Software Requirements Specification

for

E-COMMERCE WEBSITE

Thakur College of Engineering and Technology

Jagdu Singh Charitable Trust's (Regd.)
THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY
Estd. 2001
NBA and NAAC Accredited

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

E-commerce is fast gaining ground as an accepted and used business paradigm. More and more business houses are implementing web sites providing functionality for performing commercial transactions over the web. It is reasonable to say that the process of shopping on the web is becoming commonplace.
The objective of this project is to develop a general-purpose e-commerce store where product like clothes can be bought from the comfort of home through the Internet.

1.2 Document Conventions

This SRS uses standard IEEE conventions. **Bold** text highlights keywords (e.g., **must**, **shall**, **optional**) and system entities. Requirement IDs (e.g., REQ-1.2.1) provide unique tracking. Priority (High, Medium, Low) is assigned to each requirement and inherited by sub-requirements unless explicitly stated otherwise.

1.3 Intended Audience and Reading Suggestions

This document serves developers, testers, project managers, and marketing teams. Readers should start with Section 1 (Overview) and 2 (General Description) for context, then proceed to Section 3 (Specific Requirements) based on their role: Developers focus on functional/non-functional requirements (3.2, 3.3), Testers on test cases (3.4), and Managers on scope/goals (1.4).

1.4 Product Scope

This SRS details the "GlobalMart E-commerce Platform," an online store enabling users to browse, search, and purchase products, while allowing admins to manage inventory and orders. Its goals are to expand market reach, increase online revenue by **20%,** and enhance customer experience, supporting the company's digital transformation strategy.

1.5 References

1. **IEEE Std 29148-2018**, *Systems and Software Engineering — Life Cycle Processes — Requirements Engineering*, IEEE, 2018.
2. **IEEE Std 830-1998**, *Recommended Practice for Software Requirements Specifications*, IEEE.
3. **OWASP Foundation**, *OWASP Top 10 Web Application Security Risks*, Latest Edition, https://owasp.org.
4. **PCI Security Standards Council**, *PCI DSS v4.0 – Payment Card Industry Data Security Standard*, 2022.
5. **General Data Protection Regulation (GDPR)**, European Union, Regulation (EU) 2016/679.

2. Overall Description

2.1 Product Perspective

The E-commerce platform is a self-contained, web-based system designed to serve as a digital storefront for both customers and administrators. It operates within a distributed multi-tier architecture, consisting of a front-end user interface, a back-end application server, and a relational database for secure data persistence. The system is intended to function as a standalone product but maintains critical external dependencies for full operationality, including integration with third-party payment gateways (e.g., PayPal, Stripe) for transaction processing and SMTP servers for automated email notifications, such as order confirmations and shipping updates.

2.2 Product Functions

For an e-commerce website, major product functions involve **User Management** (registration, login, profiles), **Product Catalog** (browsing, searching, filtering, viewing details), **Shopping Cart & Checkout** (adding, removing, managing items, payment, order confirmation), **Order Management** (tracking, history for users; processing for admins), and **Admin Features** (product/inventory management, customer service, reporting) – all designed for seamless shopping and backend control.

2.3 User Classes and Characteristics

For an e-commerce website, the user classes are categorized based on their interaction with the platform, privilege levels, and technical expertise.

Customers: This is the most critical user class. It consists of frequent and occasional shoppers with varying technical expertise. They use the product primarily for searching items, managing carts, and processing payments. Satisfying their needs for usability and security is the highest priority for the product's success.

Administrators: These are high-privilege users with advanced technical expertise. They focus on backend functions such as inventory management, order processing, and user account oversight. While fewer in number, their ability to maintain site integrity is vital.

Guest Users: These are first-time or infrequent visitors who browse the site without an account. They use a limited subset of features (product search and viewing). While they are a secondary priority compared to registered customers, a positive experience is necessary to convert them into permanent users.

Support Staff: This class includes customer service representatives with moderate privilege levels. They use specific communication and order-tracking tools to resolve issues. Their importance is moderate, focusing on post-purchase satisfaction.

2.4 Operating Environment

The e-commerce platform will operate in a high-availability cloud environment, such as AWS or Google Cloud, utilizing a Linux-based server architecture (Ubuntu 24.04 LTS or later). The hardware must support at least 8GB of RAM and quad-core processing to handle concurrent user sessions efficiently. On the client side, the application must be accessible via any modern web browser (Chrome 120+, Firefox 125+, Safari 17+, and Edge) on both desktop and mobile devices. The software must peacefully coexist with third-party integrations, including Stripe or PayPal for payment processing and SendGrid for email dispatch services.

