

Software Requirements Specification

for

LUXORA E-Commerce Platform

Version 1.0 approved

Prepared by

Group 13

[D.T. Jayasinghe]

[WA.M. Madhushani]

[W.M.S.S. Ariyapala]

2024/12/03

Table of Contents

1. INTRODUCTION	4
1.1 Purpose	4
1.2 Document Conventions	4
1.3 Intended Audience and Reading Suggestions	4
1.4 Project Scope	4
1.5 References	5
2. OVERALL DESCRIPTION	6
2.1 Product Perspective	6
2.2 Product Features	6
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints	7
2.6 User Documentation	7
2.7 Assumptions and Dependencies	7
3. SYSTEM FEATURES	8
3.1 User Authentication and Profiles	8
3.2 Product Catalog and Search	8
3.3 Shopping Cart and Checkout	8
3.4 Order Tracking and Notifications	8
3.5 Admin Dashboard	9
4. EXTERNAL INTERFACE REQUIREMENTS	10
4.1 User Interfaces	10
4.2 Hardware Interfaces	10
4.3 Software Interfaces	10
4.4 Communications Interfaces	10
5. REQUIREMENT ANALYSIS	11
5.1 Functional Requirements	11
5.1.1 User Management	11
5.1.2 Product Catalog and Search	11
5.1.3 Shopping Cart and Checkout	11
5.1.4 Order Management and Tracking	12
5.1.5 Notifications and Alerts	12

5.1.6 Admin Dashboard	12
5.1.7 Promotions and Discounts	13
5.1.8 Analytics and Reporting.....	13
5.2 Non-functional Requirements	13
5.2.1 Performance Requirements:	13
5.2.2 Security Requirements:	14
5.2.3. Usability Requirements:.....	14
5.2.4. Reliability Requirements:	15
5.2.5. Scalability Requirements:	15
6. OTHER REQUIREMENTS	16
7. APPENDICES.....	16

Revision History

Name	Date	Reason For Changes	Version
			1.0

1 INTRODUCTION

1.1 Purpose

The purpose of the Luxora project is to create a dedicated e-commerce platform tailored for luxury goods, providing a seamless and secure online shopping experience. Luxora is designed to bridge the gap between high-end product sellers and discerning buyers, offering a curated space where users can easily browse, search, and purchase exclusive items. By incorporating advanced features like personalized dashboards, secure payment processing, and responsive design, Luxora aims to deliver an exceptional user experience.

1.2 Document Conventions

- Bold headings for sections and subsections.
- Requirements are tagged with unique IDs (e.g., REQ-001).
- Inline references are hyperlinked for quick navigation.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Developers: To design and implement features as specified.
- Testers: To validate system functionality and performance.
- Project Managers: To ensure that the project aligns with timelines and deliverables.
- Stakeholders: To confirm that the system meets business goals.

1.4 Project Scope

The scope of the Luxora project encompasses the development of a high-end e-commerce platform focused on luxury products. Luxora will enable users to register, browse product categories, view detailed product descriptions, add items to a cart, and securely complete purchases. It will also include advanced features such as a personalized shopping dashboard, product filtering, and search capabilities, designed to enhance user experience. Additionally, Luxora will provide

administrators with tools to manage products, orders, and user accounts. With an emphasis on security and performance, Luxora aims to become a trusted platform for premium online shopping.

- Register to the platform
- Log in to the platform
- Browse through the database of available product
- View user profile
- Get contact information of the seller
- View product dashboard

1.5 References

1. [Laravel Documentation] (<https://laravel.com/docs/10.x/readme>)
2. [Bootstrap Documentation] (npm i [bootstrap@5.3.3](#))
3. [<https://www.geeksforgeeks.org/software-requirement-specification-srs-format/>]
4. [<https://www.scribd.com/document/422533627/E-Commerce-Srs-document>]

2. OVERALL DESCRIPTION

2.1 Product Perspective

Luxora is a standalone web application with features optimized for mobile and desktop users. Its primary objective is to provide an intuitive interface for browsing, purchasing, and managing high-end products. The system incorporates third-party payment gateways for secure transactions.

2.2 Product Features

- User Authentication: Secure login and role-based access.
- Product Search: Dynamic search with advanced filters.
- Shopping Cart: Real-time updates and multi-item checkout.
- Order Tracking: Status updates for customers and order management for admins.

