The Islamia University of Bahawalpur

Department of Information Technology

SOFTWARE DESIGN DESCRIPTION

(SDD DOCUMENT)

for

Online Garments Store

By

Ansa Sarwar

F21BINFT1E02125

Session Fall 2022- 2026

Supervisor

Ms. Sara Fareed

Bachelor of Science in InformationTeachnology

Tables of Contents

Revision History

Name	Date	Reason for changes	Version

Application Evaluation History

Comments (by committee) *include the ones given at scope time both in doc and presentation	Action Taken

Supervised by

Ms. Sara Freed

Signature_

Introduction

The Garments e- commerce website is a user- centric platform designed to facilitate seamless online shopping for garments. Users can register, search product, manage a shopping cart, and securely complete orders, and there will be admin panel, admin can handle user accounts, manage products and add or remove products. The website prioritizes an intuitive user interface, efficient order processing, and adherence to security and regulatory standards. With a focus on reliability and user satisfaction, it aims to provide a visually appealing and responsive online shopping experience for garments. Garments Store is an ambitious endeavour to create a sophisticated and user- friendly ecommerce platform tailored specifically for the garment industry. With a focus on offering a diverse and curated collection of garments, this platform aspires to redefine online shopping experiences by providing a seamless, secure, and enjoyable journey for users seeking high- quality fashion.

Design methodology and software process model

Explanation:

The Agile methodology is chosen for the Software Design Document (SDD) of the full- stack developed website. Agile's iterative and flexible nature aligns with the dynamic requirements of web development, facilitating continuous improvements, customer collaboration, and incremental deliveries.

Justification:

- Iterative Development: Agile's short cycles refine both front- end and back- end in full- stack development.
- Flexibility: Adapts to changing requirements and client needs in the dynamic web development landscape.
- Customer Collaboration: Ensures continuous stakeholder engagement for user satisfaction and business goals.
- Incremental Deliveries: Facilitates regular feature releases, enhancing the website and addressing user feedback promptly.
- Cross- Functional Teams: Enables effective collaboration among full- stack developers, designers, and stakeholders.

Process Model: Agile Model

Explanation:

The Agile Model is selected as the software process model for the Garments store system. Agile is an iterative and incremental approach that emphasizes flexibility, collaboration, and customer feedback throughout the development process.

System overview

System Overview: Garments store

Functionality:
Garments store is an online platform designed to provide a seamless and user-friendly experience for customers to browse, select, and purchase a variety of high- quality Garments s. The system encompasses features for both customers and administrators, facilitating efficient management of products, orders, and user profiles.

Customer Features:

- Browse Garments s: Customers can explore a diverse catalog of Garments s, categorized for easy navigation.
- Search and Filter: Users have the ability to search for specific Garments s and apply filters based on categories, brands, or keywords.
- Order Placement: Customers can place orders for selected Garments s, specify quantities, and proceed through a secure payment process.
- User Profiles: Users can manage their profiles, update personal information, and configure preferences related to notifications and subscriptions.

Administrator Features:

- Product Management: Administrators can add new Garments products, including details such as names, descriptions, and pricing.
- Inventory Control: Real- time updates on product availability, ensuring accurate and up- to- date inventory management.
- Order Processing: Efficient handling of customer orders, including order confirmation, updating inventory status, and addressing order- related issues.
- User Profile Management: Administrators can view and manage user profiles, ensuring a personalized and responsive customer experience.

Design:
The system is designed with a procedural programming approach, emphasizing clear and linear procedures for tasks such as inventory management, order processing, and user profile management. The Waterfall Model is employed as the process model, ensuring a systematic and well- documented development process.

Background:
The inspiration behind Garments store arises from the growing demand for healthy and premium snack options. The platform aims to connect customers with a wide range of high- quality Garments s while ensuring a straightforward and secure online shopping experience. As the system evolves, it will continue to

integrate customer feedback and adapt to emerging trends in the Garments market.
Architectural design

Architectural design: Modular Program Structure for Garments store

1. User Interface (UI):
   - Presents the web interface to users.
   - Handles user interactions and communicates requests to the backend.
2. Business Logic:
   - Core functionality managing the product catalog, inventory, orders, and user profiles.
   - Validates user inputs and enforces business rules.
3. Database Management:
   - Stores and manages data related to products, inventory, orders, and user profiles.
   - Supports efficient retrieval and update operations.
4. Order Processing:
   - Processes customer orders.
   - Communicates with business logic for inventory and order updates.
5. Product Management:
   - Manages products and communicates with business logic for catalog updates.
6. User Profile Management:
   - Handles user authentication, registration, and profile updates.
   - Communicates with business logic for user- related actions.

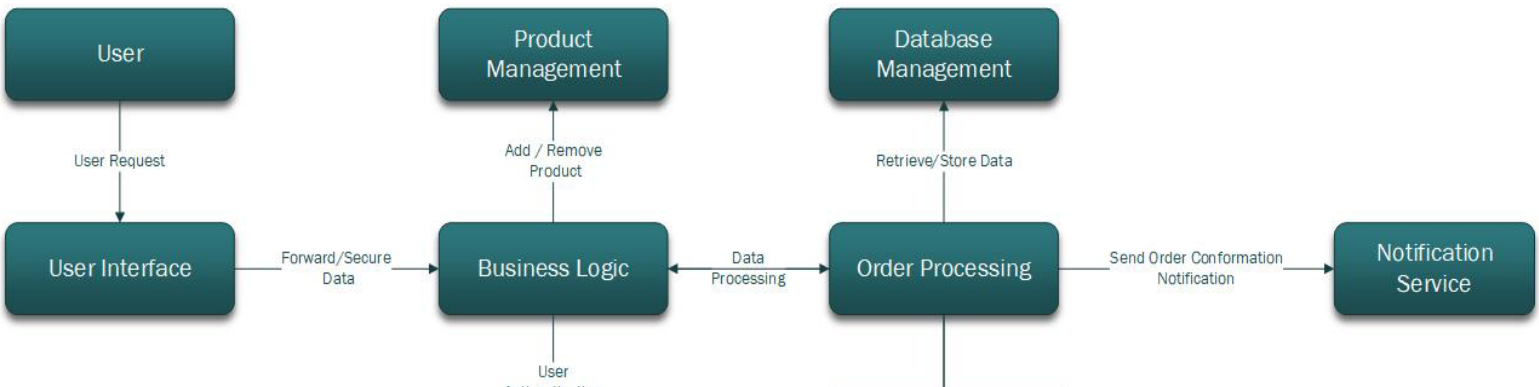
Module Relationships:

- UI communicates with Business Logic for user requests.
- Business Logic communicates with the Database for data operations.
- Order Processing, Product Management, and User Profile Management interact with Business Logic for specific tasks.

