

Contents

BookMyHotel - Comprehensive Project Documentation	2
Executive Summary	2
System Architecture Overview	2
Technology Stack	2
Multi-Tenancy Strategy	2
Hotel Booking System Analysis	2
Booking Flow Architecture	2
Key API Endpoints - Hotel Booking	2
Payment Processing	3
Guest Management	3
Shop Management System Analysis	3
Shop Architecture Overview	3
Key API Endpoints - Shop Management	3
Product & Inventory Management	4
Order Processing System	4
Security Architecture Analysis	4
Authentication & Authorization	4
Multi-Tenant Security	5
Performance & Scalability Analysis	5
Database Optimization	5
Caching Strategy	6
Identified Gaps & Improvement Recommendations	6
Security Enhancements	6
User Experience Improvements	6
Performance Optimizations	6
Monitoring & Observability	7
Business Intelligence & Analytics	7
Key Performance Indicators (KPIs)	7
Recommended Dashboard Views	7
Integration Opportunities	8
External System Integrations	8
Third-Party Service Enhancements	8
Deployment & DevOps Recommendations	8
Production Deployment Strategy	8
Security Hardening	9
Technical Specifications	9
Database Schema Overview	9
API Response Formats	9
Security Implementation Details	9
Conclusion	9
Immediate Next Steps	9
Long-term Roadmap	10

BookMyHotel - Comprehensive Project Documentation

Version: 1.0

Date: August 27, 2025

Project: BookMyHotel Multi-Tenant Platform

Executive Summary

BookMyHotel is a sophisticated multi-tenant hotel management platform with dual functionality covering hotel booking operations and in-hotel shop management. The system implements enterprise-grade architecture with robust security, multi-tenancy, and comprehensive role-based access control.

System Architecture Overview

Technology Stack

- **Backend:** Spring Boot 3.x (Java 21), Spring Security, JPA/Hibernate
- **Frontend:** React 18+ TypeScript, Material-UI v5
- **Database:** MySQL 8.0 with multi-tenant row-level isolation
- **Authentication:** JWT-based with role-based access control
- **Payment:** Stripe integration with webhooks
- **Infrastructure:** Docker, Docker Compose

Multi-Tenancy Strategy

- **Shared Database, Shared Schema** with tenant isolation via `tenant_id`
- **Automatic Tenant Context** resolution from JWT tokens
- **Hibernate Filters** for transparent row-level security
- **Thread-Local Context** management for request isolation

Hotel Booking System Analysis

Booking Flow Architecture

Guest Booking Journey

1. Guest Search Hotels
2. Select Dates & Room Type
3. Guest Information Form
4. Authentication Choice (Anonymous or Registered)
5. Payment Processing
6. Email Confirmation
7. Booking Management Token

Key API Endpoints - Hotel Booking

Endpoint	Method	Purpose	Authentication
/api/bookings	POST	Create new booking	Optional
/api/bookings/{id}	GET	Get booking details	JWT or Token
/api/bookings/{id}/modify	PUT	Modify booking	JWT or Token
/api/bookings/{id}/cancel	PATCH	Cancel booking	JWT or Token
/api/bookings/{id}/pdf	GET	Download booking PDF	JWT or Token
/api/front-desk/checkin/{id}	PATCH	Check-in guest	FRONT_DESK role
/api/front-desk/checkout/{id}	PATCH	Check-out guest	FRONT_DESK role

Payment Processing

Supported Payment Methods

- **Stripe Credit Cards:** Real-time processing with 3D Secure
- **Pay at Front Desk:** Deferred payment option
- **Ethiopian Mobile Money:** M-birr and Telebirr integration
- **Cash Payments:** Front desk processing

Payment Security

- PCI DSS compliance via Stripe
- Webhook signature verification
- Idempotency keys for duplicate prevention
- Audit logging for all transactions

Guest Management

Guest Types

1. **Anonymous Guests:** No account, token-based booking management
2. **Registered Users:** Full account with booking history
3. **Corporate Guests:** Business account with billing features

Shop Management System Analysis

Shop Architecture Overview

The shop system integrates product management, inventory control, order processing, payment tracking, and analytics dashboard with room service and front desk operations.

