Business and Management Studies
AIJEFR VOL 6 NO 1 (2024) P-ISSN 2641-4937 E-ISSN 2641-4953
Available online at www.acseusa.org
Journal homepage: https://www.acseusa.org/journal/index.php/aijbms
Published by American Center of Science and Education, USA

DEVELOPMENT AND IMPLEMENTATION OF A PYTHON-BASED HOTEL MANAGEMENT SYSTEM: A COMPREHENSIVE TOOL FOR ROOM RESERVATION, PAYMENT, AND ADMINISTRATION

Mahfuz Alam [(a)1] *Md Shafiqur Rahman* [(b)] *Mir Araf Hossain Rivin* [(c)] *Md Borhan Uddin* [(d)] *Mir Mohtasam Hossain Sizan* [(e)]

[(a)] *Department of MBA in Business Analytics, International American University (IAU), Los Angeles, CA, United States; E-mail: jnmahfuz@gmail.com*
[(b)] *Department of MBA in Management Information Systems, International American University (IAU), LA, CA, United States; E-mail: s.rahman9560@gmail.com*
[(c)] *Department of MSc in Mathematics, Louisiana Tech University, 201 Mayfield Ave, Ruston, LA, United States; E-mail: miraraf.rivin@gmail.com*
[(d)] *Department of MBA in Business Administration, International American University, Los Angeles, CA, United States; E-mail: seumuradkhan@gmail.com*
[(e)] *Department of Masters of Science in Business Analytics, University of North Texas, United States; E-mail: mhsizan855@gmail.com*

ARTICLE INFO

Article History:

Received: 28th October 2024
Reviewed & Revised: 28th October 2024 to 26th December 2024
Accepted: 27th December 2024
Published: 31st December 2024

Keywords:

Hotel, Management, User, Dashboard, Python-Django, Payment, Administration

JEL Classification Codes:

O14

Peer-Review Model:

External peer review was done through double-blind method.

ABSTRACT

The hotel industry faces challenges related to manual management processes, which can lead to inefficiencies and errors in operations such as booking, payment processing, and administrative tasks. To address these challenges, there is a growing demand for automated systems to streamline operations and improve overall efficiency. This study investigates the design and implementation of a Python-based Hotel Management System built using the Django framework and MySQL database. The system is developed to handle essential hotel operations, including room booking, payment processing, and general management, while ensuring scalability, security, and usability. The research follows a software development approach based on functional and non-functional requirements, with a focus on unit, integration, functional, and security testing to ensure the system's reliability and performance. Key features of the system include real-time booking functionality, secure payment processing, and authenticated user access for administrators and customers. The results reveal that the system effectively reduces operational errors, enhances the user experience for both hotel administrators and guests, and improves overall operational efficiency. The system also supports secure financial transactions and simplifies hotel management tasks through an intuitive interface. The findings suggest that the Python-based Hotel Management System provides an effective solution for automating hotel operations. It offers a scalable, secure, and user-friendly platform, optimizing both management and customer-facing tasks while improving the overall efficiency and effectiveness of hotel operations.

INTRODUCTION

Hotel The rapid expansion of the hospitality industry has highlighted the necessity of effective hotel management systems that can respond to the challenges created by manual and semi-automated systems. Conventional techniques for room reservations, payment processing, and administrative functions are susceptible to inefficiencies, errors, and delays. This ultimately results in customer dissatisfaction and operational penalties. Besides, with technological advancements, the hospitality industry is experiencing cut-throat competition. Tackling these issues is important for the sector's future growth and competitiveness. From a scientific perspective, the contribution of this research is in providing an integrated, real-time and scalable hotel management system with efficient operational processes while ensuring strong security and ease of use. This is one of the reasons why current solutions often only provide a partial remedy to the comprehensive issues surrounding real-time data synchronization, secure payment gateways, and role-based access control. These gaps need to be filled which calls for a complete overall system to fulfill the specific demands of the administrators, hotel owners, and customers. This study proposes the design and implementation of a hotel management system based on Python programming language, using the Django framework and MySQL database. Main goal is build reliable tool for fast room booking, safe payment

[1]Corresponding author: ORCID ID: 0009-0009-2005-8326

https://doi.org/10.46545/aijbms.v6i1.322

To cite this article: Alam, M., Rahman, M. S., Rivin, M. A. H., Uddin, M. B., & Sizan, M. M. H. (2024). DEVELOPMENT AND IMPLEMENTATION OF A PYTHON-BASED HOTEL MANAGEMENT SYSTEM: A COMPREHENSIVE TOOL FOR ROOM RESERVATION, PAYMENT, AND ADMINISTRATION. *American International Journal of Business and Management Studies*, *6*(1), 15–44. https://doi.org/10.46545/aijbms.v6i1.322

processing, and administration management. Mediate goals are the following data digital, robust identity isolation, and also real-time feedbacks for all stakeholders. The approach includes scrutiny of functional and non-functional requirements, architecture design as a modular system, and thorough testing to ensure performance and reliability. As a result, the system undergoes unit, integration, functional, and security tests that produce a valid response and are consistent with industry standards. The world has come to understand the importance of computers and the applications they can be used for. Computers' influence on society is seen in hotel reservation systems. The hotel operations' efficiency has significantly improved due to the Hotel Room Reservation System. Travelers need an information system that is easy to use, quick to respond to, and trustworthy while making travel plans. Everyone always needs shelter, whether business travelers traveling from one place to another or students studying away from home. Shelter is a basic human necessity. Professionals venture outside of their industry to pursue particular objectives, like travel. The manual hotel management method is labor-intensive and time-consuming, often leading to delays in releasing booking lists. It also increases the risk of duplicate records when clients request additional rooms for others, complicating room management. Additionally, improper handling of document transfers or records can result in the loss of entire bookings, further disrupting operations. This method also makes it difficult to efficiently track and inspect visitors, especially those with complex or challenging booking issues.

The aims and objectives of the system include simplifying data retrieval and storage, allowing for easy tracking of hotel guests' arrivals and departures. It will provide summaries of visitors who have reserved, arrived, and left the hotel on a daily, weekly, and monthly basis. Additionally, the system aims to improve data management efficiency by eliminating redundant information.

Subsequent sections discuss the design and development process for the system, share results of testing, and assess the effectiveness of the system in addressing identified problems. The results demonstrate the system's ability to enhance hotel operations and their efficiency, security, and user satisfaction.

LITERATURE REVIEW

An online hotel management system is a computerized web-based system for reservations. These systems have been developed over the years to meet the growing demands of technology and other requirements, such as secure payments. The literature review chapter discusses the evolution of hotel management systems from manual calculation to digital platforms. The chapter also discusses the influence of technology and the different deliveries of online and offline systems.

Understanding the need for hotel management systems

The exponential growth of the hotel and hospitality industry requires practical tools and approaches to satisfy the ever-growing demand from the ever-growing population. Several systems have been designed to offer solutions to hotel room reservations. These systems have been developed from the manual user of ledger cards to sophisticated systems that implement advanced technologies. The literature review section examines information concerning the online hotel reservation system.

Technology has led to various transformations in different contexts, including the hospitality industry. In hotel management, online reservations have become common in virtually all hotels worldwide. Online hotel reservation is a business operation that provides an efficient way for a customer to find a room in a hotel, check if that Room is available, and make payment, which is done through a secure online payment interface. There has been an increasing trend in hotel reliance on one platform due to a wide range of advantages associated with the operation, such as streamlining operation procedures, hen, eliminating onerous occurrences, and providing a seamless platform for users to interact with (Kim & Kim, 2004).

The advancement of technologically oriented systems is mainly attributed to the need for accurate operations, customer convenience, and the effectiveness of hotel operations. The early researchers on the use of online reservation management systems indicated that there were numerous advantages associated with online operations primarily due to the ability of the information to vary in real-time, thereby facilitating a more effective continuing satisfaction of customers' needs and the development of customer relationship management (Hu & Gu, 2013). The elimination of manual activities in hotels is also associated with the minimization of human errors. Indeed, one of the most significant benefits of using an online hotel reservation management system is eliminating human errors in many operations. The use of the management system integrated with payment processes, such as the use of credit cards, is also associated with enhanced user satisfaction. Some security measures integrated into online payment systems include securely encrypted consumer information and high accuracy of transactions since the operations are done through automated systems. The process of transaction is therefore associated with minimal mistakes. Operation systems are also essential since they allow hotels to manage their products effectively, including room inventory, prices, profit margin strategies, and customers' interests. The need for services by the hospitality industry hotels has been precipitated by a high level of competition in the sector (Delizo & Esguerra, 2013). The influence of online travel agencies such as booking.com and Expedia.com has also added to the reliance on technologies in online reservations. Necessarily, in the bid to enable users to book reservations in real-time, effectively compare prices, and check quality levels of services through reviews, hotels have yet to be left with an option but to develop their online booking engines or resort to using external booking engines.

Historical Development of HMIS

In the 20th century, there was a significant shift in the hospitality sector from manual booking methods to computerized ones; hotels relied on paper files to book clients and track the number of booked rooms and guests in and out of the Hotel. This process was inefficient, leading to double bookings and loss of files many times, among others. The computerized booking method was booming with the computerized hotel management system used in the 1990s. The first computerized system adopted basic ways of Booking, which included information about the guest, Booking, orders ready, Room ready for the guest, and checking the guest out. The system was one of the significant ways that improved the hospitality management system and helped in numerous ways. Among the benefits include proper Booking of clients, knowing which rooms are empty for new guests, and well-structured payment methods, leading to efficient collection of revenue in the hotels.