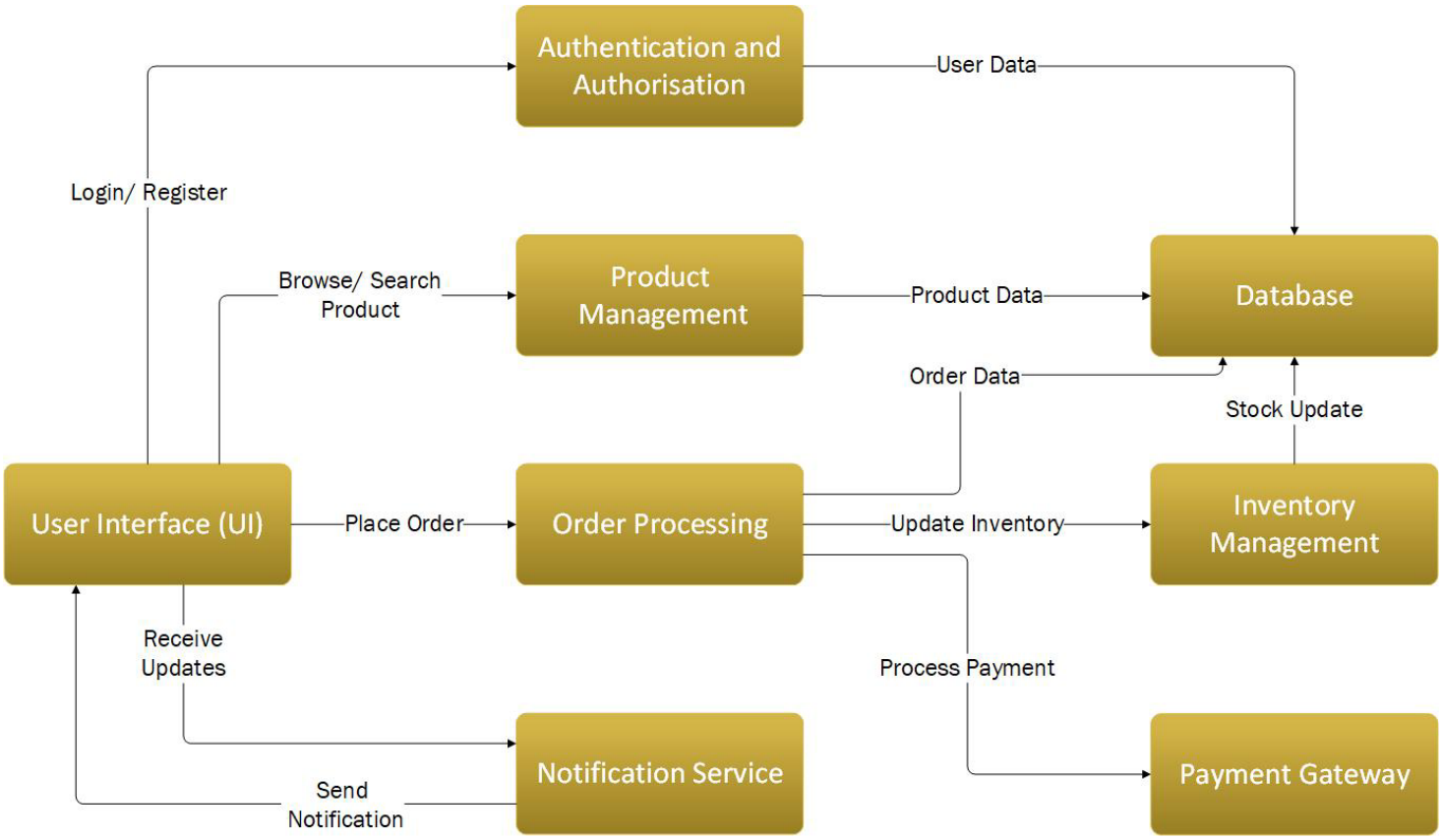
Subsystem Connections:

- Well- defined connections through APIs or service contracts.
- Modular design allows for updates, replacements, and additions without affecting the entire system.

Architectural Diagram



Process flow/Representation



Class Diagram:

User:

- Represents a general user of the Garments store platform.

Attributes:

- User Id (int): Unique identifier for each user.
- Name (string): User's name.
- Email (string): User's email address.
- Password (string): User's login password.
- Last login (string):User’ s last Login info.

Operations:

- Authenticate (credentials): Registration, login and Authenticates the user during the login process.

Customer:

- Represents a customer interacting with the Garments store platform.

Inherits from User.

Attributes:

- Browse Products: Allows the customer to browse available Garments products.
- Update profile (details): Allows the customer to update their profile information.
- Billing Address (details): Add Customer’ s billing Address.

Operations:

- Inherits attributes from the User class.

Shopping- Cart:

- Manages the items selected by the customer for purchase.

Attributes:

- userId(int): Unique identifier for each user.
- productId(int): Unique identifier for each product

Operations:

- AddToCart (product): Adds a selected product to the shopping cart.
- removeItem (product): Removes a product from the cart.