2.3 User Classes and Characteristics

User Classes and Characteristics

- Guest Users: Limited access to browse products.
- Registered Users: Can manage profiles, add products to the cart, and make purchases.
- Admins: Manage user accounts, product catalogs, and orders.

2.4 Operating Environment

- Server: PHP-compatible server (e.g., Apache) with Laravel.
- Database: MySQL.
- Frontend: HTML/CSS, JavaScript, Bootstrap for responsive design.
- Supported Browsers: Latest versions of Chrome, Firefox, Safari, and Edge.

2.5 Design and Implementation Constraints

- **Security:** Payment data encryption using AES-256 ensures all transactions are securely processed. Implement HTTPS with SSL certificates to protect communication between users and the platform.
- **Performance:** The system must handle up to 200 concurrent users during peak hours with minimal latency. Utilize caching mechanisms like Redis to optimize product page load times.
- **Storage:** Allocate scalable cloud storage to handle up to 1 TB of product images, videos, and user-generated content efficiently. Use file compression techniques for media uploads to optimize storage space without compromising quality.

2.6 User Documentation

This documentation approach will ensure that your users feel supported throughout their experience on Luxora, improving both customer satisfaction and retention.

2.7 Assumptions and Dependencies

- The platform depends on stable server hosting for continuous operation.
- Laravel, MySQL, and any third-party services (e.g., Pusher) are assumed to be stable and well-documented.

3. SYSTEM FEATURES

3.1 User Authentication and Profiles

Description: Users can register, log in, and manage profiles securely.

Requirements:

- REQ-001: Implement email-based registration and login.
- REQ-002: Enable users to reset passwords via email verification.

3.2 Product Catalog and Search

Description: Provides advanced filtering options for browsing products.

Requirements:

- REQ-003: Allow filtering by price, category, and brand.
- REQ-004: Implement predictive search functionality.

3.3 Shopping Cart and Checkout

Description: Users can add products to the cart and securely complete purchases.

Requirements:

- REQ-005: Support multiple payment options, including Stripe and PayPal.
- REQ-006: Generate detailed invoices for each order.

3.4 Order Tracking and Notifications

Description: Provides real-time updates on order status and notifications.

Requirements:

- REQ-007: Notify users of order status changes via email and in-app alerts.

3.5 Admin Dashboard

Description: The admin dashboard provides a comprehensive interface for managing product listings, user accounts, orders, and other platform-related operations.

Requirements:

- REQ-8: Admins must be able to add, edit, and remove products from the catalog.
- REQ-9: Provide analytics and reports on sales, user activity, and inventory levels.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 User Interfaces

- Homepage: Displays trending and recommended products.
- Profile Page: User details, purchase history, and settings.
- Admin Dashboard: Manage products, users, and orders.

4.2 Hardware Interfaces

- The platform is web-based, requiring no specific hardware interfaces.

4.3 Software Interfaces

- Backend: Laravel for server-side operations.
- Database: MySQL for structured data management.
- APIs: Stripe for payments and Google Analytics for tracking.

4.4 Communications Interfaces

- HTTPS for secure data transmission.
- WebSocket for real-time notifications.

5. REQUIREMENT ANALYSIS

5.1 Functional Requirements

5.1.1 User Management

- Users must be able to register with an email and create a secure password.
- The system should allow secure login and logout functionality for all users.
- Password recovery must be available through email verification.
- Users should be able to update their profile details, such as name, email, password, and shipping address.
- Admins must have the ability to manage user accounts, including activating, deactivating, or deleting accounts.

5.1.2 Product Catalog and Search

- The system should allow users to browse products by categories, brands, or featured collections.
- A keyword-based search bar must be provided for finding products.
- Filtering options should be available to narrow down products by price, ratings, availability, and other attributes.
- Product pages must include detailed information, such as specifications, images, user reviews, and availability status.
- Related or recommended products should be displayed based on user browsing behavior.

5.1.3 Shopping Cart and Checkout

- Users must be able to add products to their shopping cart and view them at any time.
- The shopping cart should allow users to adjust product quantities or remove items.
- A detailed summary of the cart, including product details, subtotal, taxes, and total price, must be displayed.
- The platform should allow users to save their cart for later access if logged in.

- The checkout process must support multiple payment methods, including credit/debit cards and PayPal.
- Users should receive a confirmation email with order details after completing a purchase.