The advances in internet technology led to the development of web-based hotel reservation systems, which hotels adopted during the late 1900s and early 2000s. The system enables the guest to know which hotels have which empty rooms, reserve the rooms, and pay for the rooms. This system was more efficient, and many hotels that had not adopted the radical mode of Booking were forced to follow suit. The system was enhanced by the development of travel and tour companies and agents, some of which provided a way to make travel arrangements and reservations, not necessarily call the Hotel for Booking.

As the technology advancements continued, hotel management systems started to be equipped with more advanced and useful features such as dynamic pricing algorithms, customer relationship management tools, and integration with social media. These features allowed hotels to understand customers' preferences better, lead to a more tailored experience for guests, and optimize pricing strategies for hotel owners based on real-time demand. Nowadays, a hotel management system is integral for hotels of every size and type. It offers an extensive solution to manage reservations, payment, and interaction with hotel guests.

Technological Advancements in Hotel Management Systems

Technological advancements in hotel management have significantly improved hotel operations' efficiency, accuracy, and customer service standards. The most game-changing innovation is cloud computing. It has allowed many hotels to migrate from on-premise hotel management systems to cloud-based solutions. The benefits of the latter solution are manifold, including scalability, which allows hotels of any size to adopt it; flexibility, which allows the management to access the system from any location; and reduced costs for maintaining additional on-site infrastructure. Furthermore, cloud storage of data is more efficient than saving it in local databases. Saving the data to remote servers makes the system real-time, which is an excellent way to facilitate and enhance the efficiency of hotel operations.

Mobile technology is another crucial solution that should be implemented in the modern hotel management system. In conjunction with the rapid growth of smartphones, many hotels have designed mobile-friendly booking platforms that allow guests to check in, check out, and book their rooms through their smart devices. Mobile apps for hotels are becoming increasingly popular due to their convenience and the possibility of booking hotel services through the app, such as room service and spa or restaurant service.

Artificial intelligence and machine learning have become used in the industry in recent years, reshaping the industry by improving customer interactions, operational efficiency, and pricing strategies. The technology has allowed using AI-powered chatbots in many hotels to manage guest inquiries, suggest recommended sites, and even book guest reservations. AI systems learn the behaviors of guests, analyze the data, and predict the potential preferences a guest may want, hence offering more personalized services to increase customer satisfaction. Further, analyzing these data helps predict the areas where the casino is losing money, with machine learning algorithms being used to change the room rates in real-time accordingly. In addition, blockchain technologies have become more commonly used to manage booking guest systems. The technology allows a secure and effective management system where transactions are managed securely and mutually. This dramatically reduces fraud and increases trust. In addition, the technologies were also used in managing customer loyalty programs, allowing the earning and redemption of customer points across different branches in the Hotel.

The Internet of Things has also allowed the incorporation of smart devices in the industry, significantly increasing the guest experience in many ways. Many hotels have smart thermostats, intelligent occupancy sensors, smart switchboards, and even innovative parking systems to make guests' stays much more convenient. Secure payment gateways are also more commonly used today, considering that more customers are paying for their slots online. The secure payment gateways use encryption, tokenization, and other essential steps to ensure customers' safety and better payment processes that ensure that the confidentiality and payments of customers are always safe and secure.

Online and Offline Hotel Reservation Systems: A Comparative Analysis

A comparison between online and offline hotel reservation systems shows that online systems present numerous advantages in terms of efficiency, convenience, and guest satisfaction. Online hotel reservation systems allow guests and prospective visitors to check room availability and make reservations and payments from any location and time without direct contact between the visiting guests and the hotel staff. Convenience is the principal factor leading guests to adopt online rather than traditional offline systems. Offline hotel reservation systems are still being used, and they involve the guests' communications with the Hotel to make reservations. This method is time-consuming and is prone to human errors, such as inadvertent double bookings, miscommunications, and misunderstandings between guests and hotel staff.

On the other hand, online systems are automating and updating the room availability status and supporting safe and secure payments using online means. Furthermore, online hotel reservation systems are more cost-effective because they significantly reduce the time and effort required from the hotel staff to prepare invoices, manage payments, and control room availability and Booking. In addition, online systems can be integrated with other systems that support customer relationship management or hotel operations. The user experience is improved by the fact that the users can compare room rates and search for reviews and special promotions. Guests have more control over their reservations and can change their reservations or cancel them without needing to contact the Hotel.

User experience and why it matters

User experience is critical in the context of the success of online hotel reservation systems. One of the significant aspects of user experience in hotel online reservation systems is a user-friendly interface. A successfully designed UI would improve customer satisfaction, increase online bookings, and stimulate brand loyalty. By contrast, poor design would result in booking cancellations, negative feedback, and loss of revenue. While navigating a hotel reservation system, customers should be able to see what rooms are available for the dates they need to book, the prices, and how they can do it in just one click. The search box that would not cause your guests to perform dozens of clicks should be easy to find and placed right in front of the search engine.

Similarly, precise wording in web design and the context of simple navigation menus is essential for achieving a simplistic UI. Additionally, online customers are impatient, and it is critical to complete most bookings without them leaving their pages, as they would get annoyed by the lengthy booking process. The need-to-have list of features is also transferable to the context of e-commerce.

Mobile responsiveness is another crucial consideration. The data show that more guests are booking hotels through their smartphones, so adjusting the website to the device is crucial. In particular, it is stated after Delizo and Esguerra that a mobile-friendly design enables guests to quickly look through the rooms available for Booking, reserve a room, and pay for it. Thus, this requirement also contributes to enhancing the hotel reservation system. Another consideration is the use of personalization. Modern AI and machine learning technologies collect guest data and allow one to make personalized offers depending on previous bookings, preferences, and search history. According to Kim and Kim, this personalization promotion can significantly improve the customers' experiences by providing them with offers that meet their specific needs.

RESULTS AND DISCUSSIONS

Requirement Analysis

Requirement analysis is the initial phase of any systems development lifecycle to collect information about customers' and administrators' needs. In systems development, it is crucial to consider the requirement analysis phase to ensure the final product meets the users' requirements. This aspect is particularly crucial in developing the hotel management system. The current part will outline the need for the Python tool to manage a hotel. The following part will consider the requirement analysis with a separate focus on the functional and non-functional requirements. The requirement analysis phase aims to develop a comprehensive hotel management tool to help hotel owners, administrators, and customers fulfill their needs in managing reserved and booked rooms.

Requirements Engineering

As the Name suggests, requirements engineering is the technological discipline that develops user needs and specifies software systems. The concept of requirements engineering is defined differently by different people. However, it is generally agreed upon that requirements involve understanding user demands and what they mean for the design of a computer system. Requirements engineering and software engineering are closely associated fields that center on the process of building systems that consumer’s desire.

User Requirements

- An app that facilitates pleasant and easy communication is required.
- An application that lets the administrator book a room with accurate guest details.
- A program that lets the administrator take online and cash payments.
- Using computers to handle all organizational functions.
- Enabling direct room reservations for users.

System Requirements

Admin Module

Home: The administrator may quickly check the total number of new bookings, authorized bookings, canceled bookings, and registered users in this section through the dashboard.

Room Category: The administrator can create, edit, and remove categories in this section.

New Room: The administrator can add, edit, and delete rooms in this section.

Booking: In this section, the administrator can inspect, approve, cancel, and add comments to new and approved

bookings.
Registered Users: The administrator can view the details of registered users in this section.
Search: Using the booking number, the Admin can look up inquiries and booking information in this part.
Reports: This section allows the administrator to examine the details of inquiries and bookings made within a specific time frame.
Invoice: The administrator can create and print an invoice following the confirmation of a reservation.

Hotel Owner Module
Home: This part allows the hotel owner to examine all new reservations, authorized reservations, and canceled reservations quickly through the dashboard.
Hotel owners can add or amend categories for their rooms in this section.
Add or amend rooms in this section for the hotel owner.
Booking: The hotel owner can view, approve, cancel, and view new reservations in this section. They can also leave comments.
Reports: Hotel owners can examine booking details for a specific period and read inquiry details in this section.
Invoice: The administrator can create and print an invoice after the reservation has been confirmed.

User Module
Home: It is a welcome page for users.
About: It is an about us page of a website.
Book Room: In this section, the user can book a hotel room by registering with the Hotel.
Sign Up: Users can register through the sign-up page.
Login: It is the login page.
Invoice: After reservation confirmation, the user can generate an invoice and print it.

Functional Requirement
- Provide users with the ability to search and discover the most pertinent booking alternatives. Link each online reservation to an account. Restrict each account to a single user.
- Accept the date and time to see if any rooms are available at that specific moment. Send the booking confirmation to the designated contact information. Compute and show the cost of lodging and other utilities. Cancel reservations. Show and modify visitor records.

Non-functional Requirement
- Encrypt data to prevent bot bookings and make sure search results appear in reasonable amounts of time.
- If users provide incorrect input, they should be suitably assisted in filling out the mandatory forms.
- Payments should be accepted by the system using various methods (optional).
- Accessible, effective, and simple to use. Monitor responses, actions, and documents.

