*Software Requirements Specification*

*-*

**Author: Harish M**

1. **Abstract…………………………………………………………………………………………………….3**
2. **Requirements…………………………………………………………………………………………….3**
   1. Functional Requirements…………………………………………………………………………………………3
   2. Non – Functional Requirements………………………………………………………………………………4
3. **Design………………………………………………………………………………………………………4**
   1. High level design………………………………………………………………………………………………….….4
      1. System Architecture……………………………………………………………………………………...4
      2. Home and search flow........................................................................5
      3. Purchase and checkout flow................................................................6
   2. Low level design........................................................................................7
4. **Flowchart............................................................................................................7**
5. **Use case diagram................................................................................................9**
6. **Class case diagram..............................................................................................9**
7. **Sequence diagram.............................................................................................11**
8. **Test cases..........................................................................................................12**
9. **Abstract**

The online shopping system has become an integral part of modern commerce, offering convenience and accessibility to consumers worldwide. This abstract outlines the design and implementation of an online shopping system, aiming to provide a seamless and user-friendly experience for customers while offering robust functionality and efficiency for administrators.

The system facilitates the browsing, selection, and purchase of products through an intuitive web interface accessible via desktop and mobile devices. Users can create accounts, browse product categories, view detailed product information, add items to their shopping carts, and proceed through a secure checkout process.

.

1. **Requirements**
   1. **Functional Requirements**

**User Registration and Authentication**:

* Users should be able to register for an account with the system.
* Users should be able to log in securely using their credentials.
* Password reset functionality should be available for users who forget their passwords.

**Product Browsing and Searching:**

* Users should be able to browse products by category, brand, or other criteria.
* Users should be able to search for products using keywords or filters.
* Product listings should include detailed information such as descriptions, prices, images, and availability.

**Shopping Cart Management:**

* Users should be able to add products to their shopping carts.
* Users should be able to view and modify the contents of their shopping carts.
* Users should be able to proceed to checkout to complete their purchases.

**Checkout Process:**

* Users should be guided through a secure checkout process.
* Users should be able to enter shipping and billing information.
* Multiple payment options (credit/debit card, PayPal, etc.) should be supported.
* Users should receive confirmation emails after completing their orders.

**Order Management:**

* Users should be able to view their order history and track the status of their orders.
* Administrators should be able to manage orders, including processing payments and updating order statuses.
* Users should be able to cancel or modify orders within a specified timeframe.

**User Account Management:**

* Users should be able to update their account information, such as email addresses and shipping addresses.
* Users should be able to manage their saved payment methods.
* Administrators should have access to user management features, such as account activation/deactivation and role assignment.

**Product Management:**

* Administrators should be able to add, edit, and remove products from the system.
* Product listings should support features such as product variations (sizes, colors), discounts, and promotions.
* Administrators should be able to manage product inventory and update product availability.

**Reviews and Ratings:**

Users should be able to leave reviews and ratings for products they have purchased.

Product listings should display average ratings and the number of reviews for each product.

* 1. **Non – Functional Requirements**

**Performance:**

* The system should be able to handle a high volume of concurrent users and transactions without significant degradation in response time.
* Page load times should be optimized to ensure a smooth and responsive user experience.
* Database queries and server-side processing should be optimized to minimize latency.

**Security:**

* User data, including personal information and payment details, should be stored securely using encryption and hashing techniques.
* The system should implement secure authentication mechanisms to prevent unauthorized access to user accounts.
* Payment transactions should be conducted over secure channels (e.g., HTTPS) to protect sensitive information from interception.

**Usability:**

* The user interface should be intuitive and easy to navigate, with clear labels, consistent layout, and minimal clutter.
* User feedback mechanisms, such as ratings and reviews, should be provided to help users make informed purchasing decisions.
* Accessibility features should be implemented to ensure that the system is usable by people with disabilities.

**Scalability:**

* The system architecture should be designed to scale horizontally and vertically to accommodate increasing numbers of users and transactions.
* Load balancing and caching strategies should be implemented to distribute traffic evenly across multiple servers and improve performance under heavy loads.

