Software Requirements Specification   
–

Document ID: ASP-CMS-SRS-02.doc

Author(s): SRI NIDHI NIVAS K

Version: 0.1

Date: March 20, 2024

[1. Objective and Scope 3](#_toc104)

[2. Project End Users 3](#_toc113)

[3. Functional Requirements 3](#_toc115)

3.1 Login to the System………………………………………………………………………………………………………….4

3.2 Search Bar……………………………………………………………………………………………………………….……….4

3.3 Filter Option………………………………………………………………………………………………………………………4

3.4 Whish List……………………………………………………………………………………………………….…………………4

3.5 Cart Details……..……………………………………………………………………………………………………………….5

3.6 Transactional Details………………………………………………………………………………………………….…….5

3.7 Checkout Process………………………………………………………………………………………………………………5

**4. Non-Functional Requirements…..…………….………………………………………………………..…6**

4.1 Security………………………….……………………………………………………………………………..……………….…6

4.2 Performance………………………………………………………………………………………………………………………6

4.3 Maintainability……………………………………………………………………………………………………………………7

**5. Design……………………………………..….……………….…………………………………………..……...7**

5.1 High level design ………………………………………………………………………………………………………….….7

5.2 Low level design…………………………………………………………….……………………………………………..….8

**6. Diagram…………………………………..….……………… …………………………………………….……..9**

6.1 Flow Chart Diagram……………………………………………………………………………………………………….….9

6.2 Sequence Diagram…………………………………………………………………………………………………….……10

6.3 Class Diagram………………………………………………………………………………………………………………...11

6.4 Use Case Diagram……………………………………………………………………………………………………………12

**7. Test Cases………………………………..….…………………………………………………..……………..12**

**1. Objective and Scope**

The core aim of an online shopping app is to offer users a seamless, personalized experience for browsing and purchasing products conveniently. It focuses on enhancing customer satisfaction, optimizing operations, and leveraging user feedback for continual improvement, while also serving as a vital tool for brand building and business expansion.

The scope of the software includes the following basic features:

**· Product Range:** Offering a wide variety of products across different categories to cater to diverse customer preferences and needs.

**· User Experience**: Providing a seamless and intuitive experience with easy navigation, quick searches, and smooth checkout processes.

**· Payment Options:** Supporting multiple secure payment methods to accommodate user preferences and ensure hassle-free transactions.

**· Security Measures:** Implementing robust security measures to protect users' personal and financial information, building trust and credibility.

**· Personalization:** Offering personalized recommendations based on user behavior and preferences to enhance engagement and encourage repeat purchases.

**· Analytics and Insights**: Utilizing data analytics to gain insights into user behavior, purchasing patterns, and trends, optimizing performance and marketing strategies.

**2. Project End Users**

The end users nivas of an online shopping app are typically consumers or shoppers who utilize the app to browse, select, and purchase products or services offered by various retailers or sellers. These

end users can include individuals of all demographics and geographic locations who prefer the convenience and accessibility of shopping online through a mobile app or website.

**3. Functional Requirements**

**3.1 Login to the system**

Each and every user should be authenticated with a User Name and Password to login into the system.

Validations for User Name and Password.

**User Name:** It accepts only Alphabets, Numbers, Dot (.) symbol and Underscore (\_) symbol.

**Password:** It can be anything of the users’ choice

**3.2 Search bar**

Search bars in online shopping apps are like personal assistants, helping users find products quickly and easily. By typing keywords, users can avoid endless scrolling and pinpoint exactly what they need. Features like auto-suggestions and filters make searches even more efficient, while voice search adds convenience, especially for mobile users.

**· Enhancing User Experience**

Placing the search bar prominently and ensuring it works well on all devices is crucial. Gathering user feedback helps improve functionality and ensures the search bar remains a valuable tool. Overall, a well-designed search bar enhances navigation and satisfaction, boosting user engagement and loyalty.

**3.3 Filter option**

Filter options in online shopping carts allow users to refine their search results based on specific criteria such as price, brand, size, and color. This feature streamlines the browsing experience, helping users quickly find products that match their preferences. By simplifying product selection and reducing search time, filter options enhance user satisfaction and contribute to higher engagement and conversion rates.

.

