A

Project Proposal

On

HotelHQ - Hotel Booking System

For the partial fulfillment of the requirements for the degree of Bachelor of Computer Application under Pokhara University

**Submitted to**

Department of Computer Application

National Academy of Science and Technology

**Under the Supervision of**

Mr. Sunil Bahadur Bist

Lecturer, Department of Computer Application

**Submitted By**

Gaurav Kathayat (23530007)

Motiram Mahato (23530015)

Pradip Panthi (235300037)

Bachelor of Computer Application, 4th Semester

April, 2025



National Academy of Science and Technology

(Affiliated to Pokhara University)

Accredited by University Grants Commission (UGC), Nepal (2022)

Uttarbehedi 4 Dhangadhi, Kailali, Nepal

# Abstract

This project aims to develop a web-based Hotel Booking System that manages room and table reservations efficiently and accurately. The system provides real-time availability, secure user authentication with role-based access, and an intuitive interface for customers and hotel staff. By digitizing the booking process, the system minimizes errors, prevents double bookings, and enhances customer satisfaction. The system is designed with scalability and user experience in mind, using modern web technologies.

***Keywords: Hotel Booking, Hotel Reservation, Hotel Booking System***

## Introduction

Efficient management of room and table bookings is critical in the hospitality industry to ensure smooth operations and high customer satisfaction. Many hotels and restaurants still rely on manual or semi-manual booking systems, which often lead to overbooking, miscommunication, and poor user experience. This project proposes a comprehensive web-based platform dedicated exclusively to managing hotel room and restaurant table bookings in real time, improving operational efficiency and customer convenience. The main functionality provided are room booking, room status, room category name and room prices, bill creation handling, payment handling,

## Problem Specification

The main problems that the current systems or manual methods in the hospitality industry face include:

* Lack of real-time updates leading to double bookings and conflicts.
* Limited access control causing data privacy issues.
* Manual processes prone to human errors and delays.
* No centralized system to manage and track all bookings effectively.

These issues result in lost revenue, decreased customer loyalty, and operational inefficiencies.

## Objective

The objectives of this project are:

1. Provides a login and sign up for customer and admin (Manager or receptionist).
2. Making Hotel Room Booking process quick and easy.
3. Providing real-time access to room/table availability and booking history.
4. Providing an appealing and easy to use booking platform to the customers.

## Related Work Review

This section reviews existing hotel and restaurant management systems to identify strengths and weaknesses relevant to the core function of booking management. While many current platforms offer comprehensive solutions encompassing billing, inventory, and housekeeping, they often lack simplicity and affordability for small to medium-sized businesses. Our proposed HotelHQ system addresses this gap by focusing exclusively on providing a streamlined, real-time booking system for rooms and restaurant tables. Following 2 cases were studied for relevant work review and feasibility study:

**1. Hotel Devotee**

This hotel situated in Dhangadhi uses an email and excel based booking system which doesn’t provide an online booking option to the customers.

**Strengths Related to Booking:**

* Less prone to loadshedding
* Saves cost of developing a booking system

**Limitations for Booking-Focused Solutions:**

* Missing out on possible online customer traffic and marketing.
* Email booking adds an unpleasing appeal to the online customers.
* Issues of payment sync in excel sheets and email.

**2. Rubus Hotel**

Rubus Hotel also uses excel sheets, phone call, and email for booking management.

**Strengths Related to Booking:**

* Easy for non-technical staff to get used to.
* Cheaper and cost saving option.

**Limitations for Booking-Focused Solutions:**

* Potential of double-booking issue due to the emails and the excel sheets not being in sync.
* Not being able to handle high number of bookings.
* More prone to Human error.
* Email booking is a hassle for the customer.

## System Requirements

The HotelHQ system requires both functional and non-functional requirements to ensure reliability, scalability, and user satisfaction. The system is designed to be simple for end-users while being robust enough for administrative and managerial operations.

**Functional Requirements**

* User registration and secure login with role-based access.
* Room booking module: check availability, book, modify, cancel.
* Table booking module: check availability, book, modify, cancel.
* Booking status and history tracking.
* Administrative access to manage bookings and view reports.
* Bill generation module: Generate bills for bookings.
* Payment handling module: Track and update payments for bills.