Key API Endpoints - Shop Management

Endpoint	Method	Purpose	Role Requirements
/api/hotels/{hotelId}/shop/products	GET/POST	Product CRUD	HOTEL_ADMIN, FRONT_DESK
/api/hotels/{hotelId}/shop/products/{id}/stock	PATCH	Update stock inventory	HOTEL_ADMIN, FRONT_DESK
/api/hotels/{hotelId}/shop/inventory/summary	GET	Inventory overview	HOTEL_ADMIN, FRONT_DESK
/api/hotels/{hotelId}/shop/orders	GET/POST	Order management	HOTEL_ADMIN, FRONT_DESK
/api/hotels/{hotelId}/shop/orders/{id}/update	PATCH	Update order status	HOTEL_ADMIN, FRONT_DESK
/api/hotels/{hotelId}/shop/orders/status	GET	Order analytics	HOTEL_ADMIN

Product & Inventory Management

Product Categories

- FOOD
- BEVERAGE
- TOILETRIES
- ELECTRONICS
- SOUVENIRS
- SNACKS
- OTHER

Inventory Control Features

- **Low Stock Alerts:** Configurable threshold monitoring
- **Out of Stock Management:** Automatic availability updates
- **Stock Movement Tracking:** Full audit trail
- **Bulk Inventory Updates:** Excel import/export capabilities

Order Processing System

Order Status Workflow PENDING → CONFIRMED → PREPARING → READY_FOR_PICKUP → DELIVERED

Alternative flows: - PENDING/CONFIRMED/PREPARING → CANCELLED - DELIVERED/CANCELLED → REFUNDED

Order Management Features

- **Real-time Status Updates:** Live order tracking
- **Room Delivery Integration:** Direct room service
- **Payment Tracking:** Multiple payment method support
- **Automatic Stock Deduction:** Inventory management integration

Security Architecture Analysis

Authentication & Authorization

JWT Token Structure

- Subject: user email
- User ID and personal information
- Roles array
- Tenant ID (null for system-wide users)
- Issued and expiration timestamps

Role-Based Access Control Matrix

Feature	GUEST	FRONT_DESK	HOTEL_ADMIN	SYSTEM_ADMIN
View own bookings	Yes	No	No	Yes
Manage all bookings	No	Yes	Yes	Yes
Product management	No	Yes	Yes	Yes
Order management	No	Yes	Yes	Yes
Analytics access	No	No	Yes	Yes
User management	No	No	Yes	Yes
Cross-tenant access	No	No	No	Yes

Multi-Tenant Security

Tenant Isolation Mechanisms

1. **Row-Level Security:** Hibernate filters on all entities
2. **JWT-Based Context:** Automatic tenant resolution
3. **API Endpoint Scoping:** Hotel ID in URL paths
4. **Database Constraints:** Foreign key tenant validation

Performance & Scalability Analysis

Database Optimization

Key Indexes

-- Booking system indexes

```
CREATE INDEX idx_reservations_tenant_hotel ON reservations(tenant_id, hotel_id);
CREATE INDEX idx_reservations_dates ON reservations(check_in_date, check_out_date);
CREATE INDEX idx_reservations_status ON reservations(status, tenant_id);
```

-- Shop system indexes

```
CREATE INDEX idx_shop_orders_tenant_hotel ON shop_orders(tenant_id, hotel_id);
CREATE INDEX idx_shop_orders_status ON shop_orders(status, tenant_id);
CREATE INDEX idx_products_tenant_active ON products(tenant_id, is_active);
```

Query Performance Optimization

- **Pagination:** All list endpoints support page/size parameters
- **Filtered Queries:** Search and status filtering capabilities
- **Eager Loading:** Strategic use of JOIN FETCH for related entities
- **Connection Pooling:** HikariCP for optimal database connections