2.5 Design and Implementation Constraints

Development is constrained by the requirement to use the MERN (MongoDB, Express, React, Node.js) stack to ensure compatibility with the organization's existing maintenance capabilities. The system must comply with GDPR for data privacy and PCI DSS v4.0 standards for secure financial transactions. Performance constraints require that all page loads complete within 2.5 seconds under standard broadband conditions. Additionally, the system must support HTTPS protocols for all communications and follow the Airbnb JavaScript Style Guide to maintain code quality for future handovers.

2.6 User Documentation

The software will be delivered with a comprehensive suite of digital documentation. This includes an integrated **Online Help System** with searchable FAQs for customers, a **PDF User Manual** detailing administrative functions for staff, and a **Quick Start Video Tutorial** for new vendors. All documentation will be provided in HTML5 and PDF formats, adhering to Microsoft Manual of Style standards to ensure technical clarity and accessibility.

2.7 Assumptions and Dependencies

Assumptions:

1. Users have access to a stable internet connection and modern web browsers.
2. Third-party services such as payment gateways and email servers remain available and reliable.
3. Administrators and support staff have basic technical knowledge to operate the system.
4. Cloud service providers ensure sufficient scalability and uptime.
5. Legal and security standards (GDPR, PCI DSS) remain consistent during development.
6. The MERN stack continues to receive long-term support.

Dependencies:

1. External payment gateways (Stripe/PayPal) for transaction processing.
2. Email services (SMTP/SendGrid) for notifications and alerts.
3. Cloud infrastructure for hosting, storage, and scalability.
4. Modern web browsers for frontend compatibility.
5. Third-party libraries and frameworks used in the MERN stack.

3. External Interface Requirements

3.1 User Interfaces

The e-commerce system will provide a responsive, web-based graphical user interface (GUI) for customers, administrators, and support staff.

1. **User Authentication Interface:**
   Users must be able to register, log in, recover passwords, and view or update their profile details.
2. **Product Browsing Interface:**
   Users must be able to view all listed products, detailed product pages (PDPs) with images, descriptions, reviews, categories, and attributes.
3. **Search and Navigation Interface:**
   Users must have a search bar and filtering options based on price, rating, style, and categories.
4. **Shopping Cart Interface:**
   Users must be able to add or remove items, update quantities, save items for later, and view a cart summary.
5. **Checkout Interface:**
   Users must be able to view cart items, manage delivery addresses, complete a secure multi-step checkout, and receive order confirmation.
6. **Order Management Interface (User View):**
   Users must be able to view order history, track orders, and initiate returns.
7. **Customer Support Interface:**
   Users should have access to FAQs, contact forms, and live chat or call support options.
8. **Usability and Accessibility:**
   The interface must be simple, intuitive, user-centric, and accessible to users of all technical skill levels.
9. **Mobile Responsiveness:** The interface must be mobile-first, touch-friendly, and optimized for different screen sizes.

3.2 Hardware Interfaces

1. The system will operate on standard client devices such as desktops, laptops, tablets, and smartphones.
2. No specialized hardware is required beyond input devices (keyboard, mouse, touchscreens).
3. The application will run on cloud-hosted servers with sufficient CPU, memory, and storage to support high traffic.
4. Communication between client devices and servers will occur over standard internet connections.

3.3 Software Interfaces

1. **Database Interface:**
   The system will use MongoDB for storing user data, product details, orders, and transaction records.

2. **Payment Gateway Interface:**
   Integration with third-party payment services such as Stripe and PayPal for secure online payments.
3. **Email and Notification Services:**
   SMTP or third-party services (e.g., SendGrid) will be used to send order confirmations and notifications.
4. **Operating System and Frameworks:**
   The backend will run on a Linux-based OS using Node.js and Express, while the frontend will use React.
5. **Third-Party Libraries:**
   The system will depend on external libraries and APIs for UI components, authentication, and security features.

3.4 Communications Interfaces

1. The system will use **HTTP/HTTPS** protocols for all web communication.
2. All sensitive data transmission must be secured using **SSL/TLS encryption**.
3. The platform will support browser-based access and server-client communication over RESTful APIs.
4. Email communication will follow standard SMTP protocols.
5. The system must support reliable data transfer with minimal latency and proper synchronization between frontend and backend components.

4. System Features

4.1 User Authentication and Account Management

4.1.1 Description and Priority

This feature allows users to register, log in, recover passwords, and manage their profile information securely. **Priority:** High

4.1.2 Stimulus/Response Sequences

1. User enters registration details → System validates data → Account is created.
2. User enters login credentials → System authenticates → User is redirected to dashboard.
3. User requests password recovery → System sends recovery email.

4.1.3 Functional Requirements

REQ-1: The system shall allow users to register using valid email and password.
REQ-2: The system shall authenticate users during login.
REQ-3: The system shall allow users to recover forgotten passwords.
REQ-4: The system shall allow users to view and update profile details.
REQ-5: The system shall display error messages for invalid credentials.