**Non-Functional Requirements**

* Secure authentication and session management.
* Responsive frontend accessible from desktop and mobile devices.
* Fast and reliable database operations.
* Clear separation of backend and frontend logic.

**Requirement Prioritization Table**

This table helps identify the core (high-priority) functions required for the HotelHQ Hotel and Restaurant Inventory Sales Management System and categorizes other useful enhancements based on development feasibility.

**Table: Requirement Prioritization Table**

|  |  |  |
| --- | --- | --- |
| **Priority** | **Requirement** | **Type** |
| High | Role-based Login (Customer, Staff, Manager) | Functional |
| High | Room Booking System | Functional |
| High | Booking Management (View, Modify, Cancel) | Functional |
| High | Centralized Booking Records | Functional |
| High | Bill Generation Module | Functional |
| High | Payment Handling | Functional |
| Medium | Password Hashing and Secure Login Sessions | Non-functional |
| Medium | Responsive UI for Multiple Devices | Non-functional |
| Medium | Error Handling and User Feedback | Non-functional |
| Low | UI Animations and Aesthetic Enhancements | Non-functional |
| Low | Basic Booking Reports for Managers | Functional (Optional) |

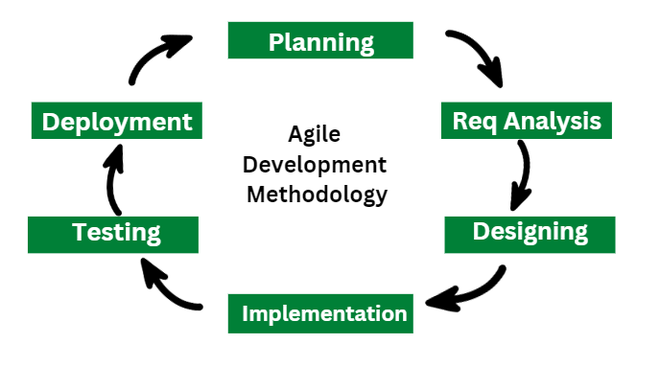
## Methodology and System Design

**Methodology**

This project adopts the Agile development methodology, promoting iterative and incremental delivery of system features. Agile enables continuous integration, regular testing, and frequent stakeholder feedback, ensuring that the system evolves effectively to meet user requirements. Each sprint will focus on delivering specific, testable functionalities such as user authentication, room booking, and table booking modules, allowing flexibility and early detection of issues.

Key advantages of our Agile approach include:

* Incremental delivery of features
* Clear role distribution and team collaboration
* Regular sprint planning and reviews
* Transparent tracking and documentation
* Reduced development risk through early testing and feedback.



**Figure 6.1: Agile Model**

**System Design**

The design of the Hotel Booking System is user-friendly and role-based, ensuring appropriate access privileges depending on the user type (e.g., customer, staff, manager).

**Frontend:**

* Developed using HTML, CSS, and JavaScript to create a responsive and intuitive interface.
* Designed for ease of use across devices including desktops and mobiles.
* Provides separate interfaces tailored to different user roles:
* Customers: Booking portal to search and reserve rooms and tables.
* Staff: Dashboard to manage and update bookings.
* Managers: Overview interface for booking reports and system monitoring.

**Backend:**

* Built with Python Flask framework to handle all server-side logic.
* Manages user authentication and session control with role-based routing to restrict access.
* Processes booking requests for rooms and tables, ensuring real-time availability and avoiding conflicts.
* Handles booking management operations, such as viewing, modifying, and canceling reservations.