- Checkout (): Initiates the checkout process.

Orders:

- Represents a customer's order.

Attributes:

- OrderId (int): Unique identifier for each order.
- Customer (Customer): Reference to the customer placing the order.
- Product List (List of Product): List of products included in the order.
- Total Price (float): Total amount for the order.

Operations:

- Process Payment(payment Details): Handles the payment process for the order.
- Place Order (): proceed the order for confirmation.
- Cancel Order(): cancel the order process.
- Confirm Order (): confirmation from the customer to be shipped to billing address.

Order- Details:

- Represents the details of each product within an order.

Attributes:

- User Id (int): Unique identifier for each user.
- OrderId (int): Unique identifier for each order detail.
- Product (Product): Reference to the product in the order.
- Quantity (int): Quantity of the product in the order.
- Subtotal (float): Subtotal for the specific product in the order.

Operations:

- Cancel Order (): cancel the order process.

Diagram:

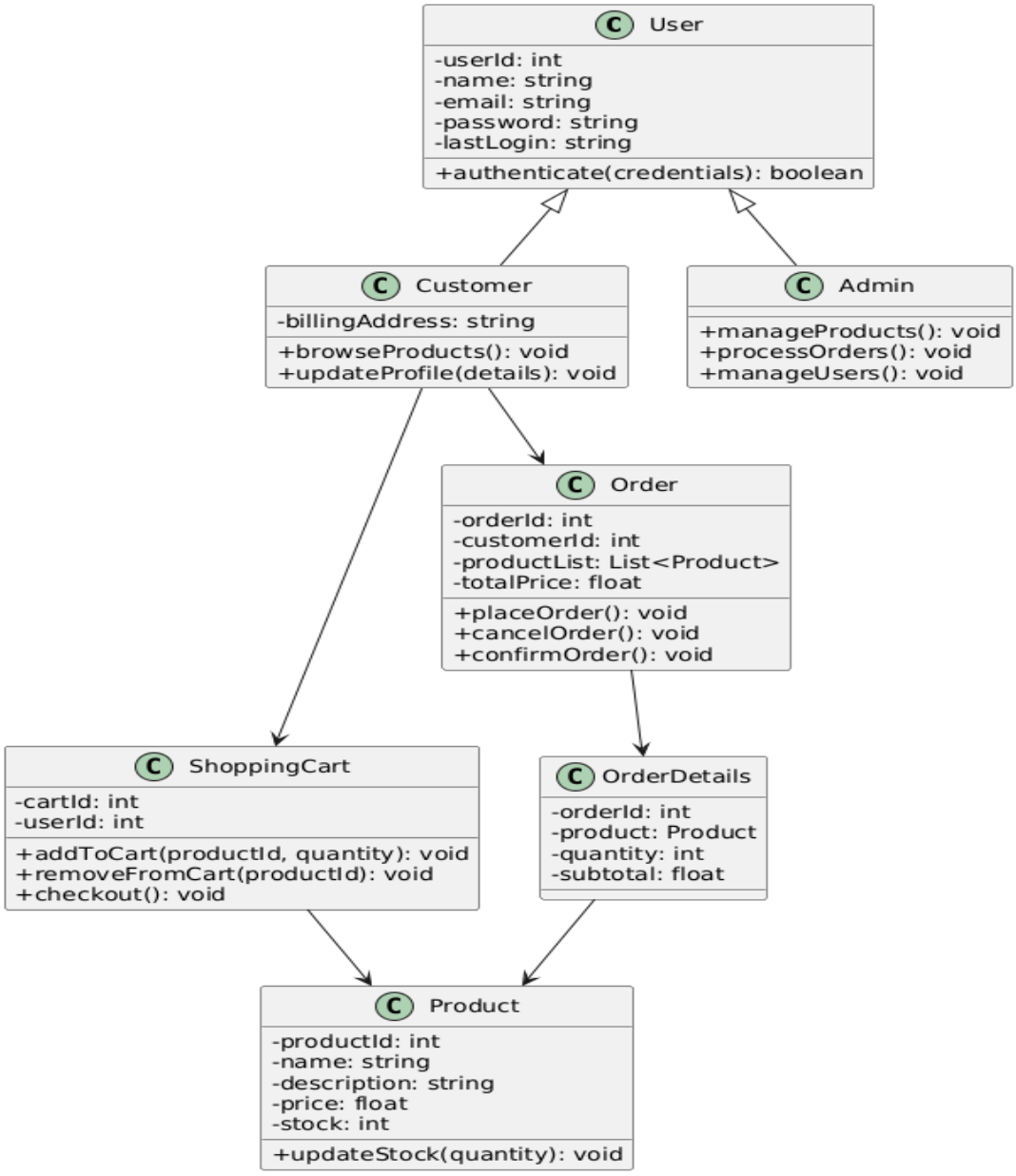


Figure: Class Diagram

Sequence Diagram:

- Customer: The person interacting with the Garments store online platform.
- Online Garments store: Represents the website or platform for purchasing Garments s.
- Shopping Cart: Manages the availability of Garments items in the store.

- Admin: Handles the process of order processing for Garments purchases.

Sequence of Events:

- The customer visits the Garments store's online platform.
- The platform requests the product list from the Garments inventory.
- The platform displays the available Garments products to the customer.
- The customer adds selected Garments items to their cart.
- The platform checks the availability of the chosen items with the inventory system.
- Upon confirmation of item availability, the platform informs the customer that the items have been added to the cart.
- The customer proceeds to checkout for the Garments items.
- The platform initiates the payment process through the Admin for Garments purchases.
- The admin confirms the order for Garments items.
- Finally, the platform sends an order confirmation to the customer for the Garments purchase.