4.2 Product Catalog and Search

4.2.1 Description and Priority

This feature enables users to browse products, view detailed product pages, and search or filter products. **Priority:** High

4.2.2 Stimulus/Response Sequences

1. User browses categories → System displays product listings.

2. User selects a product → System displays product details.
3. User applies filters → System updates product results.

4.2.3 Functional Requirements

REQ-6: The system shall display all available products with images and descriptions.
REQ-7: The system shall allow users to search products using keywords.
REQ-8: The system shall support filtering by price, rating, and category.
REQ-9: The system shall display product reviews and ratings.

4.3 Shopping Cart and Checkout

4.3.1 Description and Priority

This feature allows users to add products to a cart, manage items, and complete purchases securely.
Priority: High

4.3.2 Stimulus/Response Sequences

1. User adds product to cart → System updates cart summary.
2. User proceeds to checkout → System validates cart and address.
3. User confirms payment → System places order.

4.3.3 Functional Requirements

REQ-10: The system shall allow users to add and remove products from the cart.
REQ-11: The system shall allow users to update item quantities.
REQ-12: The system shall support a secure multi-step checkout process.
REQ-13: The system shall display order confirmation after successful checkout.

4.4 Payment Processing

4.4.1 Description and Priority

This feature handles secure online payments using third-party gateways. **Priority:** High

4.4.2 Stimulus/Response Sequences

1. User selects payment method → System redirects to payment gateway.
2. Payment is successful → System confirms order.
3. Payment fails → System displays error message.

.

4.4.3 Functional Requirements

REQ-14: The system shall integrate with Stripe and PayPal payment gateways.
REQ-15: The system shall encrypt payment-related data.
REQ-16: The system shall handle failed or cancelled transactions gracefully.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The system shall load all standard pages within **2.5 seconds** under normal network conditions.
2. Search and filtering results shall be displayed within **2 seconds**.
3. The checkout and payment process shall be completed without noticeable delays.

4. The system shall support **at least 10,000 concurrent users** during peak traffic.
5. The platform shall maintain **99.9% availability** excluding scheduled maintenance.

5.2 Safety Requirements

1. The system shall prevent accidental data loss during transactions.
2. User actions that may lead to irreversible operations (e.g. order cancellation) shall require confirmation.
3. Proper error handling shall be implemented to avoid system crashes.

5.3 Security Requirements

1. The system shall use **SSL/TLS encryption** for all data transmission.
2. User authentication shall be required for accessing personal and order-related information.
3. Payment information shall not be stored on the system servers.
4. The system shall comply with **PCI DSS** standards for payment security.
5. User data shall be protected in accordance with **GDPR** regulations.

5.4 Software Quality Attributes

1. **Usability:** The interface shall be simple and easy to use for all user types.
2. **Reliability:** The system shall operate continuously with minimal downtime.
3. **Maintainability:** The codebase shall be modular and easy to update.
4. **Scalability:** The system shall support future growth in users and data.
5. **Portability:** The application shall run on all major web browsers and devices.

5.5 Business Rules

1. Only administrators shall have permission to add, update, or remove products.
2. Customers must be registered and logged in to place orders.
3. Orders once confirmed cannot be modified by customers.
4. Refunds and returns shall follow predefined company policies.

6. Other Requirements

1. **Database Requirements:**
   The system shall use a NoSQL database (MongoDB) to store user profiles, product details, orders, and transaction metadata. Regular backups shall be maintained.
2. **Internationalization Requirements:**
   The system shall be designed to support multiple languages and currencies in future releases.
3. **Legal and Regulatory Requirements:**
   The application shall comply with applicable data protection and privacy laws such as GDPR and payment regulations such as PCI DSS.
4. **Reuse Objectives:**

The system architecture and components shall be modular to enable reuse in future projects or extensions.

Appendix A: Glossary

Term	Description
SRS	Software Requirements Specification
GUI	Graphical User Interface
MERN	MongoDB, Express, React, Node.js
PDP	Product Detail Page
API	Application Programming Interface
SSL/TLS	Secure communication protocols
GDPR	General Data Protection Regulation
PCI DSS	Payment Card Industry Data Security Standard

Appendix B: Analysis Models

The following analysis models may be used to represent the system design:

1. Use Case Diagrams for user interactions
2. Entity Relationship (ER) Diagrams for database structure
3. Data Flow Diagrams (DFD) for data movement
4. Class Diagrams for system architecture

Appendix C: To Be Determined List

1. Selection of final payment gateway configuration.
2. Final UI design and theme standards.
3. Return and refund policy rules.
4. Deployment region and cloud service configuration.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				