**Database:**

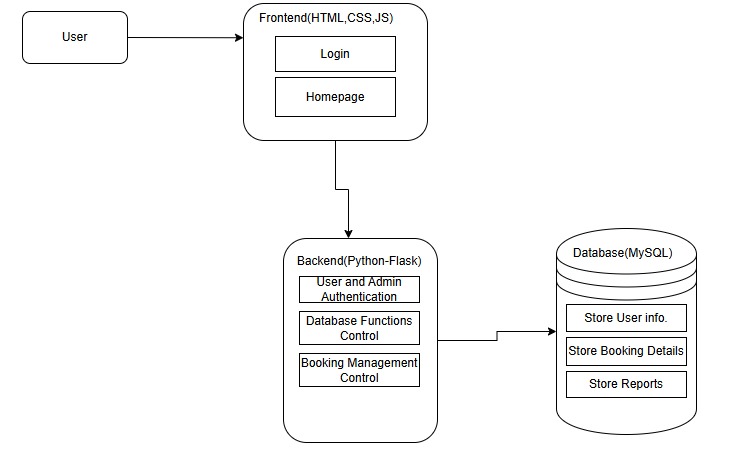
The system uses MySQL to securely store and manage all necessary data related to the hotel booking process, including:

* User credentials and roles: Information for authentication and role-based access control (e.g., customer, staff, manager).
* Room details: Information about available rooms, linked to a separate Room Category entity that defines room type, description, and pricing.
* Table details: Information about restaurant tables, seating capacity, and availability status.
* Booking details: Records of room and table bookings including customer information, booking dates and times, status, and any modifications or cancellations.
* Bill: Stores billing details including total amount, date, and association with specific bookings.
* Payment: Captures payment status, method, and links to the corresponding bill or booking.
* **Room Category**: Stores room types (standard, deluxe), prices, and features, which each Room references.

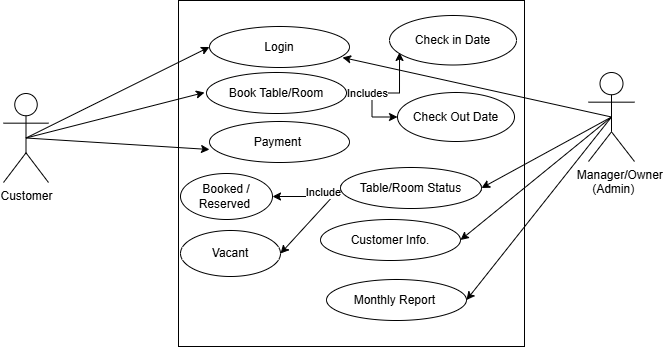
**System Architecture:**

The high-level system architecture of Hotel Booking System consists of:

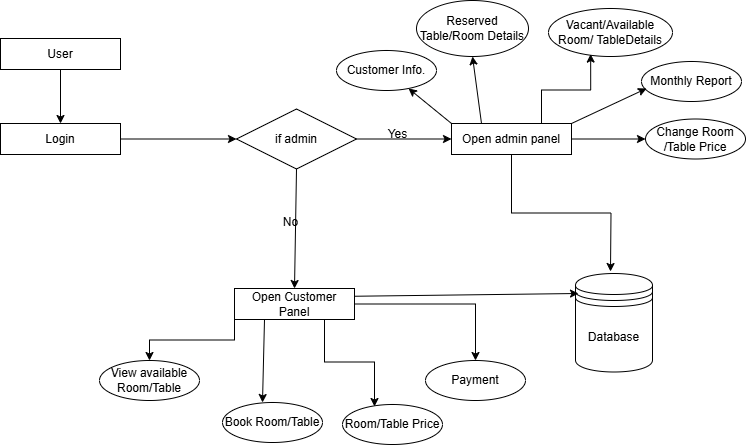
* Frontend (Client-side): A responsive UI built with HTML, CSS, and JavaScript, using frameworks like Tailwind CSS to speed up development and improve design. Interfaces are tailored for customers, staff, and managers for smooth booking and management across devices.
* Backend (Server-side): Implemented using the Flask framework in Python, responsible for handling API requests, business logic, session management, and enforcing role-based routing to secure access to system functionalities.
* Database Layer: Utilizes MySQL to efficiently manage relational data including users, rooms, tables, and bookings, ensuring integrity and fast query performance.
* Role-Based Access Control: Ensures data privacy and restricts functionality based on user roles:
* Customers can create and manage their bookings.
* Staff assist with booking management.
* Managers access all bookings and reporting features.
* Optional Future Enhancements: May integrate analytics or reporting modules as needed, focusing initially on delivering a robust booking platform, and may include integration with online payment gateways to automate transaction processing and reduce manual work.



