Software Requirements Specification

for

Amazon : E-Commerce Shopping System

Prepared by

Batch:A

Tanuja Gunjal	2023300071	tanuja.gunjal23@spit.ac.in
Alok Jadhao	2023300079	alok.jadhao23@spit.ac.in

Instructor: *Prof. Prasenjit Bhavathankar*

Course: Software Engineering

Date: 14/08/2025

1 Introduction

The Amazon E-Commerce Shopping System is an online platform designed to facilitate product browsing, purchasing, and delivery tracking for millions of users worldwide. This document outlines the system's functional and non-functional requirements to guide its development, testing, and maintenance. The section introduces the purpose, scope, audience, conventions, and references related to this SRS.

1.1 Document Purpose

This Software Requirements Specification (SRS) defines the requirements for the Amazon E-Commerce Shopping System (Revision 1.0). The system serves as Amazon's primary online retail platform, enabling customers to browse and purchase products from multiple sellers, and providing sellers with tools to manage their product listings and orders.

The document specifies the system's intended functionalities, operating environment, design constraints, and quality attributes. It will be used by developers to implement the system, testers to verify compliance with requirements, and stakeholders to ensure business objectives are met.
This SRS covers only the **E-Commerce Shopping Platform** and does not include other Amazon services such as AWS, Prime Video, or Kindle.

1.2 Product Scope

The Amazon E-Commerce Shopping System is a web and mobile application allowing customers to:

- Search and browse products by category, name, or keywords.
- Add products to a shopping cart or wishlist.
- Securely purchase products using integrated payment gateways.
- Track order status and receive delivery updates.
- Post reviews and ratings for purchased products.

Benefits and Objectives:

- **Convenience:** Provide users with 24/7 access to millions of products from anywhere in the world.
- **Efficiency:** Reduce the time from search to checkout through optimized navigation and search algorithms.
- **Security:** Protect customer data and transactions with encryption and authentication protocols.
- **Scalability:** Support high traffic during peak shopping seasons such as Black Friday or festive sales.

1.3 Intended Audience and Document Overview

Intended Audience:

- **Developers:** To understand system architecture and functional modules.
- **Testers:** To design test cases based on the specified requirements.
- **Project Managers:** To monitor progress and ensure requirement compliance.
- **UI/UX Designers:** To create interfaces that meet user needs.
- **Client/Professor:** To evaluate project feasibility and completeness.

Document Overview:

- **Section 1:** Introduction, purpose, scope, and audience.
- **Section 2:** Overall description including product perspective, functionality, and user types.
- **Section 3:** Specific functional and interface requirements.
- **Section 4:** Non-functional requirements such as performance, security, and quality attributes.
- **Section 5:** Additional system requirements.
- **Appendix A:** Data dictionary with variables, constants, and operations.

Recommended reading order: Start with **Section 1 & 2** for a general understanding, then move to **Section 3** for detailed requirements, followed by **Section 4** for performance and security details.

1.4 Definitions, Acronyms and Abbreviations

Term	**Definition**
AES	Advanced Encryption Standard
API	Application Programming Interface
AWS	Amazon Web Services
HTTPS	Hypertext Transfer Protocol Secure
OTP	One-Time Password
PCI DSS	Payment Card Industry Data Security Standard
SKU	Stock Keeping Unit
UI	User Interface

1.5 Document Conventions

Header

- Font Size: **10**
- Font Style: **Bold Italic**
- Font: **Times New Roman**

Heading

- Font Size: **20**
- Font Style: **Bold**
- Font: **Times New Roman**

Sub Heading

- Font Size: **15**
- Font Style: **Bold**
- Font: **Times New Roman**

Content

- Font Size: **12**
- Font: **Times New Roman**

1.6 References and Acknowledgments

- IEEE Std 830-1998, *IEEE Recommended Practice for Software Requirements Specifications*.

- Amazon Developer Documentation – https://developer.amazonservices.com/

- Amazon Help & Support – https://www.amazon.com/gp/help/customer/display.html

- Payment Card Industry Data Security Standard (PCI DSS) – https://www.pcisecuritystandards.org/

- UML Notation Guide – Object Management Group (OMG), Version 2.5.1

2 Overall Description

2.1 Product Perspective

The Amazon E-Commerce Shopping System is a **mature, follow-on product** in Amazon's family of retail and technology services. It builds upon Amazon's existing online retail infrastructure but is designed as a **self-contained, modular platform** that can integrate with other Amazon services like Amazon Pay, AWS hosting, and third-party logistics providers.

