Software Requirements Specification

for

Hotel Reservation and Guest Services Management System

Version 1.0 approved.

Prepared by Group 40

AHAMED M.A.H. 230022B

DEVINDU W.G.T. 230134U

G. JANUSHIKHA 230190K

IFAZ M.I.M. 230253H

JAYAMANNE J.M.A.N.N. 230272N

07-08-2025

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This SRS document explains the requirements for the first phase of SkyNest Hotels' Hotel Reservation and Guest Services Management System, which is a regional hotel chain operating in Sri Lanka. It simply gives a detailed definition of the database design and basic functionality to be captured for managing hotel branches, guest reservations, service tracking, and billing within HRGSMS.

This document details only the first phase of the project, covering branch management, reservation management, guest check-ins/outs, and service usage, including the special online booking system. The revision number for this document is 1.0.

1.2 Document Conventions

This Software Requirements Specification (SRS) follows the IEEE Std 830-1998 standard for documenting software requirements. The following conventions are used throughout the document:
Bold text is used to highlight key terms, system component names, user roles, and significant entities.
Italic text is used for system actions, screen labels, or interface-related terms.
Functional requirements are labeled sequentially using the format REQ-1, REQ-2, etc., for traceability.
Requirement priorities are classified as

- **Priority:** High—Essential for system success or regulatory compliance.
- **Priority:** Medium—Important but not critical for the initial release.
- **Priority:** Low—Optional or nice-to-have features.

1.3 Intended Audience and Reading Suggestions

The following individuals are the target audience for this document:

- **Developers**: For understanding system architecture, database design, and implementation guidelines.
- **Project Managers**: For tracking progress and ensuring compliance with the requirements.
- **QA Testers**: For developing test cases based on the specified requirements.

- **Documentation Writers**: For creating user manuals and help guides.
- **Hotel Management and Stakeholders**: For reviewing the scope and ensuring the project aligns with business goals.

It is recommended that readers begin with the overview sections (Introduction, Product Scope) before moving to the more detailed requirements and functional specifications.

1.4 Product Scope

The Hotel Reservation and Guest Services Management System (HRGSMS) is designed to replace SkyNest Hotels' outdated system with a modern, database-driven platform. This software will handle branch management, guest reservations, and service tracking, ensuring alignment with Sri Lanka's hospitality industry standards. The HRGSMS supports the hotel's strategy to improve operational efficiency, level up guest experience, and offer a better service, including online bookings and real-time room availability. The system also connects with external payment gateways, enabling smooth transactions and digital invoicing, which is important for the hotel's growth and guest satisfaction.

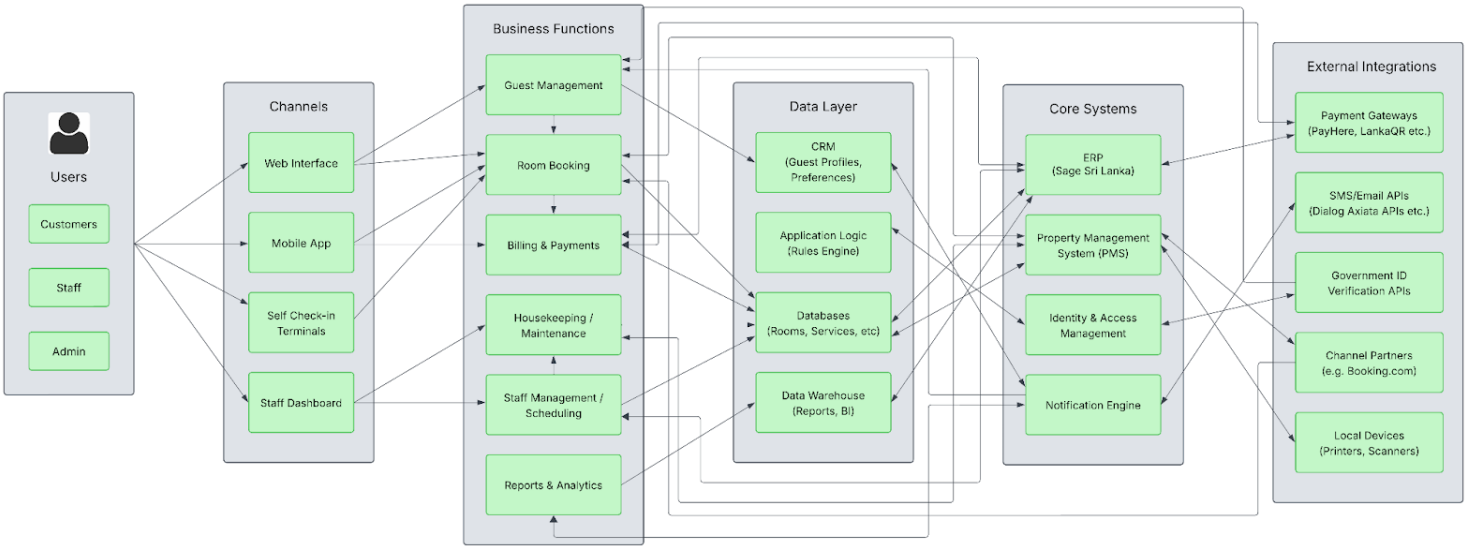
1.5 References

- IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, IEEE Computer Society, 1998
- Hotel Information Systems: A Comprehensive Guide by John Doe, 2022, ISBN: 978-1-23456-789-0
- Document detailing the hotel's strategic goals and long-term vision. (Lab description PDF)

2. Overall Description

2.1 Product Perspective

The Hotel Reservation and Guest Services Management System (HRGSMS) is a new product designed to replace the legacy desktop-based system currently used by SkyNest Hotels. The solution is aimed at modernizing the operations of the hotel to meet new hospitality standards and connect with external booking platforms and payment gateways in Sri Lanka. HRGSMS will become the centre for the management of hotel branches, processing guest reservations, and handling of chargeable guest services. This will be a self-contained system, which will be connected with other hotel systems, such as online booking portals, staff dashboards, and mobile applications. The HRGSMS is designed for easy scalability for upgrades to be made in the future, like loyalty program connection, dynamic pricing, and real-time third-party interoperability.