Specification of Each Requirement
Admin specification
Function: Registration, data addition, editing, and deletion.
Description: All system accessibility.
Inout: Admin entered his data in the designated fields.
Output: Successful information submission was made.

Hotel Owner Specification
Functions include registering, logging in, adding, editing, and deleting data.
Description: System interface access for hotel owners.
The hotel owner entered his details in the designated field.
Output: The data was successfully submitted.
Adverse consequences: Not one

User Specification
Features: Sign up, log in, search, add, and modify information
Description: System access through the user interface.
User input: The user entered his criteria and his information.
Output: The data was successfully submitted.
No adverse consequences

Database Specification
Function: Insert an entry using the database.
Enter the data as an input.
Results: Inserted data was successfully
Take Action: Provide Supporting Data
No adverse consequences

Use Case Diagram of the system

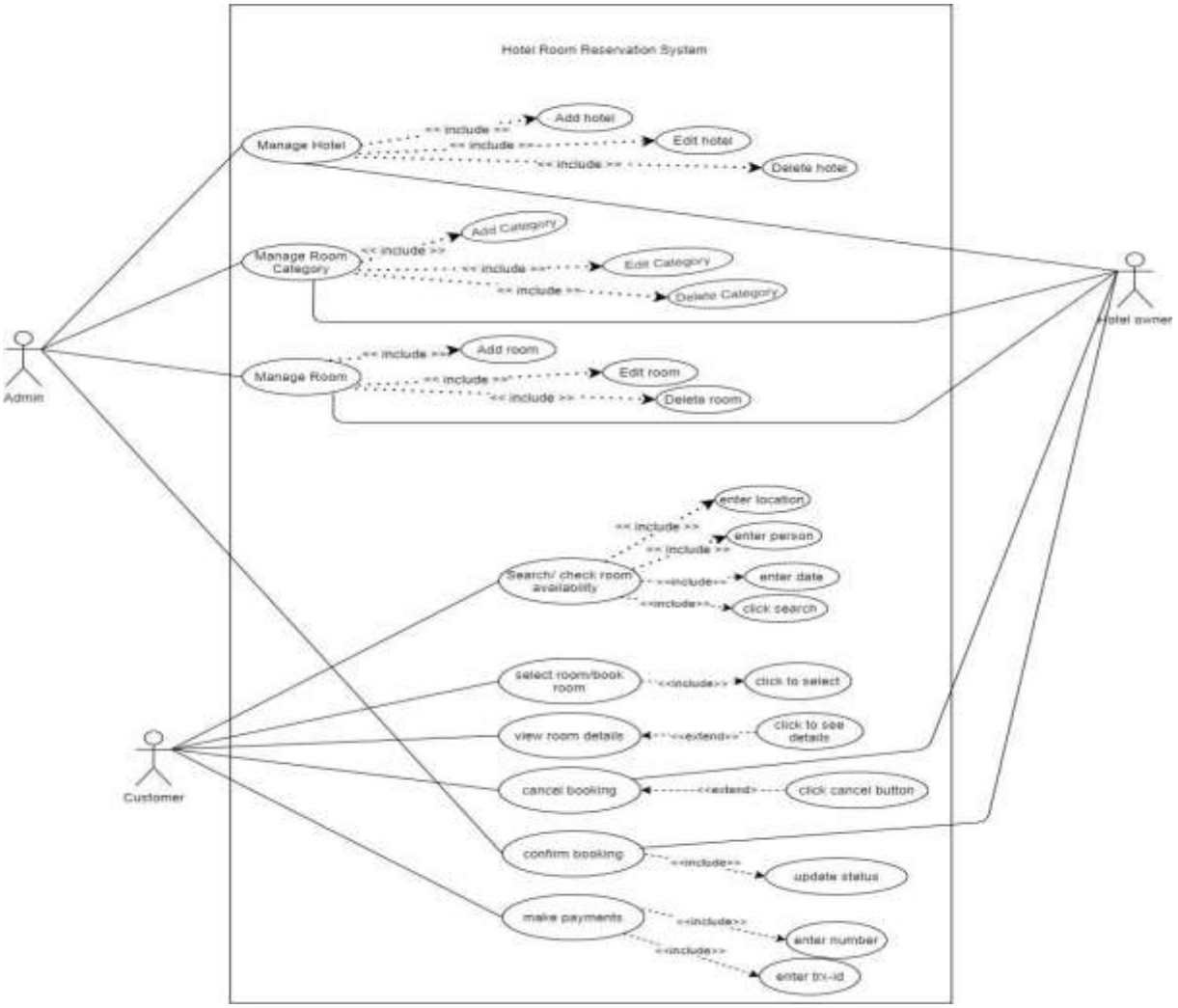


Figure 1. Use case diagram for hotel room reservation system

Figure 1 shows some of the critical interactions of the main actors, Admin, Hotel Owner, and Customer, as well as the core functionalities of such interaction. The Admin, in turn, will be involved in managing the system, rooms, bookings, reports, and users. The hotel owner will also have the following functions: a room administrator who should monitor and update the availability or selling of rooms for Booking. The customer, in turn, will search for rooms or bookings, make the reservations, and pay for them through the secure credit card payment gateway.

Project Planning
Project planning is an integral part of the software development life cycle, providing details regarding the resources, scope, activities, and timelines that have to be considered to achieve the set objectives. In the case of the Python-based hotel management system, the planning process has been logical and systematic to ensure that the identified functional and non-functional requirements are met. The current section provides an overview of such aspects of the planning process as task Estimation, resource allocation, scheduling, and a strategy for mitigating risks.

Project planning is an integral part of the software development life cycle, providing details regarding the resources, scope, activities, and timelines that have to be considered to achieve the set objectives. In the case of the Python-based hotel

management system, the planning process has been logical and systematic to ensure that the identified functional and non-functional requirements are met. The current section provides an overview of such aspects of the planning process as task Estimation, resource allocation, scheduling, and a strategy for mitigating risks. The following activities of software project planning that have followed in this project are:

- Estimation of the software project
- Task scheduling
- Personal requirements
- Resource requirements
- Estimation of the software cost
- Cost-benefit analysis

Project Estimation

The use of software has grown in importance in the design, development, and purchase of systems, especially large, intricate systems. Precise assessments of software expenses are essential to effective program administration for these kinds of systems. Most cost estimation techniques concentrate on this element and provide estimates in person months since human labor accounts for most software development costs. Accurate cost estimates are essential to both developers and clients. RFPs, contract negotiations, scheduling, monitoring, and managing are among their uses. Underestimating expenses could result in management approving suggested solutions that eventually cost more than planned, have poor quality, underdeveloped functions, and are not completed on schedule. An excessive amount of resources could be committed as a result of overestimating. The outcome for the projects, or during contract bidding, is that the contract is lost, which may result in job loss. Failures in software projects have been a popular topic in the last ten years. The majority of software project failures are caused by errors in planning and estimate rather than problems with the project itself.

Function Estimation

Functions of the Proposed System

Table 1. Functional Points

User Registration	**F1**
User login	**F2**
Admin Login	**F3**
Hotel owner Registration	**F4**
Hotel Owner Login	**F5**
Add Types	**F6**
Add Rooms	**F7**
Manage Category	**F8**
Manage Types	**F9**
Manage Rooms	**F10**
Manage guest	**F11**
Booking Rooms	**F12**
Manage Booking	**F13**
View Booking Record	**F14**
Add Payment	**F15**
Manage Payment	**F16**
View Records	**F17**
Contact Us	**F18**
Generate Report	**F19**

Ensuring that functional points measure a downloaded software system's functionality is essential. Spitzer states that functional points are critical for estimating the effort required to develop the system. Therefore, at the organizational level, functional points reflect critical tasks or features that have to be supported by the hotel management system. Table 1 explains what must be done to make the system function. The data in the table obviously reflects the results of a top-down design, stating that three types of users will use the system and 12 functions must be performed. A row in the table corresponds to the unique system functions and can be associated with the users' intended interaction or an internal operation.

An explanation of some of the Functional Points is given below:

User Registration: The system allows users to create an account by entering their details. It is a fundamental function because it is the entry point for customers to book hotel rooms, view previous bookings, and other information in the system.

User Login: After registering, users need to log in to access the system's features, such as searching for rooms or making payments. This ensures that the system can differentiate between the registered users and provides a personalized experience for them, who can view the hotel rooms they booked previously.

Admin Login: This function enables the Admin to log in to the backend system to view reports and other functions. The admin access is restricted for security reasons to ensure that only the Admin can make changes to the entire system.

Hotel Owner Registration and Hotel Owner Login: This function is similar to the user registration; however, it allows the hotel owners to register and log in. After logging in, the hotel owners can make hotel rooms available, adjust the pricing, and view bookings. These two roles need to be separated, meaning the registration of normal users and hotel room management users.

Room and Category Management: These are all the functions that allow the Admin or the hotel owners, should they log in locally, to add, update, delete, and manage the room categories and types. For example, the various types of rooms available in a certain hotel can be standard, deluxe, suites, etc. These usually have differing prices and are available on different days.

Manage Guests: These are all the functions that allow the admins or hotel owners to view and manage the records of the people who are booking the hotels. It should give information about the guests through booking details and other information.

Search and Book Rooms: This is one of the system's core functionalities. It allows users or customers to search and book rooms based on availability, dates, and room type preferences. Booking functionality ensures the system maintains bookings in real-time without overbooking or double bookings.

