Software Requirements Specification (SRS)
Otithi Vacation Rental Platform

Development Team

July 2025

Contents

Chapter 1

Introduction

1.1 Background

The tourism and hospitality industry in Bangladesh has experienced significant growth in recent years, with an increasing number of domestic and international travelers seeking alternative accommodation options beyond traditional hotels. The rise of the sharing economy globally has demonstrated the viability of peer-to-peer accommodation platforms, with services like Airbnb and Booking.com achieving massive success worldwide.

However, Bangladesh lacks a robust, localized vacation rental platform that addresses the specific needs of the local market, including support for Bangladeshi Taka, local payment methods, and cultural preferences. The existing international platforms often lack proper localization and fail to cater to the unique requirements of Bangladeshi hosts and guests.

1.2 Motivation

The development of Otithi is motivated by several key factors:

- The growing demand for alternative accommodation options in Bangladesh's expanding tourism sector
- The need for a platform that supports local payment methods and currency
- The opportunity to provide income generation for property owners across Bangladesh
- The requirement for a culturally appropriate and linguistically accessible platform
- The potential to boost domestic tourism and contribute to the local economy
- The gap in the market for a comprehensive, secure, and user-friendly vacation rental platform

1.3 Problem Statement

Currently, the Bangladeshi vacation rental market faces several challenges:

- Lack of a centralized, trusted platform for vacation rentals

- Limited support for local payment methods and Bangladeshi Taka
- Absence of real-time booking and availability management systems
- Inadequate security measures for both hosts and guests
- Limited options for hosts to showcase their properties effectively
- Difficulty for travelers to find and book suitable accommodations
- Lack of standardized review and rating systems
- Absence of proper dispute resolution mechanisms

1.4 Objectives

The primary objectives of the Otithi platform are:

- To provide a seamless, intuitive experience for both hosts and guests
- To ensure secure, real-time booking and payment processing
- To support dynamic pricing with real-time availability updates
- To offer comprehensive property showcase capabilities with high-quality images
- To implement robust user management for admins, hosts, and guests
- To create a trusted environment through verified reviews and ratings
- To support multiple languages (Bengali and English) for accessibility
- To integrate local payment methods and currency support
- To establish a scalable platform that can grow with market demands
- To contribute to the growth of Bangladesh's tourism industry

Chapter 2

System Overview

Otithi is a comprehensive full-stack web application designed specifically for the Bangladeshi vacation rental market. The platform serves as a digital marketplace connecting property owners (hosts) with travelers (guests) seeking short-term accommodation.

The system architecture follows a modern three-tier approach:

- Presentation Layer: Responsive web interface built with HTML5, CSS3, JavaScript, and Bootstrap framework
- Business Logic Layer: Python-based backend using Flask framework for API development and business rule implementation
- Data Layer: MySQL database for persistent data storage with optimized queries and indexing

Key system components include:

- User authentication and authorization system
- Property listing management module
- Search and booking engine
- Payment processing integration
- Review and rating system
- Administrative dashboard
- Notification and communication system
- Image upload and gallery management
- Real-time availability and pricing calculator

The platform supports three distinct user roles: Guests (travelers seeking accommodation), Hosts (property owners), and Administrators (system managers), each with specific functionalities and access levels.

Chapter 3

System Life Cycle

The Otithi platform development follows a structured Software Development Life Cycle (SDLC) approach:

1. Requirements Analysis
    - Stakeholder interviews and market research
    - Functional and non-functional requirements gathering
    - Use case identification and documentation
    - System constraints and assumptions definition
2. System Design
    - Architecture design and technology stack selection
    - Database schema design and optimization
    - User interface mockups and wireframes
    - API design and endpoint specifications
    - Security architecture and protocols
3. Implementation
    - Backend development using Python Flask
    - Frontend development with responsive design
    - Database implementation and migration scripts
    - Third-party integrations (payment gateways, email services)
    - API development and testing
4. Testing
    - Unit testing for individual components
    - Integration testing for system modules
    - User acceptance testing with real users
    - Security testing and vulnerability assessment
    - Performance testing under various loads

- Mobile responsiveness testing

5. Deployment
    - Production environment setup
    - Database migration and data seeding
    - SSL certificate installation and security configuration
    - Performance monitoring and logging setup
    - Backup and disaster recovery implementation
6. Maintenance and Updates
    - Regular security updates and patches
    - Performance monitoring and optimization
    - Feature enhancements based on user feedback
    - Bug fixes and issue resolution
    - System scaling and infrastructure improvements