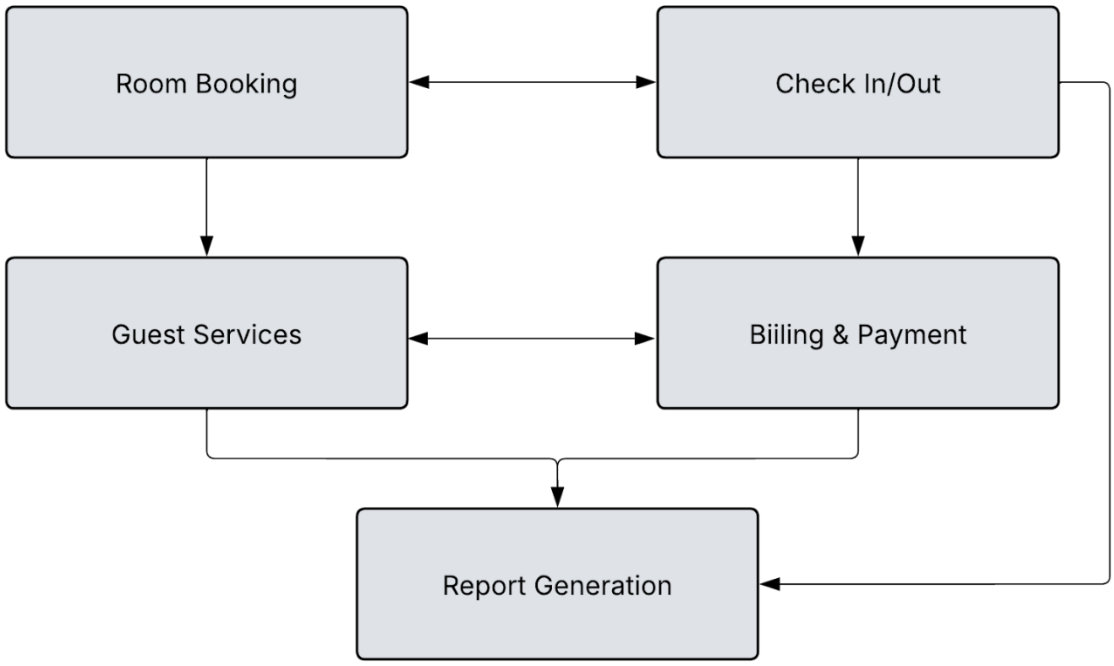


2.2 Product Functions

The major functions of the HRGSMS include:

- **Room Booking Management**: Manage room availability and reservations across multiple hotel branches. Prevent double-bookings, validate check-in/check-out dates, and synchronize booking schedules in real time.

- **Check-In/Check-Out Processing**: Update room status upon guest arrival and departure. Log guest check-ins and check-outs while maintaining correct occupancy records and connecting with the guest database.
- **Guest Services Management**: Track and manage guest-requested services such as room service, spa treatments, minibar, and laundry. Charges are recorded and automatically added to the guest's invoice.
- **Billing and Payment Handling**: Generate all in one invoices for each guest stay, including both room and service charges. Supports multi-mode payments, partial settlements, and flags any outstanding dues.
- **Reporting and Analytics**: Generate branch-wise reports on room occupancy, revenue, and guest service usage. Access to sensitive financial and operational data is restricted to authorized personnel, such as hotel managers.
- **User Management**: Enforce role-based access control for hotel staff and administrators. Ensure that users access only those modules and features relevant to their assigned roles.

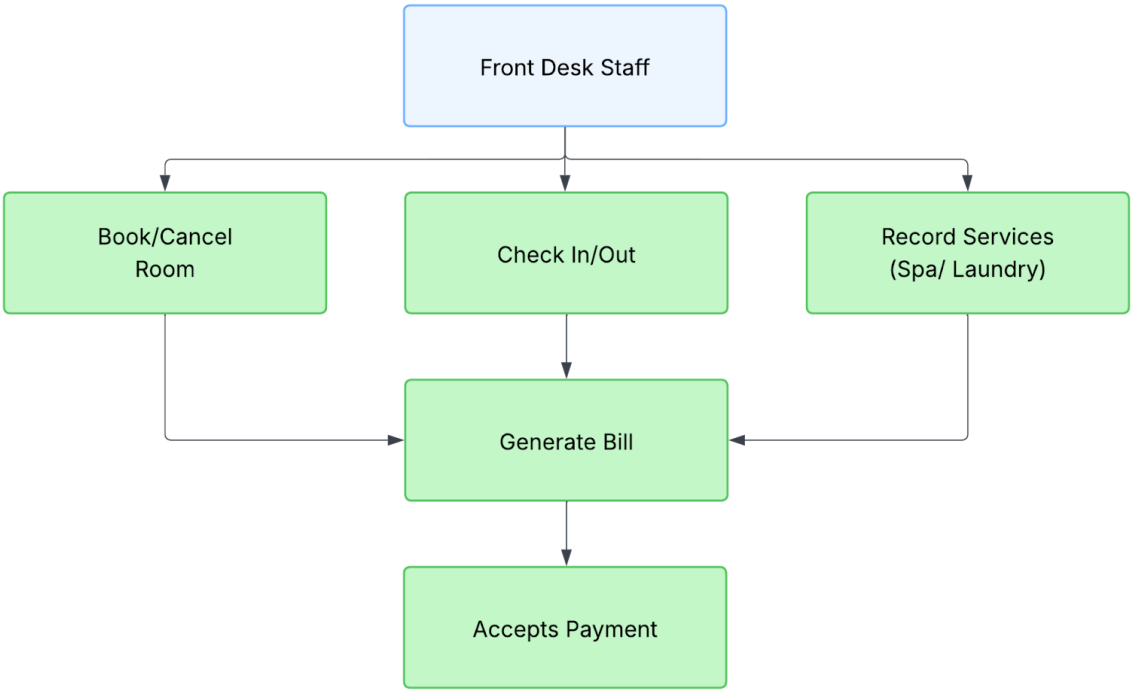


2.3 User Classes and Characteristics

The HRGSMS will be used in the following user classes:

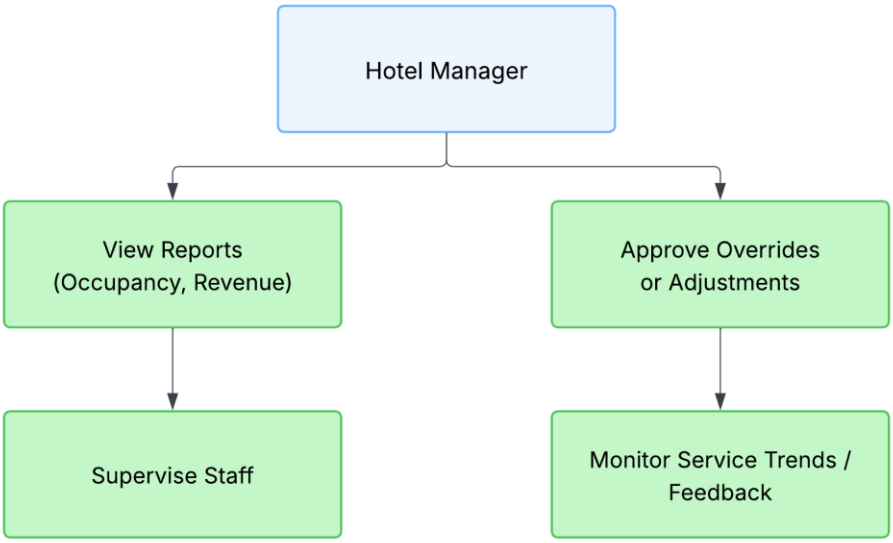
- **Front Desk Staff**:

  These individuals are responsible for the day-to-day guest-facing operations at each hotel branch. They are the primary users of the HRGSMS and are granted access to operational modules such as room booking, check-in/check-out, guest service tracking, and billing functions. They do not have access to system-level configurations or administrative controls. They are expected to have moderate technical proficiency and interact with the system frequently throughout their shift. A responsive and straightforward interface is required to support their fast-paced work environment.