**Availability:**

* The system should be highly available, with minimal downtime for maintenance and upgrades.
* Redundancy and failover mechanisms should be implemented to ensure uninterrupted service in the event of hardware or software failures.
* Monitoring and alerting systems should be in place to detect and respond to performance issues and outages promptly.

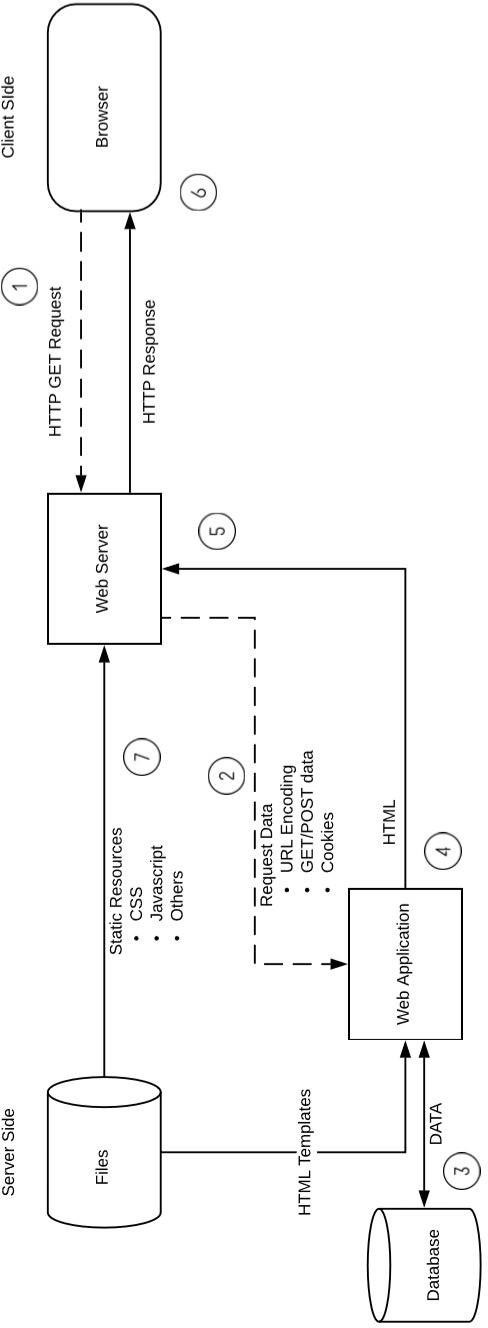
**Compliance:**

* The system should comply with relevant regulations and standards, such as GDPR (General Data Protection Regulation) for data privacy and PCI DSS (Payment Card Industry Data Security Standard) for handling payment information.
* Compliance with industry best practices and security guidelines should be ensured to protect user data and maintain trust.

1. **Design**

**3.1. High level design**

* + 1. **System Architecture**



* + 1. **Home and Search flow**

In the online shopping system, two user interfaces (UIs) are provided: the home screen and the search page. The home screen offers personalized or general recommendations based on whether the user is returning or new. Meanwhile, the search page enables users to view search results based on entered text.

The management of multiple suppliers is facilitated through Inbound Services, which interact with supplier systems via Kafka events to retrieve updated data. This data, including new suppliers or inventory items, is then ingested into the system. The Item Service listens to Kafka for new item data, storing it in MongoDB due to its unstructured nature. Additionally, a Search Consumer processes and formats new item data for storage in an ElasticSearch database, chosen for its text search efficiency and fuzzy search capabilities, thus enhancing the user experience by ensuring prompt searchability of newly added items.

The Search Service, powered by ElasticSearch, provides APIs for product filtering, sorting, and searching, ensuring users only see deliverable items. It interfaces with a Serviceability and TAT service to check delivery feasibility from the warehouse to the user’s location, including route viability and estimated delivery dates. Concurrently, a Kafka service linked to a Spark streaming consumer generates real-time analytics on consumer behavior and trends, which feed into a Hadoop cluster for advanced recommendation algorithms. These insights inform a Spark cluster to communicate with a recommendation service, offering personalized product suggestions based on user activity and preferences across different product categories.