**3.4 Wish list**

**· Convenient Product Storage:** The wish list feature enables users to save products they're interested in for future reference or potential purchase.

**· Personalized Collection**: It serves as a personalized collection of items that caught the user's attention during browsing sessions, allowing them to easily revisit and review these products later.

**· Decision Flexibility**: Users can use the wish list when they're undecided about a purchase or want to wait for a better deal or availability, providing flexibility in decision-making.

**· Time saving:** By adding items to their wish list, users avoid the need to search for them again, saving time and effort in the shopping process.

**· Notifications and Reminders**: Wish lists often provide notifications or reminders about price changes, stock availability, or special promotions related to the saved items, helping users make informed decisions.

**· Increased Conversion Rates:** The wish list feature can potentially increase conversion rates for online retailers by encouraging users to revisit and purchase saved items when they're ready to buy.

**3.5 Cart Details**

In online shopping, the cart details section is a crucial step where users can review and manage their chosen items before completing a purchase. It offers a clear breakdown of selected products, including names, images, descriptions, and any chosen variations like size or color. Users can adjust quantities as needed and view pricing information, ensuring transparency with costs before proceeding to checkout.

Additionally, the cart details may include shipping details, providing estimated delivery times and available methods, aiding users in planning their purchase. Ultimately, the cart details section streamlines the purchasing process, offering users a comprehensive overview of their order and facilitating a seamless online shopping experience.

**3.6 Transaction Details**

In online shopping, transaction details are vital for users to understand their completed purchases. They include:

**· Order Summary:** A brief overview of items purchased.

**· Total Cost:** The overall amount spent, including taxes, shipping fees, and discounts.

**· Payment Information:** Details on the payment method used.

**· Shipping Details:** Information on delivery address, estimated arrival, and chosen method

**· Order Status:** Updates on the current stage of the order, from processing to delivery.

**· Receipts/Invoices:** Access to digital documentation for record-keeping. These detailsensure transparency and help users track their orders efficiently, contributing to a smooth online shopping experience.

**3.7 Customer Support**

Customer support plays a crucial role in ensuring a positive and satisfactory online shopping experience for users. It encompasses various aspects aimed at assisting and addressing the needs and concerns of customers throughout their journey:

**· Assistance with Queries:** Customer support provides assistance to users by addressing their queries, concerns, and inquiries regarding products, orders, payments, shipping, and returns. This assistance may be provided through various channels such as live chat, email, or phone support.

**· Problem Resolution:** In case of any issues or problems encountered during the shopping process, customer support facilitates prompt resolution. This includes addressing issues such as order discrepancies, damaged or defective items, billing errors, or delivery delays, ensuring that users receive timely and effective solutions.

**· Guidance and Advice:** Customer support offers guidance and advice to users, helpingthem make informed decisions about their purchases. This may involve providing product recommendations, sizing assistance, or information about promotions and discounts.

**· Technical Support:**In addition to addressing product-related queries, customer support also provides technical assistance to users experiencing difficulties with the website or app. This includes troubleshooting issues such as login problems, navigation issues, or payment processing errors.

**· Returns and Refunds:** Customer support facilitates the returns and refunds process, guiding users through the steps required to return unwanted or faulty items and obtain refunds or exchanges. They ensure that users understand the return policy and assist them in initiating return requests smoothly.

**· Feedback and Suggestions:** Customer support serves as a channel for users to provide feedback, suggestions, and complaints about their shopping experience. This valuable input helps the online retailer identify areas for improvement and enhance the overall customer experience

**4. Non-Functional Requirements**

**4.1 Security**

For user data and transactions to be shielded from malicious activity, unauthorised access, and security breaches, security criteria are essential. These specifications cover authorization, encryption, authentication, and threat mitigation for typical security vulnerabilities.

To avoid interception or unwanted access, sensitive data, such as user passwords, payment information, and personal information, should be encrypted.User authorization and authentication should be implemented by the system to regulate access to particular capabilities according to user roles and permissions. It is important to take precautions against typical security risks including SQL injection, cross-site scripting (XSS) attacks, and other weaknesses that can jeopardise the system's confidentiality and integrity.

**4.2 Performance**

Performance standards make sure the Online Shopping System runs smoothly and offers a positive customer experience. These specifications centre on how quickly the system loads, how responsive it is, and how scalable it is to accommodate different user traffic volumes.

