Software Requirements Specification (SRS)

HanFoods E-commerce Platform

Document Information - Document Version: 1.0 - Date: June 24, 2025 - Author: Pham Nam Khanh - Business Analyst & Software Requirements Engineer - Status: Final

---

Table of Contents

---

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the software requirements for the HanFoods E-commerce Platform. It is intended for:

- **Development Team**: To understand technical implementation requirements
- **Quality Assurance Team**: To create test plans and validation criteria

- **Project Stakeholders**: To validate system functionality against business needs
- **Maintenance Team**: For future system updates and modifications

1.2 Scope

The HanFoods E-commerce Platform is a specialized web-based application that facilitates online coconut product sales with focus on fresh and dried coconut hearts (củ hủ dừa). The system encompasses:

Software Name: HanFoods Coconut Products E-commerce Platform
What the software will do: - Enable customers to browse, search, and purchase authentic Vietnamese coconut products online - Provide farmers with tools to manage coconut harvests, product listings, and farm operations - Support administrators in managing quality control, farmer certification, and platform operations - Process payments securely including international transactions for export orders - Track product freshness, origin, and cold-chain delivery status - Maintain traceability from farm to customer for quality assurance

What the software will not do: - Farm management automation or IoT monitoring systems - Physical coconut processing or packaging operations - Direct logistics operations (integration only with cold-chain providers) - International export documentation processing beyond basic support - Financial accounting beyond basic sales reporting

Benefits: - Direct market access for coconut farmers with premium pricing - Guaranteed authentic Vietnamese coconut products for consumers - Complete traceability and quality assurance for fresh products - International market expansion for Vietnamese agricultural products

1.3 Definitions, Acronyms, Abbreviations

Term	Definition
API	Application Programming Interface
COD	Cash on Delivery
CRUD	Create, Read, Update, Delete
JWT	JSON Web Token
PCI DSS	Payment Card Industry Data Security Standard
REST	Representational State Transfer
SPA	Single Page Application
SSL/TLS	Secure Sockets Layer/Transport Layer Security
UI/UX	User Interface/User Experience
Customer	End user purchasing coconut products
Farmer	Coconut farmer/supplier providing products
Coconut Heart	Fresh or dried inner part of young coconut (củ hủ dừa)

Term	Definition
Cold-chain	Temperature-controlled supply chain for fresh products
Traceability	Ability to track product from farm origin to customer
Harvest Date	Date when coconuts were harvested from trees
Quality Grade	Classification of coconut product quality (Premium, Standard, Economy)
Origin Certification	Documentation proving farm location and farming practices

1.4 References

- IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Business Requirements Specification (BRS) - HanFoods E-commerce Platform v1.0
- Vietnam E-commerce Law 2005 (amended 2013)
- PCI DSS Requirements and Security Assessment Procedures v3.2.1
- WCAG 2.1 Web Content Accessibility Guidelines

1.5 Overview

This SRS document is organized into four main sections: - **Section 2**: Provides overall system description including product perspective, functions, and user characteristics - **Section 3**: Details specific functional and non-functional requirements - **Section 4**: Describes individual system features with detailed specifications - **Section 5**: Contains appendices with additional technical specifications and diagrams

---

2. Overall Description

2.1 Product Perspective

The HanFoods E-commerce Platform is a new, standalone web application with the following system context:

2.1.1 System Context Diagram

```
[Customers] ↔ [Web Browser] ↔ [HanFoods Platform] ↔ [Database]

[Coconut Farmers] ↔ [Farmer Portal] ↔ [Payment Gateway] ↔ [Email Service]

[Administrators] ↔ [Admin Dashboard] ↔ [Cold-chain API] ↔ [File Storage]
```

```
                               [Quality Tracking] ↔ [Export Documentation]
```

2.1.2 System Architecture

- **Frontend**: React.js Single Page Application (SPA)
- **Backend**: Node.js with Express.js framework
- **Database**: MongoDB for primary data storage
- **File Storage**: Cloud-based storage for images and documents
- **Authentication**: JWT-based authentication system
- **Payment Processing**: Integration with third-party payment gateways