Another crucial component is the User Service, acting as a repository for all users and providing APIs for fetching, updating, adding, and deleting users from the system. It is backed by a MySQL database and maintains a Redis cache. When the search service needs to fetch a user’s pin code for communication with the serviceability service, the user service first checks Redis. If the information is not found in Redis, it queries the MySQL database, retrieves the user’s information, stores it in Redis for future use, and returns it to the search service, ensuring efficient retrieval of user data.

* + 1. **Purchase and checkout flow**

The purchase and checkout flow in the online shopping system is a critical aspect of the user experience, ensuring that users can easily and securely complete their purchases. It begins with users adding desired products to their shopping cart while browsing the website. Once satisfied with their selections, users proceed to the checkout process, where they may be prompted to sign in to their accounts or register if they are new users. During checkout, users provide shipping information, review their order summary, and select their preferred payment method. After entering payment details and confirming their order, users receive a confirmation message or email with details about their purchase, including an order confirmation number and estimated delivery date. Throughout the entire flow, the system prioritizes clarity, simplicity, and security, guiding users through each step and ensuring a seamless and satisfactory shopping experience.

* 1. **Low level design**

**User Interface (UI):**

* This component includes the web pages and forms that users interact with.
* It consists of pages for user registration, login, product browsing, shopping cart management, checkout, and order confirmation.
* UI elements such as buttons, input fields, and dropdown menus facilitate user interactions.

**Backend Server:**

* The backend server handles business logic, data processing, and communication with external services.
* It consists of several modules:
* Authentication Module: Responsible for user authentication and authorization. Handles user registration, login, and session management.
* Product Management Module: Manages product information, including retrieval, storage, and updates.
* Cart Management Module: Handles shopping cart operations such as adding/removing items and calculating totals.
* Order Management Module: Manages order processing, including order creation, validation, and fulfillment.
* Payment Gateway Integration: Integrates with external payment gateways to facilitate payment processing.
* Database Access Layer: Provides access to the database for storing and retrieving user data, product information, and order details.

**Database:**

* The database stores persistent data required by the system.
* It includes tables for storing user accounts, product details, shopping cart contents, and order information.
* SQL or NoSQL databases can be used, depending on the requirements and scalability needs of the system.

**External Services:**

* The system may interact with external services for functionalities such as:
* Payment processing: Integrates with payment gateways (e.g., PayPal, Stripe) for processing credit/debit card payments.
* Shipping services: Integrates with shipping carriers (e.g., UPS, FedEx) to calculate shipping costs and generate labels.
* Product information: Integrates with external APIs or databases to fetch additional product details or reviews.

**Security Mechanisms:**

* The system implements security measures to protect user data, prevent unauthorized access, and ensure secure transactions.
* This includes encryption of sensitive data (e.g., passwords, payment information), secure communication over HTTPS, and protection against common security threats

**Error Handling and Logging:**

* The system incorporates mechanisms for handling errors and logging relevant information for debugging and auditing purposes.
* Error handling routines catch and handle exceptions gracefully to prevent system crashes and ensure a smooth user experience.
* Logging mechanisms record events, errors, and user activities for analysis and troubleshooting.

1. **Flowchart**



**Start:** The process begins.

**Enter the Website:** The user accesses the website from their browser.

**Registered:** If the user is already registered, they sign in; if not, they sign up.

**Search for Book:** On the homepage, the user searches for the book of their choice.

**Select Book:** The user selects their preferred book.

**Buy New or Old:** A decision point asking if the user wants to buy a new book.

* If yes, the user selects a new book, adds it to the cart, and reviews the order.
* If no, the user selects an old book and enters shipping information.