The system acts as the **central hub** connecting customers, sellers, payment gateways, and delivery services. It includes subsystems for **User Management**, **Product Catalog Management**, **Shopping Cart & Checkout**, **Order Tracking**, and **Seller Portal Management**.

2.2 Product Functionality

At a high level, the Amazon E-Commerce Shopping System provides the following major functions:

- **User Account Management** – Registration, login, profile updates, authentication.
- **Product Search & Browsing** – Keyword search, category browsing, filtering, and sorting.
- **Recommendation Engine** – Personalized product suggestions.
- **Shopping Cart Management** – Add, update, and remove items.
- **Secure Checkout & Payment** – Integration with multiple payment gateways.
- **Order Management** – Order placement, status updates, and cancellations.
- **Shipment Tracking** – Real-time tracking via logistics APIs.
- **Review & Rating System** – Customer feedback on products.
- **Seller Tools** – Inventory updates, pricing, sales analytics.
- **Customer Support Integration** – Chatbots, email support, and dispute resolution.

2.3 Users and Characteristics

1. **Customers**

- **Profile:** General public, all age groups, varying technical expertise.
- **Usage Frequency:** Occasional to daily.
- **Requirements:** Fast browsing, easy checkout, secure transactions.

2. **Sellers**
   - **Profile:** Small businesses, large brands, and individual sellers.
   - **Usage Frequency:** Daily for product updates and order processing.
   - **Requirements:** Bulk upload tools, real-time inventory updates, sales analytics.
3. **Administrators**
   - **Profile:** Internal Amazon staff managing system operations.
   - **Usage Frequency:** Continuous monitoring.
   - **Requirements:** Access to all modules, system analytics, and dispute management.

Priority Users: Customers and Sellers are the highest priority since they generate platform activity.

2.4 Operating Environment

The system must operate seamlessly in a multi-platform environment:

- **Client-Side:**
  - Web browsers: Chrome, Firefox, Safari, Edge (latest two versions).
  - Mobile apps: iOS 13+ and Android 9+.
- **Server-Side:**
  - Hosting: AWS EC2 with load balancing.
  - OS: Linux (Ubuntu 20.04 LTS).
  - Database: Amazon RDS, DynamoDB.
- **Network:**
  - Minimum bandwidth: 1 Mbps for customer access.
  - Secure HTTPS/TLS 1.3 for all communications.

2.5 Design and Implementation Constraints

1. **Compliance Requirements:** Must meet PCI DSS and GDPR for payments and data privacy.
2. **Scalability Limits:** Must handle up to 1 million concurrent users without performance degradation.
3. **Technology Constraints:** Backend in Java/Spring Boot; frontend in ReactJS; database AWS RDS/DynamoDB.
4. **Integration Standards:** Must use REST APIs for communication with external systems.
5. **Security Standards:** All sensitive data must be encrypted using AES-256.

2.6 User Documentation

The following user documentation will be provided:

- **Buyer User Guide:** Web-based help with screenshots for navigation, purchasing, and tracking.
- **Seller Onboarding Manual:** PDF and online tutorial videos for product listings and sales management.
- **FAQ & Knowledge Base:** Hosted on Amazon's Help Center.
- **In-App Tutorials:** Interactive guides for first-time users.
- **Admin Manual:** Internal document with system monitoring and troubleshooting guidelines.

2.7 Assumptions and Dependencies

Assumptions:

- Users have stable internet connectivity.
- Payment gateway services are available and functional 99.9% of the time.
- Logistics partners provide real-time status updates.

Dependencies:

- AWS services (EC2, RDS, S3) remain operational.
- Third-party APIs (payment, logistics) maintain backward compatibility.
- Mobile app stores (Google Play, Apple App Store) approve app updates promptly.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The Amazon E-Commerce Shopping System will have intuitive, consistent, and responsive interfaces accessible via **web browsers** and **mobile apps**.
Key UI elements and screens include:

- **Home Page:**
  - Search bar at the top, navigation menu with product categories.
  - Dynamic banners for offers and recommended products.
  - Quick links to orders, cart, and profile.
- **Product Listing Page:**
  - Grid/list view of products with thumbnails, name, price, and rating.
  - Filters for price, brand, ratings, and delivery options.
  - Sort options (e.g., Price: Low to High).
- **Product Detail Page:**
  - Large image carousel, detailed specifications, seller information.
  - “Add to Cart” and “Buy Now” buttons.
  - Customer reviews and Q&A section.
- **Shopping Cart Page:**
  - List of selected items with quantity controls and total cost.
  - “Proceed to Checkout” button.
- **Checkout Page:**
  - Address selection, payment method selection, order summary.
  - “Place Order” button with confirmation prompt.
- **Order Tracking Page:**
  - Real-time status (Processing, Shipped, Out for Delivery, Delivered).
  - Tracking ID and expected delivery date.
- **Seller Dashboard:**
  - Product listing management, inventory updates, sales analytics.

