Software Requirements Specification (SRS) for eCommerce Application

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive overview of the requirements for the development of an eCommerce application. This document will serve as a guide for developers, testers, and other stakeholders involved in the project.

1.2 Scope

The eCommerce application is intended to facilitate online buying and selling of products and services. It will include features such as user registration, product catalog, shopping cart, payment processing, order management, and user account management.

* 1. Definitions, Acronyms, and Abbreviations

eCommerce: Electronic Commerce

SRS: Software Requirements Specification

UI: User Interface

API: Application Programming Interface

2. System Overview

2.1 System Description

The eCommerce application will be a web-based platform accessible through standard web browsers. It will provide a user-friendly interface for both customers and administrators.

2.2 System Features

The key features of the eCommerce application include:

User Registration and Authentication

Product Catalog Management

Shopping Cart and Checkout

Payment Gateway Integration

Order Management

User Account Management

Search and Filter Functionality

Rating and Review System

Admin Dashboard

3. Functional Requirements

3.1 User Registration and Authentication

3.1.1 Users shall be able to register for an account using a valid email address and password.

3.1.2 The system shall provide secure authentication mechanisms, such as two-factor authentication.

3.2 Product Catalog Management

3.2.1 The application shall display a categorized product catalog with images, descriptions, and prices.

3.2.2 Admins shall be able to add, edit, or remove products from the catalog.

3.3 Shopping Cart and Checkout

3.3.1 Users shall be able to add products to a shopping cart.

3.3.2 The application shall calculate the total cost and provide a secure checkout process.

3.4 Payment Gateway Integration

3.4.1 The system shall integrate with a secure and reliable payment gateway for processing transactions.

3.4.2 Multiple payment methods (credit card, PayPal, etc.) shall be supported.

3.5 Order Management

3.5.1 Users shall be able to view and track their order status.

3.5.2 Admins shall have access to an order management system for processing and updating orders.

3.6 User Account Management

3.6.1 Users shall be able to update their profiles and change account settings.

3.6.2 Admins shall have the ability to manage user accounts and permissions.

3.7 Search and Filter Functionality

3.7.1 The application shall provide a search bar for users to find products efficiently.

3.7.2 Users shall have the option to filter products based on various criteria.

3.8 Rating and Review System

3.8.1 Users shall be able to rate and review products.

3.8.2 The application shall display average ratings and reviews for each product.

3.9 Admin Dashboard

3.9.1 Admins shall have access to a dashboard for monitoring site analytics, managing products, and handling user-related activities.

