**Software Requirements Specification**

**for**

Online Marketplace for Handcrafted Goods

Prepared by

Aravindhan T (23CSR020)  
Dhakshanesh R (23CSR042)  
Dineshbalaji A (23CSR054)

Kongu Engineering College  
  
27/02/2025

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Intended Audience 1

1.3 Project Scope 1

1.4 Definitions,Acronyms and Abbreviations 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Features 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

3. System Features 3

3.1 User Authentication for Buyers and Sellers…………………………………………………………3

3.2 Product Listing and Approval Workflow……………………………………………………………3

3.3 Shopping Cart and Checkout Flow………………………………………………………………….3

3.4 Payment Gateway and Order Tracking……………………………………………………………..3

3.5 Review and Feedback System……………………………………………………………………....3

3.6 Dashboard for Sellers……………………………………………………………………………..…3

4. External Interface Requirements 4

4.1 User Interfaces 4

4.2 Hardware Interfaces 4

4.3 Software Interfaces 5

4.4 Communications Interfaces 5

5. Other Nonfunctional Requirements 5

5.1 Performance Requirements 5

5.2 Security Requirements 5

5.3 Usability Requirements 5

5.4 Scalability Requirements 5

6. Other Requirements 5

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Introduction**

**1.1 Purpose**

The purpose of this document is to define the software requirements for the **Online Marketplace for Handcrafted Goods**. This platform allows artisans to list and sell their handcrafted products directly to consumers. It includes features such as user authentication, product listing and approval workflows, shopping cart and checkout functionalities, a secure payment gateway, order tracking, and a seller dashboard.

**1.2 Intended Audience**

The intended audience for this document includes:

* **Developers** (Frontend, Backend, and Database Engineers)
* **UI/UX Designers**
* **Project Managers and Stakeholders**
* **Handcraft Artisans**

**1.3 Project Scope**

The system will serve as a marketplace where artisans can create profiles, list their handmade goods, and connect with potential buyers. Key functionalities include:

* Secure user authentication for buyers and sellers
* A workflow for product approval and listing
* An integrated shopping cart and checkout process
* A secure payment gateway with order tracking
* A rating and review system for buyers to provide feedback
* A seller dashboard for tracking sales and inventory

**1.4 Definitions, Acronyms, and Abbreviations**

* **User**: Any person registered on the platform (Buyer or Seller)
* **UI**: User Interface
* **CRUD**: Create, Read, Update, Delete
* **Backend**: The server logic built using Node.js and Express.
* **Database**: MongoDB for storing user data and meal information.

**1.5 References**

* MongoDB Documentation: https://www.mongodb.com/docs/
* React Documentation: https://reactjs.org/docs/
* Node.js Documentation: <https://nodejs.org/en/docs/>
* Razorpay Payment API Documentation

**2. Overall Description**

**2.1 Product Perspective**

The system is a web-based marketplace allowing artisans to showcase and sell their handcrafted products. The platform will use React.js for frontend development, Node.js for backend logic, and MongoDB for data storage.

**2.2 Product Features**

1. **User Authentication: Secure login and registration for buyers and sellers.**
2. **Product Listing: Sellers can create product listings with images, descriptions, and prices.**
3. **Approval Workflow: Admins review and approve product listings before publishing.**
4. **Shopping Cart: Buyers can add products to a cart and proceed to checkout.**
5. **Payment & Order Tracking: Secure payment processing and real-time order tracking.**
6. **Reviews & Feedback: Buyers can rate and review sellers and products.**
7. **Seller Dashboard: Real-time sales analytics and inventory management.**

**2.3 User Classes and Characteristics**

* **Buyers: Can browse, purchase products, and leave reviews.**
* **Sellers: Can list products, manage inventory, and track orders.**
* **Admins: Oversee the platform, approve product listings, and manage disputes.**

**2.4 Operating Environment**

* **Frontend**: Built using React.js, HTML, CSS, Bootstrap, JavaScript.
* **Backend**: Built using Node.js and Express.
* **Database**: MongoDB.
* **Browser Compatibility**: Chrome, Firefox, Edge, and mobile browsers.
* **Mobile App Compatibility**: Android and iOS.

**2.5 Design and Implementation Constraints**

* Must support high traffic loads for future scalability.
* Secure payment transactions through third-party gateways.
* Multi-platform support (Web and Mobile browsers).

**3. System Features**

**3.1 User Authentication**

1. **Users can register, log in, and manage profiles.**

### 3.2 Product Listing and Approval Workflow

1. **Sellers can add products with images, descriptions, and pricing.**
2. **An admin must approve product listings before they go live.**

**3.3 Shopping Cart and Checkout Flow**

1. **Buyers can add multiple items to a cart.**
2. **Checkout process with address and payment information.**

**3.4 Payment Gateway and Order Tracking**

1. **Integration with Razorpay for secure transactions.**
2. **Users can track orders and receive notifications.**

### 3.5 Review and Feedback System

1. **Buyers can review purchased items.**
2. **Ratings impact seller reputation.**

### 3.6 Dashboard for Sellers

1. Sales insights and order management.
2. Inventory tracking and product performance analytics.

**4. External Interface Requirements**

**4.1 User Interfaces**

The application should have an intuitive and responsive UI across web and mobile platforms:

* **Home Page**: Displays featured and recommended products.
* **Product Page**: Shows product details, pricing, and seller info.
* **Cart Page**: Displays selected items.
* **Checkout Page**: Facilitates payment and order confirmation.
  + **Dashboard**: Seller analytics and order management

**4.2 Hardware Interfaces**

The system will be compatible with the following hardware:

* Desktop and mobile compatibility.
* Support for barcode scanning (optional for inventory management).

**4.3 Software Interfaces**

The system will integrate with:

* **Integration with Razorpay for payments.**
* **RESTful APIs for data retrieval.**
* **MongoDB as the database.**

**4.4 Communication Interfaces**

* **Frontend-Backend: REST API calls.**
* **Backend-Database: CRUD operations via MongoDB.**
* **Email & SMS: Notifications for order updates.**

**5. Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system must handle at least **1,000 concurrent users**.
* Transactions should process in **under 2 seconds**.

**5.2 Security Requirements**

* Data encryption for sensitive user information.
* Secure payment processing through **SSL encryption**.

**5.3 Usability Requirements**

* User-friendly and responsive UI.
* Easy navigation across devices.

**5.4 Scalability Requirements**

* Ability to scale up with increasing seller and buyer activity.
* Support for cloud hosting with autoscaling features.

**6. Other Requirements**

* Secure session management to prevent unauthorized access.
* Logging of critical actions for auditing purposes.