Chapter 4

Functional Requirements

4.1 List of Features

1. User Registration and Authentication
    - Email and phone number based registration
    - Secure login with password hashing
    - Email and SMS verification
    - Password reset functionality
    - Social login integration (Google, Facebook)
2. Host Listing Management
    - Create new property listings
    - Edit existing listing details
    - Upload multiple property images
    - Set pricing and availability calendar
    - Manage listing status (active/inactive)
3. Guest Booking System
    - Search properties by location, dates, and guests
    - View detailed property information
    - Real-time availability checking
    - Secure booking process
    - Booking confirmation and management
4. Real-Time Price Calculation
    - Dynamic pricing based on dates and occupancy
    - Automatic tax and fee calculation
    - Discount and promotion code support
    - Multi-currency support with BDT as primary

5. Review and Rating System
    - Post-stay review submission
    - Star rating system (1-5 stars)
    - Review verification through booking history
    - Host response to reviews
    - Review moderation and filtering
6. Admin Dashboard
    - User management and monitoring
    - Listing approval and moderation
    - Booking oversight and dispute resolution
    - Analytics and reporting
    - System configuration and settings
7. Image Upload and Gallery
    - Multiple image upload support
    - Image compression and optimization
    - Gallery management and ordering
    - Image quality validation
8. Search and Explore Listings
    - Advanced search filters
    - Map-based property exploration
    - Sorting options (price, rating, distance)
    - Saved searches and favorites
9. Profile Management
    - User profile creation and editing
    - Profile picture upload
    - Booking history and management
    - Host dashboard with earnings overview
10. Secure Payment Integration
    - Multiple payment gateway support
    - bKash and Nagad integration
    - Credit/debit card processing
    - Payment history and receipts
    - Refund processing

4.2 Feature Table

Feature	Description
User Registration	Comprehensive sign-up process with email/phone verification, secure password creation, and profile setup for both guests and hosts
User Authentication	Secure login system with password hashing, session management, password reset, and optional social login integration
Listing Management	Complete property management suite allowing hosts to create, edit, delete, and manage their property listings with images and details
Property Search	Advanced search functionality with filters for location, dates, price range, property type, and amenities
Booking System	End-to-end booking process from property selection to payment confirmation with real-time availability checking
Price Calculation	Dynamic pricing system calculating total costs including base price, taxes, cleaning fees, and service charges in real-time
Payment Processing	Secure payment integration supporting local methods (bKash, Nagad) and international cards with BDT currency support
Reviews and Ratings	Comprehensive review system allowing guests to rate and review properties with host response capabilities
Admin Dashboard	Complete administrative interface for user management, listing moderation, booking oversight, and system analytics
Image Gallery	Robust image management system with upload, compression, gallery creation, and quality validation features
Profile Management	User profile system for personal information, booking history, host earnings, and account settings management
Notifications	Real-time notification system for bookings, messages, payment confirmations, and system updates

Chapter 5

System Specification Diagrams

5.1 Use Case Diagram

The use case diagram illustrates the interactions between different actors (Guest, Host, Admin) and the system functionalities. The diagram shows the primary use cases and their relationships.

Note: Use Case Diagram would be inserted here showing actors and their interactions with the system.

5.2 Use Case Narrative Table

Actor	**Use Case**	**Description**
Guest	Register/Login	Create new account or authenticate existing user with email/phone and password
Guest	Search Listings	Find available properties using filters like location, dates, price range, and amenities
Guest	View Property Details	Access comprehensive information about a property including images, amenities, and reviews
Guest	Book Property	Complete booking process including date selection, payment, and confirmation
Guest	Manage Bookings	View, modify, or cancel existing bookings and access booking history
Guest	Leave Review	Submit ratings and written reviews for properties after completed stays
Guest	Update Profile	Edit personal information, preferences, and account settings
Host	Register as Host	Complete host onboarding process with identity verification and property details
Host	Create Listing	Add new property with descriptions, images, pricing, and availability
Host	Manage Listings	Edit, activate, deactivate, or delete existing property listings

Host	View Bookings	Access booking requests, confirmations, and guest information
Host	Manage Calendar	Update property availability and pricing for different dates
Host	Respond to Reviews	Reply to guest reviews and feedback about their properties
Host	View Earnings	Access financial dashboard with booking income and payout information
Admin	Manage Users	Oversee user accounts, handle verification, and manage user permissions
Admin	Moderate Listings	Review and approve new listings, ensuring quality and compliance
Admin	Handle Disputes	Resolve conflicts between guests and hosts through mediation
Admin	View Analytics	Access system statistics, user behavior, and performance metrics
Admin	Manage Content	Moderate reviews, manage reported content, and maintain platform quality
Admin	System Configuration	Update system settings, payment methods, and platform policies

5.3 Data Flow Diagram (DFD)

The Data Flow Diagram shows how data moves through the system, including data stores, processes, and external entities.