2.2 Product Functions

The major functions of the HanFoods Platform include:

2.2.1 Customer Functions

- **Account Management**: Registration, login, profile management, delivery address management
- **Product Discovery**: Browse coconut product categories, search by origin/quality, filter by freshness/type
- **Shopping Experience**: Add to cart with freshness tracking, quality comparison, farmer information
- **Order Management**: Checkout with cold-chain delivery options, payment, freshness tracking, order history
- **Quality Feedback**: Product reviews focused on freshness and authenticity, farmer ratings

2.2.2 Farmer Functions

- **Farmer Portal**: Registration with farm verification, profile management, certification uploads
- **Product Management**: Add coconut products with harvest dates, quality grades, origin details, pricing
- **Harvest Planning**: Schedule harvests, manage inventory levels, track quality batches
- **Order Processing**: View orders, update harvest/processing status, coordinate with logistics
- **Farm Analytics**: Sales performance, seasonal trends, quality feedback analysis

2.2.3 Administrative Functions

- **User Management**: Manage customer and farmer accounts, verification processes

- **Quality Control**: Farmer certification management, product quality monitoring, compliance tracking
- **Platform Management**: Coconut product categories, origin database, certification standards
- **Logistics Coordination**: Cold-chain partner management, delivery route optimization
- **Export Management**: International order processing, documentation support, compliance monitoring

2.3 User Characteristics

2.3.1 Customer Users

- **Demographics**: Health-conscious adults aged 25-55, Vietnamese diaspora, natural food enthusiasts
- **Technical Expertise**: Basic to intermediate computer/smartphone skills
- **Usage Frequency**: 1-3 times per month for coconut product ordering
- **Primary Goals**: Authentic Vietnamese coconut products, quality assurance, origin transparency
- **Device Preferences**: Mobile-first for browsing, desktop for detailed origin research

2.3.2 Farmer Users

- **Demographics**: Coconut farmers and agricultural cooperatives in Mekong Delta region
- **Technical Expertise**: Basic smartphone/computer skills, agricultural focus
- **Usage Frequency**: Daily for harvest updates, weekly for new product listings
- **Primary Goals**: Direct market access, fair pricing, simplified sales process
- **Device Preferences**: Mobile for quick updates, basic computer for detailed management

2.3.3 Administrator Users

- **Demographics**: Platform employees with agricultural and e-commerce background
- **Technical Expertise**: Advanced technical and agricultural knowledge
- **Usage Frequency**: Daily platform monitoring and quality control
- **Primary Goals**: Quality assurance, farmer support, international expansion
- **Device Preferences**: Desktop for comprehensive management, mobile for field verification

2.4 Constraints

2.4.1 Regulatory Constraints

- Must comply with Vietnam E-commerce Law and consumer protection regulations
- Payment processing must meet international security standards (PCI DSS)
- Data privacy must comply with applicable data protection laws
- Food safety information display requirements

2.4.2 Hardware Constraints

- Must operate on standard web browsers (Chrome, Firefox, Safari, Edge)
- Mobile responsiveness required for devices with minimum 320px screen width
- Server infrastructure must support minimum 1,000 concurrent users
- Database performance requirements for sub-100ms query response times

2.4.3 Software Constraints

- Backend development using Node.js v16+ and Express.js framework
- Frontend development using React.js v18+ with modern JavaScript (ES6+)
- Database using MongoDB v5.0+ with Mongoose ODM
- RESTful API design principles for all service endpoints

2.4.4 Security Constraints

- All data transmission must use HTTPS/TLS encryption
- User passwords must be hashed using bcrypt with minimum 12 rounds
- Session management using secure JWT tokens with expiration
- Input validation and sanitization for all user inputs