When a user clicks a link, adds anything to their basket, or completes the checkout process, the system ought to react quickly.In order to minimise waiting times for consumers and improve their browsing and buying experience, pages and content should load rapidly. It should be possible for the system to grow in size to handle spikes in user traffic during busy times, like holidays or special occasions, without noticeably degrading performance.

**4.3 Maintainability**

Platforms for online shopping are frequently built using a modular architecture, in which various parts and modules are rationally arranged and divided. Since modifications to individual modules don't impact the system as a whole, this modular architecture makes maintenance and updates easier.

Developers can successfully communicate, monitor changes to the software, and roll backto earlier versions as needed by implementing version control systems like Git. This guarantees that changes and updates may be handled methodically and that rollback processes are available in case they are required. Detailed documentation helps developers, administrators, and other stakeholders understand the platform's functionality, operation, and structure. This documentation includes code documentation, system architectural diagrams, and user manuals. Systems with thorough documentation are simpler to maintain and debug.

**5. Design**

**5.1 High level design**

An outline of the main parts and architecture of the online shopping system is given in the High-Level Design (HLD). It focuses on the system's general structure and the interactions between its many modules.

* **System architecture:** It describes the system's general architecture, including the placement of its parts and their interactions. Recognises the use of architectural patterns or styles, such as layered architecture, client-server architecture, or micro services architecture.
* **Data Flow and Control Flow:** It shows the data flow through the system from input to output and processing. Defines the control flow, which includes the steps in the system's decision-making process and order of actions.
* **External Interfaces:** It describes the interfaces that connect to outside services, systems, or users. Describes the way the system communicates with outside resources including databases, user interfaces, and APIs.
* **Scalability and Performance Considerations:** Focuses on scalability needs, like the capacity to support future expansion or handle growing loads. Takes into account speed optimisations such load balancing, parallel processing, and caching techniques.
* **Security and Reliability:** It describes the security steps that are taken to guard the system from risks likeunauthorised access and data leaks. Focuses on aspects of reliability that are necessary, like redundancy, fault tolerance, and disaster recovery plans.
* **Technology stack:** It describes the tools, frameworks, and technologies that will be utilised to put the system into place. Explains why each technology was chosen in light of the restrictions and requirements of the system.
* **Development approach**: It specifies the technique or development methodology to be used, such as DevOps, Waterfall, or Agile. Describes the stages, deliverables, and milestones of the development lifecycle.
* **Documentation and Communication:** It makes certain that the high-level design is thoroughly documented and shared with development teams, architects, and stakeholders in an efficient manner.

