address inquiries, complaints, and feedback in a timely manner.



- **Admin:**
  - **Description:** Represents an administrator or manager who oversees the operation of the fashion e-commerce website.
  - **Responsibilities:** Manages product listings, updates website content, handles customer inquiries, and ensures the smooth functioning of the platform.
  - **Example Operations:** Adding new fashion products, updating product details, managing customer orders, viewing sales reports, and responding to customer inquiries.

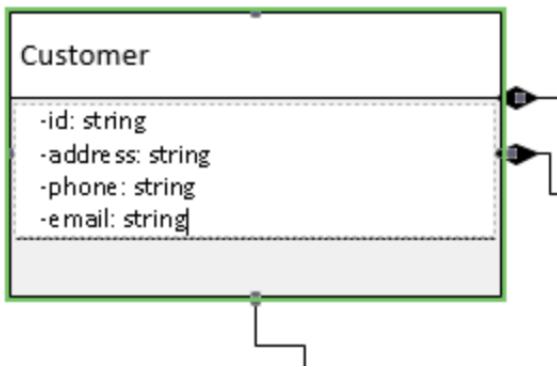


- **Seller:**
  - **Description:** Represents a fashion retailer or vendor who provides products to sell on the website.
  - **Responsibilities:** Supplies fashion items to be listed on the platform, manages inventory, and ensures timely delivery of products to customers.
  - **Example Operations:** Adding new fashion products, updating product availability, managing shipping information, and providing customer support for product-related queries.

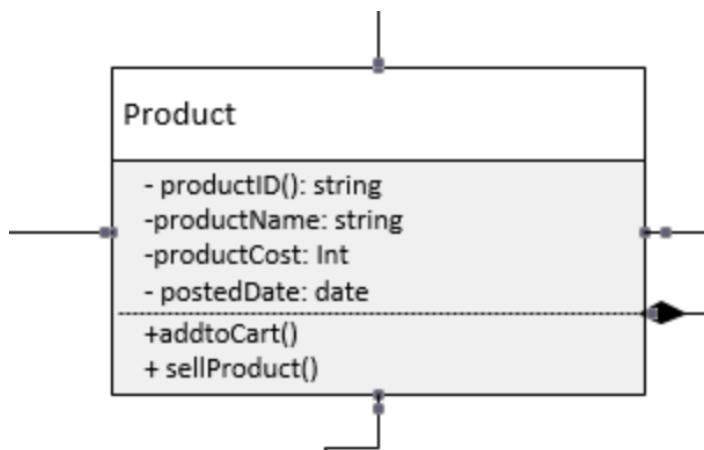


- **Customer:**

- **Description:** Represents an end-user or shopper who visits the fashion e-commerce website to browse and purchase products.
- **Responsibilities:** Explores the product catalog, selects items for purchase, completes the checkout process, and tracks order status.
- **Example Operations:** Browsing products, adding items to the shopping cart, proceeding to checkout, making payments, and tracking order shipments.

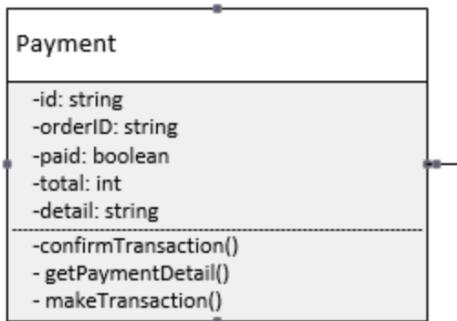


- **Product:**
  - **Description:** Represents individual fashion items available for sale on the website.
  - **Attributes:** May include details such as product name, description, price, category, size, color, and availability.
  - **Example Operations:** Adding new products to the inventory, updating product information, managing stock levels, and categorizing products for easy navigation.