Note: DFD would be inserted here showing data flow between processes, stores, and external entities.

5.4 Sequence Diagram

Sequence diagrams illustrate the interaction between different system components over time for key processes like booking, payment, and user authentication.

Note: Sequence diagrams would be inserted here showing object interactions over time.

5.5 Activity Diagram

Activity diagrams show the workflow of key system processes including user registration, property booking, and listing creation.

Note: Activity diagrams would be inserted here showing process workflows.

5.6 Class Diagram

The class diagram represents the object-oriented structure of the system, showing classes, attributes, methods, and relationships.

Note: Class diagram would be inserted here showing system classes and their relationships.

Chapter 6

Non-Functional Requirements

6.1 Security Requirements

1. All data transmission must be encrypted using HTTPS/TLS 1.3 protocol
2. User passwords must be hashed using bcrypt with minimum 12 rounds
3. Payment information must comply with PCI DSS standards
4. API endpoints must implement rate limiting to prevent abuse
5. User sessions must expire after 24 hours of inactivity
6. All user inputs must be validated and sanitized to prevent injection attacks
7. System must implement proper authentication and authorization mechanisms
8. Security audit logs must be maintained for all critical operations

6.2 Performance Requirements

1. Page load time must not exceed 3 seconds under normal load conditions
2. Database queries must execute within 500 milliseconds
3. Search results must be displayed within 2 seconds of query submission
4. Image upload and processing must complete within 10 seconds
5. System must support minimum 1,000 concurrent users
6. API response time must not exceed 200 milliseconds for standard requests

6.3 Scalability Requirements

1. System must support horizontal scaling to handle increased load
2. Database must efficiently handle growth to 100,000+ users and properties

3. Architecture must support load balancing across multiple servers
4. System must be designed for cloud deployment with auto-scaling capabilities
5. Code must be modular to support addition of new features and modules

6.4 Usability Requirements

1. User interface must be intuitive for users with basic computer literacy
2. System must provide clear error messages and user guidance
3. Mobile interface must be fully responsive and touch-friendly
4. System must support both Bengali and English languages
5. Help documentation and FAQs must be easily accessible
6. User workflow must require minimum clicks to complete tasks

6.5 Reliability Requirements

1. System uptime must be 99.5 percent or higher
2. Mean Time Between Failures (MTBF) must be at least 720 hours
3. System must gracefully handle and recover from failures
4. Data backup must be performed daily with point-in-time recovery
5. System must recover from failures within 4 hours maximum

6.6 Availability Requirements

1. System must be available 24/7 with minimal planned downtime
2. Planned maintenance windows must not exceed 2 hours monthly
3. System must have redundant components to prevent single points of failure
4. Emergency response procedures must be in place for system outages

6.7 Compatibility Requirements

1. System must work on all modern web browsers (Chrome, Firefox, Safari, Edge)
2. Mobile compatibility must support iOS 12+ and Android 8+
3. System must be responsive across all device sizes and orientations
4. Database must be compatible with MySQL 8.0 or higher
5. API must follow REST standards for third-party integrations

6.8 Maintainability Requirements

1. Code must follow established coding standards and best practices
2. System must have comprehensive documentation for all components
3. Code must be modular with clear separation of concerns
4. System must have automated testing coverage of at least 80 percent
5. Database schema must be version-controlled with migration scripts
6. System must have logging and monitoring for troubleshooting

Chapter 7

Risk Analysis

7.1 SWOT Analysis

7.1.1 Strengths

- Localized Solution: Specifically designed for the Bangladeshi market with local payment methods, currency support, and cultural considerations
- Real-time Data Processing: Advanced booking system with real-time availability and pricing calculations
- Robust Security: Implementation of modern security practices including encryption, secure payments, and data protection
- Comprehensive Feature Set: Complete ecosystem covering all aspects of vacation rental management
- Mobile-First Design: Responsive interface optimized for mobile users, reflecting local usage patterns
- Multi-language Support: Bengali and English language support for broader accessibility
- Scalable Architecture: Modern technology stack designed for growth and expansion