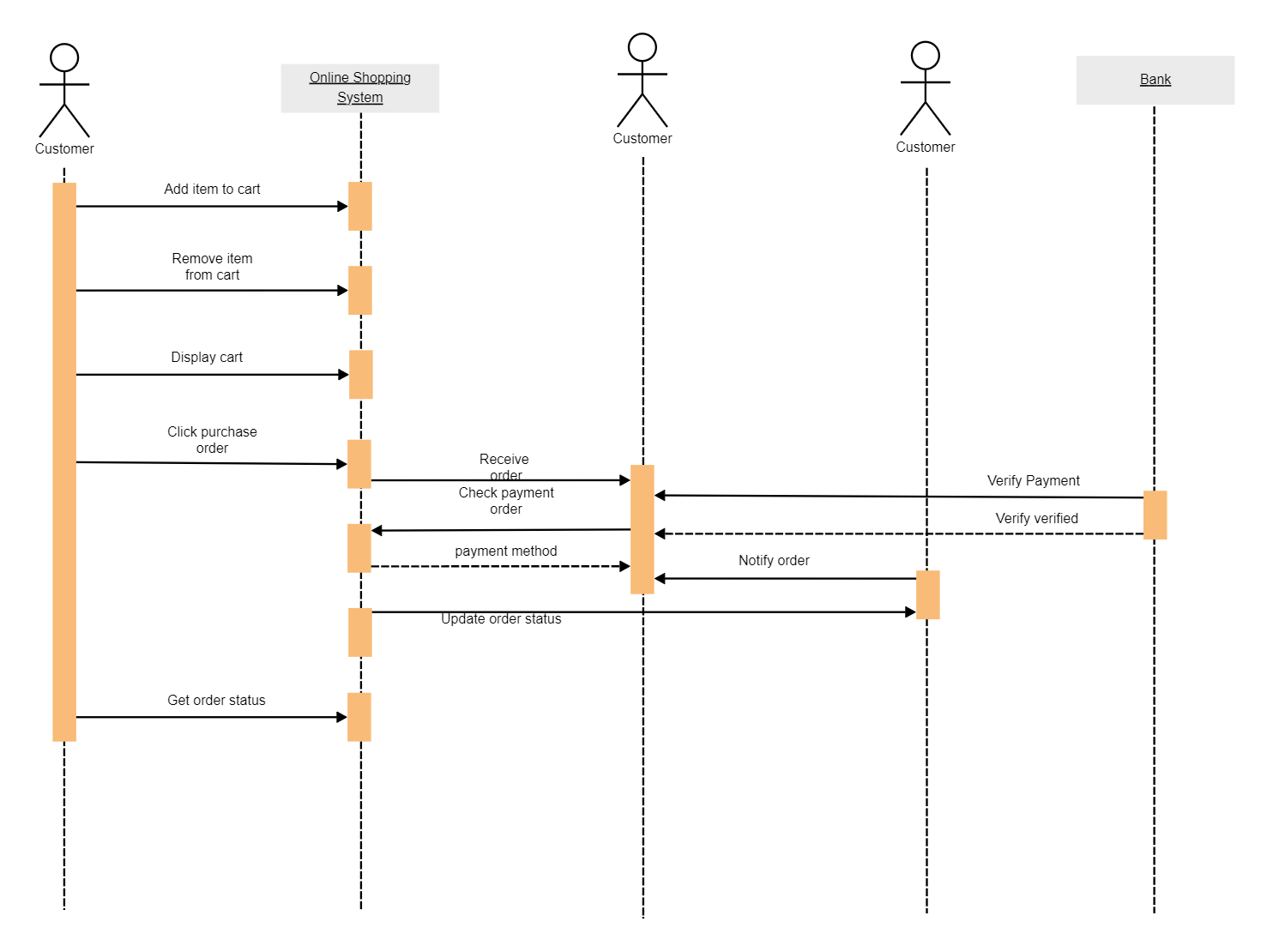
**5.2 Low level design**

Low-Level Design (LLD), which focuses on the internal operations of specific components defined in the high-level design, goes deeper into the implementation specifics of the system. It offers thorough specifications that programmers can utilise as a guide for creating and implementing code.