2.5 Assumptions and Dependencies

2.5.1 Assumptions

- Users have reliable internet connectivity (minimum 1 Mbps)
- Target market has widespread smartphone and internet adoption
- Vendors have basic inventory management capabilities
- Payment gateway services will remain available and stable
- Cloud hosting services will provide 99.9% uptime SLA

2.5.2 Dependencies

- **Third-party Payment Gateways**: Integration with VNPay, MoMo, or similar services
- **Email Service Provider**: For transactional emails and notifications

- **Cloud Storage Service**: For product images and document storage
- **SMS Gateway**: For mobile verification and order notifications
- **Maps API**: For delivery address validation and distance calculations

---

3. Specific Requirements

3.1 Functional Requirements

3.1.1 User Authentication and Authorization FR-AUTH-001: User Registration - ID: FR-AUTH-001 - **Priority**: High - **Description**: System shall allow new users to create accounts with email and phone verification - **Preconditions**: User has valid email address and phone number - **Main Flow**: 1. User navigates to registration page 2. User enters required information (name, email, phone, password) 3. System validates input format and uniqueness 4. System sends verification email and SMS 5. User verifies email and phone 6. System creates user account and sends welcome message - **Alternative Flows**: - If email already exists, system displays error message - If verification fails, system allows resend verification - **Postconditions**: User account created and can access system - **Acceptance Criteria**: - Registration form validates all required fields - Email and phone verification required before account activation - Password must meet security requirements (8+ characters, mixed case, numbers) - System prevents duplicate registrations

FR-AUTH-002: User Login - ID: FR-AUTH-002 - **Priority**: High - **Description**: System shall authenticate users with email/phone and password - **Preconditions**: User has active account - **Main Flow**: 1. User enters email/phone and password 2. System validates credentials 3. System generates JWT token 4. System redirects to appropriate dashboard - **Alternative Flows**: - If credentials invalid, system displays error message - If account locked, system displays appropriate message - **Postconditions**: User logged in with active session - **Acceptance Criteria**: - Login supports both email and phone number - System implements rate limiting (5 attempts per 15 minutes) - JWT tokens expire after 24 hours - Remember me option extends session to 30 days

FR-AUTH-003: Password Reset - ID: FR-AUTH-003 - **Priority**: Medium - **Description**: System shall allow users to reset forgotten passwords - **Preconditions**: User has registered account - **Main Flow**: 1. User clicks "Forgot Password" link 2. User enters email address 3. System sends password reset email 4. User clicks reset link and enters new password 5. System updates password and confirms change - **Acceptance Criteria**: - Reset links expire after 1 hour - New password must meet security requirements - System logs password reset events

3.1.2 Product Management FR-PROD-001: Product Catalog Display - ID: FR-PROD-001 - **Priority**: High - **Description**: System shall display products in organized categories with search and filtering - **Main Flow**:

1. User browses product categories or searches 2. System displays products with images, names, prices 3. User can filter by price, category, ratings, dietary preferences 4. User can sort by price, popularity, ratings, newest - **Acceptance Criteria**: - Products display with high-quality images (minimum 400x400px) - Price shown in VND with proper formatting - Availability status clearly indicated - Search supports partial matching and typo tolerance

FR-PROD-002: Product Details - **ID**: FR-PROD-002 - **Priority**: High - **Description**: System shall show comprehensive product information - **Main Flow**: 1. User clicks on product 2. System displays detailed product page 3. User can view multiple images, description, ingredients, nutrition info 4. User can read reviews and ratings 5. User can add to cart or wishlist - **Acceptance Criteria**: - Multiple product images with zoom functionality - Detailed description with formatting support - Customer reviews with star ratings - Related products suggestions

FR-PROD-003: Vendor Product Management - **ID**: FR-PROD-003 - **Priority**: High - **Description**: Vendors shall manage their product listings through admin portal - **Preconditions**: Vendor has approved account - **Main Flow**: 1. Vendor logs into admin portal 2. Vendor navigates to product management 3. Vendor can add, edit, or delete products 4. Vendor uploads images and sets pricing 5. System validates and publishes products - **Acceptance Criteria**: - CRUD operations for all product attributes - Image upload with automatic resizing - Inventory management with low stock alerts - Bulk product import/export functionality