Manage Bookings: This function allows administrators or hotel owners to manage and modify existing bookings. It allows viewing new bookings, accepting or canceling them, and putting instructions and remarks if needed.

View Booking Records: It is another critical feature for administrators and hotel owners to get a summary or detailed view of all bookings. Not viewing the Booking will allow the administration to see the trend in occupancy and customer information.

Add and Manage Payment: The system also needs to process payment, allowing users and admins to control the payment related to the Booking. The payment process needs to accept payment online, and recording in the system should see through the different payment methods.

View Record and Generate Report: This functionality is vital for administrators and hotel owners to watch guest bookings, payments, and other records. Report functionality should also be available where the report can be generated to indicate how the Hotel runs in a given period, such as a daily, weekly, and monthly summary.

Function-Oriented Metrics

Instead of emphasizing software values, function point-based estimate concentrates on information domain values. Five information domain properties are compared in order to compute function points. The following are the information domain values:

The total number of external inputs (EI) is the sum of all user inputs that give the program unique application-oriented data. It is essential to distinguish between inquiries and inputs.
Total number of external outputs (EO): The total number of external outputs (EO) is tallied for every user output that gives the user information specific to their application.

Number of external inquiries (EQ): An online input that yields results is considered an inquiry—creating an online output that is an instantaneous software reaction. Every unique question was recorded.

Counting each logical master file determines the number of internal logical files (ILF), which are database tables that a program modifies based on input.

The total number of machine-readable interfaces used to send data to another system was tallied as the external interface files (EIF).

The domain weights are fixed and can be found in the corresponding table. Three categories can be used to group weights based on how the system works. They are average, complicated, and straightforward. Although a component of the more extensive system, the entire system is complex. Each count is given a complexity rating once the data has been gathered.

The FP count is calculated using the following formula:

The Value Adjustment Factor (VAF) equals 0.65 plus (.01X TDI). Adjusted Function Point Count (AFP) = UFP X VAF Effort for PHP = AFP x Productivity; UFP = UFP (Data Fn) + UFP (Transaction Fn).

Function Point Estimation

This table shows the functionality of the input and output of Admin.

Table 2. Functional point Estimation (Admin)

Functionality	Input	Output
Login	Email, Password	Enter the Admin Dashboard
Add Categories	Category id, Category name	Added into a Database table
View Categories	Click on View categories	Display all categories listed as a table
Update Category	Click on the Edit button	Updated successfully
Delete Category	Click on the Delete button	The record has been removed
Add Types	Type id, Type name	Added into a Database table
View Types	Click on View Types	Display all Types list as a table
Update Types	Click on the Edit button	Updated successfully
Delete Types	Click on the Delete button	The record has been removed
Add Rooms	Room ID, Room Name, Category, Type, Facility, Details, Price	Added into a Database table
View Rooms	Click on View rooms	Display all Rooms listed as a table
Update Rooms	Click on the Edit button	Updated successfully
Delete Rooms	Click on the Delete button	The record has been removed
Add Booking	Booking ID, Guest ID, Room ID, Member ID, Check Time, Check Out Time, Grand Total	Added into a database table
View Booking	Click on view booking	Display all Booking lists as a table
Delete Booking	Click on the Cancel button of Booking	Data has been deleted
View payments	Click on View Payment	Display payments as a table
Delete payments	Click on the delete button	Data has been deleted
Update Payment	Click on the update button	Data has been updated successfully

Table 3. Functional Point Estimation (Customer)

Functionality	Input	Output
Registration	User ID, user name, user email, user phone, password	Added into Database Table
Login	Email, Password	Enter the user panel
Search rooms	By filtering Dates, categories, and types	Display rooms info
Add Booking	Select rooms and book that Room	Added into the database
Payment	Choose offline payment or online payment	The Room will be listed on the account as booked
Booking Details view	Booking Id, Guest Id, Room id, Member id, Check-In Time, Check-Out Time, Grand Total	Display order detail

Table 4. Identify Complexity (Hotel Owner)

Transition function	Field/ file involve	FTRs	DETs
Registration	Hotel owner ID, user name, user email, user phone, password	2	6
Login (EI)	Fields- Email, password File- login	1	2
Add Categories (EI)	Fields- Category id, Category name File-Category	1	2
View Rooms (EO)	Fields- Room ID, Room Name, Category, Type, Facility, Details, Price, created at File- View Room, category, type, facility	4	8
Update Rooms (EI)	Fields- Room ID, Room Name, Category, Type, Facility, Details, Price, created at File- Room, category, type, facility	4	8
Delete Rooms (EI)	Fields- Room ID, Room Name, Category, Type, Facility, Details, Price, created at File- Room	1	8
Add Booking (EI)	Fields- Booking ID, Guest ID, Room ID, Member ID, Payment ID, Check-In Time, Check-Out Time, Grand Total, Issue date File- Booking, Room, Guest, Payment	4	10
View Booking (EO)	Fields- Booking ID, Guest ID, Room ID, Member ID, Payment ID, Check-In Time, Check-Out Time, Grand Total, Issue date File- View Booking, Room, Guest, Payment	4	10
Delete Booking (EI)	Fields- Booking ID, Guest ID, Room ID, Member ID, Payment ID, Check-In Time, Check-Out Time, Grand Total, Issue date File- Booking, Room, Guest, Payment	4	10
View payments (EO)	Fields- Payment ID, subtotal, VAT, total, discount, paid, due File- View Booking, Payment	2	7
Delete payments (EI)	Fields- Payment ID, subtotal, VAT, total, discount, paid, due File- Payment	1	7
Update Payment (EI)	Fields- Payment ID, subtotal, VAT, total, discount, paid, due	2	7

	File- Booking, Payment		
Add Guest info (EI)	Fields- Guest ID, Name, address, phone, adult, child File- Guest, Booking	2	6

Table 5. Identify Complexity (Customer)

Transition function	Field/ file involve	FTRs	DETs
Registration (EI)	Fields- User ID, user name, user email, user phone, password File- User	1	6
Login (EI)	Fields- Email, password File- login	1	2
Search rooms (EQ)	Fields- Check-in date, check-out date, Categories, Types, Rooms, Price File- Category, Type, Room	3	6
Add Booking (EI)	Fields- Booking ID, Guest ID, Room ID, Member ID, payment ID, Check Time, Check-Out Time, Grand Total, Issue date File- Booking, Room, Guest, Payment	4	10
Payment (EI)	Fields- Payment ID, subtotal, VAT, total, discount, paid, due File- Booking, Payment	2	7
Booking Details view (EI)	Fields- Booking ID, Guest ID, Room ID, Member ID, Payment ID, Check-In Time, Check-Out Time, Grand Total, Issue date File- View Booking, Room, Guest, Payment	4	10

Identify Complexity of Data Function

The table shows the identified complexity of the Data Function

Table 6. Identify Complexity (DF)

Data function	Field/ file involve	RETs	DETs
Manage Categories (ILF)	Fields- Category id, Name, edit, delete File- View Categories	1	2
Manage Type (ILF)	Fields- Facility id, Name, edit, delete File- View Facilities	1	2
Manage Types (ILF)	Fields- Type id, Name, edit, delete File- View Types	1	2
Manage Rooms (ILF)	Fields- Room ID, Room Name, Category, Type, Facility, Details, Price, created at, edit, delete File- View rooms	1	3
Manage Booking (ILF)	Fields- Booking ID, Guest ID, Room ID, payment ID, Check-In Time, Check-Out Time, Grand Total, Issue date, edit, delete File- View Booking	1	6
Manage Guests (ILF)	Fields- Guest ID, Name, address, phone, adult, child File- Guest, Booking	2	6
Manage Payment (ILF)	Fields- Payment ID, subtotal, VAT, total, discount, paid, due, edit, delete File- Booking, Payment	2	5
Manage Users (ILF)	Fields- User ID, user name, user email, user phone, password File- User	1	5
Manage admin users (ILF)	Fields- id, Name, Email, password, address, edit, delete File- Admin	1	5

Unadjusted Function Point Contribution

The table shows the Unadjusted function Point Contribution for the Transaction Function.

Table 7. Unadjusted Function Point Contribution for Transaction Function

#	Transition Function	FTRs	DETs	Complexity	UFP
1	Registration (EI)	1	6	Low	3
2	Login (EI)	1	2	Low	3
3	Add Categories (EI)	1	2	Low	3
4	View Categories (EO)	1	3	Low	4
5	Update category (EI)	1	2	Low	3
6	Delete Category (EI)	1	3	Low	3
7	Add Types (EI)	1	2	Low	3

8	View Types (EO)	1	3	Low	4
9	Update Types (EI)	1	2	Low	3
10	Delete Types (EI)	1	3	Low	3
11	Add Facilities (EI)	1	2	Low	3
12	View Facilities (EO)	1	3	Low	4
13	Update Facility (EI)	1	2	Low	3
14	Delete Facilities (EI)	1	3	Low	3
15	Add Rooms (EI)	4	8	High	6
16	View Rooms (EO)	4	8	High	7
17	Update Rooms (EI)	4	8	High	6
18	Delete Rooms (EI)	1	8	Low	3
19	Add Booking (EI)	4	10	High	6
20	View Booking (EO)	4	10	High	7
21	Delete Booking (EI)	4	10	High	6
22	View payments (EO)	2	7	Average	5
23	Delete payments (EI)	1	7	Low	3
24	Update Payment (EI)	2	7	Low	3
25	Add Guest info (EI)	2	6	Low	3
26	Add Member info (EI)	2	5	Low	3
27	Search rooms (EQ)	3	6	Average	4
TOTAL					**107**

Unadjusted Function Point Contribution

Table show the Unadjusted function Point Contribution for Data Function

Table 8. Unadjusted Function Point Contribution for Data Function

Data function	RETs	DETs	Complexity	UFP
Manage Categories (ILF)	1	2	Low	**7**
Manage Facilities (ILF)	1	2	Low	**7**
Manage Types (ILF)	1	2	Low	**7**
Manage Rooms (ILF)	1	3	Low	**7**
Manage Booking (ILF)	1	6	Low	**7**
Manage Guests (ILF)	2	6	Low	**7**
Manage Payment (ILF)	2	5	Low	**7**
Manage Users (ILF)	1	5	Low	**7**
Manage admin users (ILF)	1	5	Low	**7**
Total				**63**

Performance and Environmental Impact

The table shows the Performance and environmental impact here.