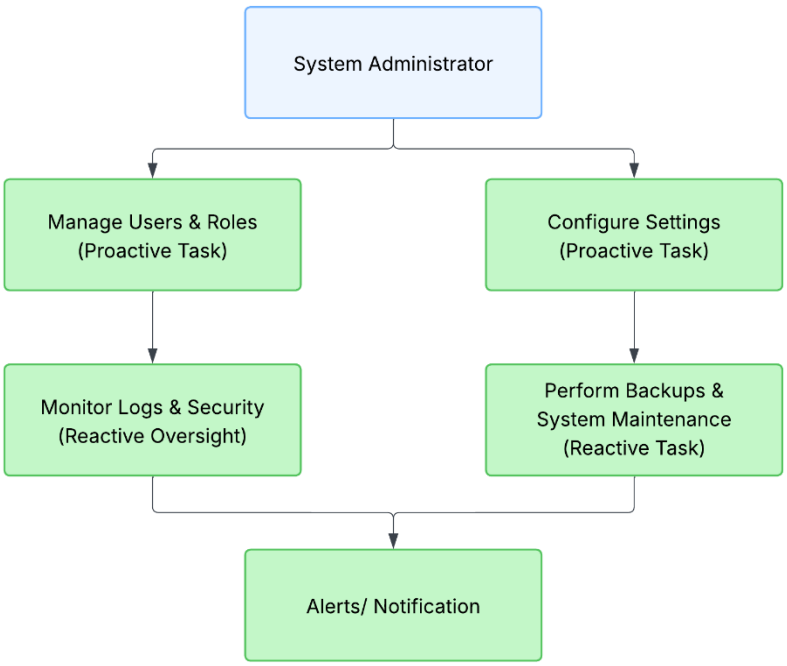
- **Hotel Managers:**

  These are responsible for overseeing operational activities at the branch level. They are provided with elevated access privileges to review business performance, supervise staff actions, and ensure compliance with hotel policies. Although their system usage is less frequent than that of front desk staff, their role requires access to reporting tools, approval workflows, and service trend monitoring features.



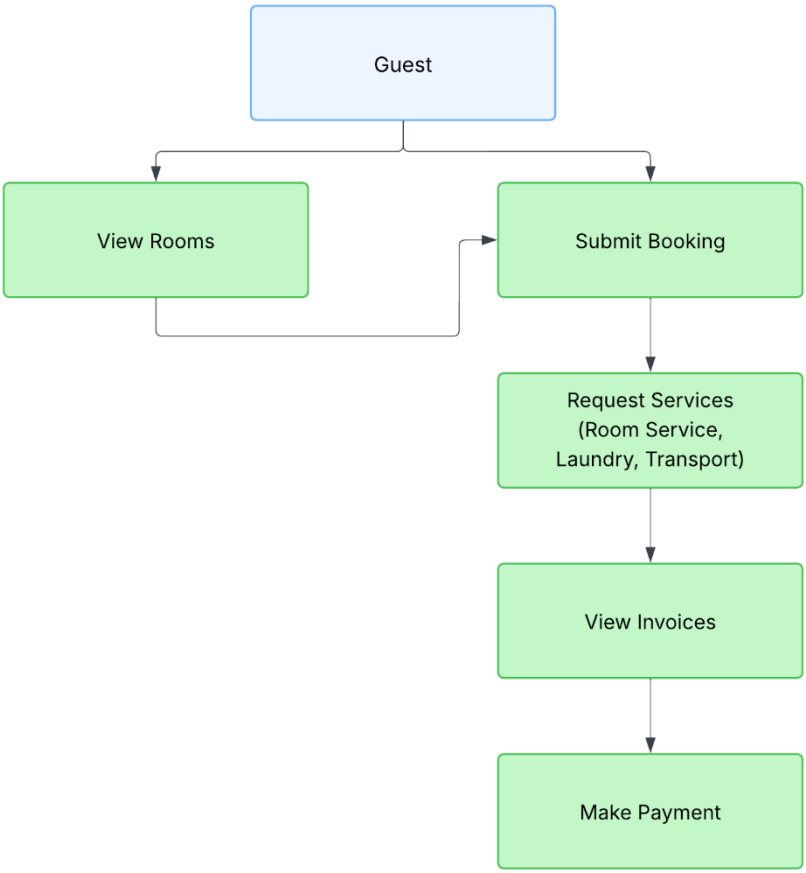
- **System Administrators:**

  These are responsible for the technical maintenance, configuration, and overall security of the HRGSMS. They are granted full system privileges to manage configurations, monitor system health, and enforce IT compliance standards. Though they access the system less frequently than operational users, their activities have a significant impact on system performance and integrity.



- **Guests:**

  These are the end-users of the HRGSMS who will interact with the system via a dedicated web portal or mobile application. Their system requirement is a simplified and user-friendly interface that requires minimal technical knowledge. Guest-facing functionality is part of the planned future scope and is intended to enhance service convenience and user engagement.



2.4 Operating Environment

The HRGSMS will operate within the following environment:

- **Hardware**: Centralized servers hosted in a secure infrastructure, accessible remotely by all hotel branches. Front desk and managerial staff will use standard desktop or laptop computers with optional peripherals such as receipt printers, barcode scanners, and cash drawers.
- **Operating System**: Linux-based server environments (e.g., Ubuntu or CentOS) for backend deployment. Client machines may operate on Windows 10 or later, macOS, or Linux distributions.

- **Database**: A relational database management system (RDBMS) such as PostgreSQL or MySQL, configured with backup and replication support. All transactions will adhere to ACID principles to ensure data integrity and consistency.
- **Network**: Secure, high-speed communication across hotel branches using centralized WAN or VPN connectivity. All data transmission will be encrypted using TLS 1.3 over HTTPS.
- **Other Software**: The system integrates with various third-party services, including.
    - Email and SMS APIs for notifications and booking confirmations.
    - payment gateways (e.g., PayHere, LankaQR)
    - Optional government ID verification services and booking channels (e.g., Booking.com)
- **Security**: Role-Based Access Control (RBAC) enforced session management, system activity logging, and audit trails are implemented to ensure secure operations.
- **Backup and Recovery**: Daily automated backups with off-site storage and log archiving policies to support data recovery and regulatory compliance.
- **Scalability**: The environment is designed for scalable deployment across new hotel branches and future integration with mobile apps and guest-facing portals.

2.5 Design and Implementation Constraints

The design of the Hotel Reservation and Guest Services Management System (HRGSMS) is subject to the following design constraints:

- **Database Technology:** The system uses a MySQL relational database to store all the hotel's data.
- **Avoid Double Bookings:** To make sure no room is booked twice, the system uses checks at both the database level and the application level.
- **Partial Payments:** Partial Payments will be added to the Database linked to the booking. The System will flag the booking with outstanding dues.
- **Security:** Only authenticated users like the Hotel Manager and Front Desk Staff will have access to make changes to the Database.
- **Performance Expectations:** System should work smoothly with ability to handle concurrent bookings across multiple branches.

- **For multiple hotels:** Since there may be several hotel branches, the system must handle all their activity at once while keeping the data consistent. This can be done using a shared or distributed database system.

2.6 User Documentation

The system has to be delivered with the following documentation for users:

- **User Manual:** A step-by-step guide for hotel staff on how to do bookings, check-ins, check-outs, logging services and taking payments.
- **Technical Documentation:** This includes the SQL database structure, ER diagrams and how to deploy the system for each Hotel branch.

All these documents will be available in standard terms so they're easy to understand.