3.1.3 Shopping Cart and Orders **FR-CART-001: Shopping Cart Management** - **ID**: FR-CART-001 - **Priority**: High - **Description**: System shall allow users to manage items in shopping cart - **Main Flow**: 1. User adds products to cart 2. System saves cart items (persistent across sessions) 3. User can modify quantities or remove items 4. System calculates totals including taxes and shipping - **Acceptance Criteria**: - Cart persists for logged-in users across devices - Real-time price and availability updates - Quantity limits based on inventory - Clear breakdown of costs (subtotal, tax, shipping, total)

FR-ORDER-001: Order Checkout Process - **ID**: FR-ORDER-001 - **Priority**: High - **Description**: System shall process customer orders from cart to payment - **Preconditions**: User has items in cart and valid delivery address - **Main Flow**: 1. User proceeds to checkout 2. System displays order summary 3. User selects delivery address and payment method 4. User confirms order details 5. System processes payment 6. System creates order and sends confirmation - **Alternative Flows**: - If payment fails, user can retry or change payment method - If items unavailable, user can modify order - **Postconditions**: Order created and payment processed - **Acceptance Criteria**: - Guest checkout option available - Multiple payment methods supported - Order confirmation sent via email and SMS - Estimated delivery time provided

FR-ORDER-002: Order Tracking - **ID**: FR-ORDER-002 - **Priority**: Medium - **Description**: System shall provide real-time order status updates - **Main Flow**: 1. Customer places order 2. System creates order with "Pending" status 3. Vendor updates status as order progresses 4. System sends notifications for status changes 5. Customer can track order through account or link - **Acceptance Criteria**: - Order statuses: Pending, Confirmed, Preparing, Ready, Out for Delivery, Delivered - Email and SMS notifications for major status changes - Estimated delivery time updates - Order history accessible in user account

3.1.4 Payment Processing **FR-PAY-001: Payment Gateway Integration** - **ID**: FR-PAY-001 - **Priority**: High - **Description**: System shall process payments through multiple secure gateways - **Preconditions**: User has completed order checkout - **Main Flow**: 1. User selects payment method 2. System redirects to payment gateway 3. Payment gateway processes transaction 4. System receives payment confirmation 5. System updates order status and sends confirmation - **Acceptance Criteria**: - Support for credit/debit cards, bank transfer, digital wallets, COD - PCI DSS compliant payment processing - Payment retry mechanism for failed transactions - Refund processing capability

3.1.5 User Reviews and Ratings **FR-REV-001: Product Reviews** - **ID**: FR-REV-001 - **Priority**: Medium - **Description**: Customers shall rate and review purchased products - **Preconditions**: Customer has purchased and received product - **Main Flow**: 1. Customer navigates to order history 2. Customer selects product to review 3. Customer enters rating (1-5 stars) and written review 4. System validates and publishes review - **Acceptance Criteria**: - Only verified purchasers can leave reviews - Reviews moderated for inappropriate content - Average ratings calculated and displayed - Helpful/unhelpful voting system

3.2 Non-functional Requirements

3.2.1 Performance Requirements **NFR-PERF-001: Response Time** - **Requirement**: 95% of page requests shall load within 3 seconds - **Measurement**: Server response time + client rendering time - **Testing**: Load testing with 1,000 concurrent users

NFR-PERF-002: Throughput - **Requirement**: System shall handle 1,000 concurrent users without degradation - **Measurement**: Concurrent user load testing - **Testing**: Stress testing with gradual load increase

NFR-PERF-003: Database Performance - **Requirement**: 95% of database queries shall complete within 100ms - **Measurement**: Query execution time monitoring - **Testing**: Database performance profiling