Table 9. Performance and Environmental Impact

	GSC	TDI
1	Data Communication	2
2	Distributed Data Processing	0
3	Performance	3
4	Heavily Used Configuration	1
5	Transaction Rate	0
6	Online Data Entry	3
7	End-user Efficiency	4
8	Online Update	2
9	Complex Processing	2
10	Reusability	3
11	Installation Ease	3
12	Operational Ease	3
13	Multiple Sites	3
14	Facilitate Change	3
Total Degree of Influence (TDI) (Range 0 to 70->influence size by +-32%)		32

Value adjustment factor (VAF) = (0.65+ (0.01* TDI))
= (0.65+ (0.01* 32))
= 0.97

UFP= UFP (Data function) + UFP (Transaction function)
= 107 + 63 = 170

AFP= UFP * VAF = 170 * 0.97 = 165 Approx.

Total time calculation frame = 165 * 15.5 [Productivity of Python is 15.5] = 2558 per hour
= 2558-person hours / 9 hours
= 284-person days / 3 [person in a group]
= 95 days per person
= 3 months 5 days for one person

Language	Hours Per Function Point
ASP*	06.1
Visual Basic	08.50
Java	10.6
SQL	10.8
C++	12.4
C	13.0
C#	15.5
Python	15.5

Three people are required to finish the project in approximately three months, and we have completed the project within the estimated timeframe.

Process-Based Estimation

In process-based Estimation, the process is decomposed into a relatively small set of tasks, and the effort required to accomplish each task is estimated. Process-based Estimation begins with delineating software functions obtained from the project's scope. A series of software process activities must be performed for each function.

Table 10. Process Based Estimation

Activity	CC	Planning	Engineering		Construction		Imp.	Total
Function			Analysis	Design	Code	Test		
F1	0.011	0.053	0.115	0.104	0.133	0.021	0.032	.354
F2	0.010	0.051	0.165	0.129	0.164	0.052	0.024	.595
F3	0.016	0.030	0.102	0.175	0.139	0.031	0.016	.656
F4	0.013	0.023	0.049	0.192	0.238	0.057	0.025	.697
F5	0.015	0.016	0.102	0.147	0.297	0.018	0.012	.607
F6	0.016	0.021	0.151	0.113	0.234	0.063	0.026	.624
F7	0.010	0.039	0.123	0.121	0.232	0.039	0.027	.691
F8	0.012	0.032	0.112	0.295	0.136	0.016	0.022	.625
F9	0.014	0.061	0.125	0.192	0.215	0.032	0.029	.864
F10	0.013	0.064	0.185	0.282	0.233	0.061	0.047	.798
F11	0.010	0.025	0.117	0.105	0.135	0.014	0.014	.987
F12	0.011	0.022	0.128	0.125	0.142	0.025	0.021	.896
F13	0.012	0.052	0.043	0.172	0.176	0.020	0.018	.769
F14	0.010	0.035	0.122	0.185	0.240	0.032	0.021	.785
F15	0.013	0.016	0.106	0.112	0.134	0.044	0.019	.876
F16	0.015	0.024	0.101	0.285	0.158	0.018	0.031	.876
F17	0.014	0.033	0.118	0.108	0.155	0.055	0.042	.520
F18	0.015	0.034	0.119	0.109	0.156	0.056	0.043	.514
F19	0.016	0.035	0.120	0.110	0.157	0.057	0.044	.897
F20	0.013	0.033	0.101	0.165	0.125	0.045	0.042	.513
F21	0.015	0.022	0.115	0.138	0.186	0.068	0.027	.534
Total	0.32	0.90	2.80	3.84	4.23	0.90	0.64	13.63
Effort	2%	8%	20%	25%	35%	6%	4%	100%

Effort Distribution for the Project

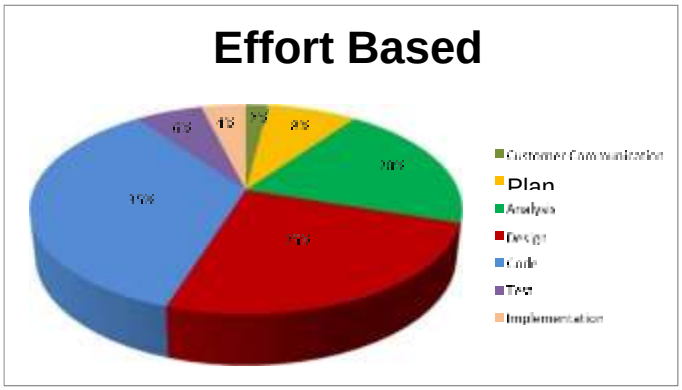


Figure 2. Effort Based Estimation

Description

•1 (2% - Customer Communication)
•2 (8% -Planning)
•3 (20% -Analyzing)
•4 (25% -Designing)
•5 (35% -Coding)
•6 (6% -Testing).
•7 (4% -Implementation).

Project Scheduling

The process of allocating the estimated work within the allotted project length is known as project scheduling. Project scheduling follows a few fundamental guidelines. They are listed in the following order:

Partitioning: The project must be divided into several doable jobs and activities.

Interdependency: It is necessary to ascertain how each divided task or activity is interdependent. While specific jobs must be completed in order, others can be completed concurrently.

Time allocation: Each task that needs to be scheduled must be allotted a certain number of work units.

Validation of effort: Each project has a set number of employees. It should guarantee that no more individuals have scheduled at any particular moment than those allotted.

Clearly defined roles: Each scheduled task should be assigned to a designated team member.

Clearly defined goals: Each work plan should have a clear goal. Usually, the result is a work product or a portion of a work product.

Project Scheduling Chart

The construction of a whole system involves several different tasks. These assignments must be completed on time and in order. The project timetable serves as the system developer's guide. This project's timetable chart is as follows:

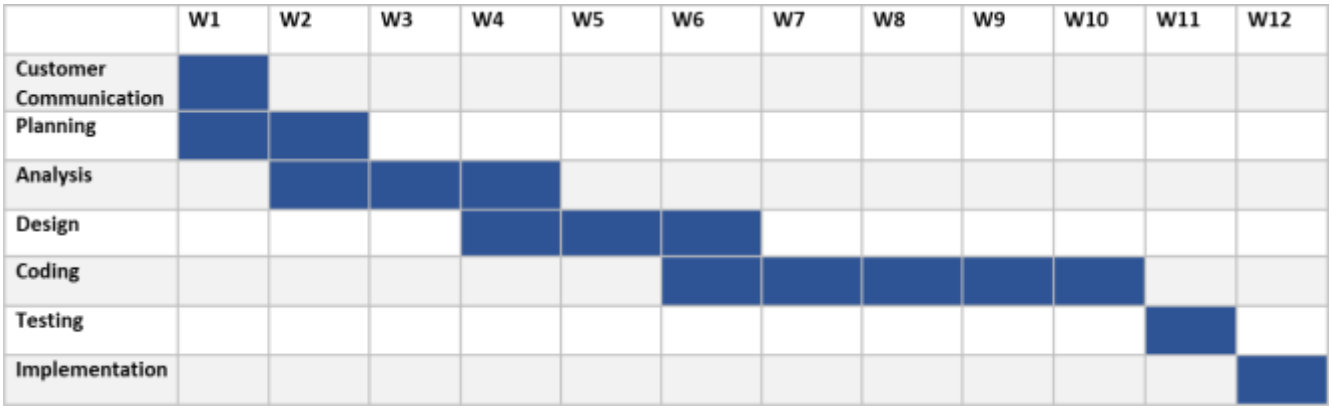


Figure 3. Project Schedule Part

Cost Estimation

The approximation of the cost of a program is cost estimation. This project has five factors to analyze and calculate the cost. Bellowed,

- Personnel cost
- Software cost
- Hardware cost
- Another cost

Personnel Cost

- Three hundred sixty-five days make up a year. There are 24 federal holidays. There are 52 weekly holidays.
- 365-(52+24) = 289 days is the total working days needed to complete the project.

- 289/12 = 24.083 days are the total monthly working days required to complete the project.
- The working hours of the organization are as follows: 9 hours per day x 24.083 * 9 = 216.747 hours per month.

Table 11. Personal Cost

Type	No. of Members	Salary
System Analyst	1	40,000.00
Designer	1	20,000.00
Coder, Tester & Customer Communicator	1	35,000.00
Total		95,000.00

Hardware Cost

Cost of the computer that was used to complete the project.