Diagram:

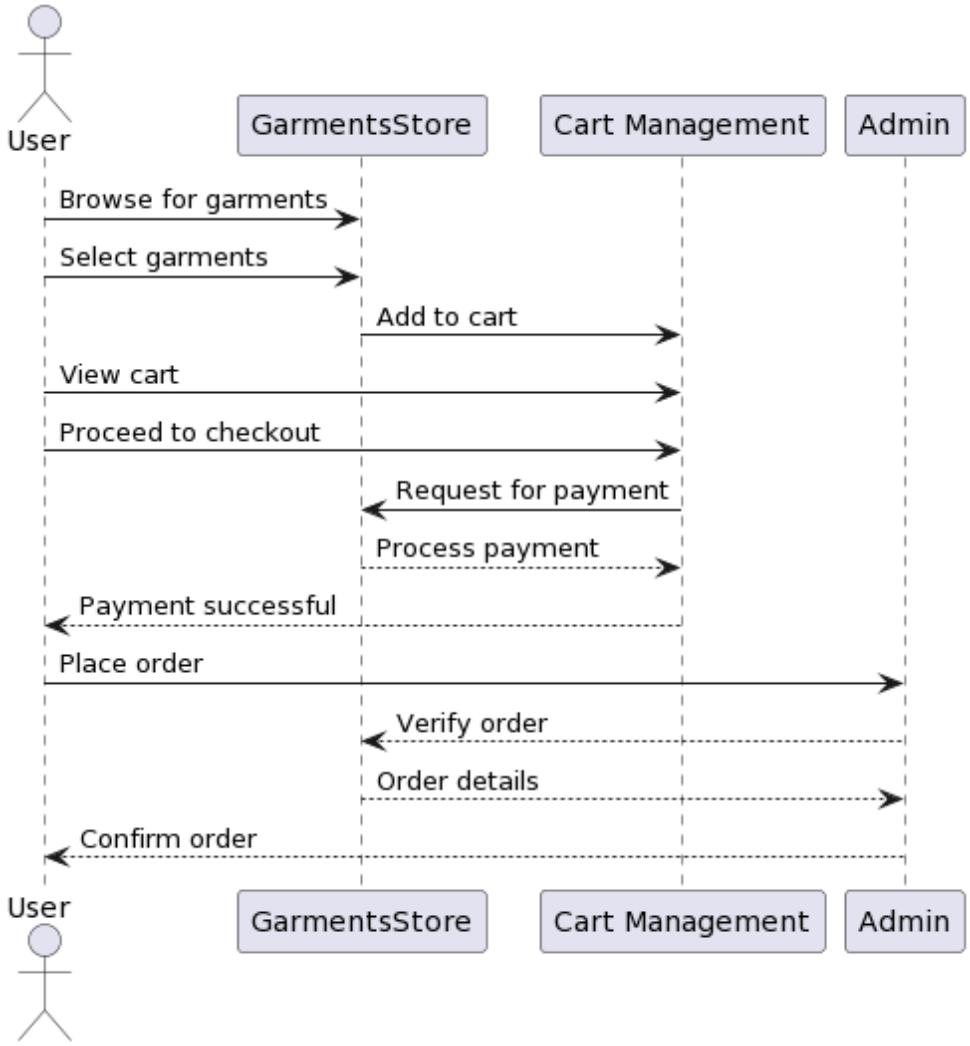


Figure: Sequence diagram

State Transition Diagram:

1. User Not Logged In (Initial State):

Description: This is the initial state when the user is not logged into the Garments store platform.

Transitions:

If the user chooses to log in, transition to the "User Logged In" state.

If the user decides to browse products without logging in, transition to the "Browsing Products (Guest)" state.

2. Browsing Products (Guest):

Description: In this state, a non- logged- in user can browse the product catalog as a guest.

Transitions:

If the user decides to log in, transition to the "User Logged In" state.

If the user adds items to the cart, transition to the "Shopping Cart (Guest)" state.

3. User Logged In:

Description: The user is successfully logged into the platform.

Transitions:

If the user starts browsing products, transition to the "Browsing Products (Logged In)" state.

If the user accesses their profile, transition to the "User Profile Management" state.

4. Browsing Products (Logged In):

Description: A logged- in user is browsing the product catalog.

Transitions:

If the user adds items to the cart, transition to the "Shopping Cart (Logged In)" state.

If the user proceeds to the checkout, transition to the "Placing Order" state.

5. Placing Order:

Description: The user is in the process of placing an order.

Transitions:

If the user completes the order, transition to the "Order Confirmation" state.

If the user cancels the order, transition back to the previous state.

Diagram:

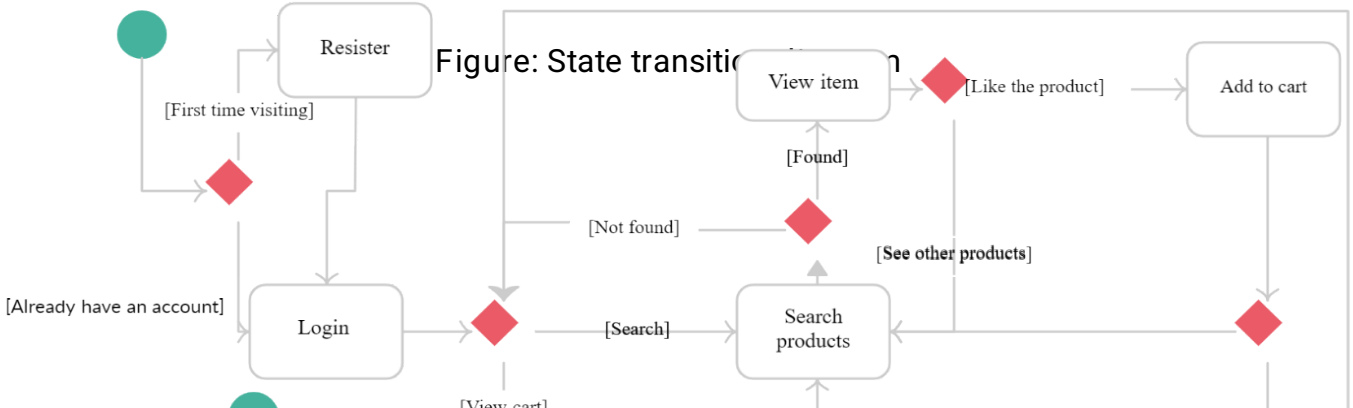


Figure: State transitio n

Data Flow Diagram:

The Data Flow Diagram (DFD) provides a graphical representation of how data flows through the Used Car Price Prediction System. It shows interactions between external entities (e.g., users and admins), processes (e.g., input validation, prediction generation), and data stores (e.g., car details, prediction results).