UI Standards:

- Responsive design for all screen sizes.
- Consistent color themes and typography across all modules.
- Standard buttons (e.g., *Cancel*, *Confirm*, *Save*) on every actionable screen.
- Error messages displayed in red with user-friendly explanations.

3.1.2 Hardware Interfaces

- **Customer Devices:** Desktop computers, laptops, smartphones, tablets.

- **Seller Devices:** PCs or laptops with optional barcode scanners for inventory updates.

- **Servers:** Hosted on AWS EC2 instances with load balancers.

- **Peripheral Support:** Printers for invoice printing (seller side).

3.1.3 Software Interfaces

- **Operating Systems Supported:**
  - Client: Windows 10+, macOS 10.15+, iOS 13+, Android 9+.
  - Server: Linux (Ubuntu 20.04 LTS).
- **Backend Dependencies:**
  - Java 17 (Spring Boot Framework).
  - AWS SDK for integration with Amazon S3, EC2, RDS.
  - REST APIs for payment and logistics integration.
- **Database:**
  - Amazon RDS for relational data.
  - DynamoDB for high-speed non-relational data operations.

3.1.4 Communications Interfaces

- **Web Protocols:** HTTPS/TLS 1.3 for all communications.
- **API Communication:** RESTful APIs with JSON data format.
- **Email/SMS:** SMTP for email notifications, SMS API for mobile alerts.
- **Security:** All sensitive data (passwords, card info) encrypted using AES-256 before transmission.
- **Synchronization:** Real-time inventory and order status updates via API polling and WebSockets.

3.2 Functional Requirements

The functional requirements are grouped as follows:

1. **Product Management**
    - Add, update, and remove products from the catalog.
    - Manage product categories and stock levels.
    - Upload and manage product images.
2. **Order Processing**
    - Allow users to add items to the cart and proceed to checkout.
    - Generate order IDs and store order details.
    - Enable payment through multiple methods (credit card, debit card, UPI, net banking, wallet).
    - Send confirmation emails upon successful order placement.
3. **User Account Management**
    - User registration, login, and logout functionality.
    - Password reset via email.
    - View and update personal profile details.
    - Track current and past orders.
4. **Search and Recommendations**
    - Provide keyword-based and category-based product searches.
    - Suggest products based on user browsing and purchase history.

3.3 Behaviour Requirements

3.3.1 Use Case View

Actors:

- **Customer**: Browses, searches, orders products, and manages account.
- **Admin**: Manages products, processes orders, handles customer queries.
- **Payment Gateway**: Processes online transactions.

Example Use Cases:

- **UC1 – Place an Order**: Customer selects a product, adds it to the cart, checks out, and makes payment.
- **UC2 – Manage Products**: Admin logs in and updates product catalog.
- **UC3 – Track Order**: Customer checks current status of their order.

4 Other Non-functional Requirements

4.1 Performance Requirements

🎬 **Search Speed** – Product search results must be displayed within **2 seconds** for a database containing up to 500,000 items.

🎬 **Checkout Completion Time** – The entire checkout process, from cart confirmation to payment acknowledgment, should take **no longer than 5 seconds** under normal network conditions.

🎬 **Concurrent User Handling** – The system must support at least **10,000 concurrent users** without noticeable performance degradation.

🎬 **Page Load Time** – The home page and product detail pages must load within **3 seconds** on a standard broadband connection.

🎬 **Inventory Update Speed** – Stock availability updates after an order should occur within **2 seconds** to prevent overselling.

4.2 Safety and Security Requirements

Safety Requirements:

1. Transactions must be **atomic** to prevent partial payments or incomplete orders due to system or network failures.
2. The system must implement **automatic session timeouts** after 15 minutes of inactivity to avoid unauthorized access.
3. All failed payment attempts must be logged, and inventory should remain unchanged until a successful payment confirmation is received.