5.1.4 Order Management and Tracking

- Users must have access to their order history, showing details of all previous purchases.
- The system should allow users to track the status of their orders, such as Processing, Shipped, and Delivered.
- Tracking information, including tracking numbers, must be provided for shipped orders.
- Admins should have the ability to update order statuses, process returns, and issue refunds as necessary.

5.1.5 Notifications and Alerts

- Users must receive notifications for critical events, such as order updates, payment confirmations, and delivery status.
- Promotional notifications, such as discounts and flash sales, should be sent via email or SMS based on user preferences.
- Users should have the option to customize notification settings to suit their needs.

5.1.6 Admin Dashboard

- Admins must be able to add, edit, or remove products from the catalog.
- The system should allow admins to manage categories, brands, and product attributes.
- An analytics dashboard must be provided for admins to monitor sales trends, inventory levels, and customer activity.
- Admins should be able to create and manage discount codes, promotions, and flash sales.
- The system must include role-based access control, ensuring only authorized users can perform sensitive administrative actions.

5.1.7 Promotions and Discounts

- Admins should be able to create promotional campaigns, such as percentage-based discounts, flat-rate discounts, or free shipping offers.
- Discount codes must be validated and applied automatically during checkout if eligible.
- Flash sale banners and countdowns must be displayed on the homepage and product pages.

5.1.8 Analytics and Reporting

- The system must provide detailed sales reports, customer insights, and inventory status to admins.
- Admins should be able to export reports for external analysis or auditing.
- Key performance indicators (KPIs) must be displayed, such as total sales, top-selling products, and customer engagement metrics.

5.2 Non-functional Requirements

5.2.1 Performance Requirements:

- Page Load Time

The platform should load within two seconds for 95% of users, even under peak traffic conditions. This includes optimizing images, scripts, and CSS for fast delivery.

- Concurrent User Support

Luxora must support up to 10,000 concurrent users during peak shopping hours or promotions without degradation in response time or user experience.

- System Response Time

The system should respond to user requests within one second for most actions, such as clicking, browsing products, and submitting the cart.

5.2.2 Security Requirements:

- Data Encryption

All sensitive data, such as user credentials and payment information, must be encrypted using AES-256 encryption, both during transmission and while stored.

- Authentication and Authorization

Implement secure login mechanisms, including password hashing, and offer two-factor authentication (2FA) for users performing sensitive actions.

- Data Privacy Compliance

The platform must comply with data privacy regulations, such as GDPR, allowing users to access, update, and delete their personal information.

- Payment Card Security

The platform must integrate with payment gateways like Stripe and PayPal that comply with PCI-DSS standards for payment processing.

5.2.3. Usability Requirements:

- User Interface (UI)

The user interface must be responsive and user-friendly, providing a seamless experience on all devices, including mobile, tablet, and desktop. The design should be intuitive, enabling users to easily navigate through product catalogs and complete purchases.

- Accessibility Compliance

The platform must meet WCAG 2.1 accessibility guidelines, ensuring that users with disabilities can access and use the platform effectively, including visual support and screen reader compatibility.

- Onboarding and Support

Include in-app tutorials, tooltips, and help sections to guide users through the first-time use of key features, such as placing an order, managing the cart, and updating their profiles.

5.2.4. Reliability Requirements:

- Uptime

The platform must maintain 99.9% uptime, ensuring that the site is available and functional at all times, with minimal downtime for maintenance.

- Backup and Recovery

The platform should have automatic daily backups for all critical data, such as user data, orders, and product information, to ensure recovery in case of system failure.

- Fault Tolerance

The system must be designed with redundancy in key areas, such as database replication and server failover, to prevent a single point of failure and ensure operational continuity during failures.

5.2.5. Scalability Requirements:

- Horizontal Scalability

The system should be designed to scale horizontally, allowing additional servers to be added easily as user demand increases. This can be achieved through load balancing and distributed databases.

- Elastic Resource Management

Use cloud-based infrastructure that can automatically scale resources up or down to handle unexpected traffic spikes, ensuring consistent performance during high-traffic events.

6. OTHER REQUIREMENTS

The platform will comply with data privacy regulations and use HTTPS for secure data transmission.

7. APPENDICES

Appendix A: Glossary

- Timeline: A feed of posts from users the individual follows.
- Like: An action indicating a user's positive reaction to a post.
- Follower: A user who subscribes to see another user's posts.