Table 12. Hardware Cost

Name	Number	Price	Number *Price	Total
Computer	2	10,000	10,000*2	20,000

Total Hardware Cost = 20,000.00 TK

Software Cost

It is the cost of the software used in this project.

Table 13. Software Cost

SL.	Software	Number	Amount	Total
1	OS (Windows 10)	1	Free	Free
2	MS Office 2013	1	Free	
3	Atom	1	Free	
4	HeidiSQL	1	Free	

Other Cost

Table 14. Other cost

Name	Price
Pen and paper	300 Tk.
Mobile	200 Tk.
Transport	500 Tk.
Total	1000 Tk.

Accounts Table for the whole project are shown in table 4.

Table 15. Total cost

Particulars	TK
Salary-	95,000.00 /=
Total Hardware Cost –	
• Computer	20,000.00/=
Software Cost –	free
Other Costs-	1,000.00 /=
Total cost	1,16,000.00 /=

Risk Management

The effective execution of my project depends on risk management, which identifies potential issues before they become serious and develops mitigation plans. Risks related to scope, technology, security, and resource limits were identified during the planning of the Python-based hotel management system. The hazards related to each component and the management of those risks are listed in the current study.

The following steps are taken in order to create a risk management model:

Identification: Risk identification is identifying possible risks or hazards by gathering data. Many tools and methods are available for gathering and manipulating data. The group is gathering information and starting to identify possible threats to Web resources through the use of both automated and manual methods. One efficient method of gathering data regarding

the condition of websites and Web pages is web crawling.

Risk classification creates an organized model that includes observable risk characteristics and events. The team uses quantitative and qualitative methodologies to describe and categorize the hazards to web pages, websites, and hosting servers.

Risk assessment involves identifying pertinent risk scenarios or events that can potentially cause harm or loss and the likelihood that these events will occur. Rosenthal lists the qualities essential to a general risk assessment standard as "Transparent, coherent, consistent, complete, comprehensive, impartial, uniform, balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance."

Analysis: Risk analysis establishes the likely extent of loss, the possible impact of risk patterns or scenarios, and the direct and indirect recovery costs. In this step, vulnerabilities are identified, mitigation strategies are developed, and the organization's willingness to accept risk in light of possible repercussions is considered.

Implementation: To manage and address identifiable risks, policies, processes, and methods are defined by risk management implementation. The implemented program should balance the worth of assets and the direct and indirect expenses of averting harm or recovering from loss.
To take comprehensive care of a web-based system, we must consider the following points:

- The hardware and software environment includes any operating system and web server updates, security patch installations, insecure service removals, firewall usage, etc.
- Administrative processes, including renewing domain name registration and entering into agreements with reliable service providers.
- Configuring and maintaining the network, including usage tracking, traffic control, and load balancing.
- Policies and procedures for archiving and backups include the type of backup media to use, how often to replace it, how many backups to make, and where to store them.

Every software project should consider various risk categories. In my project, the following risk categories have been considered.

- **Project risks:** The project plan is in danger from these threats. The project timeline is likely to be delayed, and costs will likely rise if these risks materialize. Project risks are those that indicate possible issues with the software project's budget, timeline, staffing, resources, customers, and requirements.
- **Technical risks:** These concerns threaten the timeliness and quality of the software generated. Should a technological risk materialize, implementation can become challenging or unfeasible. Technical hazards identify potential design, implementation, interface, verification, and maintenance issues. In addition, risk factors include technological uncertainty, specification ambiguity, and technical obsolescence.
- **Business risks:** These risks threaten the software's viability. Market risks, creating a system that no one actually wants to use, and creating a system that no longer aligns with the company's overarching business plan are examples of business risks. Management risks losing senior management's support if their priorities or personnel shift. Risks to the budget, loss of manpower, or budgetary commitment.

The RMMM Plan

- **Risk Mitigation:** proactive preparation to reduce danger.
- **Risk Monitoring:** evaluating whether or not anticipated risks materialize, making sure preventive measures are implemented appropriately, gathering data for potential future risk analysis, and making an effort to identify which risks led to which issue.
- **Risk Management:** What should be done if the risk has materialized and the mitigation measures have failed?

Type of Impact: Catastrophic (1), Marginal (2), Tolerable (3), Critical (4).

Type of Probability: deficient (75%).
Project Risks: endanger the project schedule. The project hazards listed below were ones I needed to control in my system.

Table 16. Project Risk (P01)

Project Risk (P01)		Date: 22-02-2024
Name	Changes the requirements	
Probability	Low (25%)	
Impact	Marginal (2)	
Description	The needs of the customer could alter.	
Mitigation & Monitoring	The organization redefines requirements in response to business needs or schedule constraints. Regular meetings will be held with the company, guaranteeing that the product we are making addresses an issue.	
Management	An emergency meeting between both parties to identify a new project	

	Both sides convene in an emergency meeting to determine the new parameters and objectives of the project.
Status	Not Occur

Table 17. Project Risk (P02)

Project Risk (P02)		Date: 27-02-2024
Name	Poor Quality Documentation	
Probability	Low (15%)	
Impact	Catastrophic (1)	
Description	Poor quality documentation of the members.	
Mitigation & Monitoring	Meetings will be held routinely to offer documentation suggestions and topics. The progress on documentation will also have a monitor in each meeting.	
Management	Adding new topics or removing unnecessary topics from the documentation will be assigned to the responsible person.	
Status	I was monitoring it.	

Table 18. Project Risk(P03)

Project Risk (P03)	Date: 08-03-2024
Name	Lack of Development Experience.
Probability	Moderate (30%)
Impact	Catastrophic (1)
Description	Lack of developmental experience of the members.
Mitigation & Monitoring	Each member of the team should watch and see areas where another team members may be weak.
Management	The members who have the most experience in a particular area will be required to help to overcome problems arising from this risk.
Status	We have not encountered such issues yet

Table 19. Project Risk (P04)

Project Risk (P04)	Date: 18-03-2024
Name	Poor Comments in Code
Probability	Low (15%)
Impact	Marginal (2)
Description	The code of the developed system is not up to the mark.
Mitigation & Monitoring	A formal written standard must be established to ensure the quality of comments in all code.
Management	We should call a meeting with the development team to get rid of this problem and improve the quality.
Status	We are monitoring the issue.

Technical Risks threaten product quality and the timeliness of the schedule. As this is my practicum project, these types of risks need to be addressed properly.

Table 20. Technical Risk (T01)

Technical Risk (TR01)	Date: 24-03-2024
Name	Computer Crash
Probability	Moderate (25-40%)
Impact	Catastrophic (1)
Description	The computer may crash due to several reasons.
Mitigation & Monitoring	We should adequately follow up on computers. We also take regular data backups every day. We can use IPS to stop unexpected shutdown.
Management	If our computer crashes, then we will restore the backup.
Status	We have not encountered such an issue yet

Table 21. Technical Risk (T02)

Technical Risk (TR02)	Date: 30-03-2024
Name	Technology Does Not Meet Specifications.
Probability	Low (25%)
Impact	Catastrophic (1)
Description	The customer does not have the technology to their desired specification.
Mitigation & Monitoring	That ensures that the product we are producing and the specifications of the customers are equivalent.
Management	The customer should be immediately notified, and whatever steps. Necessary to rectify this problem should be done. Preferably a meeting should be held between the development team and the customer is to discuss this issue at length.
Status	We have yet to encounter such an issue.

Table 22.Technical Risk (T03)

Technical Risk (TR03)		Date: 07-04-2024
Name	Poor Training Skill in Team Members.	
Probability	Moderate (30%)	
Impact	Catastrophic (1)	
Description	Poor Training Skills in Team Members to Train the Client. The entire functionality of the software. System analysts need to Ensure and monitor it while the training session starts.	
Management	We should arrange a meeting with the train team and come to A point to solve this problem.	
Status	We have yet to encounter such an issue.	

Business Risk: put the software's viability in jeopardy (risks related to the market, strategy, management, budget, and management). I am developing this project as my practicum; thus, there will not be any traditional business risks involved. As a result, it is decided that all business risks have a low probability.

Table 23. Business Risk (BR01)

Business Risk (B01)		Date: 14-04-2024
Name	Insufficient Budget	
Probability	Low (10%)	
Impact	Marginal (2)	
Description	If the budget is low, the project may not be completed.	
Mitigation & Monitoring	The project requires a pricey streaming server setup. To lower the budget risk, we've streamlined services.	
Management	Refinement in project goal. A new plan for regulating the budget	
Status	Not encountered	

Table 24. Business Risk (BR02)

Business Risk (B02)		Date: 20-04-2024
Name	End Users Accept System	
Probability	Low (10%)	
Impact	Critical (4)	
Description	The system fails to gain the user's faith.	
Mitigation & Monitoring	In order to prevent this from happening, the software will be developed with the end user in mind. The user interface will be designed to make the program convenient and pleasurable.	
Management	Training the users to familiarize them with the new system. They are releasing patches/bug fixes for greater user satisfaction.	
Status	The risk has	

Table 25. Yet to be.5.10: Business Risk (B03)

Business Risk (B03)		Date: 26-04-2024
Name	Not pay the Installment of Software Cost.	
Probability	Very Low (05%)	
Impact	Catastrophic (1)	
Description	The customer does not pay for the installation of Software Costs.	
Mitigation & Monitoring	We should make good communication between customers and ensure that the entire Installment will be completed	
Management	The only course of action available would be to find out the	
Status	Not encountered.	