Security Requirements:

- All data transmitted between client and server must be encrypted using **TLS 1.3** or higher.
- User passwords must be hashed and salted using secure algorithms such as **bcrypt**.
- Multi-factor authentication should be supported for user accounts.
- Access control must be role-based to ensure only authorized personnel can manage products, view sensitive customer data, or modify orders.
- Sensitive data (e.g., payment information) must never be stored in plain text and must comply with **PCI-DSS** standards.

4.3 Software Quality Attributes

4.3.1 Reliability

The system must ensure **99.9% uptime** excluding planned maintenance. Automatic failover servers will be used to maintain service continuity during outages. Daily backups will be performed, and a rollback system will be in place for data recovery.

4.3.2 Usability

The interface will follow consistent design guidelines, with intuitive navigation, clear labels, and mobile-responsive layouts. A help section, onboarding tutorials, and tooltips will assist new users.

4.3.3 Maintainability

The codebase will be structured into **modular components**, with clear API documentation to allow developers to update or replace features without impacting unrelated modules. Version control practices will be followed strictly.

4.3.4 Scalability

The architecture will support **horizontal scaling**, allowing the system to handle increased loads by adding more servers or cloud instances without major architectural changes.

5 Other Requirements

🎬 **Database Requirements:** Must support multi-region replication for faster access across geographic locations.

🎬 **Internationalization Requirements:** System must support multiple languages (English, Spanish, French, Hindi) and multiple currencies with real-time exchange rate updates.

🎬 **Legal Requirements:** Must comply with GDPR (EU), CCPA (California), and local e-commerce regulations.

🎬 **Reuse Objectives:** Core modules (Authentication, Payment Processing, Review System) should be designed for reuse in other Amazon platforms.

🎬 **Accessibility Compliance:** System must meet **WCAG 2.1 Level AA** accessibility standards for differently-abled users.

Appendix A – Data Dictionary

The following table lists the key variables, constants, states, inputs, and outputs used in the Amazon E-Commerce Shopping System. Each entry includes its description, possible values, and related operations.

Name	**Type**	**Description**	**Possible Values**	**Related Operations / Requirements**
user_id	Integer	Unique identifier for each registered user.	Auto-generated numeric ID	Create during registration; used in authentication and profile retrieval.
username	String	Unique display name for the user.	3–30 characters, alphanumeric with underscores	Create, Update, Validate uniqueness.
email	String	User's email address for communication and login.	Valid email format	Used in registration, login, notifications.
password_hash	String	Encrypted password for authentication.	Encrypted string (bcrypt, Argon2)	Create during registration; Update on password change.
product_id	Integer	Unique identifier for each product.	Auto-generated numeric ID	CRUD operations for product management.
product_name	String	Name of the product displayed to users.	Up to 100 characters	Create, Update, Search operations.
category_id	Integer	Identifier for the product category.	Auto-generated numeric ID	Used in product filtering and categorization.
price	Decimal	Product price in selected currency.	Positive decimal value	Display in product pages; update by seller.
currency_code	String	Currency format used for price display.	USD, EUR, INR, GBP, etc.	Conversion during display and checkout.
stock_quantity	Integer	Number of units available for sale.	Non-negative integer	Update on purchase, restock by seller.
cart_id	Integer	Identifier for a customer’s shopping cart.	Auto-generated numeric ID	Created on first cart action; linked to user_id.
order_id	Integer	Unique identifier for each placed order.	Auto-generated numeric ID	Create after successful checkout; used in tracking.
order_status	String	Current stage of order processing.	Pending, Processing, Shipped, Out for Delivery, Delivered, Cancelled	Update via logistics integration.
payment_id	Integer	Identifier for a payment transaction.	Auto-generated numeric ID	Create during checkout; validate with payment gateway.

payment_status	String	State of payment transaction.	Pending, Completed, Failed, Refunded	Updated after payment confirmation.
tracking_id	String	Shipment tracking number from logistics provider.	Alphanumeric string	Display in order tracking page.
review_id	Integer	Unique identifier for a customer review.	Auto-generated numeric ID	Create when review is submitted.
rating	Float	Customer's rating for a product.	1.0 to 5.0	Calculate average product rating.
review_text	String	Customer's written feedback on product.	Up to 1000 characters	Display on product page; moderation checks.
language_code	String	Language setting for the interface.	en, es, fr, hi, etc.	Used for internationalization and localization.
exchange_rate	Decimal	Currency exchange rate for conversions.	Positive decimal value	Retrieved from financial API for real-time conversion.