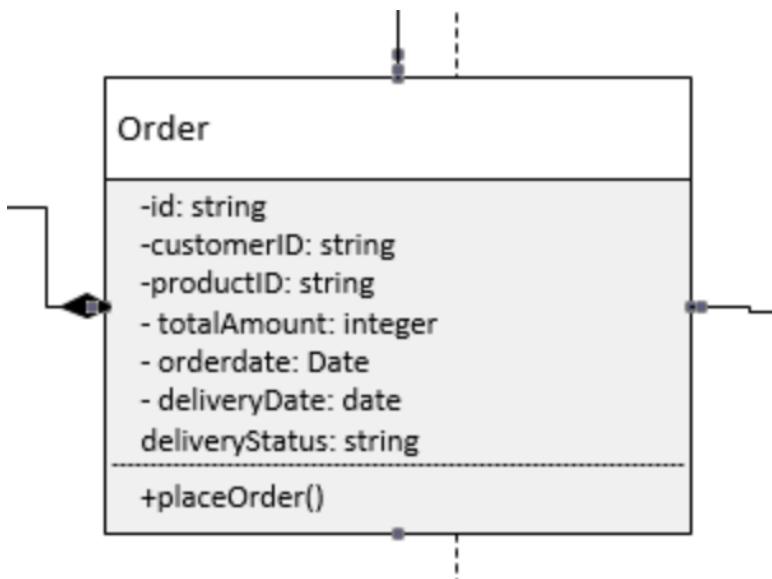
- **Payment:**

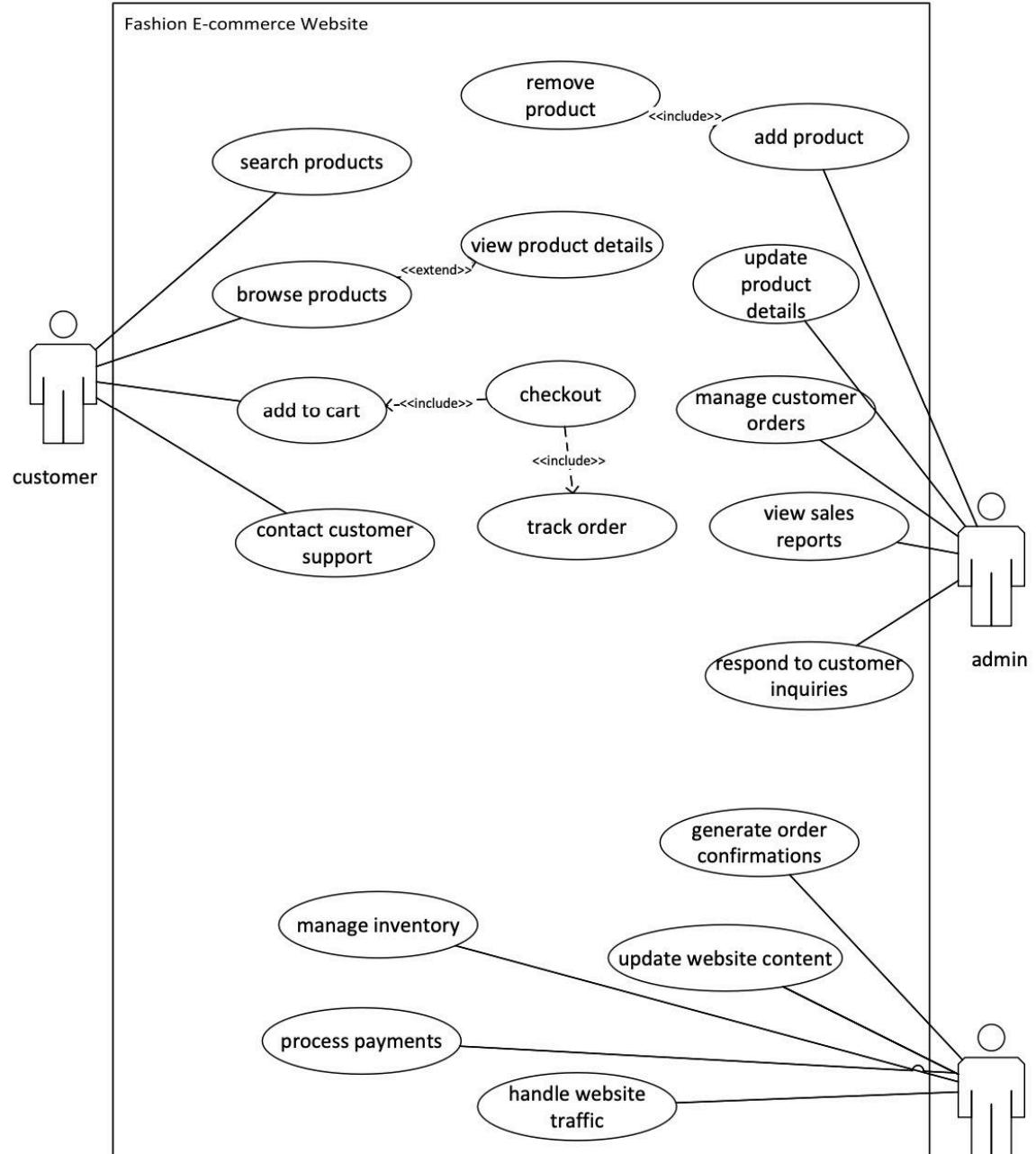
- **Description:** Represents the process of completing a financial transaction for purchasing products on the website.
- **Attributes:** Includes payment-related information such as transaction amount, payment method, billing address, and payment status.
- **Example Operations:** Processing payments securely, verifying payment information, generating payment confirmations, and handling payment disputes.