Analysis Modeling

Since analysis modeling explains how the system will operate, it is the most crucial step in software development. During this stage, developers create Use Case Diagrams, Data Flow Diagrams, and Entity-Relationship Diagrams to show the various parts of the system, how they work together, and their interactions. While the second and third types of models "illustrate the system's processes and the amounts of data flowing through them," the first type of model "captures the dynamic aspects of a system" and demonstrates the functionality of the system. Models guarantee that all system requirements are met and serve as a template for the construction process.

Activity Diagram

Activity diagrams are graphical procedures that show how sequential actions and activities with concurrency, iteration, and selection are carried out. In the Unified Modeling Language, activity diagrams are intended to represent both organizational

and computational operations. An activity diagram can be applied in the same way as the other four UML diagrams. The application's specific purpose is to mimic the control flow between tasks, which does not involve messages.

It is appropriate to model the system's activity flow when using the activity diagram. Multiple systems may be included in a single application. An activity diagram, which illustrates the progression from one system to the next, also includes these systems. Alternative diagrams do not show this exact application. These systems could be external queues, databases, or any other type of system.

We will do it right now. Examine the activity diagram's applications in real life. The previous discussion makes it clear that an activity diagram is made at a relatively high level. It thus offers a high-level overview of a system. The target audience for this high-level viewpoint is mostly business users or non-technical individuals.

Admin Activity diagram

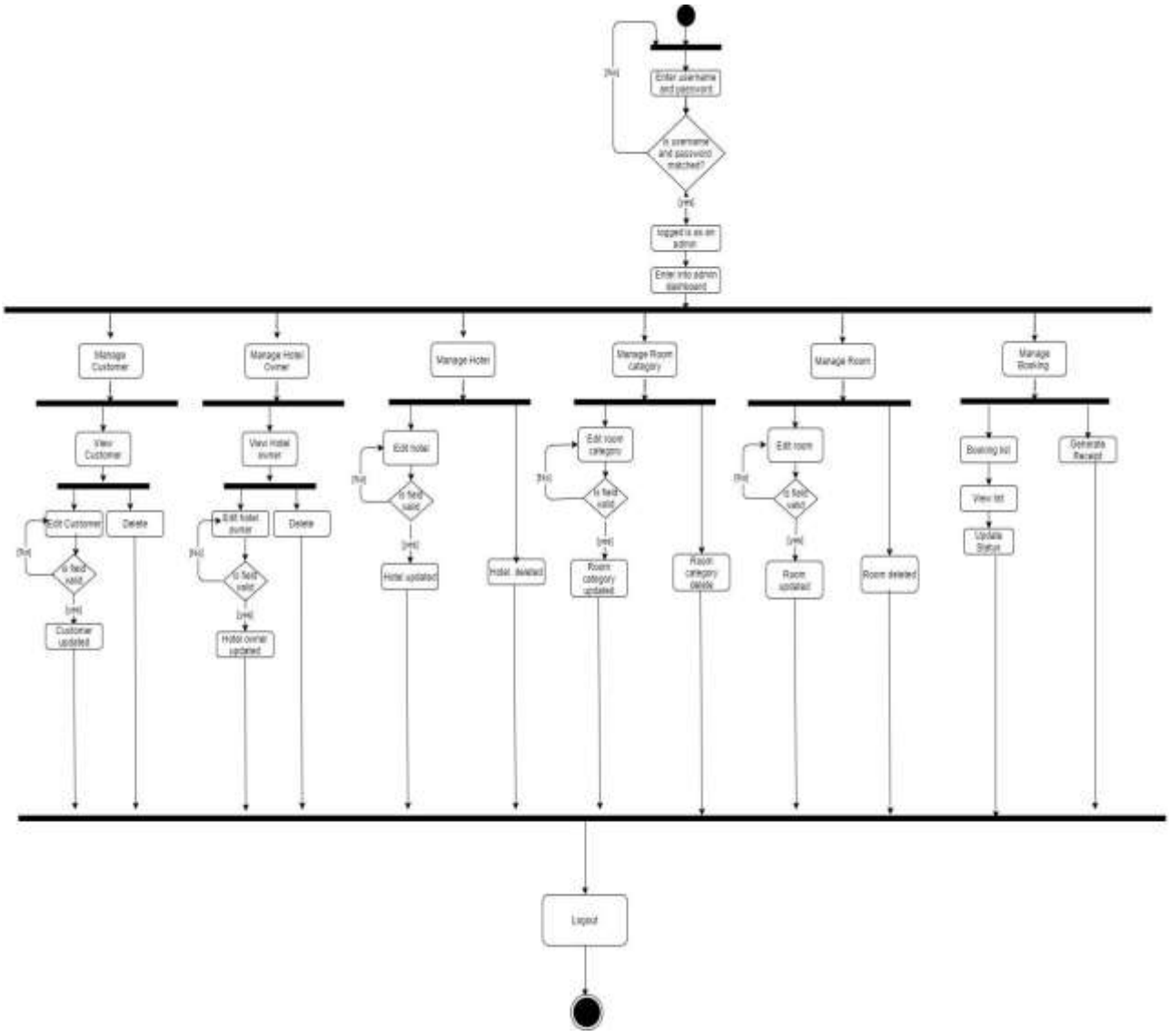


Figure 4. Admin activity diagram

Figure 4 the administrator, the hotel owner, and the user are the three critical actors in the above diagram. The administrators oversee the Booking, administer the room category, and monitor the Booking. Hotel managers are responsible for keeping the Room tidy and orderly. Bookings, payments, and invoices are the responsibility of the clients. All three roles are playable by the same user, and all three performers can be the same individual. There should be one feature available to the assistant: search. This feature allows you to view available rooms and equipment. Select the desired model and apply. In the event that they pay and inspect the Room, they will receive the "invoices" use-case. The assistant may also announce the reservation over the phone, pay for it at the Hotel, and request that the invoice be provided.