2.7 Assumptions and Dependencies

Here are a few things we're assuming or depending on while building this system

- **Hotel Policies:** Things like taxes and charges for late check-outs are expected to be set and changeable by the admin.
- **User Proficiency:** Hotel staff members are expected to be trained to use the system and able to resolve minor issues that may occur.
- **Internet Connection:** Each hotel branch is assumed to have a stable internet connection with minimum latency (higher latency can be a major cause for double booking).
- **Third-party Services:** SMS and email notifications related to bookings and service availability and service usage can be integrated if needed. Which will be required to use the Network provider for those services.
- **Browser Support**: The Web application for booking and Hotel food menu is expected to be accessed via modern web browsers (Chrome, Firefox, Edge)

3. External Interface Requirements

3.1 User Interfaces

The Hotel Reservation and Guest Services Management System (HRGSMS) will demand a web-based user interface designed for two primary user groups to improve efficiency:

1. Hotel Staff (front desk, housekeeping, and management)
2. Customers/Guests

The staff interface will be used internally across the Colombo, Kandy, and Galle branches. while the customer interface will be publicly accessible via the hotel's website globally. The user interface will prioritize simplicity, responsiveness, and role-based access to ensure efficient operations for staff and a seamless booking experience for guests guarantee customers to give fully customized experience.

3.1.1 Staff User Interface

Standard GUI Elements:

- ***Buttons:***

  Save, Edit, Delete, Cancel, Search, and Add Service.

- ***Data Tables:***

  Used for managing bookings, rooms, and guest information.

- ***Modals:***

  Quick action pop-ups for tasks like check-in, adding a service, or authorizing payment gateways.

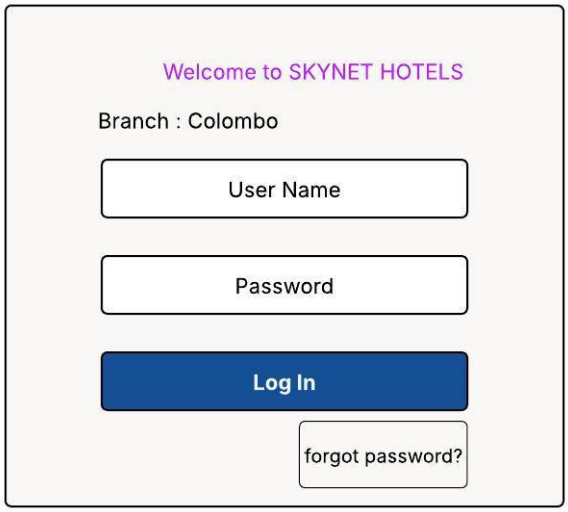
- ***Calendar Pickers:***

  For selecting dates easily within the interface.

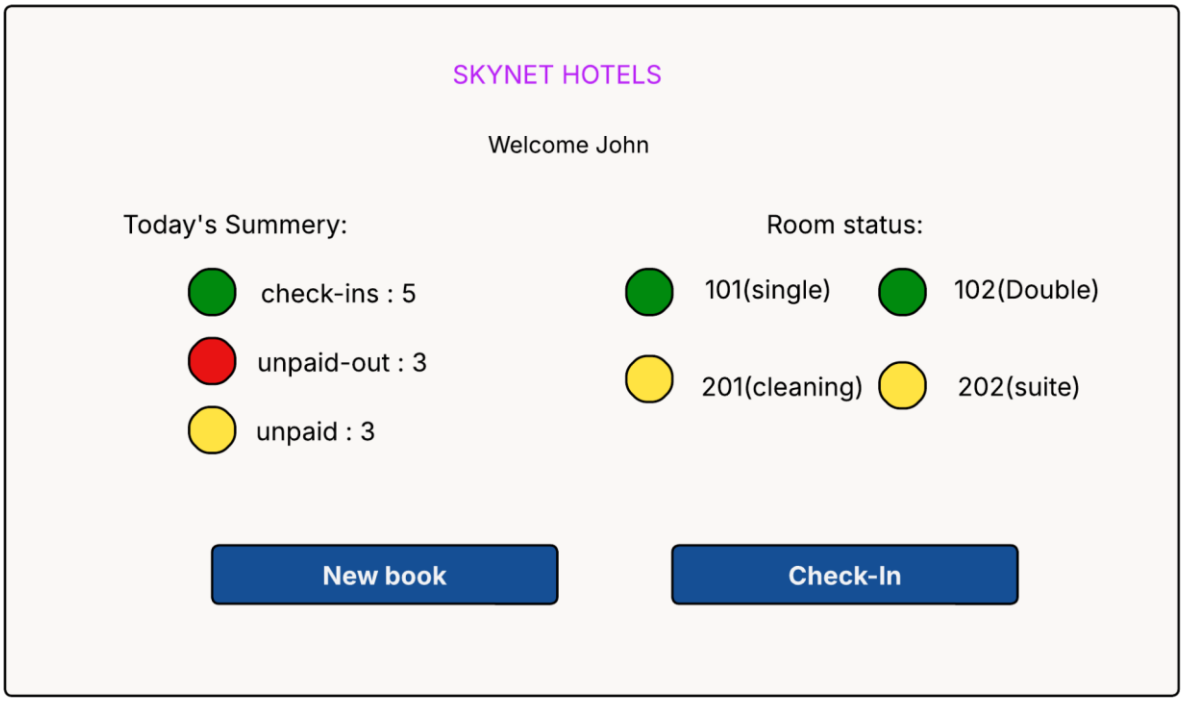
- ***Real-time Room Status Indicators:***
  - **Green:** Available
  - **Red:** Occupied

Sample Screens:

