
Software Requirements Specification

for

Resort Management System

Version 1.0 approved

Prepared by Jacinth Manuel J

Mepco Schlenk Engineering College

14.07.2025

Table of Contents

Table of Contents	ii
Revision History	iii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope	2
1.5 References.....	2
2. Overall Description	2
2.1 Product Perspective.....	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints.....	3
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	4
3.1 User Interfaces	4
3.2 Hardware Interfaces	4
3.3 Software Interfaces	4
3.4 Communications Interfaces	4
4. System Features	4
4.1 Secure Login System	4
4.2 Guest Information Management	5
4.3 Room Booking and Management	5
4.4 Check-In and Check-out Process	5
4.5 Room Details and Availability.....	5
4.6 Facility and Service Management.....	5
4.7 Billing and Payment System.....	5
4.8 Admin Dashboard & Reports.....	5
4.9 Feedback and Rating Module	6
4.10 Staff Management.....	6
4.11 Backup and Settings.....	6
5. Other Nonfunctional Requirements	6
5.1 Performance Requirements	6
5.2 Safety Requirements	6
5.3 Security Requirements	6
5.4 Software Quality Attributes	6
5.5 Business Rules	7
6. Other Requirements	7
Appendix A: Glossary.....	7
Appendix B: Analysis Models	7
Appendix C: To Be Determined List.....	8

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to provide a comprehensive overview of the functional and non-functional requirements for the Resort Management System (RMS). This system is designed to enhance the operational efficiency of resorts by automating various processes such as reservations, guest management, billing, and service delivery.

Additionally, the RMS incorporates advanced AI features, including a Room Recommendation System and a Customer Support Chatbot, which aim to improve guest satisfaction and streamline operations. This document serves as a guide for developers, testers, and stakeholders to ensure that the final product meets the specified requirements and fulfills the needs of the resort management.

1.2 Document Conventions

This document employs specific conventions to ensure clarity and consistency. Key terms and acronyms are defined in the glossary section. Functional requirements are presented in a structured format, detailing the inputs, processes, and expected outputs. Non-functional requirements are categorized into performance, safety, security, and quality attributes. All requirements are numbered for easy reference, and any changes or updates to the document will be tracked in the revision history section.

1.3 Intended Audience and Reading Suggestions

The intended audience for this SRS includes:

Project Developers & Engineers: They will use this document to understand the requirements and implement the system accordingly.

Testers & QA Team: This group will validate the functionalities against the requirements outlined in this document to ensure quality and performance.

Project Managers: They will track the scope, timelines, and deliverables based on the requirements specified.

Clients or Resort Owners: This audience will gain insights into the software capabilities and functionalities, helping them understand how the system can meet their operational needs.

1.4 Product Scope

The Resort Management System is designed to automate and manage various aspects of resort operations, including room booking, customer management, payment processing, housekeeping, and feedback handling. The system aims to provide a seamless experience for both guests and staff, reducing manual effort and minimizing errors. The scope includes the development of a user-friendly interface and robust reporting features for management oversight..

2. Overall Description

2.1 Product Perspective

The Resort Management System is envisioned as a standalone application that can be deployed as either a desktop or web application. It will consist of several internal modules, such as booking, inventory management, and billing, as well as external modules that leverage AI technologies, including a chatbot for customer support and a room recommendation engine.

The system will be designed to integrate with existing resort management tools and databases, ensuring a smooth transition and minimal disruption to current operations.

2.2 Product Functions

Booking Management: Facilitating room reservations and managing availability.

Customer Registration & Check-In/Check-Out: Streamlining the guest arrival and departure process.

Room Allotment: Efficiently assigning rooms based on guest preferences and availability.

Chatbot for Customer Support: Providing instant assistance to guests through an AI-driven interface.

Billing Process: Automating the generation of invoices and payment processing.

Admin Dashboard & Reports: Offering insights into operational metrics and performance.

Feedback and Ratings: Collecting guest feedback to improve services.