**Figure 6.1.2: System Architecture**

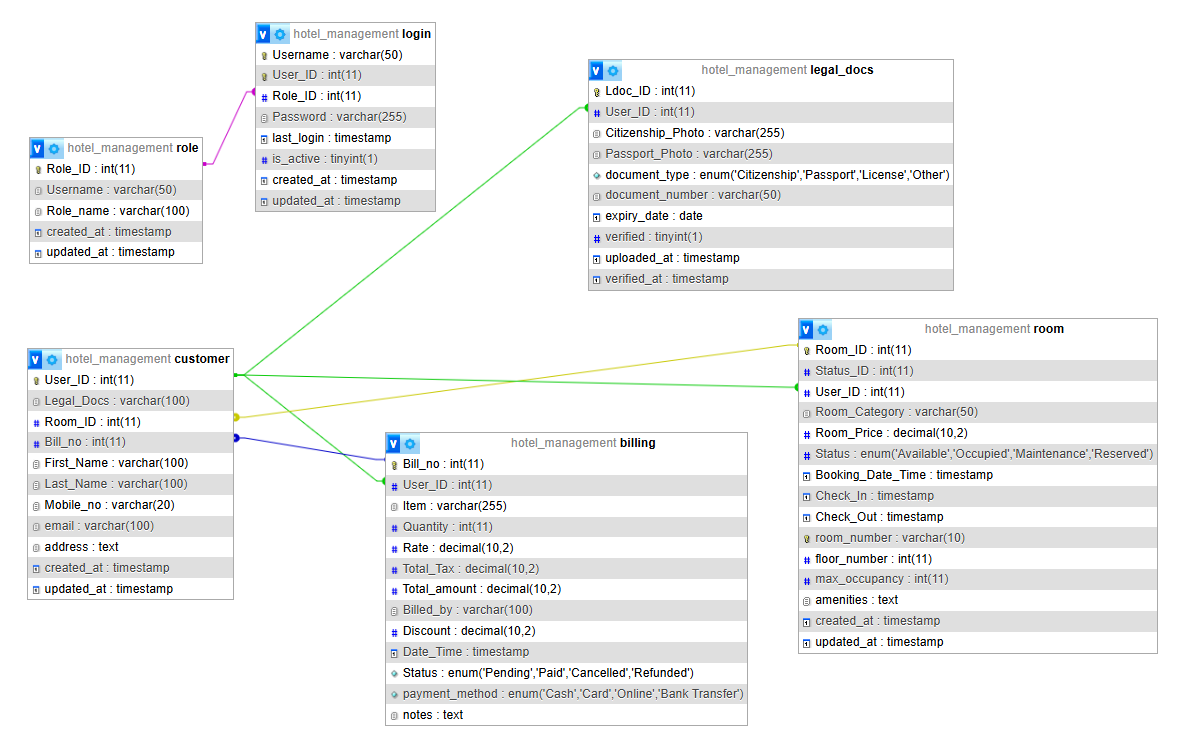
**Use Case Diagram**

**Figure 6.1.3: Use-case Diagram**

**Basic Work Flow Diagram**

**Figure 6.1.5: Basic Workflow Diagram**

**ER Diagram**

****

**Figure 6.2.6: ER Diagram (database schema)**

## Project Schedule

The project schedule follows the system development life cycle, spanning approximately two months. Key phases include requirement analysis, proposal defense, feasibility study, system design, development, testing, documentation, and final presentation. Each phase has a designated timeframe to ensure timely completion. An Agile approach is adopted, allowing overlapping tasks, iteration, and flexibility throughout development.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Activities | May | May | May | May | June | July | July | July |
| 01-09 | 10-13 | 14-20 | 21-30 | 01-30 | 01-15 | 16-22 | 23-30 |
| Project  analysis |  |  |  |  |  |  |  |  |
| Proposal  Defense |  |  |  |  |  |  |  |  |
| Feasibility  study |  |  |  |  |  |  |  |  |
| Designing |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |
| Final  testing |  |  |  |  |  |  |  |  |
| Documentation |  |  |  |  |  |  |  |  |
| Final  Submission |  |  |  |  |  |  |  |  |

## Development Cost Estimation

This section estimates the total resources required, including time, manpower, infrastructure, and miscellaneous expenses to complete the Hotel Booking System. A bottom-up approach is used, where costs for individual roles are calculated and then aggregated.