4. Non-Functional Requirements

4.1 Performance

4.1.1 The application shall support a minimum of 1000 concurrent users without performance degradation.

4.2 Security

4.2.1 User data shall be stored securely using encryption techniques.

4.2.2 The application shall comply with industry-standard security practices to prevent unauthorized access.

4.3 Reliability

4.3.1 The system shall have a backup and recovery mechanism to ensure data integrity.

4.4 Usability

4.4.1 The user interface shall be intuitive and user-friendly.

4.5 Scalability

4.5.1 The application architecture shall support easy scalability for handling increased user loads.

5. Constraints

5.1 Technology

5.1.1 The application shall be developed using [specific technology stack (MERN)].

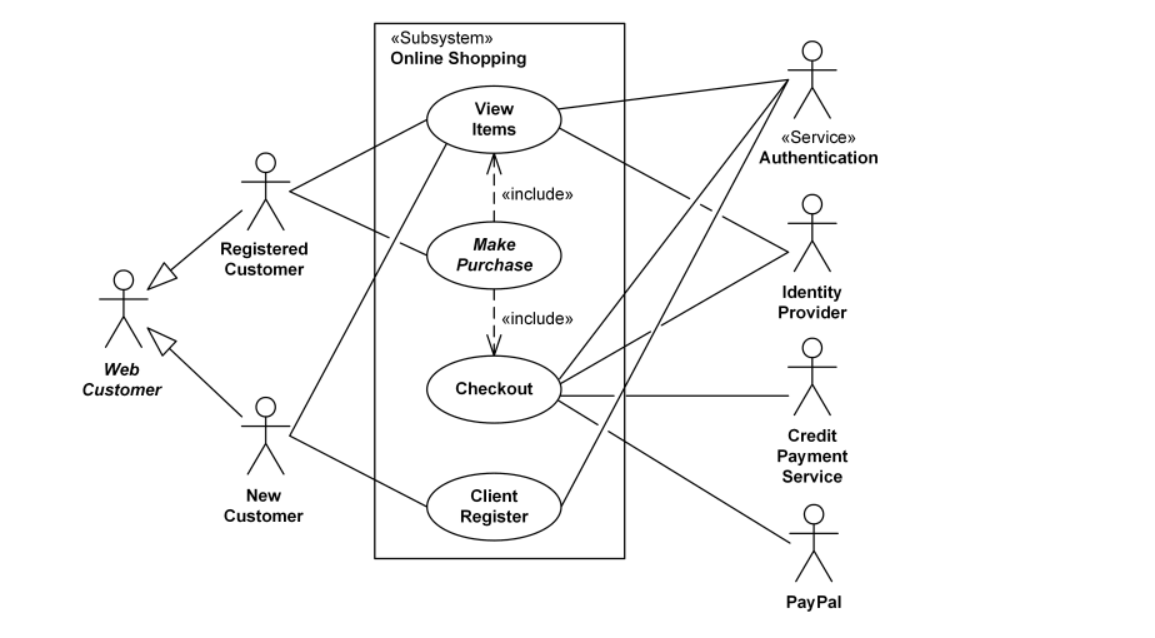
5.2 Legal

5.2.1 The system shall comply with all relevant data protection and privacy laws.

6. Unified Modeling Language (UML) Diagrams

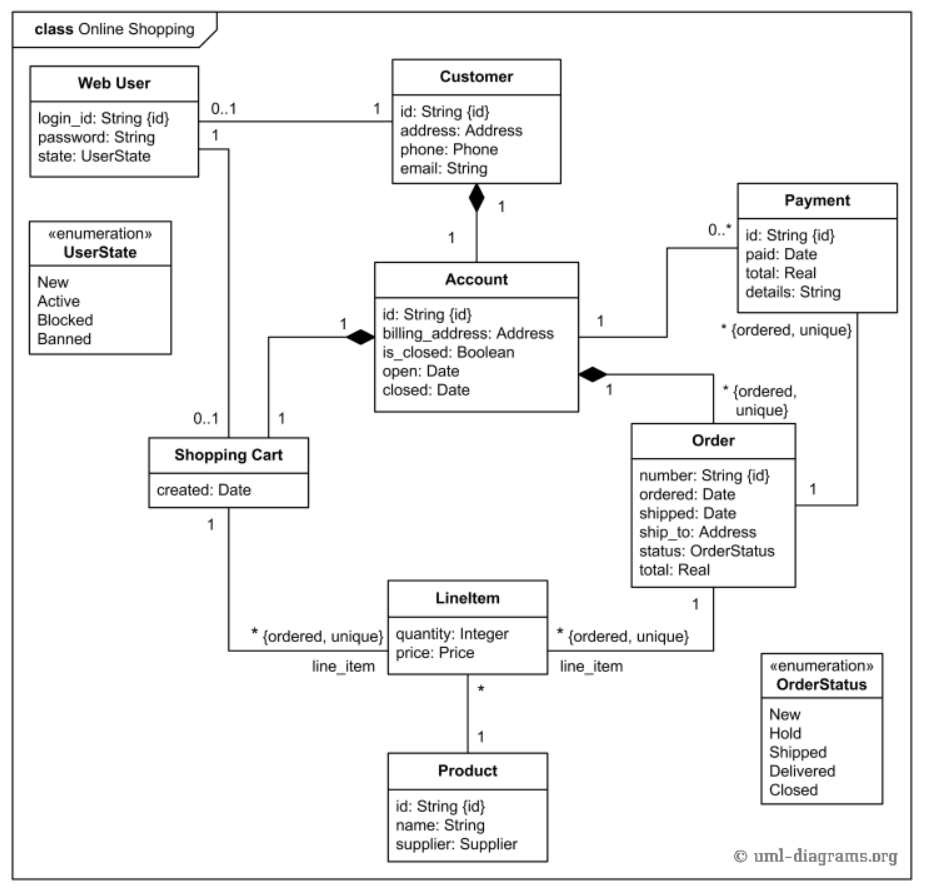
6.1 Use Case Diagram

6.1.1 A use case diagram shall illustrate the interactions between actors and the system, depicting major functionalities from the user's perspective.