Multi-user Roles: Supporting different access levels for various staff roles.

Inventory & Staff Management: Managing resources and personnel effectively.

Event & Hall Booking: Allowing guests to reserve spaces for events.

Emergency/Help Desk: Providing immediate assistance in case of emergencies.

2.3 User Classes and Characteristics

The system will cater to various user classes, each with distinct roles and access levels:

Admin: Has full access to all system functionalities, including user management, reporting, and system settings.

Receptionist: Responsible for handling guest check-ins and check-outs, managing room bookings, and providing customer service.

Customer: Guests who can access the booking portal to make reservations and manage their profiles.

Support Staff: Personnel who have access to housekeeping and transport modules to ensure guest needs are met.

2.4 Operating Environment

The RMS will be developed using a combination of technologies:

Frontend: The user interface will be built using Java or Python, ensuring a responsive and intuitive design.

Backend: The system will utilize MySQL or MongoDB for data storage, providing flexibility and scalability.

AI Modules: The integration of AI functionalities will be achieved using libraries such as Scikit-learn, NLTK, or RASA.

Operating System: The application will be compatible with Windows 10/11.

Integrated Development Environment (IDE): Development will be conducted using tools like VS Code or PyCharm.

2.5 Design and Implementation Constraints

The system must be designed to function offline with limited features, ensuring that essential operations can continue without internet connectivity. A mobile application version is considered for future development, allowing users to access the system on-the-go.

2.6 User Documentation

User documentation will be provided to assist users in navigating the system effectively. This will include:

User Manuals: Detailed guides for each user class, outlining functionalities and procedures.

Online Help: Contextual help within the application to assist users in real-time.

Training Materials: Resources for onboarding staff and ensuring they are familiar with the system's capabilities.

2.7 Assumptions and Dependencies

The successful operation of the RMS relies on certain assumptions and dependencies:

An active internet connection is required for the chatbot functionality, especially if it utilizes the OpenAI API for natural language processing.

The MongoDB server must be operational to ensure data accessibility and integrity.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be designed to be responsive and user-friendly, utilizing frameworks such as Tkinter/Applet for desktop applications or Flask for web applications. The interface will feature clean navigation, intuitive icons, and tooltips to guide users through various functionalities.

3.2 Hardware Interfaces

The system will require a desktop or laptop with a minimum of 4GB RAM to ensure smooth operation and responsiveness during peak usage times.

3.3 Software Interfaces

The RMS will interface with:

Databases: MySQL or MongoDB for data storage and retrieval.

Programming Languages: Python/Java for backend development and integration of AI modules.

Optional APIs: Third-party services for payment processing, SMS notifications, or additional functionalities.

3.4 Communications Interfaces

The system will utilize standard communication protocols to ensure secure data transmission between the client and server. This may include HTTPS for web applications and secure socket connections for desktop applications.

4. System Features

The system will utilize standard communication protocols to ensure secure data transmission between the client and server. This may include HTTPS for web applications and secure socket connections for desktop applications.

4.1 Secure Login System

The RMS will implement a secure login system that differentiates access levels based on user roles. Admins, receptionists, and guests will have separate login credentials, ensuring that sensitive information is protected. Passwords will be encrypted, and role-based access control will be enforced to restrict unauthorized access.

4.2 Guest Information Management

The system will allow staff to add, edit, or delete guest profiles, storing essential information such as name, contact details, identification proof, and stay history. A quick search and filtering feature will enable staff to retrieve guest records efficiently, enhancing operational efficiency.

4.3 Room Booking and Management

The RMS will facilitate room bookings based on type, availability, and guest preferences. Staff will have the ability to modify or cancel bookings as needed, and the system will maintain real-time updates on room status, indicating whether rooms are occupied, vacant, or being cleaned.

4.4 Check-In and Check-Out Process

The check-in process will involve assigning available rooms to guests, uploading and verifying identification documents, and generating a final bill along with a feedback form during check-out. This streamlined process will enhance the guest experience and ensure accurate billing.