3.2.2 Security Requirements **NFR-SEC-001: Data Encryption** - **Requirement**: All data transmission shall use TLS 1.2 or higher - **Implemen-**

tation: HTTPS enforced for all endpoints - **Testing**: SSL/TLS certificate validation

NFR-SEC-002: Password Security - **Requirement**: User passwords shall be hashed using bcrypt with minimum 12 rounds - **Implementation**: Password hashing in authentication module - **Testing**: Password hash verification

NFR-SEC-003: Input Validation - **Requirement**: All user inputs shall be validated and sanitized - **Implementation**: Server-side validation for all API endpoints - **Testing**: Penetration testing for injection vulnerabilities

NFR-SEC-004: Session Management - **Requirement**: User sessions shall use secure JWT tokens with expiration - **Implementation**: JWT with 24-hour expiration, refresh token mechanism - **Testing**: Session security testing

3.2.3 Usability Requirements **NFR-USA-001: Mobile Responsiveness** - **Requirement**: System shall be fully functional on devices with screen width 320px - **Implementation**: Responsive CSS design with mobile-first approach - **Testing**: Cross-device testing on multiple screen sizes

NFR-USA-002: Accessibility - **Requirement**: System shall meet WCAG 2.1 AA accessibility standards - **Implementation**: Semantic HTML, ARIA labels, keyboard navigation - **Testing**: Accessibility audit tools and user testing

NFR-USA-003: Browser Compatibility - **Requirement**: System shall work on Chrome, Firefox, Safari, Edge (latest 2 versions) - **Implementation**: Cross-browser testing and polyfills - **Testing**: Automated browser compatibility testing

3.2.4 Reliability Requirements **NFR-REL-001: System Availability** - **Requirement**: System shall maintain 99.5% uptime - **Measurement**: Monthly uptime calculation - **Implementation**: Redundant infrastructure, health monitoring

NFR-REL-002: Error Handling - **Requirement**: System shall gracefully handle errors with user-friendly messages - **Implementation**: Comprehensive error handling and logging - **Testing**: Error simulation and recovery testing

NFR-REL-003: Data Backup - **Requirement**: System data shall be backed up daily with 30-day retention - **Implementation**: Automated database backups to cloud storage - **Testing**: Backup restoration testing

3.2.5 Scalability Requirements **NFR-SCAL-001: Horizontal Scaling** - **Requirement**: System architecture shall support horizontal scaling - **Implementation**: Stateless application design, load balancer support - **Testing**: Load testing with scaled infrastructure

NFR-SCAL-002: Database Scaling - **Requirement**: Database shall support read replicas and sharding - **Implementation**: MongoDB replica sets and sharding configuration - **Testing**: Database performance testing under load

3.3 External Interface Requirements

3.3.1 User Interface Requirements UIR-001: Web Interface - **Description**: Responsive web interface accessible via modern browsers - **Requirements**: - Clean, intuitive design following modern UI/UX principles - Consistent navigation and layout across all pages - Fast loading with progressive enhancement - Mobile-first responsive design - Dark/light theme support

UIR-002: Admin Interface - **Description**: Administrative dashboard for vendors and platform administrators - **Requirements**: - Role-based interface customization - Comprehensive analytics dashboards - Bulk operation capabilities - Real-time data updates - Export functionality for reports

3.3.2 Hardware Interface Requirements HIR-001: Server Requirements - **Minimum Specifications**: - CPU: 4 cores, 2.5GHz - RAM: 8GB - Storage: 100GB SSD - Network: 1Gbps connection - Operating System: Linux (Ubuntu 20.04 LTS or CentOS 8)

3.3.3 Software Interface Requirements SIR-001: Database Interface - **Database**: MongoDB v5.0+ - **ODM**: Mongoose for Node.js - **Connection**: MongoDB connection string with authentication - **Requirements**: Connection pooling, automatic reconnection