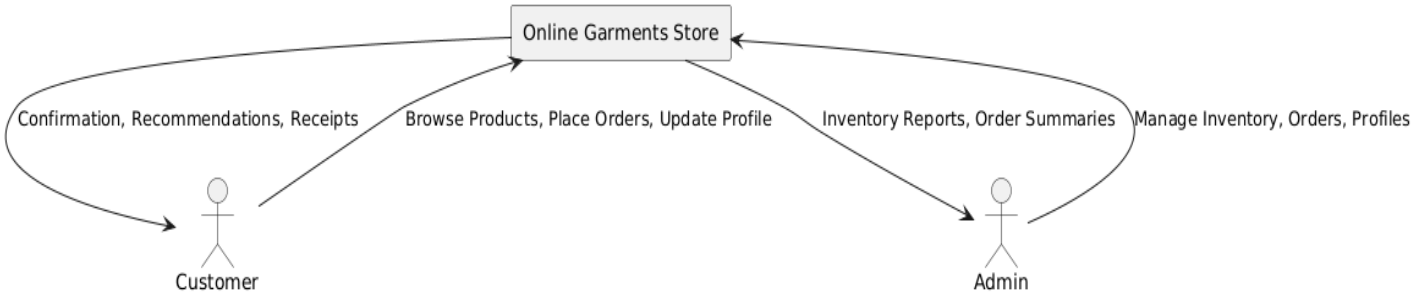


Figure: Data Flow Diagram (DFD) for Online Garment Store Level 0

Table 4.4.1:Level 0 Data Flow Diagram (DFD) Explanation

Level	Actor	Process	Data store	Description
Level 0	Customer	Browse Products, Place Orders, Update Profile	None	Customers interact with the Online Garments Store to browse, order, and manage profiles.

Level 0	Admin	Manage Inventory, Orders, Profiles	None	Admins handle inventory, user accounts, and orders through the platform.

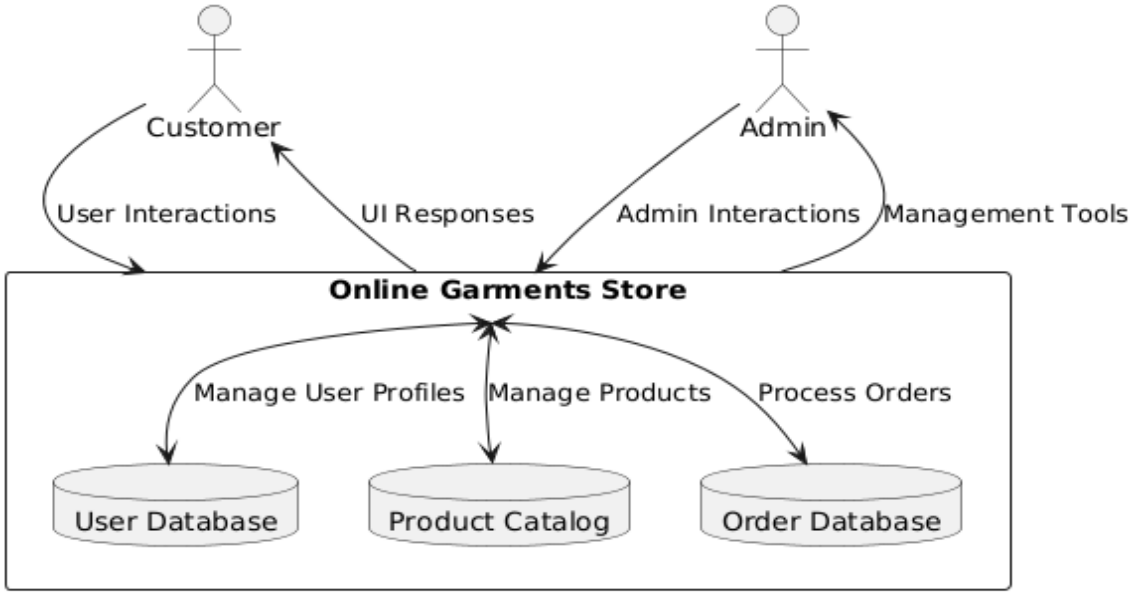


Figure : High- Level Data Flow Diagram (Level 1) for online garment store

Table : High- Level Data Flow Diagram (Level 1) for Online Garment store

Level	Actor	Process	Datastore	Description
Level 1	Customer	User interactions	User database	Customers log in, browse products, and interact with the store.

Level 1	Admin	Admin interactions	Product Catalog	Admins manage product catalogs, update inventories, and process orders.
Level 1	System	Process orders	Order database	The system processes customer orders and updates inventory.