4.5 Room Details and Availability

The system will allow staff to add or update room types, such as Single, Double, or Deluxe, and track real-time availability. Guests will be able to filter rooms based on specific features, such as air conditioning, television, or balcony access, to find accommodations that meet their needs.

4.6 Facility and Service Management

The RMS will display available facilities, such as gym, pool, Wi-Fi, and restaurant services. Staff will be able to add extra charges to guest bills for utilizing these services and schedule appointments or services while assigning staff accordingly.

4.7 Billing and Payment System

An automated billing system will generate invoices during the check-out process, including room charges, food, and additional services. The system will support multiple payment methods, including cash, UPI, and credit/debit cards, ensuring convenience for guests.

4.8 Admin Dashboard & Reports

The admin dashboard will provide a comprehensive overview of operational metrics, including total bookings, room occupancy rates, and daily/weekly revenue. Graphs and reports will be generated to visualize guest visits, payments, and service usage, with options to export reports for management review.

4.9 Feedback and Rating Module

The system will collect guest feedback at check-out, allowing guests to provide star ratings and suggestions. Admins will have the ability to review this feedback and implement improvements to services based on guest input, fostering a culture of continuous improvement.

4.10 Staff Management

The RMS will include features for managing staff, allowing admins to add new personnel, assign roles, update shift timings, and remove inactive staff. This functionality will ensure that the resort operates efficiently and that staff responsibilities are clearly defined.

4.11 Backup and Settings

The system will include a backup feature to securely store all system data, ensuring that information can be restored in case of data loss. Admins will have exclusive access to system settings, allowing them to reset configurations or export data as needed.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The RMS must be capable of handling multiple concurrent users without performance degradation. The system should maintain a response time of less than two seconds for all user interactions, ensuring a smooth and efficient user experience.

5.2 Safety Requirements

Safety requirements will include measures to protect guest data and ensure compliance with relevant regulations, such as data protection laws. The system will implement secure data handling practices to prevent unauthorized access and data breaches.

5.3 Security Requirements

The RMS will enforce role-based access control to restrict user permissions based on their roles. Passwords will be stored in an encrypted format, and the system will include backup and restore capabilities to safeguard against data loss.

5.4 Software Quality Attributes

Quality attributes of the software will include reliability, maintainability, and usability. The system will be designed to minimize downtime, facilitate easy updates and maintenance, and provide an intuitive user interface that enhances user satisfaction.

5.5 Business Rules

Business rules will define the operational principles governing the system, such as which roles can perform specific functions under certain conditions. For example, only admins may have the authority to modify system settings, while receptionists can manage guest bookings.

6. Other Requirements

6.1 Database Requirements

The RMS will utilize a NoSQL database, such as MongoDB, to store guest records, booking history, payment details, and feedback data. The database must support fast read/write operations, data scalability, and replication for backup purposes, ensuring data integrity and availability.

6.2 Performance Requirements

The system should be capable of handling concurrent requests from multiple users without noticeable delays. Page loading times must be kept under three seconds for all modules to ensure a responsive user experience.

Appendix A: Glossary

Term	Description
RMS	Resort Management System
NoSQL	Non-relational database model used for scalability and performance
Admin	System manager with full privileges
Guest	A customer who books rooms or services in the resort
Check-in/Check-out	The process of arrival and departure of guests
MongoDB	A NoSQL database solution used in this project

Appendix B: Analysis Models

This section may include various analysis models that visually represent the system's architecture and data flow. Examples include:

Data Flow Diagrams (DFD): Illustrating how data moves through the system.

Class Diagrams: Showing the relationships between different classes in the system.

State-Transition Diagrams: Depicting the states of the system and transitions based on events.

Entity-Relationship Diagrams (ERD): Representing the data entities and their relationships within the database.

Appendix C: To Be Determined List

This section will contain a numbered list of items that require further clarification or decision-making before the project can proceed. Examples may include:

Finalization of payment gateway options.

Selection of AI technologies for the chatbot.

Determination of mobile app features for future versions.