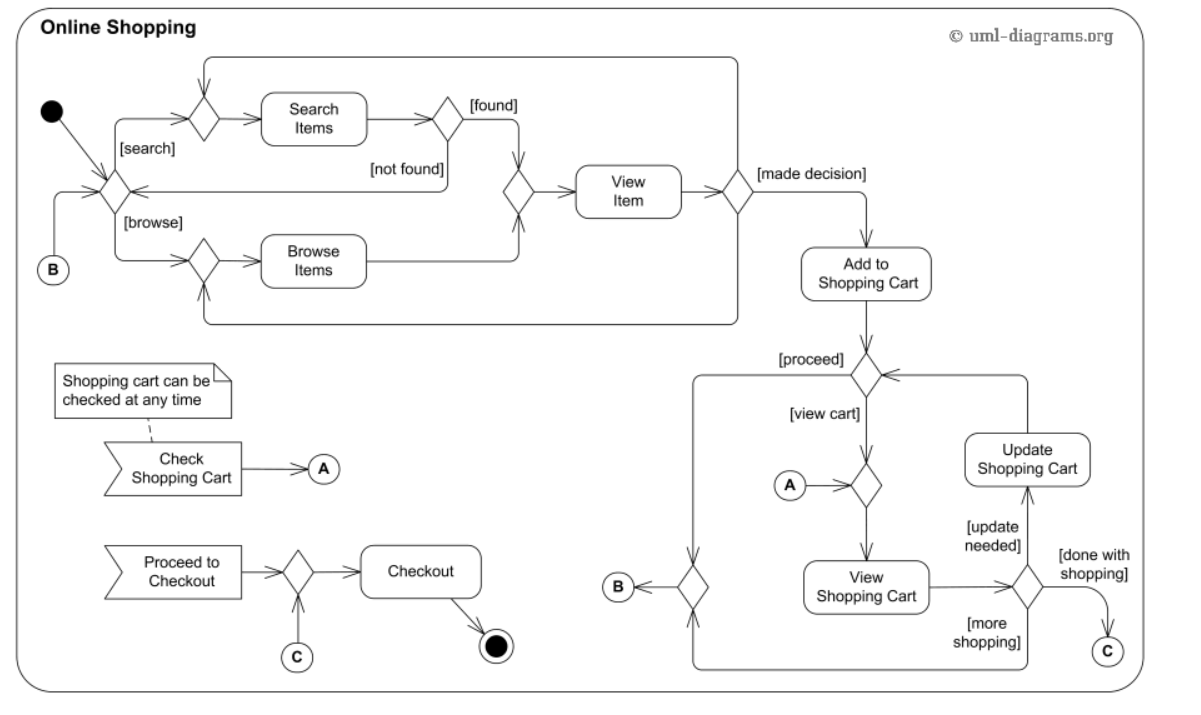
6.2 Class Diagram

6.2.1 The class diagram shall model the relationships and attributes of major system classes, such as User, Product, Order, etc.



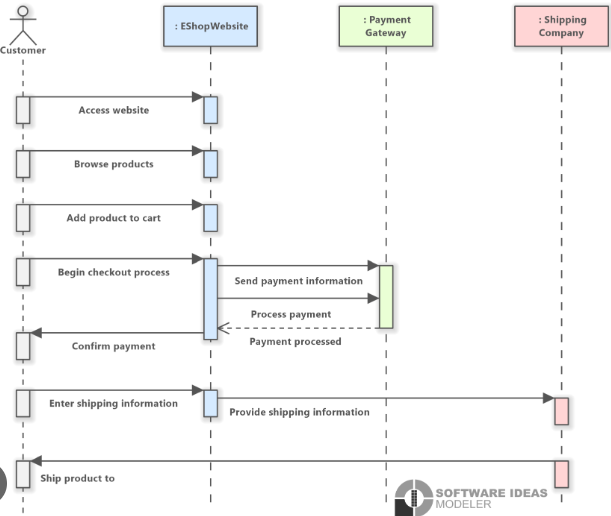
6.3 Activity Diagram

6.3.1 An activity diagram shall outline the workflow of key processes within the application, such as the shopping cart management and order fulfillment process.



10.3 Sequence Diagrams

10.3.1 Sequence diagrams shall represent the flow of interactions between system components, focusing on specific scenarios like user registration, product purchase, and order processing.



7. External Interface Specification

7.1 User Interfaces

7.1.1 The application shall have a responsive and visually appealing user interface accessible through standard web browsers.

7.1.2 Users shall interact with the system through intuitive navigation menus and forms.

7.2 Hardware Interfaces

7.2.1 The application shall be hosted on servers meeting the following specifications: [Server specifications].

7.2.2 The system shall be compatible with devices running standard web browsers.

7.3 Software Interfaces

7.3.1 The application shall integrate with the following third-party software components:

Payment Gateway API

Database Management System

7.4 Communication Interfaces

7.4.1 The system shall use HTTPS for secure communication between the client and server.

8. Technical Specification

8.1 Architecture

8.1.1 The application shall follow a three-tier architecture, comprising presentation, application logic, and data storage layers.

8.1.2 The back-end shall be developed using [Backend Framework], and the front-end shall use [Frontend Framework].

8.2 Database Design

8.2.1 The system shall use a relational database (e.g., MySQL) for storing user data, product information, and order details.

8.2.2 Database schema and relationships are outlined in the Database Design document

8.3 Programming Language

8.3.1 The application shall be primarily developed using [Programming Language].

8.4 Security Measures

8.4.1 Data transmission shall be encrypted using HTTPS.

8.4.2 Passwords shall be securely hashed before storage.

8.5 Performance Optimization

8.5.1 Caching mechanisms shall be implemented to enhance system performance.

8.5.2 Images and static resources shall be optimized for quick loading.

8.6 Testing

8.6.1 The application shall undergo unit testing, integration testing, and system testing.

8.6.2 Test cases and test scenarios are documented in the Testing Specification [Reference to Testing Specification Document].

9. Appendices

9.1 Glossary

9.2 References