7.1.2 Weaknesses

- Initial User Adoption: Challenge of building user base from scratch in a competitive market
- Internet Dependency: Requires stable internet connectivity which may be limited in rural areas
- Limited Brand Recognition: New platform competing against established international brands
- Resource Constraints: Limited budget for marketing and rapid feature development
- Technical Complexity: Complex system requiring skilled developers for maintenance and updates

- Trust Building: Need to establish credibility and trust among users for financial transactions

7.1.3 Opportunities

- Growing Tourism Industry: Bangladesh's expanding tourism sector provides significant market potential
- Digital Payment Adoption: Increasing acceptance of digital payments (bKash, Nagad) creates favorable conditions
- Smartphone Penetration: Rising smartphone usage enables broader platform accessibility
- Government Support: Government initiatives promoting digital Bangladesh and tourism
- Post-pandemic Travel Recovery: Recovery of travel industry after COVID-19 restrictions
- Domestic Tourism Growth: Increasing domestic travel creating demand for local accommodations
- Partnership Opportunities: Potential collaborations with local businesses, tourism boards, and travel agencies

7.1.4 Threats

- International Competition: Established platforms like Airbnb and Booking.com entering the market
- Regulatory Changes: Potential changes in tourism, taxation, or digital payment regulations
- Economic Instability: Economic downturns affecting travel and tourism spending
- Data Security Breaches: Risk of cyber attacks and data breaches damaging reputation
- Technology Disruption: Rapid changes in technology requiring continuous platform updates
- Market Saturation: Potential oversaturation of the vacation rental market
- Seasonal Fluctuations: Tourism seasonality affecting platform usage and revenue

7.2 Risk Mitigation Strategies

- User Adoption: Implement comprehensive marketing strategy, referral programs, and competitive pricing

- Security Measures: Regular security audits, penetration testing, and compliance with international standards
- Technology Reliability: Robust infrastructure, redundancy, and disaster recovery procedures
- Market Competition: Focus on unique value propositions, superior customer service, and local market understanding
- Regulatory Compliance: Continuous monitoring of regulatory changes and proactive compliance measures
- Financial Risk: Diversified revenue streams, careful financial planning, and investor relations

Chapter 8

Conclusion

The Otithi Vacation Rental Platform represents a comprehensive solution specifically designed to address the unique needs of the Bangladeshi vacation rental market. This Software Requirements Specification document provides a detailed roadmap for developing a robust, secure, and user-friendly platform that can compete effectively in the growing tourism industry.

The platform's success will depend on several key factors:

- Technical Excellence: Implementation of all specified functional and non-functional requirements with attention to security, performance, and scalability
- User Experience: Creating an intuitive, accessible interface that caters to both tech-savvy users and those with limited technical experience
- Local Market Understanding: Leveraging deep understanding of Bangladeshi culture, payment preferences, and travel patterns
- Trust and Safety: Building a secure platform that protects both hosts and guests through verified reviews, secure payments, and dispute resolution
- Continuous Improvement: Regular updates based on user feedback, market trends, and technological advancements

The development team must carefully follow the specified requirements while remaining flexible enough to adapt to changing market conditions and user needs. Regular stakeholder reviews and iterative development approaches will ensure the platform meets its objectives and delivers value to all users.

The platform's contribution to Bangladesh's digital economy and tourism sector could be significant, providing income opportunities for property owners and convenient accommodation options for travelers. Success in this venture could also pave the way for expansion into other markets and additional services.

This SRS document should be treated as a living document, subject to updates and revisions as the project progresses and requirements evolve. All stakeholders must remain committed to the project's success through careful planning, skilled execution, and continuous monitoring of progress against the specified requirements.

References

- IEEE Standard 830-1998: IEEE Recommended Practice for Software Requirements Specifications
- Flask Documentation: https://flask.palletsprojects.com/
- MySQL Documentation: https://dev.mysql.com/doc/
- Python Software Foundation: https://www.python.org/
- Bootstrap Framework: https://getbootstrap.com/
- bKash API Documentation: https://developer.bkash.com/
- Nagad Payment Gateway: https://nagad.com.bd/
- Bangladesh Tourism Board: https://www.parjatan.gov.bd/
- Airbnb Engineering Blog: https://medium.com/airbnb-engineering
- Booking.com Technology: https://blog.booking.com/
- OWASP Security Guidelines: https://owasp.org/
- PCI DSS Standards: https://www.pcisecuritystandards.org/
- Web Content Accessibility Guidelines (WCAG) 2.1: https://www.w3.org/WAI/WCAG21/
- Bangladesh Association of Software and Information Services (BASIS): https://basis.org.bd/
- International Organization for Standardization (ISO) Standards for Software Engineering