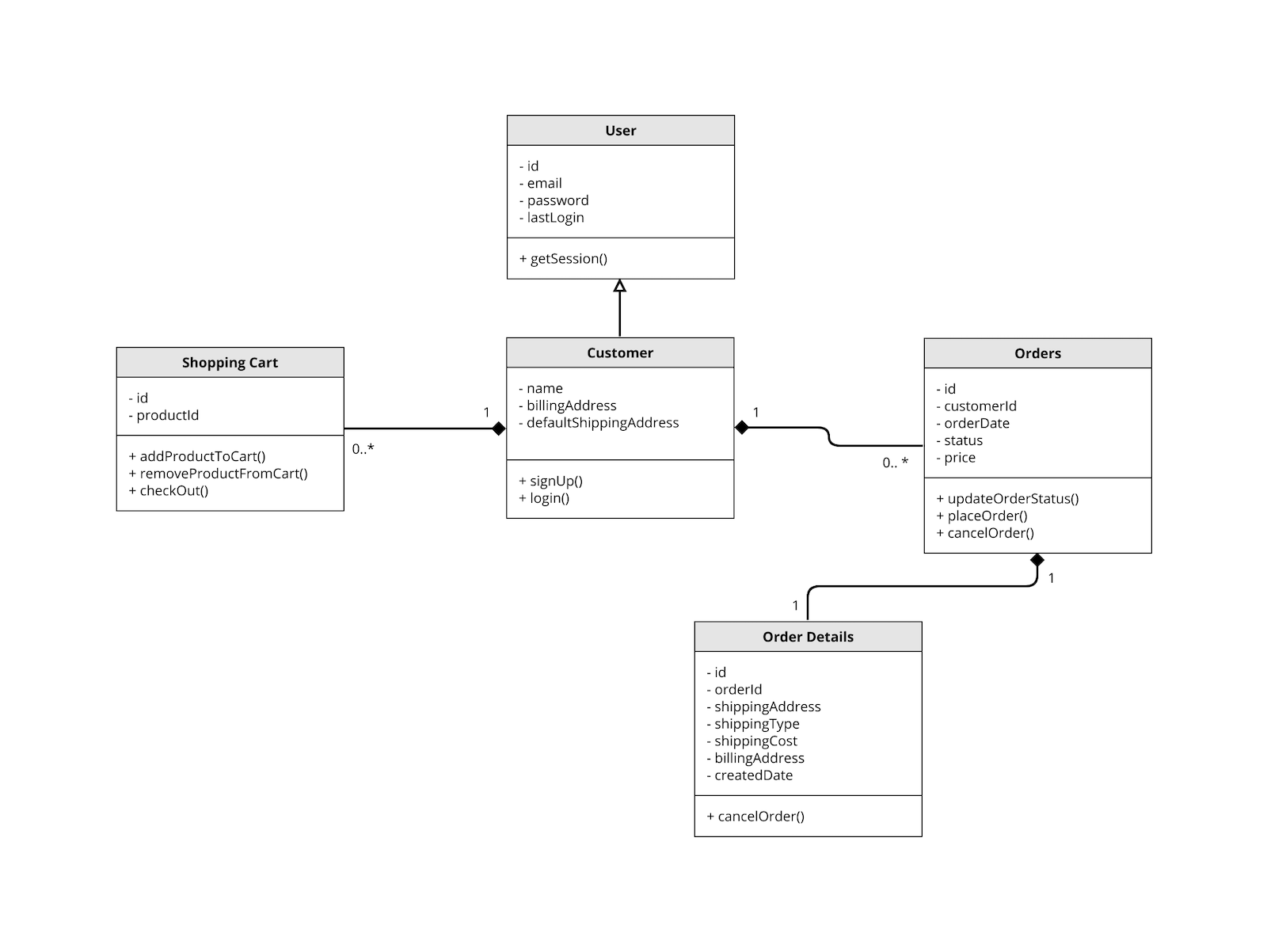
* **Module Details:** It divides each significant element included in the high-level design into more manageable modules or sections. Describes the functions, inputs, outputs, and processing logic of each module.
* **Data Structures and Algorithms**: It describes the data structures that will be utilised by the system to store and handle data. Explains algorithms and algorithms for basic tasks like data manipulation, sorting, and searching. Maximises the performance and efficiency of data structures and algorithms.
* **Database Design:** Creating the tables, columns, constraints, and relationships that make up the database structure. Outlines triggers, stored procedures, and SQL queries for retrieving and manipulating data. Describes the many kinds of data, indexing schemes, and normalisation methods that are used to guarantee data performance and integrity.
* **Class Design:** It show the classes and objects in the system. Creates interfaces and class hierarchies to encourage reusability and encapsulate behaviour.
* **Sequence Diagrams:** It shows how data and control are transferred across objects or components in a variety of contexts. Describes how parameters and messages are passed between objects in both synchronous and asynchronous communication.
* **Interfaces and APIs:** Creating the APIs and interfaces to allow external users to communicate with the system. Details each API endpoint's parameters, error handling, return types, and method signatures.
* **Security and Implementation:** It implements the security mechanisms in places to guard the system against typical security flaws and attacks. Uses security best practices to stop security breaches, like parameterized queries, output encoding, and input validation