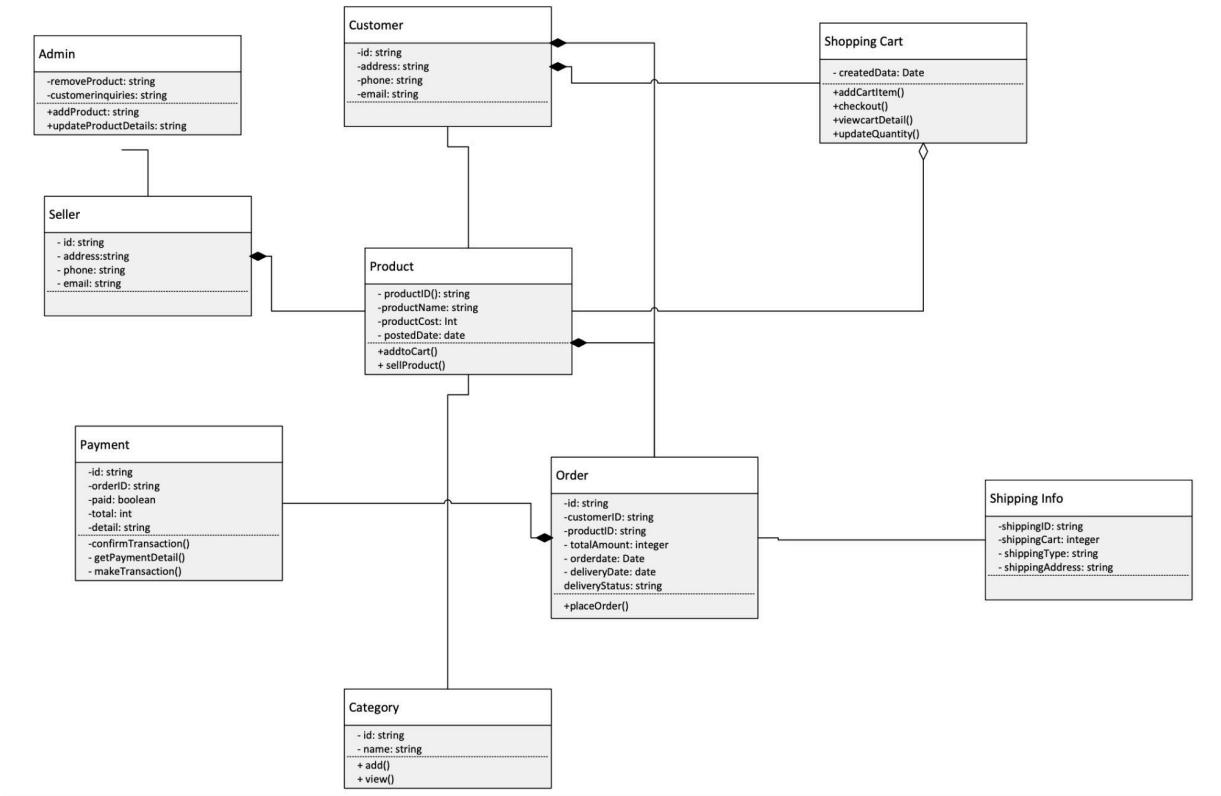


- **Order:**

- **Description:** Represents a customer's purchase request for one or more fashion products.
- **Attributes:** Contains details such as order ID, product list, order status, shipping information, and payment details.
- **Example Operations:** Creating new orders, updating order status, managing order fulfillment, generating order invoices, and handling returns or cancellations.







## Use Case 1: Customer Making a Purchase

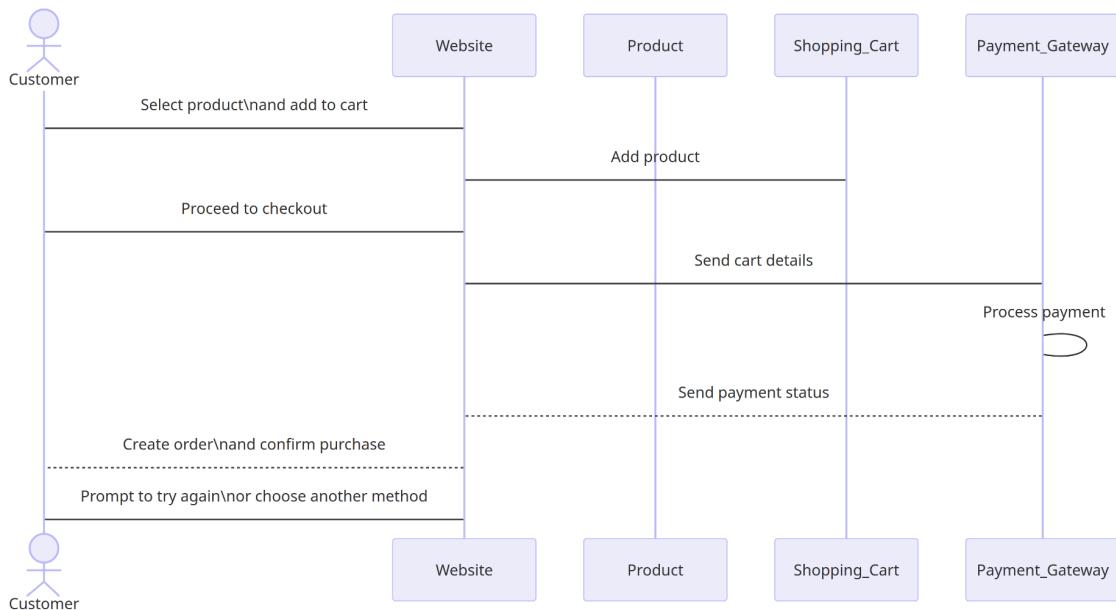
Actor: Customer

Objects: Website, Product, Shopping Cart, Payment Gateway, Order

Steps:

1. Customer selects a product from the shop page and adds it to the shopping cart.
2. Customer proceeds to checkout.
3. Website sends shopping cart details to the Payment Gateway.
4. Payment Gateway processes payment using customer's selected payment method.
5. Payment Gateway sends payment status to the website.
6. If payment is successful, website creates an order and confirms the purchase.
7. If payment fails, website prompts the customer to try again or choose a different payment method.