SIR-002: Payment Gateway Interface - **Primary**: VNPay API v2.1.0 - **Secondary**: MoMo API v2.0 - **Protocol**: HTTPS REST API - **Data Format**: JSON request/response - **Security**: API keys, request signing

SIR-003: Email Service Interface - **Service**: SendGrid API v3 - **Purpose**: Transactional emails (registration, orders, notifications) - **Format**: HTML and text emails - **Tracking**: Open rates, click tracking

SIR-004: SMS Gateway Interface - **Service**: Twilio SMS API - **Purpose**: Phone verification, order notifications - **Format**: UTF-8 text messages - **Features**: Delivery status tracking

3.3.4 Communication Interface Requirements CIR-001: API Communication - **Protocol**: HTTPS REST API - **Format**: JSON request/response - **Authentication**: JWT Bearer tokens - **Rate Limiting**: 1000 requests per hour per user - **Documentation**: OpenAPI 3.0 specification

CIR-002: Real-time Communication - **Technology**: WebSocket connections - **Purpose**: Real-time order updates, notifications - **Fallback**: Server-Sent Events (SSE) - **Security**: Authenticated connections only

---

4. System Features

4.1 User Authentication System

4.1.1 Feature Description Comprehensive authentication system supporting multiple user roles with secure session management.

4.1.2 Functional Requirements

- Multi-factor authentication support
- Role-based access control (Customer, Vendor, Admin)
- Social login integration (Google, Facebook)
- Password policy enforcement
- Account lockout mechanism

4.1.3 Use Case: User Registration

```
Use Case ID: UC-AUTH-001
Primary Actor: New User
Goal: Register new account on platform
Preconditions: User has valid email and phone
Main Success Scenario:
1. User navigates to registration page
2. User fills registration form
3. System validates input
4. System sends verification email/SMS
5. User verifies account
6. System creates account and logs user in
Extensions:
3a. Validation fails - system shows error messages
4a. Email/SMS delivery fails - system provides retry option
5a. Verification expires - system allows new verification request
```

4.2 Product Catalog System

4.2.1 Feature Description Comprehensive product management system with search, filtering, and categorization capabilities.

4.2.2 Functional Requirements

- Hierarchical category structure
- Advanced search with filters
- Product comparison functionality
- Inventory management
- Bulk product operations

4.2.3 Data Model

```
// Product Schema
{
  productId: String (unique),
  name: String (required),
  description: String,
  category: ObjectId (ref: Category),
  vendor: ObjectId (ref: Vendor),
  price: Number (required),
  images: [String],
  inventory: {
    quantity: Number,
    lowStockThreshold: Number,
    status: Enum ['available', 'out_of_stock', 'discontinued']
  },
  attributes: {
    ingredients: [String],
    nutritionInfo: Object,
    dietaryRestrictions: [String],
    spiceLevel: Number
  },
  ratings: {
    average: Number,
    count: Number
  },
  createdAt: Date,
  updatedAt: Date
}
```

4.3 Order Management System

4.3.1 Feature Description End-to-end order processing system from cart to delivery with real-time tracking.

4.3.2 State Machine

```
Order States:
PENDING → CONFIRMED → PREPARING → READY → OUT_FOR_DELIVERY → DELIVERED
                    ↓
                 CANCELLED (from PENDING, CONFIRMED, PREPARING)
                    ↓
                 REFUNDED (from CANCELLED)
```

4.3.3 Business Rules

- Orders can only be cancelled within 5 minutes of placement
- Refunds processed within 7 business days
- Order total must include applicable taxes and delivery fees

- Delivery time estimates based on vendor location and order complexity

4.4 Payment Processing System

4.4.1 Feature Description Secure, multi-gateway payment processing with support for various payment methods.

4.4.2 Payment Flow

```
1. User initiates payment
2. System calculates total (items + tax + delivery)
3. System creates payment intent
4. User redirected to payment gateway
5. Gateway processes payment
6. System receives webhook confirmation
7. Order status updated to CONFIRMED
8. Confirmation sent to user and vendor
```