Caching Strategy

Cacheable Operations

- **Hotel Information:** Rarely changes, 1-hour TTL
- **Room Types:** Static data, 4-hour TTL
- **Product Categories:** Static data, 24-hour TTL
- **User Permissions:** User-specific, 30-minute TTL

Identified Gaps & Improvement Recommendations

Security Enhancements

High Priority

- ☐ **Rate Limiting:** Implement API rate limiting per tenant/user
- ☐ **Input Validation:** Enhanced validation for all user inputs
- ☐ **Audit Logging:** Comprehensive action logging for compliance
- ☐ **Session Management:** Redis-based session store for scalability

Medium Priority

- ☐ **Two-Factor Authentication:** SMS/email OTP for admin accounts
- ☐ **API Versioning:** Support for backward compatibility
- ☐ **CORS Configuration:** Production-ready CORS policies
- ☐ **Security Headers:** Implement security headers (HSTS, CSP, etc.)

User Experience Improvements

Hotel Booking System

- ☐ **Real-time Availability:** WebSocket-based live room availability
- ☐ **Booking Modification:** Enhanced modification workflow with pricing updates
- ☐ **Guest Portal:** Comprehensive self-service portal
- ☐ **Mobile Optimization:** Progressive Web App (PWA) capabilities

Shop Management System

- ☐ **Inventory Forecasting:** AI-based demand prediction
- ☐ **Barcode Scanning:** Mobile barcode support for inventory
- ☐ **Automated Reordering:** Low-stock automatic purchase orders
- ☐ **Customer Analytics:** Purchase pattern analysis

Performance Optimizations

Backend Improvements

- ☐ **Database Connection Pooling:** Optimize HikariCP settings
- ☐ **Query Optimization:** Review and optimize N+1 queries
- ☐ **Async Processing:** Background job processing for emails/notifications
- ☐ **CDN Integration:** Static asset delivery optimization

Frontend Improvements

- ☐ **Code Splitting:** Route-based code splitting for faster loads
- ☐ **Image Optimization:** WebP format and lazy loading
- ☐ **State Management:** Redux Toolkit for complex state
- ☐ **Error Boundaries:** Comprehensive error handling

Monitoring & Observability

Metrics & Alerting

- ☐ **Business Metrics:** Booking conversion rates, revenue tracking
- ☐ **Performance Monitoring:** APM integration (New Relic/DataDog)
- ☐ **Error Tracking:** Sentry integration for error reporting
- ☐ **Health Checks:** Comprehensive health check endpoints

Logging Enhancements

- ☐ **Structured Logging:** JSON-formatted logs with correlation IDs
- ☐ **Log Aggregation:** ELK stack for centralized logging
- ☐ **Sensitive Data Masking:** PII protection in logs
- ☐ **Log Retention Policies:** Compliant log retention

Business Intelligence & Analytics

Key Performance Indicators (KPIs)

Hotel Booking Metrics

- **Occupancy Rate:** Room utilization percentage
- **Average Daily Rate (ADR):** Revenue per occupied room
- **Revenue Per Available Room (RevPAR):** Overall revenue efficiency
- **Booking Conversion Rate:** Visitor to booking conversion
- **Cancellation Rate:** Booking cancellation percentage

Shop Management Metrics

- **Inventory Turnover:** Stock movement efficiency
- **Order Fulfillment Time:** Average order processing time
- **Customer Satisfaction:** Order rating and feedback
- **Revenue Per Guest:** Shop revenue contribution
- **Popular Products:** Best-selling item analysis

Recommended Dashboard Views

Executive Dashboard

- Hotel metrics: occupancy rate, total revenue, active bookings, average daily rate
- Shop metrics: total orders, shop revenue, inventory value, low stock alerts
- Trends: revenue growth, booking trends, popular products

Operations Dashboard

- **Real-time Room Status:** Occupancy, cleaning, maintenance
- **Pending Orders:** Order queue with status tracking
- **Staff Task Management:** Housekeeping and maintenance tasks
- **Inventory Alerts:** Low stock and reorder notifications