Sequence Diagram:



## Use Case 2: Customer Browsing and Selecting Products

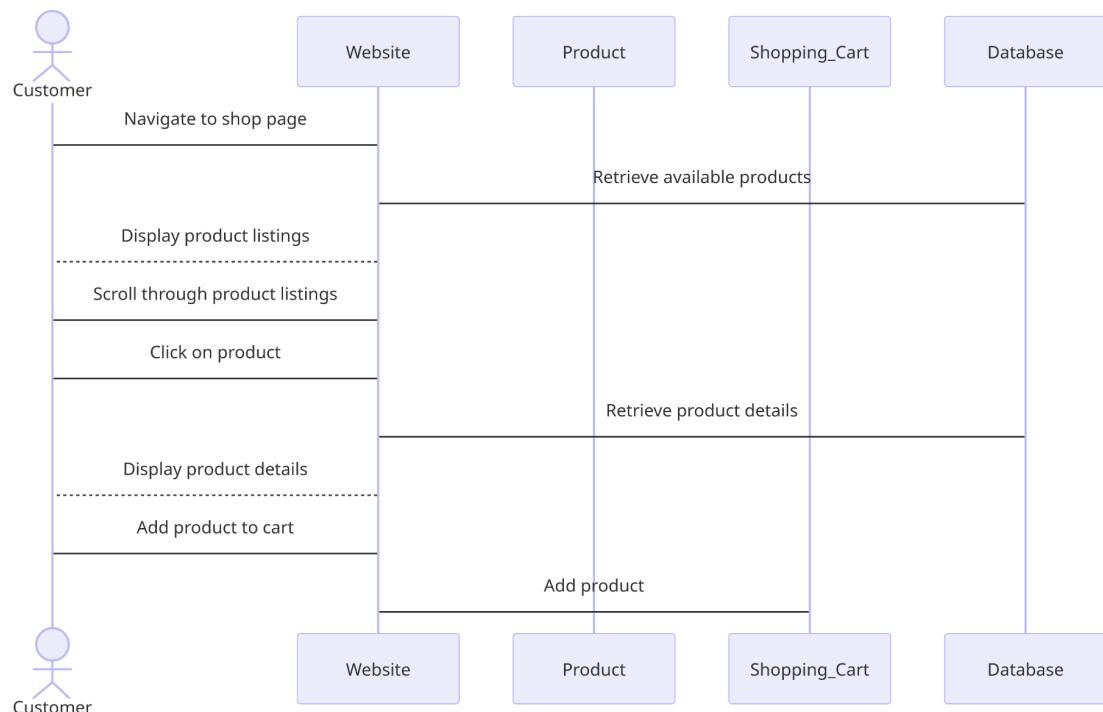
Actor: Customer

Objects: Website, Product, Shopping Cart

Steps:

1. Customer navigates to the shop page from the homepage.
2. Website retrieves the list of available products from the database.
3. Website displays the product listings to the customer.
4. Customer scrolls through the product listings.
5. Customer clicks on a product to view its details.
6. Website retrieves the details of the selected product from the database using the Product object.
7. Website displays the product details, including description, price, and images.
8. Customer decides to add the selected product to the shopping cart.
9. Website adds the selected product to the customer's shopping cart using the Shopping Cart object.

Sequence Diagram:



Use Case 1: Customer Choosing a Product and Checking Out

States:

- Initial State: Customer accesses the homepage
- Final State: Transaction is completed, and the customer receives order confirmation and shipping details

Actions:

Customer accesses the homepage of the fashion e-commerce website.  
Customer navigates to the shop page.  
Customer scrolls through the product listings.  
Customer selects a product of interest.  
Customer views product details, including description, price, and images.  
Customer selects the desired quantity and adds the product to the shopping cart.  
Customer proceeds to checkout.

If the customer is logged in:

- System retrieves customer details and shipping information.
- Customer confirms shipping details.

If the customer is not logged in:

- Customer is prompted to log in or create an account.
- Customer logs in or creates an account.
- System retrieves customer details and shipping information.
- Customer confirms shipping details.

Customer selects a payment method and provides payment details.

Customer reviews the order summary and confirms the purchase.

System processes the payment and generates an order confirmation.

Customer receives the order confirmation and shipping details.