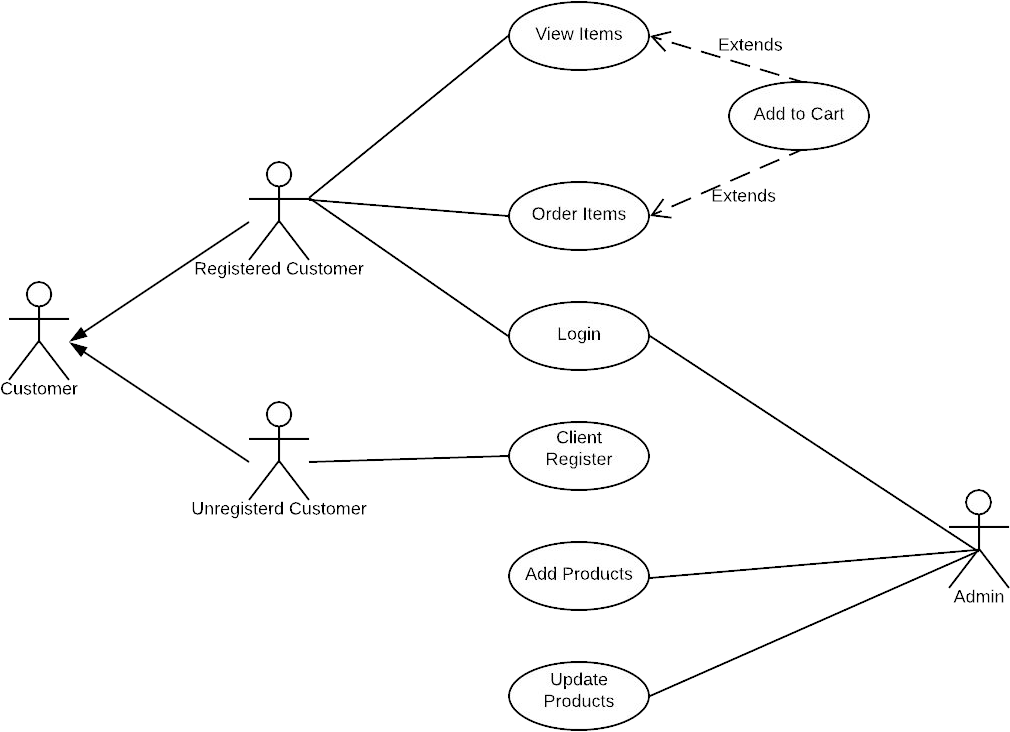
**Information Correct:** Another decision point to confirm the shipping information.

* If no, the user goes back to change the information.
* If yes, the user proceeds to select a payment method.