**Effort and Timeline**

Total Duration: 10 weeks

Team Members: 3

Weekly Working Hours per Member: 36

Total Estimated Effort: 36 hours × 10 weeks × 3 members = 1080 hours

|  |  |  |
| --- | --- | --- |
| **Role** | **Duration** | **Estimated Cost (Rs.)** |
| Project Manager | 2.4 months | 35000 |
| Frontend Developer | 2.4 months | 35000 |
| Backend Developer | 2.4 months | 40000 |
| QA Engineer *(shared role)* | 2.4 months | 28,000 *(approx.)* |
| UI/UX Designer *(shared role)* | 2.4 months | 23,000 *(approx.)* |
| Total |  | 1,61,000 |

**Table: Personnel Cost Estimation**

**Infrastructure and Miscellaneous Costs**

Includes server hosting, electricity, internet, domain names, version control services, etc. Estimated Total: Rs. 10,000

**Grand Total Development Cost**

Rs. 1,71,000

## Deliverables

The deliverables of this project encompass all tangible and intangible outputs produced during the entire development process, demonstrating the successful realization of the Hotel Booking System. These deliverables serve as evidence of project progress, completion, and readiness for deployment and use.

**List of Project Deliverables:**

## Complete project report and documentation detailing design, development, and testing.

## A fully functional web-based Hotel Booking System for managing rooms and tables.

## Organized source code for frontend, backend, and database components.

## Presentation slides for project defense and evaluation.

## Deployment instructions to guide system installation.

## Optional user and admin manuals to support system use.

## Conclusion

This Hotel Booking System offers a scalable and efficient solution for managing room and table reservations. By incorporating role-based access, real-time availability, and secure login, it enhances operational efficiency and improves customer experience. Automating the booking process reduces manual errors and streamlines hotel operations. Future enhancements could include expanded reporting and integration with payment systems, positioning the system as a reliable tool for hospitality management. The addition of bill and payment modules further enhances the system’s commercial viability and makes it suitable for real-world deployment.

## Refrences

[1] “System Design - System Analysis and Design,” *TutorialsPoint*. Available: <https://www.tutorialspoint.com/system_analysis_and_design/system_design.htm>. [Accessed: May 8, 2025].

[2] “Welcome to Flask — Flask Documentation (2.0.x),” *Pallets Projects*. Available: https://flask.palletsprojects.com/en/2.0.x/. [Accessed: May 8, 2025].

[3] “MariaDB Foundation,” *MariaDB.org*. Available: <https://mariadb.org/>. [Accessed: May 8, 2025].

[4] “Pandas - Python Data Analysis Library,” *pandas.pydata.org*. Available: https://pandas.pydata.org/. [Accessed: May 8, 2025].

[5] “Matplotlib: Visualization with Python,” *matplotlib.org*. Available: <https://matplotlib.org/>. [Accessed: May 8, 2025].

[6] “Scikit-learn: Machine Learning in Python,” *scikit-learn.org*. Available: https://scikit-learn.org/stable/. [Accessed: May 8, 2025].

[7] “Hotelogix Property Management System,” *Hotelogix.com*. Available: <https://www.hotelogix.com/>. [Accessed: May 8, 2025].

[8] “Odoo Hotel Management Module,” *Odoo.com*. Available: https://www.odoo.com/page/hotel-management. [Accessed: May 8, 2025].

[9] “eZee Absolute - Hotel Management Software,” *ezeeabsolute.com*. Available: <https://www.ezeeabsolute.com/>. [Accessed: May 8, 2025].

[10] “Agile Methodology Guide,” *Atlassian*. Available: https://www.atlassian.com/agile. [Accessed: May 8, 2025].

[11] “IEEE Reference Guide,” *IEEE Author Center*. Available: <https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf>. [Accessed: May 8, 2025].