Activity Diagram:

Customer accesses the homepage

Navigate to

Shop Page

Scrolls through

Product Listings

Selects

Product of Interest

Views

Product Details

Selects desired quantity and adds to cart

Shopping Cart

Proceeds to checkout

Not logged in

Customer prompted to log in or create account

Logs in or creates account

Logged in

System retrieves customer details

Confirms shipping details

Payment Page

Provides payment details

Order Summary

Confirms purchase

System processes payment and generates order confirmation

Sends order confirmation and shipping details

Customer

## Use Case 2: Customer Browsing Products

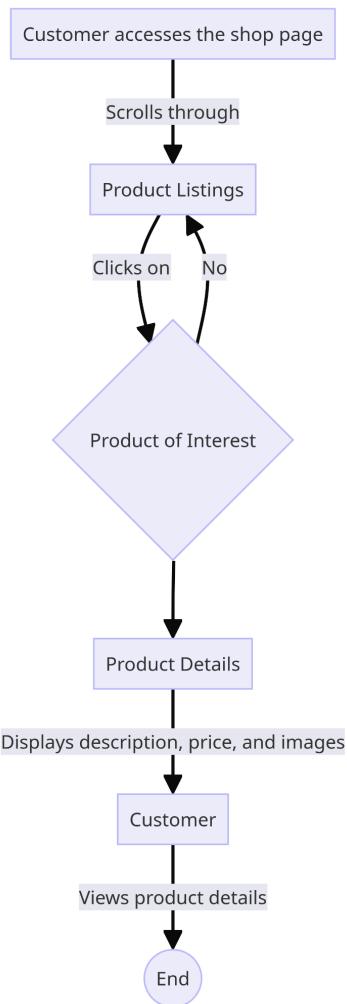
States:

- Initial State: Customer accesses the shop page
- Final State: Customer views product details

Actions:

Customer accesses the shop page from the homepage.  
Customer scrolls through the product listings.  
Customer clicks on a product to view its details.  
Website displays the product details, including description, price, and images.

Activity Diagram:

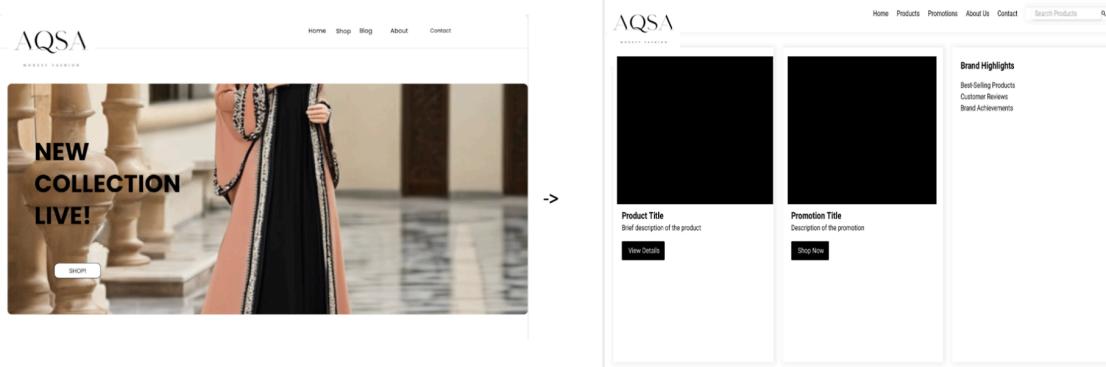


## User Interface Specifications

### Preliminary Design |

- **Home Page:**

- **Use Case:** Customer Exploring Featured Products
  - **Actor:** Customer
  - **Description:** This use case describes the scenario where a customer visits the home page of the fashion e-commerce website to explore featured products and promotions.
  - **Steps:**
    - Customer accesses the home page of the website.
    - Customer views featured products, promotions, and brand highlights displayed on the home page.
    - Customer clicks on a featured product or promotion to view more details.
    - Customer navigates to other sections of the website using the navigation menu.



---

## User Effort Estimation

- **Home Page:** Approximately 2 clicks to explore featured products and promotions.
- **Shop Page:** Approximately 4 clicks to navigate to the desired product category, apply filters, and view product details.
- **Product Pages:** Approximately 2 clicks to select product options and add it to the shopping cart.
- **Shopping Cart:** Approximately 2 clicks to review and proceed to checkout.
- **Contact Page:** Approximately 2 clicks to access the contact form and submit an inquiry.

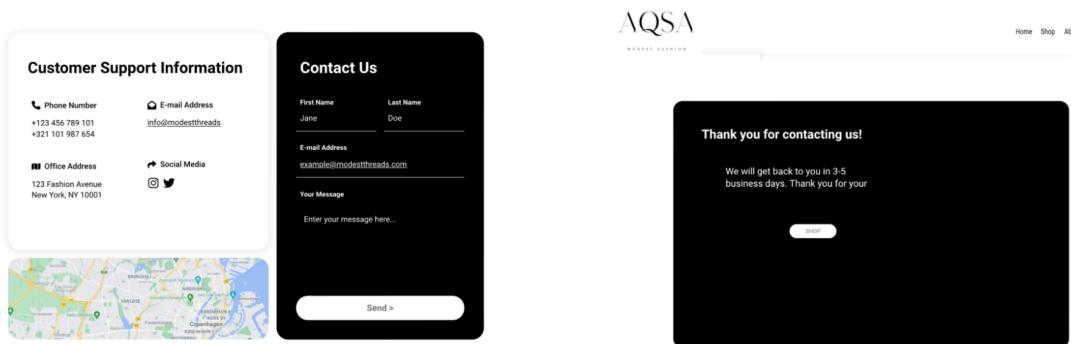
---

## Additional Information

All interactions described above assume that the user is accessing the website from a desktop device. The website will prioritize responsive design principles to ensure optimal viewing and usability across various screen sizes and devices. Additionally, the inclusion of a blog page featuring articles related to modest fashion trailblazers will provide added value to customers seeking fashion inspiration and insights.