**6. Diagram**

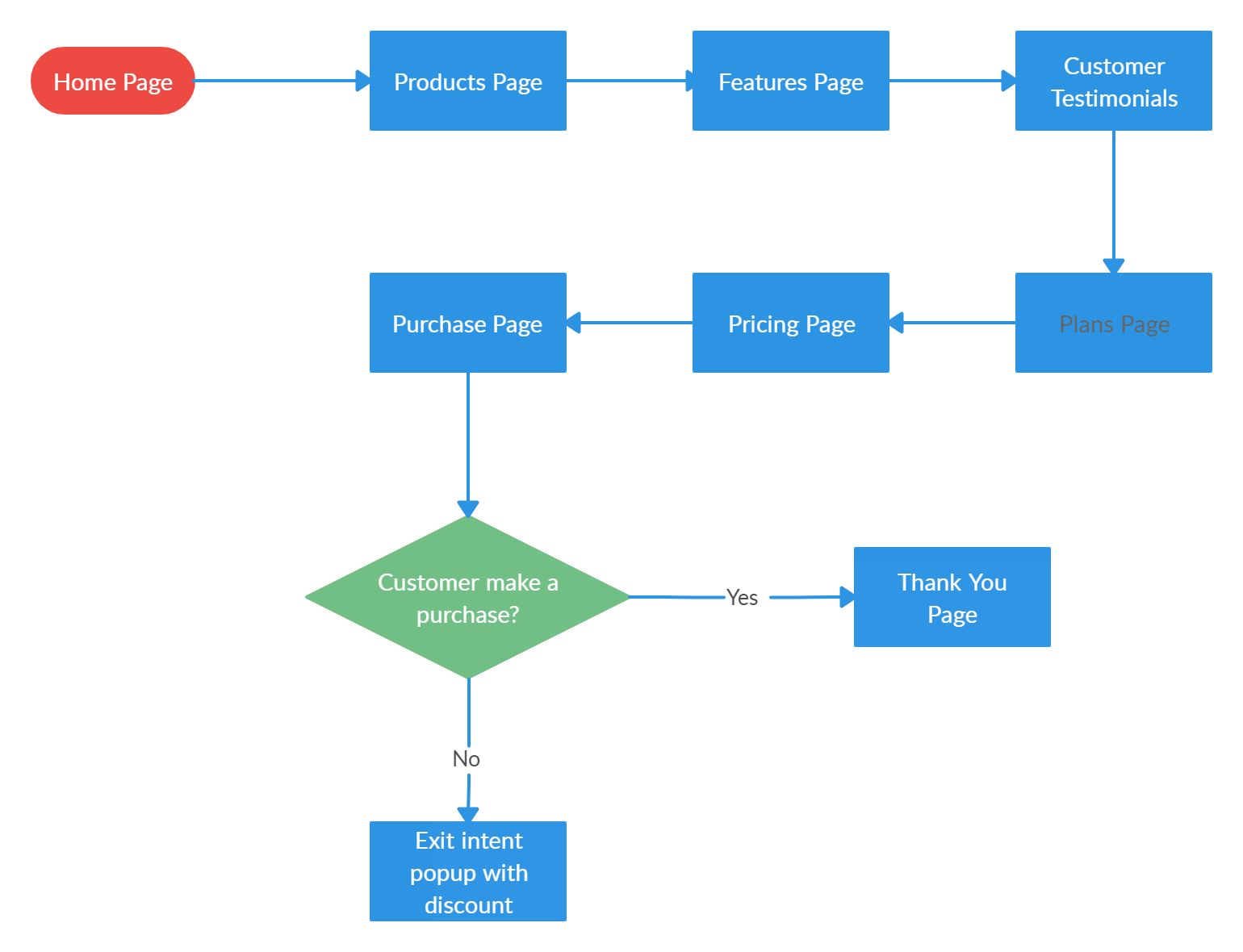
**6.1 Sequence Diagram**

****

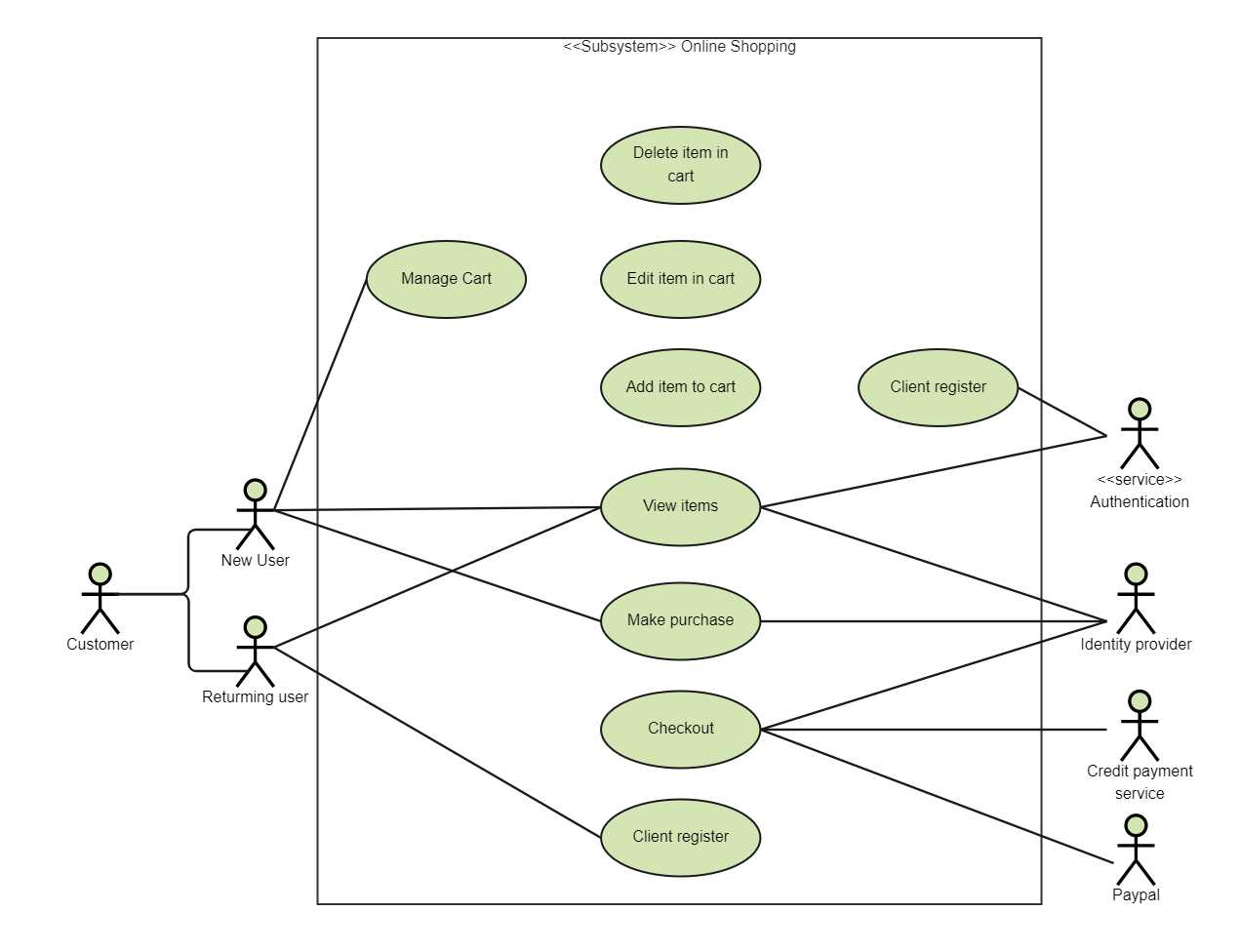
**6.2 Class Diagram**

****

**6.3 Flow Chart Diagram**

****

**6.4 Use Case Diagram**

****

**7. Test Cases**

**1.User Registration:**

**Test case 1**: Verify successful registration with valid credentials.

**Test case 2**: Verify registration failure with an invalid email format.

**Test case 3:** Verify registration failure with a weak password.

**Test case 4:** Verify registration failure with an existing username.

**2.User Login:**

Test case 5: Verify successful login with correct credentials.

**Test case 6**: Verify login failure with incorrect password.

**Test case 7**: Verify login failure with a non-existing username.

**Test case 8:** Verify login failure for a deactivated account.

**3.Product Browsing:**

**Test case 10:** Verify users can view the list of available products.

**Test case 11:** Verify users can filter products by category.

**Test case 12:** Verify users can sort products by price or popularity.

**Test case 13:** Verify users can view product details by clicking on a product.

**Test case 14:** Verify users can view product reviews and ratings.

**4.Shopping Cart:**

**Test case 15**: Verify users can add products to the shopping cart.

**Test case 16**: Verify users can remove products from the shopping cart.

**Test case 17:** Verify users can update the quantity of products in the shopping cart.

**Test case 18:** Verify the shopping cart retains items after user logout/login.

**5.Checkout Process**:

**Test case 19:** Verify users can proceed to checkout with items in the shopping cart.

**Test case 20:** Verify users can enter shipping details.

**Test case 21**: Verify users can select a payment method.

**Test case 22:** Verify users receive an order confirmation after successful checkout.