

1. Introduction

1.1 Purpose

This document describes the Software Requirements Specification (SRS) for ShopEase, an online e-commerce platform. It will serve as the basis for creating detailed test cases to ensure ShopEase meets its functional and non-functional requirements.

1.2 Scope

ShopEase allows customers to browse products, manage a shopping cart, place orders, and view order history. Administrators can manage products, categories, and view sales reports.

1.3 Definitions, Acronyms, Abbreviations

- **SKU:** Stock Keeping Unit
 - **UI:** User Interface
 - **API:** Application Programming Interface
 - **SSL:** Secure Sockets Layer
-

2. Overall Description

2.1 User Classes and Characteristics

- **Guest:** Browse products but must register or log in to purchase.
- **Registered Customer:** Full access to cart, checkout, order history.
- **Administrator:** Manage product catalog, view sales analytics.

2.2 Operating Environment

- Web browsers (Chrome, Firefox, Edge) on desktop and mobile.

- Backend runs on Linux servers with MySQL database.

2.3 Constraints

- All payment transactions must use SSL/TLS.
 - Product catalog must support up to 100,000 SKUs.
-

3. Functional Requirements

3.1 User Registration & Authentication

- **FR-1.1:** Users can register with email, password, and name.
- **FR-1.2:** Login/logout using JWT tokens.
- **FR-1.3:** “Forgot password” flow via email reset link.

3.2 Product Catalog

- **FR-2.1:** Display product list with name, image, price, and stock status.
- **FR-2.2:** Filter products by category, price range, and search keywords.
- **FR-2.3:** Product detail page shows full description, SKU, reviews, and related products.

3.3 Shopping Cart

- **FR-3.1:** Add/remove/update quantity of items in cart.
- **FR-3.2:** Persist cart between sessions for logged-in users.
- **FR-3.3:** Display cart subtotal, taxes, and shipping estimate.

3.4 Checkout & Payment

- **FR-4.1:** Collect billing and shipping address.
- **FR-4.2:** Integrate with Stripe for credit-card payments.
- **FR-4.3:** Show order confirmation page and email receipt.

3.5 Order Management

- **FR-5.1:** Customers can view past orders with status (Pending, Shipped, Delivered).
- **FR-5.2:** Administrators can update order status and track shipments.

3.6 Reviews & Ratings

- **FR-6.1:** Customers can submit a star-rating (1–5) and text review per purchased product.
 - **FR-6.2:** Administrators can moderate (approve/reject) reviews.
-

4. Non-Functional Requirements

4.1 Performance

- **NFR-1:** Page load time ≤ 2 seconds under 1,000 concurrent users.
- **NFR-2:** API response time ≤ 200 ms for 95% of calls.

4.2 Security

- **NFR-3:** All sensitive data encrypted at rest.
- **NFR-4:** Protect against OWASP Top 10 vulnerabilities.

4.3 Scalability

- **NFR-5:** System must scale horizontally to support up to 10,000 orders per hour.

4.4 Usability

- **NFR-6:** UI must be responsive on mobile and desktop.

4.5 Maintainability

- **NFR-7:** Codebase must adhere to PEP8 (Python) and use automated lint checks.
-

5. External Interfaces

5.1 User Interface

- Responsive web UI built with React.

5.2 APIs

- **GET** /api/products
 - **POST** /api/cart
 - **POST** /api/checkout
 - **GET** /api/orders/{orderId}
-

6. Data Requirements

- **Users:** id, name, email, password_hash, created_at
 - **Products:** sku, name, description, price, stock_quantity, category_id
 - **Orders:** order_id, user_id, status, total_amount, created_at
-

7. Assumptions & Dependencies

- Third-party payment gateway (Stripe) credentials available.
 - Email SMTP service for password resets.
-