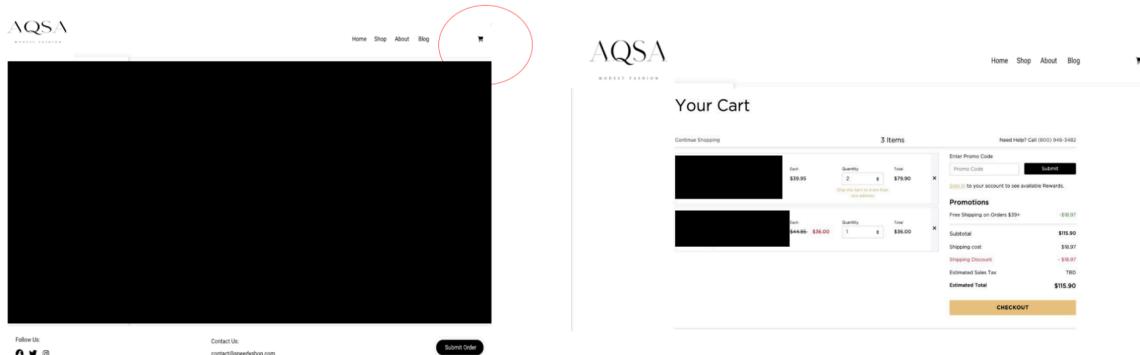
- **Contact Page:**

- **Use Case:** Customer Contacting Support
  - **Actor:** Customer
  - **Description:** This use case describes the scenario where a customer contacts support or submits an inquiry through the contact page.
  - **Steps:**
    - Customer navigates to the contact page from the navigation menu
    - Customer fills out the contact form with their name, email, subject, and message.
    - Customer selects the type of inquiry (e.g., general question, product support).
    - Customer submits the contact form.
    - Customer receives a confirmation message or email upon successful submission.



- **Shopping Cart:**

- **Use Case:** Customer Managing Shopping Cart
  - **Actor:** Customer
  - **Description:** This use case describes the scenario where a customer manages items in the shopping cart before proceeding to checkout.
  - **Steps:**
    - Customer clicks on the shopping cart icon or navigates to the shopping cart page.
    - Customer reviews items currently in the shopping cart.
    - Customer updates quantities, removes items, or adds new items to the cart.
    - Customer proceeds to checkout after reviewing and confirming the items in the shopping cart.



- **Product Pages:**

- **Use Case:** Customer Viewing Product Details
  - **Actor:** Customer
  - **Description:** This use case describes the scenario where a customer views detailed information about a specific product on the product page.
  - **Steps:**
    - Customer clicks on a product thumbnail or product name from the shop page.
    - Customer lands on the product page displaying high-quality images, descriptions, pricing, and available sizes/colors.
    - Customer reviews product details, including descriptions and specifications.
    - Customer selects desired options (e.g., size, color) and adds the product to the shopping cart.

The screenshot shows two pages of the AQSA website. On the left, a product page for an 'Elegant Dress' is displayed. It features a large image of a woman wearing a long, flowing dress with a black hijab. The product title is 'Elegant Dress' and the price is \$99.99. A short description states: 'A stunning abaya dress perfect for special occasions.' Below the description are size options: 'Color: Black Size: M, L, XL'. A 'Add to Cart' button is visible. On the right, a 'Ready to Checkout?' summary page is shown. It lists two items: 'Modest Dressing' (1 piece of modest attire, \$56) and 'Modest Skirt' (1 piece of modest clothing, \$45). The total amount is \$101. At the bottom, there is a checkbox for 'Agree to Terms of Service by checking' and a large blue 'Proceed to Pay' button.

- **Shop Page:**

- **Use Case:** Customer Browsing Product Catalog
  - **Actor:** Customer
  - **Description:** This use case describes the scenario where a customer navigates to the shop page to browse the catalog of fashion items available on the website.
  - **Steps:**
    - Customer accesses the shop page from the home page or navigation menu.
    - Customer browses through the product categories listed on the shop page.
    - Customer applies filters or sorting options to refine search results.
    - Customer clicks on a product thumbnail to view more details.

The screenshot shows three views of the AQSA website's shop page. The left view features a large image of a woman in a long, patterned dress with the text 'NEW COLLECTION LIVE!' overlaid. Below the image are social media icons and a 'SHOP' button. The middle view shows a detailed product listing for three items, each with a thumbnail, product name, price (\$xx.XX), and a 'View Details' button. Above the products are filters for 'Filters' (Price Range, Brand, Ratings, Delivery Time) and 'Most Popular' with an 'Apply' button. The right view shows a grid of products under the heading 'Explore nearby' with a 'Delivery' filter set to 'Fast'. The grid includes categories like 'Dresses', 'Shoes', and 'Accessories', each with its own filters. At the bottom of all three views are footer links for 'Follow Us' (Facebook, Twitter, Instagram), 'Contact Us' (email: contact@aqshop.com), and a 'Submit Order' button.