**Place Order:** The user places the order.

**End:** The conclusion of the process.

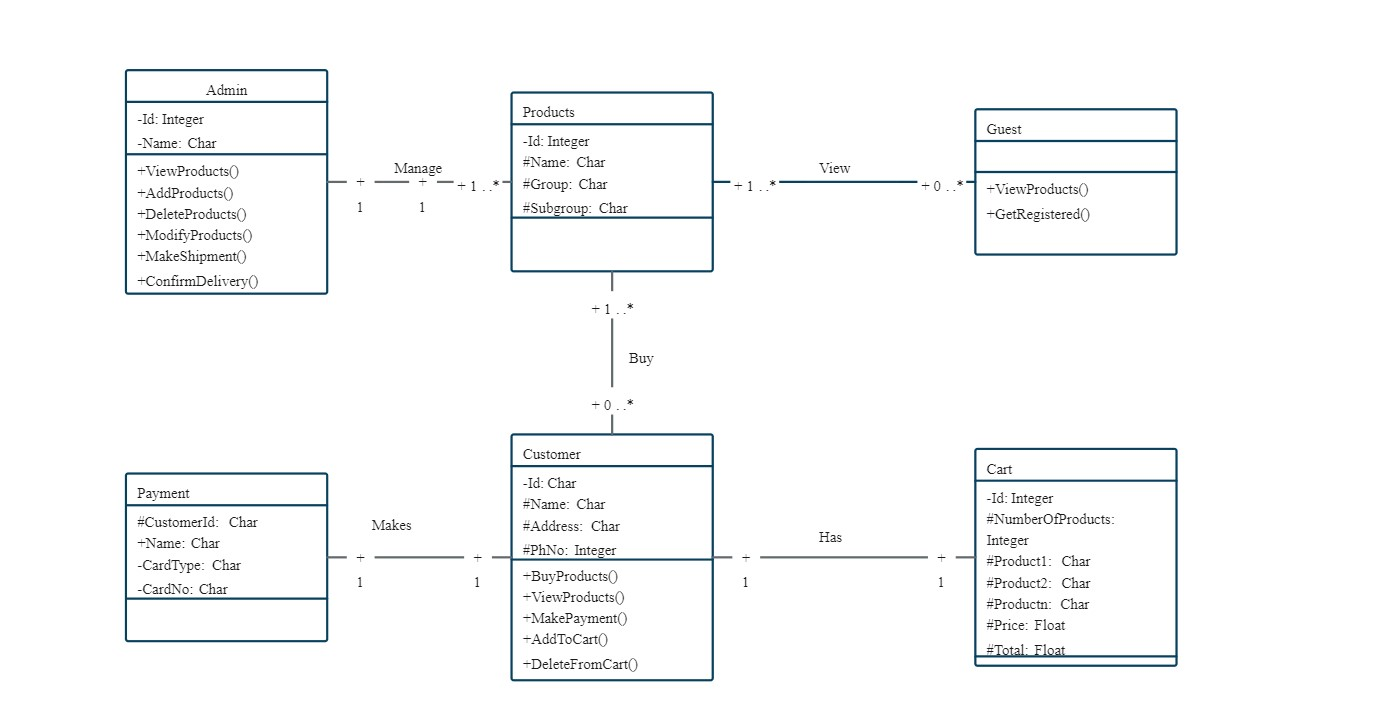
1. **Use Case diagram**



The use case diagram outlines the process flow from customer registration to product checkout, including various user types and authentication methods

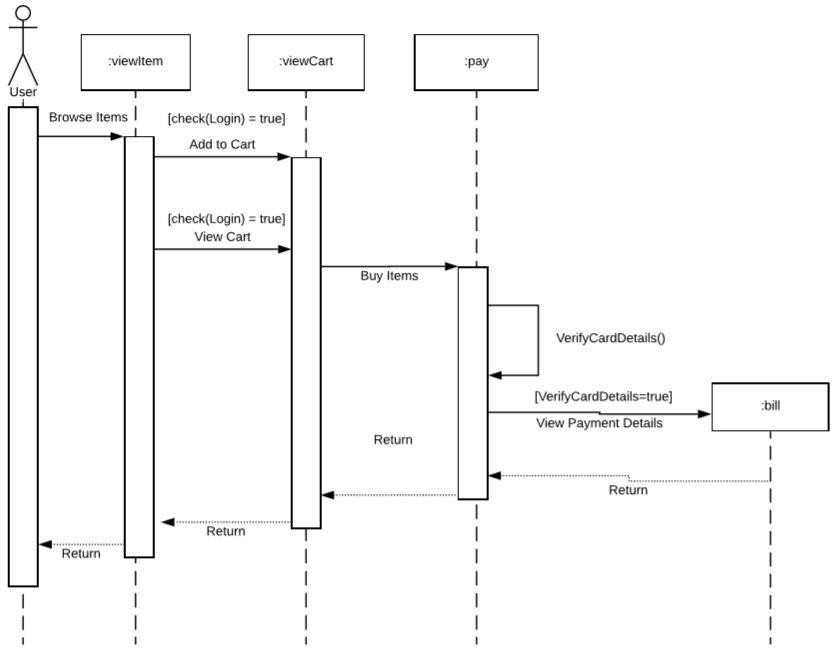
1. **Class diagram**

Online Shopping System Class Diagram provides software developers with a visual representation of the components and their relationships used in the development of an online shopping system. It shows how the classes interact with each other when a customer adds an item to their cart, pays for the order, and receives their purchase. The diagram also shows how classes manage customers, products, and orders. This clear overview helps software developers plan and implement a secure and efficient online shopping process. It helps identify potential issues and design solutions to enhance the customer experience.