Integration Opportunities

External System Integrations

Recommended Integrations

- **Channel Manager:** Booking.com, Expedia integration
- **Property Management System (PMS):** Legacy system migration
- **Accounting Software:** QuickBooks, Xero integration
- **Customer Relationship Management (CRM):** Salesforce integration
- **Business Intelligence:** Power BI, Tableau connectors

Third-Party Service Enhancements

Communication Services

- **SMS Notifications:** Twilio integration for booking confirmations
- **Push Notifications:** Firebase for mobile app notifications
- **Voice Calls:** Automated booking confirmations
- **Live Chat:** Customer support integration

Advanced Features

- **AI Chatbot:** Automated customer service
- **Dynamic Pricing:** Revenue optimization algorithms
- **Loyalty Program:** Points and rewards system
- **Review Management:** TripAdvisor, Google Reviews integration

Deployment & DevOps Recommendations

Production Deployment Strategy

Infrastructure Requirements

- **Application:** 4 cores minimum, 8GB RAM minimum, SSD-based storage
- **Database:** MySQL 8.0 with replication, daily automated backups, read replicas for analytics
- **Load Balancer:** HTTPS termination, health check configuration, session affinity support

CI/CD Pipeline

- **Source Control:** Git with feature branch workflow
- **Build Automation:** GitHub Actions or Jenkins
- **Testing:** Automated unit, integration, and e2e tests
- **Deployment:** Blue-green deployment strategy
- **Monitoring:** Post-deployment verification

Security Hardening

Production Security Checklist

- ☐ **HTTPS Everywhere:** Force HTTPS with HSTS headers
- ☐ **Database Security:** Encrypted connections and storage
- ☐ **Secrets Management:** HashiCorp Vault or AWS Secrets Manager
- ☐ **Network Security:** VPC with private subnets
- ☐ **Backup Encryption:** Encrypted backup storage
- ☐ **Compliance:** GDPR, PCI DSS compliance verification

Technical Specifications

Database Schema Overview

Core Entities The system uses a multi-tenant architecture with the following key tables:

- **tenants:** Tenant configuration and metadata
- **hotels:** Hotel information (tenant-scoped)
- **reservations:** Booking data with comprehensive indexes
- **products:** Shop inventory with category management
- **shop_orders:** Order processing with status tracking
- **users:** User accounts with role-based permissions

All tenant-scoped tables include `tenant_id` for proper isolation and performance indexes.

API Response Formats

Standard API Response Structure Success responses include success flag, data payload, optional message, and timestamp. Error responses include success flag (false), error object with code and message, and timestamp. Paginated responses include content array, total elements, pagination metadata.

Security Implementation Details

JWT Token Generation Tokens include user information, roles, tenant context, and proper expiration handling.

Multi-Tenant Filter Implementation Automatic tenant filtering through Hibernate filters with pre-persist tenant ID assignment.

Conclusion

BookMyHotel represents a well-architected, enterprise-grade hotel management platform with robust multi-tenancy, comprehensive security, and dual functionality for both booking and shop management. The identified improvements focus on enhancing user experience, strengthening security, optimizing performance, and adding advanced business intelligence capabilities.

Immediate Next Steps

1. **Implement rate limiting** for API protection
2. **Add comprehensive audit logging** for compliance

3. **Optimize database queries** for better performance
4. **Enhance error handling** and user feedback
5. **Set up production monitoring** and alerting

Long-term Roadmap

1. **Mobile application development** (React Native)
2. **Advanced analytics and AI integration**
3. **Multi-language and multi-currency support**
4. **Advanced booking modification workflows**
5. **Integration with major OTA platforms**

The platform provides a solid foundation for scaling hotel operations and can accommodate future enhancements while maintaining security and performance standards.

Document Generated: August 27, 2025

Version: 1.0

Project: BookMyHotel Multi-Tenant Platform