## **PROJECT PLAN**

Weeks 1-2: Define project scope, gather requirements, and create wireframes/mockups for key pages (home, shop, product, cart, checkout, contact).

Weeks 3-4: Set up development environment, implement basic frontend structure and styling using HTML/CSS.

Weeks 5-6: Develop interactive features using JavaScript to enhance user experience and functionality.

Weeks 7-8: Implement shopping cart functionality, checkout process, and contact form for customer inquiries.

Weeks 9-10: Populate product catalog, refine design elements, and optimize user experience based on feedback.

Weeks 11-12: Conduct comprehensive testing, including functional testing, usability testing, and performance testing.

Weeks 13-14: Fine-tune design elements, optimize website performance, and address any bugs or issues identified during testing.

Week 15: Finalize documentation, prepare for deployment, and conduct final presentation/demo.

## Functional Requirements:

No.	Priority Weight	Description
REQ1	High, 4	The website should have a visually appealing and intuitive interface.
REQ2	High, 5	Customers should be able to browse and filter products efficiently.
REQ3	High, 3	Product pages should include descriptions and high-quality images.
REQ4	High, 5	The shopping cart functionality should allow customers to add/remove items.
REQ5	High, 4	The checkout process should be streamlined and user-friendly.
REQ6	Low, 1	The website should provide personalized recommendations.
REQ7	Medium, 2	Customers should have access to engaging content and service.

## Use Cases:

Use Case	Description
UC1	Browse products: Explore the variety of fashion items available on the website.
UC2	Search products: Use search functionality to find specific fashion items based on preferences.
UC3	View product details: Access detailed information, including images, descriptions, and prices, for selected fashion items.
UC4	Add to cart: Add desired fashion items to the shopping cart for purchase.
UC5	Checkout: Proceed to checkout and complete the purchase transaction.
UC6	Contact customer support: Reach out to customer support for assistance with orders, inquiries, or issues.
UC7	Add product: Add new fashion items to the website inventory, including images, descriptions, and prices.
UC8	Remove product: Remove fashion items from the website inventory.
UC9	Update product details: Modify product information such as name, description, and price.

UC10	Manage customer orders: View and manage customer orders, including processing, shipping, and cancellations.
UC11	Respond to customer inquiries: Address customer inquiries, complaints, and feedback.
UC12	Update website content: Allow administrators to update website content, including images, text, and promotional banners.

Traceability Matrix:

REQ	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11	UC12
REQ 1	4	X		X									
REQ 2	5	X	X		X	X							
REQ 3	3	X		X									
REQ 4	5	X			X	X							
REQ 5	4			X		X							
REQ 6	1						X	X	X	X			

REQ 7	2						X	X	X	X	X	X
<b>Max PW</b>	5	5	4	5	5	1	2	2	2	1	1	1
<b>Total PW</b>	17	5	11	10	14	1	3	3	3	2	2	2

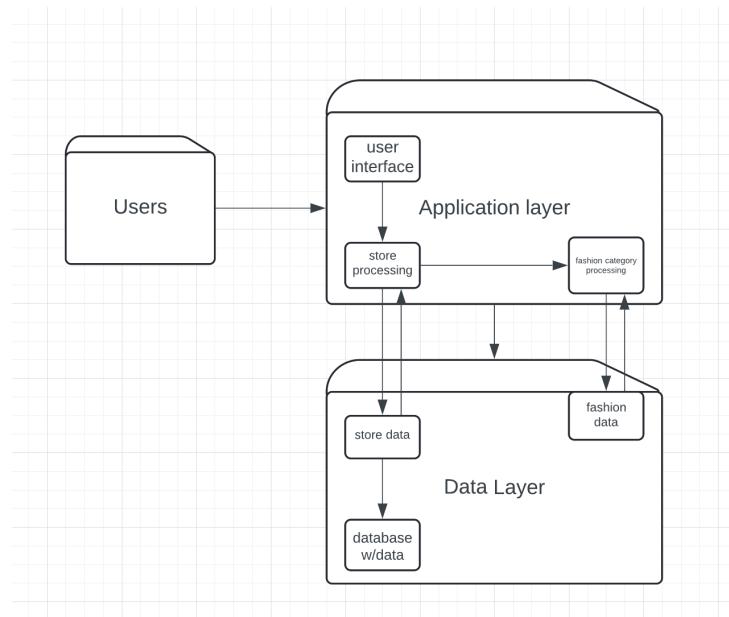
# System Architecture and System Design

Architectural Styles:

The architecture of the Fashion Ecommerce Website follows the **Client-Server** architectural style. This style involves the division of functionality between two types of entities: clients and servers. The client, typically a web browser, interacts with the server to request and receive data, while the server handles the processing of requests, manages data, and provides responses.