4.4.3 Security Measures

- PCI DSS compliant payment processing
- No storage of credit card information
- Payment tokenization for recurring payments
- Fraud detection and prevention
- Secure payment webhooks with signature verification

---

5. Appendices

5.1 Glossary

Term	Definition
Checkout	Process of finalizing purchase from shopping cart
Fulfillment	Process of preparing and delivering ordered items
Gateway	Third-party service for processing payments
Inventory	Stock quantity available for each product
Merchant	Food vendor selling products on platform
ODM	Object Document Mapper for database operations
SKU	Stock Keeping Unit - unique product identifier
Webhook	HTTP callback for real-time event notifications

5.2 Use Case Diagrams

```
Customer Use Cases:
- Register/Login
- Browse Products
- Search Products
- Add to Cart
- Checkout
- Track Order
- Leave Review

Vendor Use Cases:
- Manage Products
- View Orders
- Update Order Status
- View Analytics
- Manage Profile

Admin Use Cases:
- Manage Users
- Manage Categories
- View Platform Analytics
- Manage Content
- System Configuration
```

5.3 Entity Relationship Diagram

```
User ||--o{ Order : places
User ||--o{ Review : writes
User ||--o{ Address : has
Vendor ||--o{ Product : sells
Product ||--o{ OrderItem : includes
Product ||--o{ Review : receives
Order ||--o{ OrderItem : contains
Order ||--|| Payment : has
Category ||--o{ Product : categorizes
```

5.4 API Endpoint Specifications

Authentication Endpoints

```
POST /api/auth/register
POST /api/auth/login
POST /api/auth/logout
POST /api/auth/refresh-token
POST /api/auth/forgot-password
POST /api/auth/reset-password
```

Product Endpoints

```
GET /api/products
GET /api/products/:id
POST /api/products (vendor/admin)
PUT /api/products/:id (vendor/admin)
DELETE /api/products/:id (vendor/admin)
GET /api/products/search?q={query}
GET /api/categories
```

Order Endpoints

```
GET /api/orders (user orders)
POST /api/orders
GET /api/orders/:id
PUT /api/orders/:id/status (vendor)
POST /api/orders/:id/cancel
GET /api/orders/:id/track
```

Payment Endpoints

```
POST /api/payments/create-intent
POST /api/payments/confirm
POST /api/payments/webhook
GET /api/payments/:id/status
POST /api/payments/:id/refund
```

5.5 Database Schema Details

Collections Overview

- **users**: User accounts and profiles
- **vendors**: Vendor business information
- **products**: Product catalog
- **categories**: Product categories
- **orders**: Order information
- **order_items**: Individual order items
- **payments**: Payment transaction records
- **reviews**: Product reviews and ratings
- **addresses**: User delivery addresses
- **sessions**: User session management

5.6 Security Specifications

Authentication Flow

```
1. User login with credentials
2. Server validates credentials
3. Server generates JWT access token (24h expiry)
4. Server generates refresh token (30d expiry)
5. Client stores tokens securely
```

```
6. Client includes access token in API requests
7. Server validates token on each request
8. Client refreshes token when expired
```

Password Policy

- Minimum 8 characters
- Must include uppercase and lowercase letters
- Must include at least one number
- Must include at least one special character
- Cannot be common passwords (dictionary check)
- Cannot reuse last 5 passwords

5.7 Performance Benchmarks

Response Time Requirements

Operation	Target	Maximum
Page Load	2 seconds	3 seconds
Search Results	1 second	2 seconds
Add to Cart	500ms	1 second
Checkout Process	3 seconds	5 seconds
Payment Processing	5 seconds	10 seconds

Scalability Targets

Metric	Year 1	Year 2	Year 3
Concurrent Users	1,000	5,000	10,000
Daily Orders	500	2,500	5,000
Products	1,000	10,000	50,000
Vendors	50	500	1,000

This document contains confidential and proprietary information. Distribution is restricted to authorized development team members and stakeholders only.