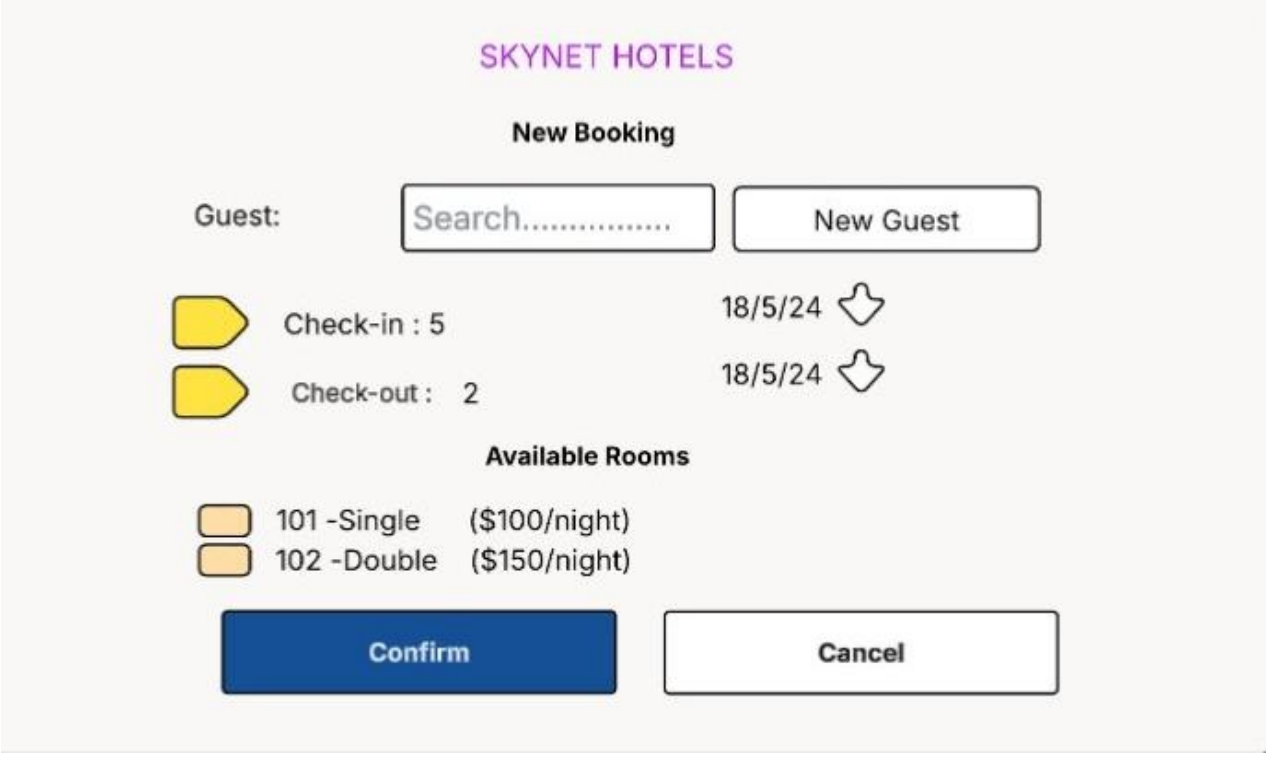
1. ***Login Page***



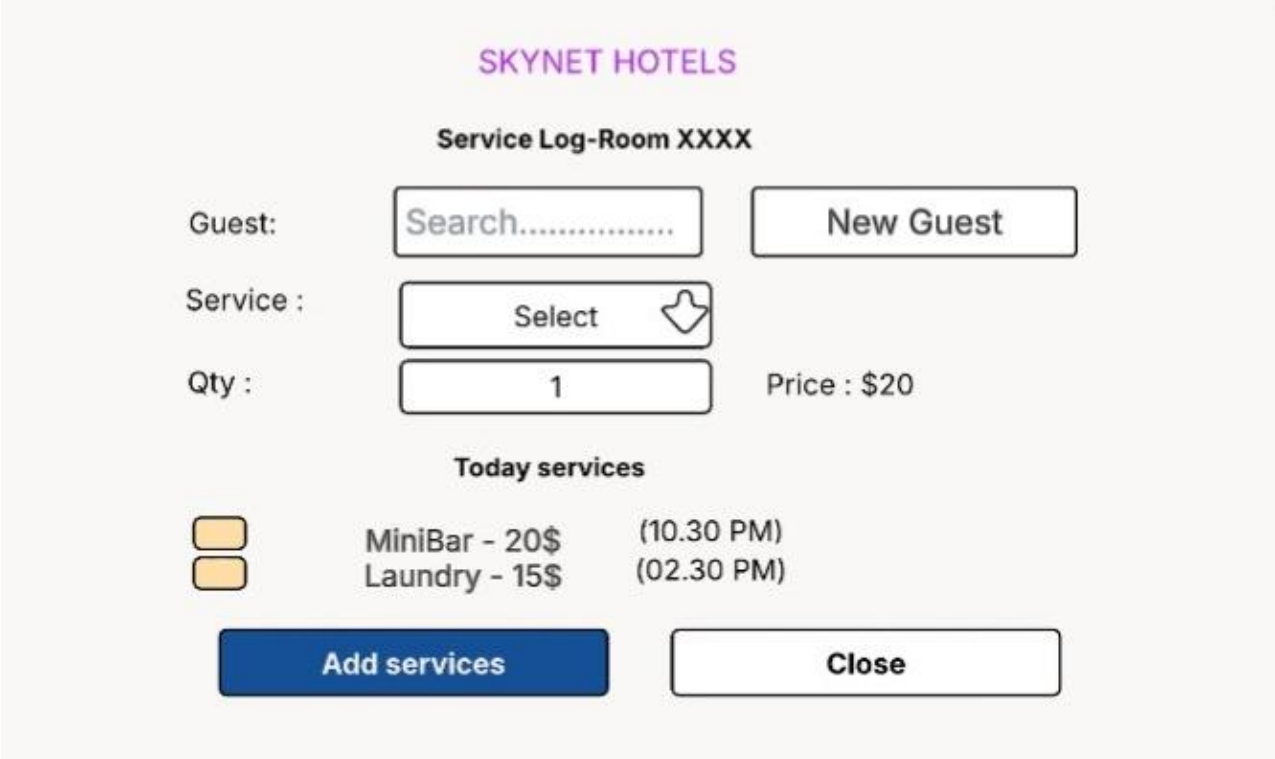
2. ***Dashboard***



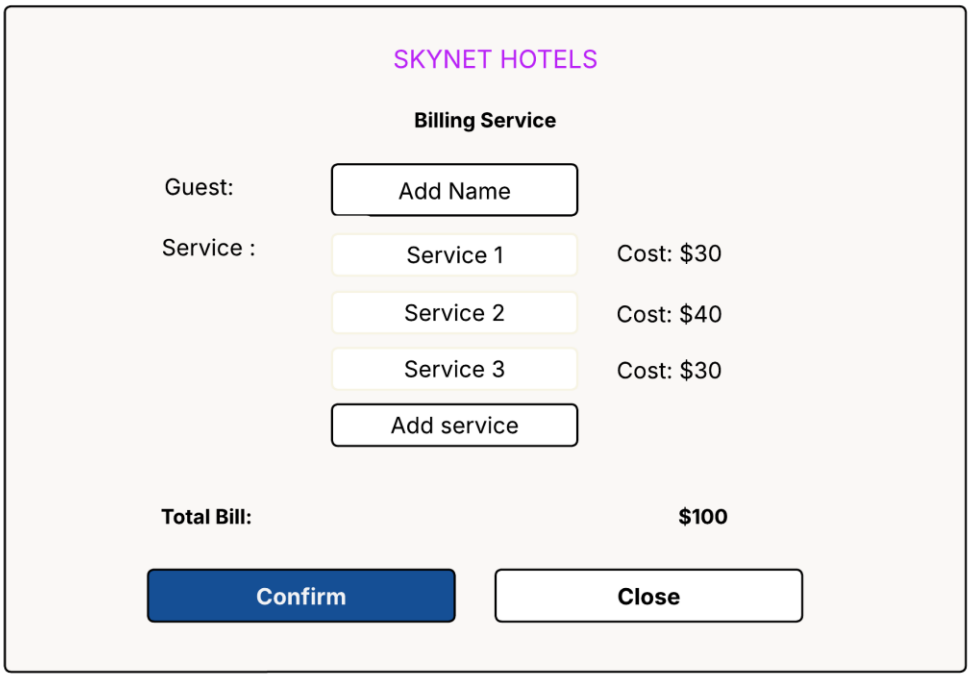
3. New Booking



4. Service Login



5. Billing Service



6. Report Dashboard

Keyboard Shortcuts

- Ctrl + S: Save current form.
- Ctrl + F: Focus on filter/search
- Alt + N: New booking

Error & Feedback

- Inline validation with red highlights for errors like invalid date.
- Toast notifications for success/failure actions.
- Confirmation modals for destructive actions e.g.: - delete, checkout.

3.1.2 Customer/Guest User Interface

Interface for Guest/Customer created with user-friendly manner without any trouble let customers to get online services.

Logical Characteristics:

- Public-facing portal accessible from the SkyNest Hotels website
- Designed with mobile-first principles to ensure usability on phones and tablets.
- Simple booking workflow that minimizes steps to complete reservation

Key Features:

- Browse available rooms by location, date, and type.
- View room details, amenities, and pricing.
- Register or book as a guest.
- Select check-in/check-out dates and optional services.
- Pay online (full or partial) using supported payment methods.
- Receive email confirmation and booking reference.
- View/modify booking (if account or booking reference is provided)
- Following elements implemented to improve efficiency when dealing with UI.