Identifying Subsystems:

Below is a UML package diagram representing the subsystems of the Fashion Ecommerce Website:



- **Application Layer:** This layer consists of the user interface components responsible for presenting information to the users, including web pages, forms, and interactive elements. This layer contains the business logic and application components responsible for processing user requests, managing sessions, and orchestrating interactions between the presentation layer and the data access layer.
- **Data Access Layer:** This layer handles the retrieval and storage of data from the underlying data sources, such as databases or external APIs.

#### Mapping Subsystems to Hardware:

The Fashion Ecommerce Website will run on multiple computers, with the following subsystems distributed across different machines:

- **Client Side (Web Browser):** Presentation Layer
- **Server Side (Web Server):** Application Layer, Data Access Layer

#### Persistent Data Storage:

Yes, the system requires persistent data storage to save data beyond a single execution. The persistent objects include user accounts, product listings, orders, and session information. The storage management strategy will utilize a relational database management system (RDBMS) such as MySQL or PostgreSQL to ensure data integrity, scalability, and efficient retrieval of structured data.

#### Network Protocol:

The system will use the HTTP (Hypertext Transfer Protocol) protocol for communication between the client (web browser) and the server (web server). HTTP is a widely adopted protocol for transmitting data over the internet and is supported by all major web browsers and web servers. It provides a standardized way for clients to request resources from servers and receive responses.

#### Global Control Flow:

The system is event-driven, where user actions trigger events that are processed asynchronously by the server. Each user can generate actions in a different order, and the system responds to these actions dynamically. The control flow is not strictly linear and adapts to user interactions and system events.

#### Time Dependency:

The system does not have strict time constraints or timers. It operates in an event-response fashion, where actions by users trigger responses from the server in real-time. There are no periodic tasks or scheduled events within the system.

#### Concurrency:

Yes, the system utilizes multiple threads to handle concurrent user requests and ensure responsiveness. Objects such as the session manager, request handler, and database connections may have separate threads of control to handle concurrent operations efficiently. Synchronization mechanisms such as locks or semaphores are employed to manage shared resources and prevent data corruption in multi-threaded environments.

#### Hardware Requirements:

The system depends on the following hardware resources:

- **Screen Display:** Clients (web browsers) require a screen display to render the user interface. The system should support various screen sizes and resolutions for optimal user experience. Minimum resolution: 640 x 480 pixels. Any computer (desktop or PC) or mobile device.
- **Communication Network:** Reliable internet connectivity is essential for clients to communicate with the server over HTTP. The system should ensure adequate network bandwidth and low latency to deliver responsive web applications.
- **Server Hardware:** The web server hosting the application and database requires sufficient computing resources (CPU, memory, storage) to handle incoming requests, process business logic, and manage data storage effectively. Additionally, regular backups and redundancy measures may be implemented to ensure high availability and fault tolerance. 2 Gb RAM and hard drive space.

## User Interface Design and Implementation:

For the User Interface Design and Implementation, I revised the initial screen mock-ups created in Module 7 - User Interface Specification to ensure they aligned with the system requirements and user expectations. I focused on optimizing user interaction and reducing user effort by streamlining navigation and enhancing usability.

Some significant changes in the user interface include:

1. **Refined Navigation:** I improved the navigation flow to ensure users can easily browse products, add them to the cart, and proceed to checkout without encountering any friction points.
2. **Simplified Checkout Process:** I redesigned the checkout process to minimize the number of steps required for users to complete a purchase. This includes consolidating information input fields and providing clear instructions at each stage.
3. **Enhanced Product Presentation:** I optimized the product detail pages to provide comprehensive information, including high-quality images, detailed descriptions, and pricing, to help users make informed purchasing decisions.
4. **Improved Feedback Mechanism:** I implemented interactive elements such as feedback forms and customer support chatbots to facilitate communication between users and administrators, reducing response times and improving overall user satisfaction.

## Design of Tests:

For the Design of Tests, I have planned the following test cases for unit testing and integration testing:

1. **Unit Testing Test Cases:**

- Test Case 1: Verify that the product browsing functionality displays the correct list of products according to the selected category.
- Test Case 2: Ensure that adding a product to the shopping cart increments the cart count appropriately.
- Test Case 3: Validate that the checkout process calculates the total order amount correctly.
- Test Case 4: Test the functionality of the search feature to accurately retrieve relevant products based on user input.
- Test Case 5: Verify that the user authentication system works correctly, allowing registered users to log in and access their accounts.

## 2. Integration Testing Strategy:

- Integration testing will focus on validating the interactions between different subsystems, such as the user interface, application logic, and data access layer.
- Test scenarios will include testing end-to-end user workflows, data integrity checks, and error handling mechanisms.
- Integration testing will be conducted using both automated testing tools and manual testing procedures to ensure comprehensive coverage.

## 3. Testing Plans for Non-functional Requirements:

- Performance Testing: Measure the system's response time under various load conditions to ensure it meets performance requirements.
- Usability Testing: Gather feedback from real users to evaluate the system's ease of use, navigation flow, and overall user experience.
- Compatibility Testing: Test the system across different devices, browsers, and operating systems to ensure compatibility and responsiveness.

By implementing these test cases and strategies, I aim to ensure the reliability, functionality, and performance of the system while meeting the specified requirements and user expectations.