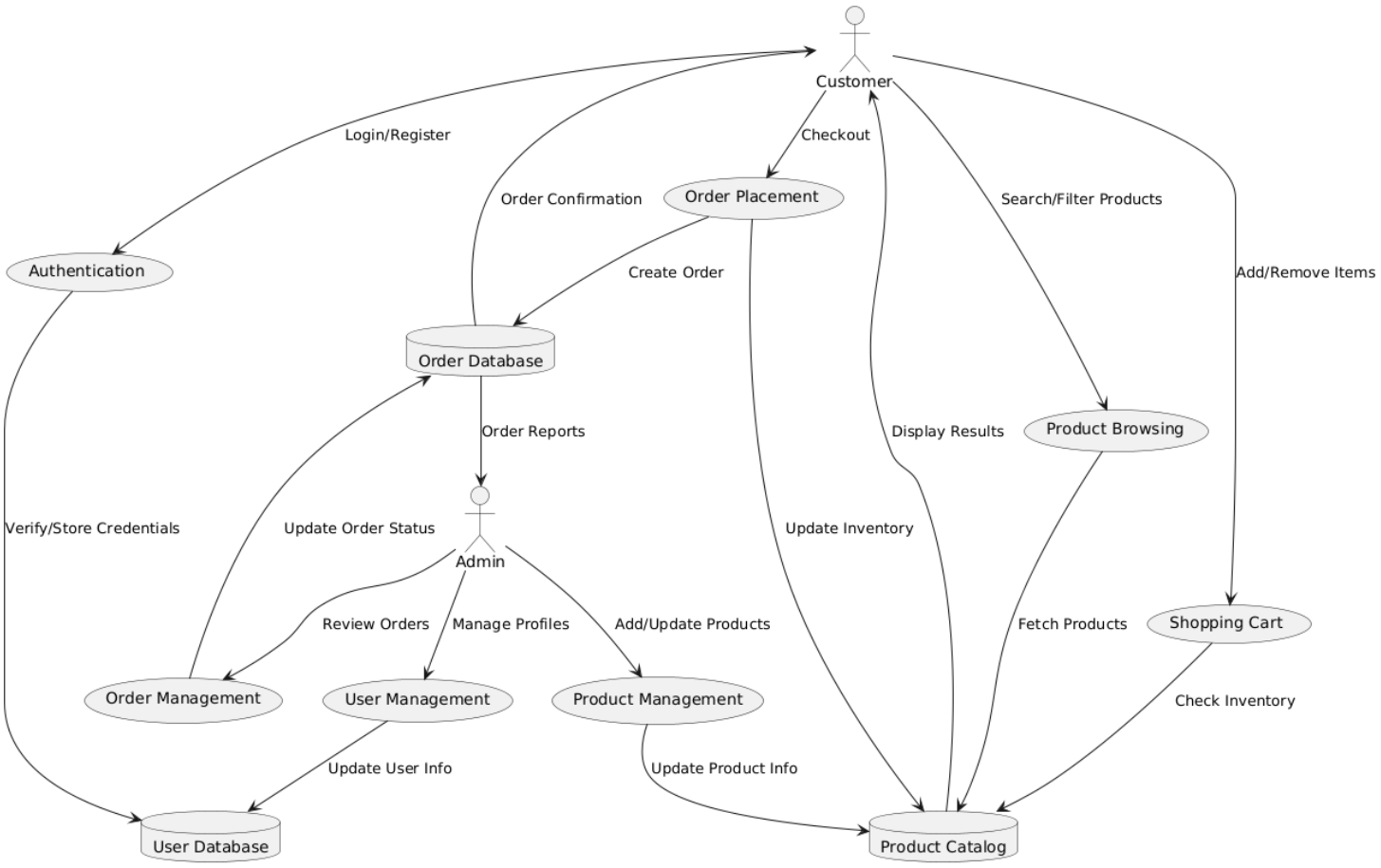


Figure : Detailed Data Flow Diagram (Level 2) for online garment store

Table : Detailed Data Flow Diagram (Level 2) for online garment store

Level	Actor	Process	Datastore	Description
Level 1	Customer	Login/Register	User Database	Customers register or log in, and their credentials are verified or stored.
Level 1	Customer	Search/Filter Products	Product Catalog	Customers search for products, and the catalog fetches the relevant data.
Level 1	Customer	Add/Remove items	Product Catalog	Customers add or remove items from the shopping cart, and inventory is checked.
Level 1	Customer	Checkout	Order Database, Product Catalog	Customers place orders, and inventory and orders are updated accordingly.

Level 1	Admin	Add/Update product	Product Catalog	Admins add or update product information in the catalog.
Level 1	Admin	Review orders	Order Database	Admins review or update order statuses.
Level 1	Admin	Manage Profile	User Database	Admins manage user profiles in the system.

Data design

In Garments store, the information domain is transformed into structured data through the use of databases and data storage items. The major data entities are organized, processed, and stored to ensure efficient functionality. Here's an overview of the data design:

1. User Profiles:

- Data Structure: Structured tables in the MySQL Database.
- Storage: User profiles are stored with attributes like name, email, password, and preferences.
- Processing: User authentication, registration, and profile updates are processed through the User Profile Management module.
- Organization: Organized categorically based on user roles and preferences.

2. Product Catalog:

- Data Structure: Product Catalog Data Store.
- Storage: Product information is stored with details like names, descriptions, and pricing.
- Processing: Managed through the Product Catalog Management module, handling addition, update, and removal of products.

- Organization: Categorized by product types (e.g., nuts, dried berries) for easy

navigation.

3. Orders:

- Data Structure: Order Database in MySQL.
- Storage: Order details, including customer information, products ordered, and order status.
- Processing: Order Processing module handles the processing of customer orders, updates inventory, and manages payments.
- Organization: Stored chronologically, supporting historical records and order tracking.

Inventory:

- Data Structure: Real- time inventory maintained in system memory.
- Storage: Tracks the availability of Garments products.
- Processing: Dynamically updated during order processing to reflect changes in stock.
- Organization: Organized to ensure real- time availability status.

Temporary Payment Information:

- Data Structure: Temporary storage for secure payment transactions.
- Storage: Holds payment details during the payment processing stage.
- Processing: Used during the payment confirmation process.
- Organization: Ensures secure handling of payment information.

Logs and History:

- Data Structure: Integrated into respective databases.
- Storage: Captures logs and historical records for user activities, orders, and system events.
- Processing: Supports tracking and analysis of user interactions and system performance.
- Organization: Stored to facilitate audit trails and historical data retrieval.