* **Customer Class:** Represents a customer with attributes like name, address, email, credit card information, shipping information, and account balance. It includes methods to register, login, and update the profile.
* **User Class:** A base class for users with attributes for user ID, password, login status, and registration date. It has a method to verify login credentials.
* **Administrator Class:** Represents an administrator with attributes for admin name and email. It includes a method to update the catalog.
* **Shopping Cart Class:** Contains attributes for cart ID, product ID, quantity, and date added. Methods include adding cart items, updating quantity, viewing cart details, and checking out.
* **Shipping Info Class:** Contains attributes for shipping ID, type, and cost. It is associated with the Order class.

1. **Sequence diagram**



The sequence of interactions is as follows:

* The customer browses items using the Browse Interface.
* The Browse Interface queries the Item DB for item details.
* The customer selects items to purchase.
* The Purchase Interface retrieves the selected item details and price information.
* The customer initiates the checkout process.
* The Checkout process interacts with the Credit Card Authorization system to authorize the purchase.
* Upon successful authorization, an immediate notification is sent to the customer, followed by an email confirmation

1. **Test Cases**

**User Registration and Authentication:**

* Test Case 1: Verify that users can successfully register for an account with valid information.
* Test Case 2: Verify that users cannot register with invalid or incomplete information (e.g., missing email address or password).
* Test Case 3: Verify that users can log in with valid credentials.
* Test Case 4: Verify that users cannot log in with invalid credentials.

**Product Browsing and Searching:**

* Test Case 5: Verify that users can browse products by category.
* Test Case 6: Verify that users can search for products using keywords.
* Test Case 7: Verify that product listings display accurate and up-to-date information (e.g., descriptions, prices, availability).

**Shopping Cart Management:**

* Test Case 8: Verify that users can add products to their shopping carts.
* Test Case 9: Verify that users can view and modify the contents of their shopping carts.
* Test Case 10: Verify that users cannot proceed to checkout with an empty shopping cart.

**Checkout Process:**

* Test Case 11: Verify that users can proceed through the checkout process smoothly.
* Test Case 12: Verify that users can enter shipping and billing information accurately.
* Test Case 13: Verify that users receive confirmation emails after completing their orders.

**Order Management:**

* Test Case 14: Verify that users can view their order history and track the status of their orders.
* Test Case 15: Verify that administrators can manage orders, including processing payments and updating order statuses.
* Test Case 16: Verify that users can cancel or modify orders within a specified timeframe.

**User Account Management:**

* Test Case 17: Verify that users can update their account information.
* Test Case 18: Verify that users can manage their saved payment methods.
* Test Case 19: Verify that administrators can manage user accounts, including activation/deactivation and role assignment.

**Product Management:**

* Test Case 20: Verify that administrators can add, edit, and remove products from the system.
* Test Case 21: Verify that product listings support features such as product variations, discounts, and promotions.
* Test Case 22: Verify that administrators can manage product inventory and update product availability.

**Security:**

* Test Case 23: Verify that user data is stored securely and encrypted.
* Test Case 24: Verify that payment transactions are conducted over secure channels (e.g., HTTPS).
* Test Case 25: Verify that authentication mechanisms are secure and prevent unauthorized access.

**Performance:**

* Test Case 26: Verify that the system can handle a high volume of concurrent users and transactions without significant degradation in response time.
* Test Case 27: Verify that page load times are optimized for a smooth and responsive user experience.
* Test Case 28: Verify that database queries and server-side processing are optimized to minimize latency.

**Usability:**

* Test Case 29: Verify that the user interface is intuitive and easy to navigate.
* Test Case 30: Verify that user feedback mechanisms, such as ratings and reviews, are accessible and functional.
* Test Case 31: Verify that accessibility features are implemented to ensure usability for people with disabilities.