Standard Elements:

- Home page with branch selection and promotional banners
- Room list with filter (room type, capacity, rate)
- Booking form with calendar input and guest detail section
- Booking summary and secure payment screen
- Downloadable invoice or confirmation

UI Style Standards:

- Clean, modern UI (following Material Design or Bootstrap guidelines)
- All screens use a consistent color scheme, icon set, and font family.
- Accessible design (keyboard navigation, screen reader labels)

Error Handling:

- Clear, non-technical error messages and keep backup servers to provide uninterrupted service.
- Visual cues (red borders, tooltips) to guide corrections.

Shared UI Components:

- Navigation menu with logo and contact info.
- Footer with FAQs, branch details, and support links.
- Help button or chatbot (extendable function)

3.2 Hardware Interfaces

The HRGSMS is designed as a web-based SaaS application, minimizing local hardware dependencies. The hardware interfaces are primarily between user devices (staff and customers) and the centralized backend system, with additional support for peripheral devices at hotel branches backend also has a backup server to keep uninterrupted service.

Supported Devices

1. ***Client Devices:***
   Desktop or Laptop Computers: Used by staff for full system access and by customers for booking via web browsers (Chrome, Firefox, Safari, Edge). Compatible with Windows, macOS, and Linux.

2. ***Tablets:***
   Used by service staff (e.g., housekeeping) for mobile room status updates
   and by customers for booking on the go (iOS, Android).

3. ***Mobile Phones:***
   Used by customers for accessing the public booking interface (IOS, Android)

4. ***Peripheral Devices:***
   Receipt Printers: For printing invoices and bills at front desks (staff only).

Data and Control Interactions

***Input**:*

Staff and customers input data via keyboard, mouse, or touch interfaces (e.g., booking details, service requests, payment information).

***Output**:*

- Rendered via the web UI (HTML/CSS) on client device screens for both staff and customers.
- Printable PDF bills and reports are generated by the system and sent to receipt printers (staff only).

Hardware Requirements:

- Client Devices: Minimum 4GB RAM, 2GHz processor, 10GB storage for local caching (staff); any modern smartphone, tablet, or computer with a web browser (customers).
- Internet: Stable connection (minimum 5 Mbps) for cloud access.
- Receipt Printers: Compatible with standard USB or network printing protocols.

Protocols

- **Device Communication**:
  USB and Network Printing: For connecting receipt printers to staff devices, using standard drivers (e.g., ESC/POS for thermal printers).

- **USB/Bluetooth**:
  For optional barcode scanners to transmit scanned data to the system.

- **Server Communication**:
  HTTPS: For secure interaction between client devices (staff and customers) and the centralized backend server hosted on AWS.

3.3 Software Interfaces

The HRGSMS integrates with several software components to support both staff and customer functionality also we expect to improve following components with newly released technologies to improve system efficiency.

Software Components

1. ***Database:*** MySQL (v8.0 or higher)
   Stores all data related to rooms, bookings, services, payments, and reports, also keeping.

2. ***Data items:***
   Room details (ID, type, status), Booking details (ID, guest, dates, status), usage (ID, booking, service, price), Payments (ID, booking, amount, status).

3. ***Customer-specific data:***
   User accounts (email, hashed password) for managing bookings.

4. ***Backend Server:*** Node.js with

5. **Express.js:** Handles API requests for booking management, service tracking, report generation, and customer bookings.

6. ***Exposes RESTful endpoints:***

7. ***Staff:*** bookings, services, /reports.

8. ***Customers***: /public/rooms/availability, /public/bookings.

9. ***Frontend Framework:*** React (v18)
   Uses JSX for building reusable UI components for both staff and customer interfaces.
   Hosted via CDN (e.g., cdn.jsdelivr.net/npm/react).

10. ***Operating Systems:***
    Browser-based, compatible with Windows, macOS, Linux, iOS, and Android.

11. ***Third-Party Libraries:***

- Tailwind CSS: For responsive and consistent UI styling.
- Chart.js: For visualizing reports (e.g., service usage trends, revenue) for staff.
- Moment.js: For date handling bookings and reports.

12. Integrated Commercial Components:

- Payment Gateway: Stripe API (v3) for processing credit card and online payments (staff and customers).
- Incoming: Payment details (amount, card info).
- Outgoing: Payment confirmation or error messages.
- Email API: SendGrid for sending booking confirmations, billing summaries, and account verification emails.
- Outgoing: Emails with booking details, bills, or promotional offers.

13. Implementation Constraints:

- Data sharing uses JSON via REST APIs.
- Customer authentication uses JWT (JSON Web Tokens) for secure access to the "My Bookings" section.
- No global data areas: State management is handled via React's Context API or Redux for complex scenarios.

3.4 Communications Interfaces

The HRGSMS relies on secure and reliable communication protocols to support staff operations and customer interactions across branches also we guarantee to implement service to improve the safety for user.

Communication Functions

1. ***Web Browser:***
   Protocol: HTTPS for secure data transfer between the client (browser) and the backend server.
   Used for all UI interactions (staff: booking, service requests, reports; customers: room search, bookings).

2. ***Email Notifications:***
   Protocol: SMTP via SendGrid API.
   Message Formatting: HTML emails with booking details, bills, account verification, or promotional offers.

3. ***Payment Processing:***
   Protocol: HTTPS for Stripe API calls.

Ensures secure transmission of payment data (e.g., card details, amount) for both staff and customer payments.

4. ***Internal Branch Communication:***
   Protocol: HTTPS for REST API calls between branches and the central cloud server.
   Example: Kandy branch queries room availability in Colombo for a guest transfer or customer searches availability across all branches.

Communication Standards

- ***HTTP/HTTPS:*** For all web-based interactions and API calls.
- ***SMTP:*** For email notifications.
- ***REST:*** For structured data exchange between frontend and backend.

Security and Encryption

- ***SSL/TLS:*** Ensures all HTTPS communications are encrypted.
- ***Stripe API:*** Uses tokenization to secure payment data.
- ***JWT Authentication:*** Secures customer access to account-related features.
- ***Data Transfer Rates:*** Optimized for low latency (target < 200ms for API responses).
- ***Synchronization:*** Database transactions use ACID compliance (via PostgreSQL) to ensure consistent data across branches and prevent double-bookings.

System Interaction Diagram (Description)

The system follows a client-server architecture:

1. **Client (Browser):**
   - Staff: Access React-based UI via HTTPS for internal operations.
   - Customers: Access public React-based UI for room searches and bookings.

2. **Backend Server (Node.js/Express)**:
   Processes request, queries the PostgreSQL database, and integrates with Stripe/SendGrid APIs.

3. **Database (PostgreSQL)**:
   Stores all data with triggers for preventing double-bookings and updating room status.

4. **External Services**:
   - Stripe: Manages payment processing for staff and customer transactions.
   - SendGrid: Sends email confirmations for bookings and account actions.

5. **Flow Example (Customer Booking)**:

- Customer submits booking requests.

→ Frontend sends POST requests to /public/bookings.
→ Backend validates room availability.
→ Database saves booking and updates room status.
→ Stripe processes payment
→ SendGrid sends confirmation email.