Databases and Data Storage Items:

- MySQL Database for user profiles, orders, and product catalog.
- Product Catalog Data Store for efficient product management.
- System- level memory for real- time inventory tracking.
- Temporary storage for secure payment transactions.

Algorithm & Implementation

This is a description of algorithms and implementation:

Customer Interactions:

- User authentication (password hashing, secure session management)
- Session management (secure storage and retrieval of session data)

Product Management:

- Catalog management (product listings, categorization, search)
- Inventory control (real- time stock tracking and updates)

Orders Processing:

- Shopping cart management (adding, removing, modifying items)
- Order placement and fulfillment (order processing, status management, fulfillment workflows)

Payment Handling:

- Payment gateway integration (secure communication with payment providers)
- Transaction handling (reliable payment processing, data integrity)

Implementation:

Frontend Development:

- Technology stack: HTML, CSS, JavaScript, and a chosen frontend framework
- User interface implementation (product displays, shopping cart, order management)
- User interactions and client- side logic

Backend Development:

- Technology stack: chosen server- side language and framework
- RESTful API development for communication between frontend and backend
- Backend logic implementation (user authentication, product management, order processing)
- Database Implementation
- Database selection (MySQL, or other suitable options)
- Database schema design
- Data access logic, CRUD operations, and transaction management

Software requirements traceability matrix

This section should contain a table that summarizes how each software requirement has been met in this document. The tabular format permits one- to-one and one- to- many relationships to be shown.

Table 7 Requirements Traceability Matrix

Req. Number	Ref. Item	Design Component	Component Items
FR01	Class Diagram	User	CreateUser(), AuthenticateUser(), GetUserSession()
FR02	Class Diagram	Customer	signUp() Login()
FR03	Class Diagram	Orders	PlaceOrder(), CancelOrder(), ConfirmOrder(), ProcessPayment(paymentDetails)
FR04	Class Diagram	Shopping Cart	addToCart(product),removeItem(product), checkout()
FR05	Class Diagram	Order Details	Cancel Order()

Human interface design

Homepage:

- Clean and modern layout: The reference image features a spacious layout with large product visuals and minimal text. This creates a clean and modern aesthetic.
- Hero banner: The image prominently features a banner showcasing a selection of fruits. You could use this space to highlight seasonal offerings, new products, or special promotions.
- Navigation: The reference image has a simple navigation bar at the top with clear categories like "Nuts & Garments s," "Chocolates," and "Gift Hampers." Consider maintaining this simplicity and clarity for easy user browsing.
- Search bar: The reference image includes a search bar for users to find specific products. This is a crucial feature to maintain.

Product Browsing:

- Product cards: Each product is displayed in a separate card with an image, name, price, and weight. This format is clear and user- friendly, so consider keeping it.
- Product filtering: The reference image doesn't show any filtering options. You could consider adding filters by category, price range, brand, or dietary restrictions to help users narrow down their choices.

Product Details:

- Detailed information: The reference image only shows product names and prices. You could add more details like descriptions, nutritional information, recipes, and reviews to provide a richer user experience.
- Call to action: The reference image has a "Buy Now" button on each product card. This is a clear call to action, so consider keeping it consistent.

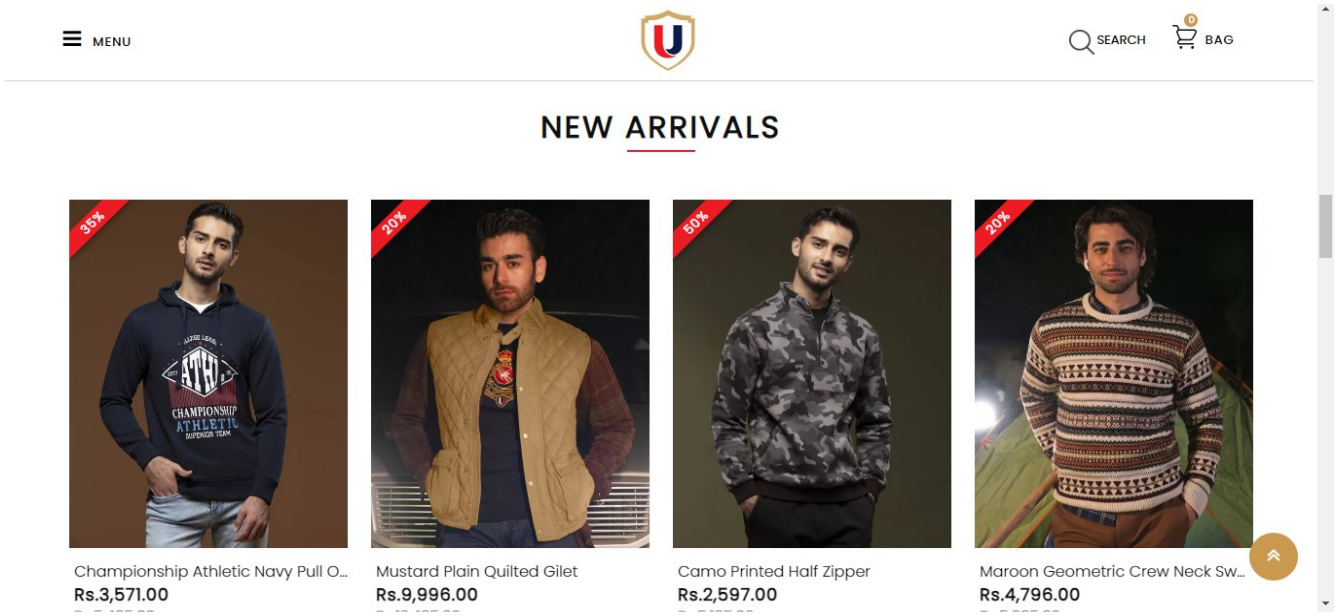
Additional considerations:

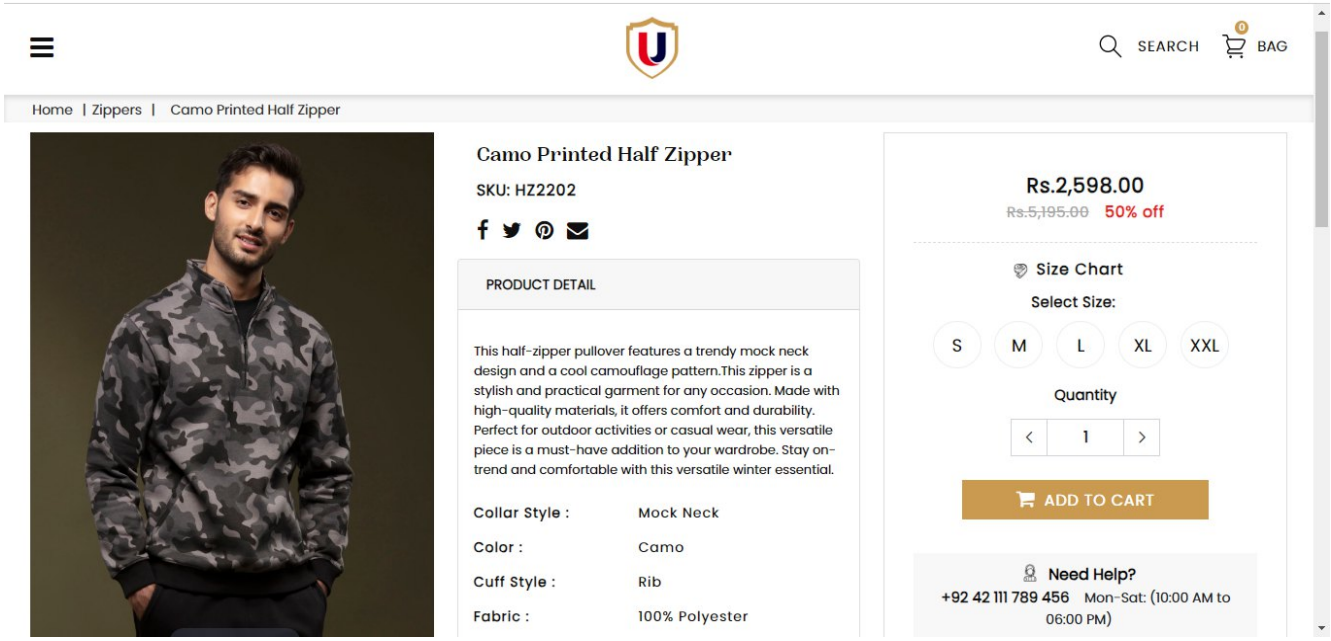
- Mobile responsiveness: Ensure the website adapts seamlessly to different screen sizes and devices for optimal mobile user experience.
- Branding: The reference image doesn't showcase any specific branding elements. If you have a logo or color palette in mind, incorporate it to create a cohesive brand identity.
- Call to action throughout the site: Include clear calls to action throughout the

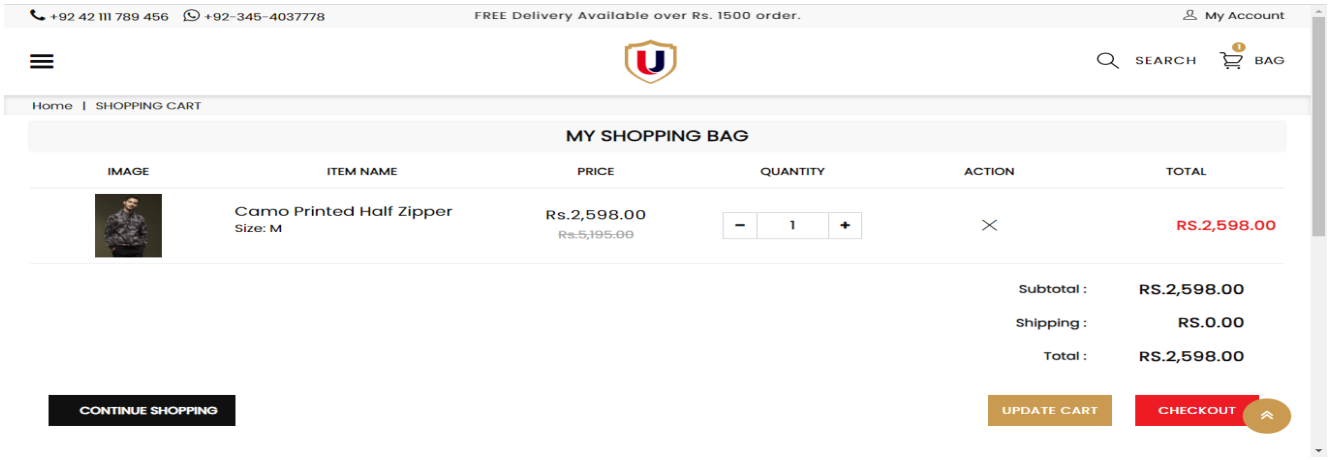
website, such as "Add to Cart" buttons on product pages or "Checkout" buttons in the cart.

- Describe the functionality of the system from the user's perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

Screen images







Appendix I

- How to design using UML (OOP): For guidance please follow the instructions mentioned in the link: http://agilemodeling.com/artifacts/
- How and when to design ER diagrams: For guidance please follow the instructions mentioned in the link: http://people.inf.elte.hu/nikovits/DB2/Ullman_The_Complete_Book.pdf
- Data flow diagrams: For guidance please follow the instructions mentioned in the link and book:

ohttp://www.agilemodeling.com/artifacts/dataFlowDiagram.htm
oSoftware Engineering – A Practitioner‟s approach by Roger Pressman

- Architecture diagram: For guidance please follow the instructions mentioned in the link and book:

olan Sommerville – Software Engineering 9th Edition– Chapter 6