List of Technologies/Third-Party Systems
(Given technologies can be changed to improve the system enhancement)

- ***Frontend***: React, CSS, java script, HTML.
- ***Backend***: Node.js, Next.js.
- ***Database***: MYSQL.
- ***Third-Party Services****:*
  - Stripe API (https://stripe.com/docs/api) for payments.
  - SendGrid API (https://sendgrid.com/docs) for emails.
- ***Hosting***: AWS (EC2 for backend, RDS for PostgreSQL, S3 for static assets).
- ***Communication Protocols***: HTTPS, SMTP.
- ***Browser Support***: Chrome, Firefox, Safari, Edge (latest versions).

4. System Features

4.1 Room Booking

4.1.1 Description and Priority

Guests can book rooms for a given check-in and check-out period. They can only book if there are available rooms

Priority: HIGH

The system must prevent double-booking of the same room.

4.1.2 Stimulus/Response Sequences

- **Stimulus:** Guest requests to book a room by specifying dates and room type

- **Response:** The System creates a booking if a room is available and the request is valid

4.1.3 Functional Requirements

- REQ-1: The system shall allow a guest to create a booking by providing check-in/check-out dates, room type, and guest details.
- REQ-2: The system shall validate that the selected room is not already booked during overlapping dates.
- REQ-3: The system shall prevent booking a room that is already occupied for the selected dates.

4.2 Guest Check-in

4.2.1 Description and Priority

At the time of check-in, the system must be updated to change the room status to Occupied.

Priority: HIGH

The system must not check into a room that is not prepared or unavailable.

4.2.2 Stimulus/Response Sequences

- **Stimulus:** The receptionist checks in a guest with a valid booking.
- **Response:** System marks the room as checked in.

4.2.3 Functional Requirements

- REQ-4: The system shall allow staff to check in guests for active bookings.
- REQ-5: The system shall update room status to "Occupied" upon check-in.

4.3 Guest Check-out

4.3.1 Description and Priority

Once the guest checks out, the room must be updated to 'Available.'

Priority: HIGH

The system must not initialize checkouts while the guests are in.

4.3.2 Stimulus/Response Sequences

- **Stimulus:** Guest initiates check-out
- **Response:** The system marks the room as checked out.

4.3.3 Functional Requirements

- REQ-6: The system shall prevent marking the booking as "Checked-Out" unless all dues are cleared.

4.4 Service Management—Room service, Spa, Laundry, Minibar usage

4.4.1 Description and Priority

Guests can request chargeable services, and those will be added to the final bill.

Priority: MEDIUM

Services are not a functional requirement of the hotel.

4.4.2 Stimulus/Response Sequences

- **Stimulus:** Guest requests a service
- **Response:** System logs the use of the service and links it to the booking

4.4.3 Functional Requirements

- REQ-7: The system shall allow staff to add service usage to a booking with service name, date, quantity, and price.
- REQ-8: The system shall link each service record to both the booking and the service catalog.

4.5 Partial Payments

4.5.1 Description and Priority

Guests can pay half of the full payment beforehand.

Priority: HIGH

Failure to add the data to the database can cause major issues with customers.

4.5.2 Stimulus/Response Sequences

- **Stimulus:** Guests pay a portion of the bill
- **Response:** System flags bookings with outstanding dues

4.5.3 Functional Requirements

- REQ-9: The system shall allow partial payments and store the amount, date, and payment method.
- REQ-10: The system shall flag bookings with unpaid balances after the check-out date.

4.6 Billing and Payments

4.6.1 Description and Priority

At checkout, the system calculates the final bill.

Priority: HIGH

The hotel must accurately create the final bill; otherwise, it'll cause dissatisfaction among customers.

4.6.2 Stimulus/Response Sequences

- **Stimulus:** System adds partial payments for the booking; the staff member confirms the check-out after a room inspection
- **Response:** The System calculates the total bill as Room charges + Service charges – Partial payments

4.6.3 Functional Requirements

- REQ-11: The system shall calculate the total bill as room charges + service charges.
- REQ-12: The system shall allow multiple partial payments to be recorded against a booking.

4.7 Reporting (Occupancy, Revenue, Guest Summaries, etc.)

4.7.1 Description and Priority

For the management, reports about the performance of the hotel are essential.

Priority: MEDIUM

Failing to make an accurately reflecting report only reduces the further improvement of the hotel.

4.7.2 Stimulus/Response Sequences

- **Stimulus:** Admin requests a report
- **Response:** The System generates a report from the current database

4.7.3 Functional Requirements

- REQ-13: The system shall generate a room occupancy report for a selected period.
- REQ-14: The system shall generate a billing summary report showing unpaid balances.
- REQ-15: The system shall generate a monthly revenue report for each hotel branch.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Speed

- Search room within 2 seconds.
- Calculate bill within 1 second.
- Provide the reports within 5 seconds.
- Load the UI in under 3 seconds.

5.1.2 Capacity

- Capable of handling 100+ users at once without lagging.
- Could process up to 50 bookings per minute.

5.1.3 Scalability

- New hotel branches and services can be added easily.
- System performance must hold steady up to 10000 bookings per branch.

5.1.4 Efficiency

- Optimize SQL using indexing for faster searches.
- Even during peak hours maintain CPU usage under 70% and memory space under 80%.

5.1.5 Availability and Backups

- Daily availability updates done in 30 minutes.
- 99.9% uptime required (system, server or service should be continuously available and run without any interruptions).
- Monthly revenue reports under 1 hour.
- Full backups done within 4 hours and crash recovery done within 2 hours.

5.2 Safety Requirements

5.2.1 Data Integrity

- Validate all the financial and booking transactions.
- Prevent check-out if there are any billing issues.

5.2.2 Safe operations

- Confirm the actions again before processing like payment or booking changes.
- Bookings which are overdue are flagged for staff review.

5.2.3 Common

- Auto logout after 10 minutes.
- Have a history of all login/ logout activities.
- Re-Authenticate when switching roles. (admin, receptionist, accountant)

5.3 Security Requirements

5.3.1 Data protection

- Encrypt all the sensitive information.
- Mask the details. (Show only the last 4 digits of card numbers.)
- Use TLS 1.3(Transport Layer Security version 1.3, the latest version) for secure transmission.

5.3.2 User access

- Role based access (admin, receptionist, accountant).
- Strong passwords (8+ characters with symbols, capitals and numbers)

5.3.3 Audit trails

- Keep the history of critical actions for nearly 2 years.
- Alert for suspicious activities.

5.3.4 Vulnerability handling

- Security scans every month.
- Protect against common attacks (SQL injection, XSS, etc)

5.4 Software Quality Attributes

5.4.1 Maintainability

- Clean modular code and an admin panel allow updates without affecting other parts.

5.4.2 Portability

- Compatible with both desktop and tablet hotel systems.

5.4.3 Testability

- Clear separation of logic makes unit testing of billing, login, and service modules easy.

5.4.4 Reliability

- Minimal downtime, quick recovery, and precise transaction handling.

5.4.5 Usability

- Easy to use for staff with basic technical skills. (Simple UI)
- Core tasks like booking, billing etc are done in less than 5 steps.
- Help icons are available on all the screens.
- Supports all three languages. English, Sinhala, and Tamil.
- Built to use in both mobile phones and laps.

5.5 Business Rules

5.5.1 Booking and Check-In

- Guests must pay an advance before Check-In.
- Rooms cannot be booked twice for overlapping dates.
- Status of the rooms automatically updates for every Check-In and Check-Out.
- Bookings, which are overdue for more than 24 hours, are flagged for staff to take further action.

5.5.2 Service and Billing

- All the billing details of the services taken by the guest are linked to their booking.
- Final bill which is the summation of room charge, and service charges are calculated at the check-out.
- Even though partial payments are allowed, balance must be cleared before leaving.
- Apply discounts, taxes or late fees as per the policy of HRGSMS.

5.5.3 Payments

- Track every payment with mode of payment, amount and date.
- Outstanding amounts should be shown clearly with bigger font size.
- Payment history should be viewable by the staff or admin.

5.5.4 Access control

- Only authorized staff could create, cancel, or update bookings.
- Admins can manage services, room types, pricing, and reports.
- Receptionists manage check-ins, check-outs, and service records.
- Accountants can generate bills and process payments.

6. Other Requirements

1. Legal & Compliance Requirements

- The system shall adhere to Sri Lankan laws pertaining to data privacy concerning the storage and processing of guest information.
- Payment transactions shall be processed according to the security requirements of PCI-DSS.
- An audit log for all financial transactions shall be kept for 5 years

2. Data Management Requirements

- Booking records shall be kept for 5 years for reporting and auditing.
- Daily automated backups of databases shall be performed with a retention period of 30 days.
- Personal data of guests shall be anonymized after 5 years, in compliance with Sri Lankan privacy laws and auditing requirements.

3. Localization & Accessibility

- The system shall have user interfaces in English, Tamil, and Sinhala.
- The user interfaces must meet the WCAG 2.1 AA requirements.
- All error messages must be provided in text and audio formats.

4. Disaster Recovery

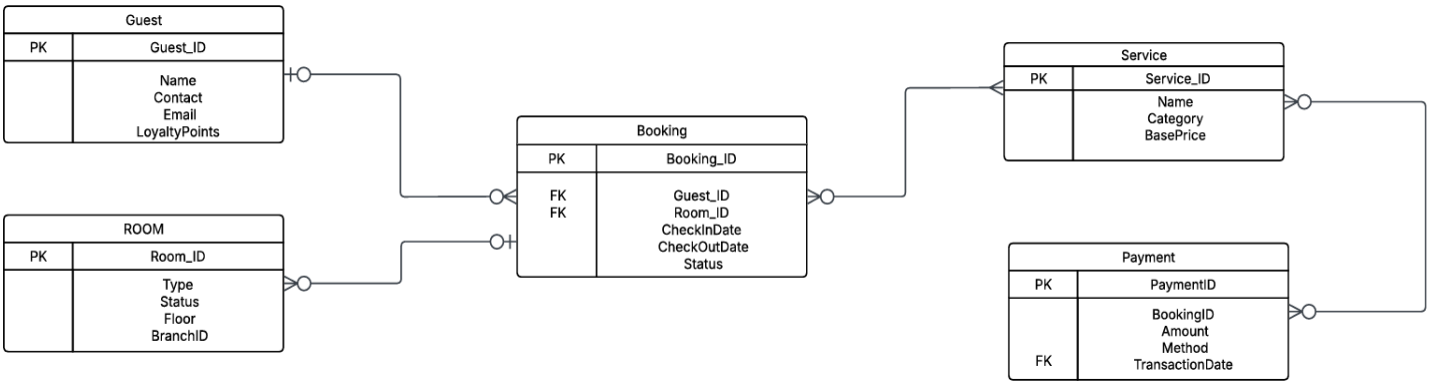
- The system must be capable of restoring backup within 4 hours after a failure.
- Critical functions (check-in/out) must be available during network outages with limited functionality.

Appendix A: Glossary

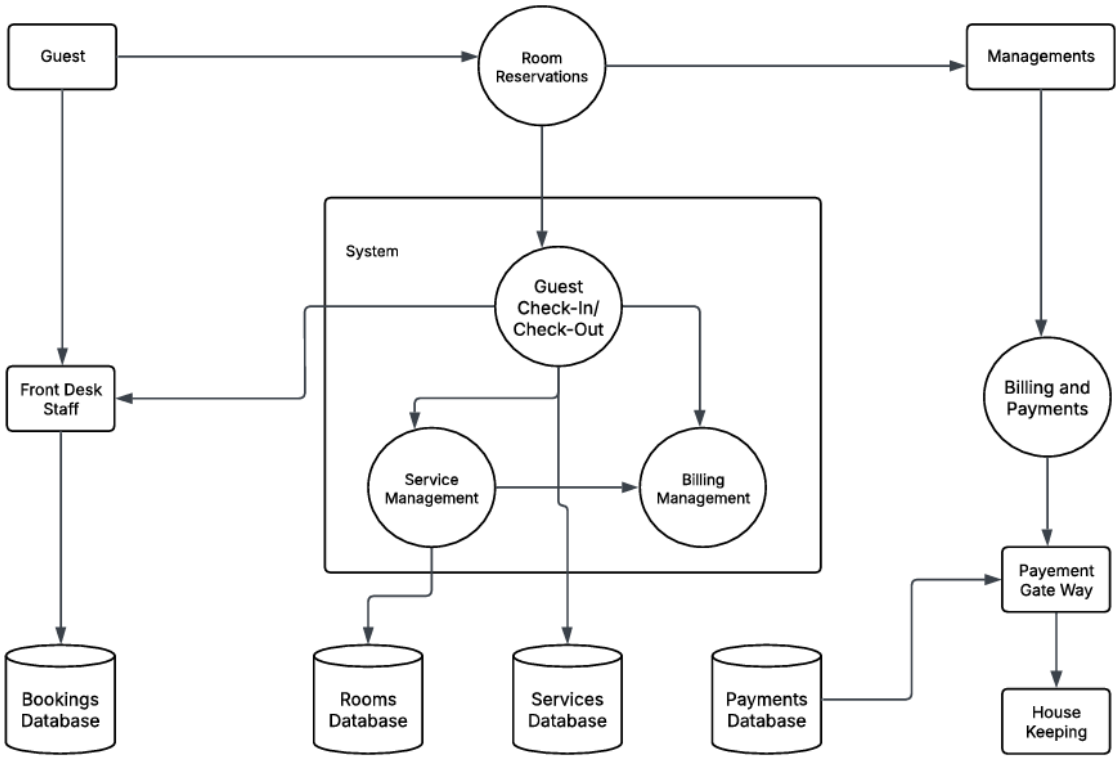
Term	Definition
Booking Status	Current status of reservation (Confirmed, Checked-In, checked-out, or Cancelled)
Double Booking	Error condition where more than one guest is booked into the same room for overlapping dates
Occupancy Rate	Percentage of available rooms that are occupied over a period of time
Partial Payment	Payment made against a booking that doesn't represent the full amount owing
Service Catalog	Master list of all chargeable services and current prices
Upsell Opportunity	System-suggested recommendation for ancillary services based on guest preferences

Appendix B: Analysis Models

Analysis Models



Data Flow Diagram



Appendix C: To Be Determined List

ID	Open Question	Impact Area	Owner
TBD-1	Should the system be interfaced with government tax systems for automatic VAT reporting?	Compliance	CFO
TBD-2	What are the detailed data retention requirements for different types of guest data?	Legal	DPO
TBD-3	Will the system need to offer dynamic pricing based on demand?	Revenue	Revenue Manager
TBD-4	Should housekeeping staff use mobile devices for real-time room status updates?	Operations	IT Director
TBD-5	What are the features of the loyalty program to be launched in Phase 1?	Marketing	CMO