Hotel Owner Activity Diagram

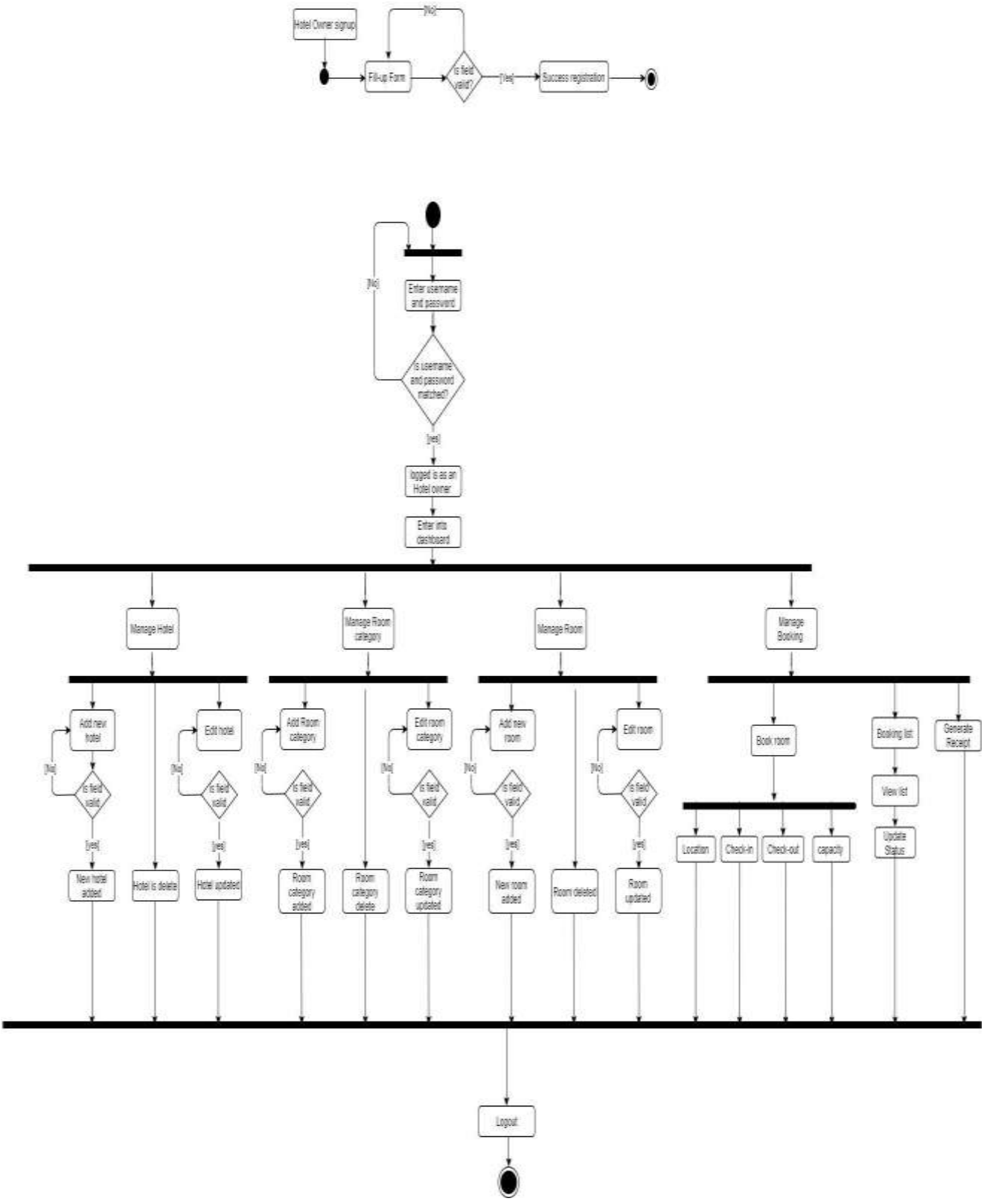


Figure 5. Hotel owner activity diagram

Customer Activity Diagram

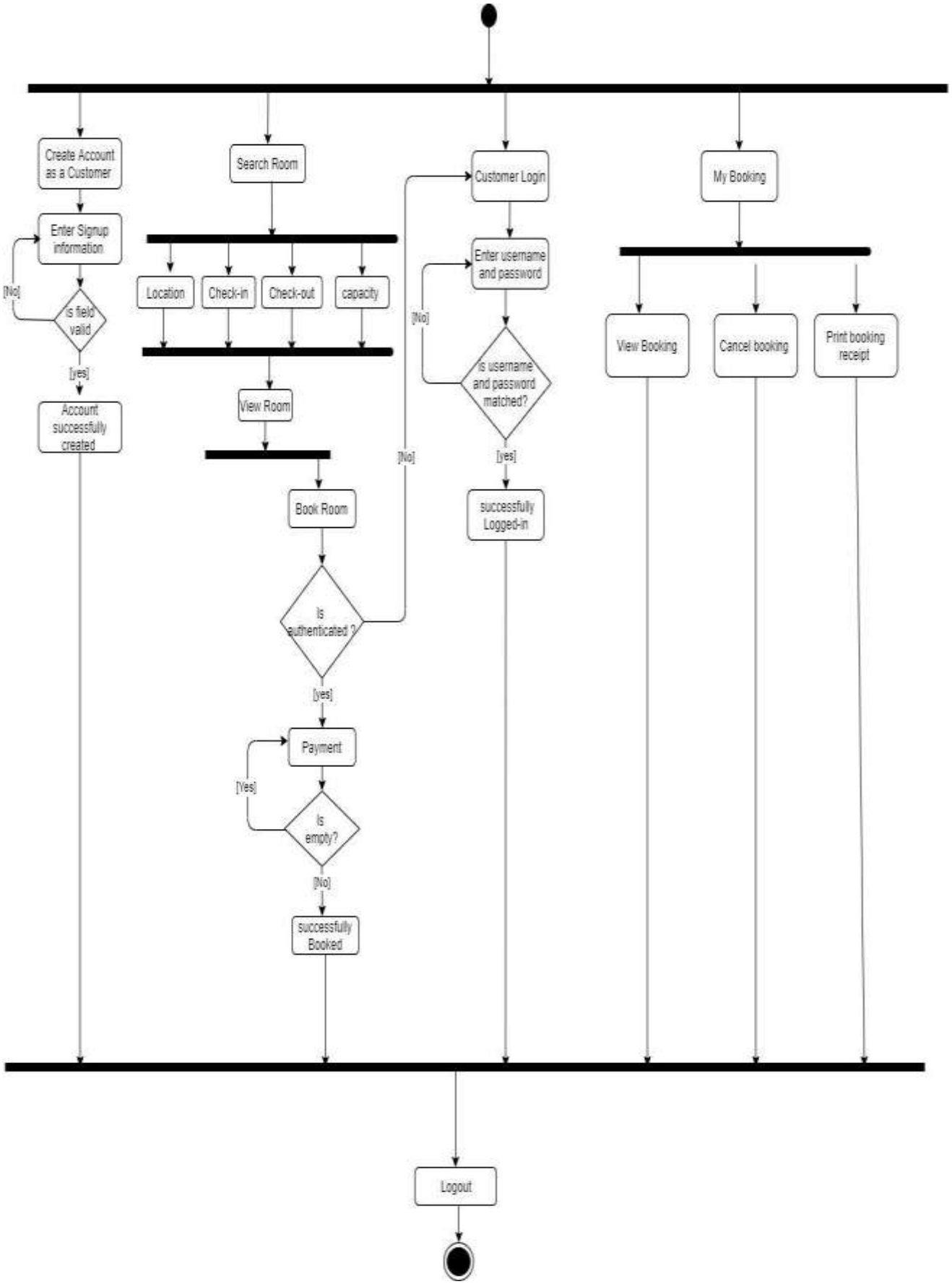


Figure 6. Activity diagram for user (ERD)

Entity-Relationship Diagram

Entity-relationship diagrams, or ERDs, are crucial components of system database designs. They illustrate the relationships between the various entities—users, rooms, and bookings—on the current diagram and show how the data will be arranged and stored within the database in this fashion.

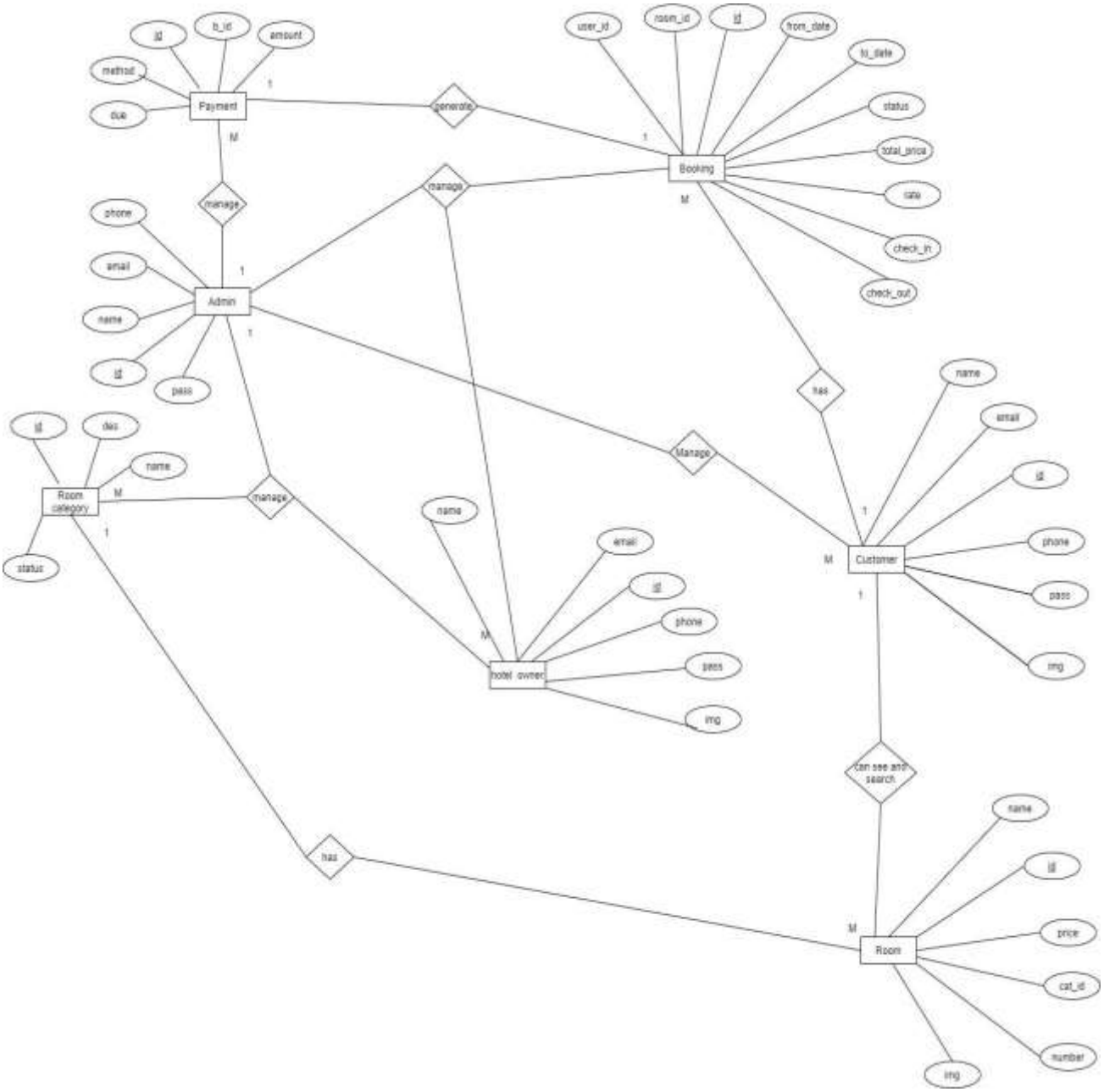


Figure 7. ER diagram

This diagram shows in Figure 6 the relations between the main entities within the hotel management system:

User. The system has two types of users, including the Admin and the Customer, who can manage and interact with the system in different ways and have diverse functions.

Room. The room entity includes data concerning distinct room types and categories as well as their availability in the Hotel. Additionally, it is possible to identify that each Room is usually related to one or several bookings.

Booking. The entity provides data on the customers' reservations, implying the booking date, the number of rooms, and their status in the system, such as whether they are approved, pending, or canceled.

Payment. This entity shows that each Booking is connected with a corresponding payment or the record describing the payment status and method used.

DFD Diagram

Context Level Diagram

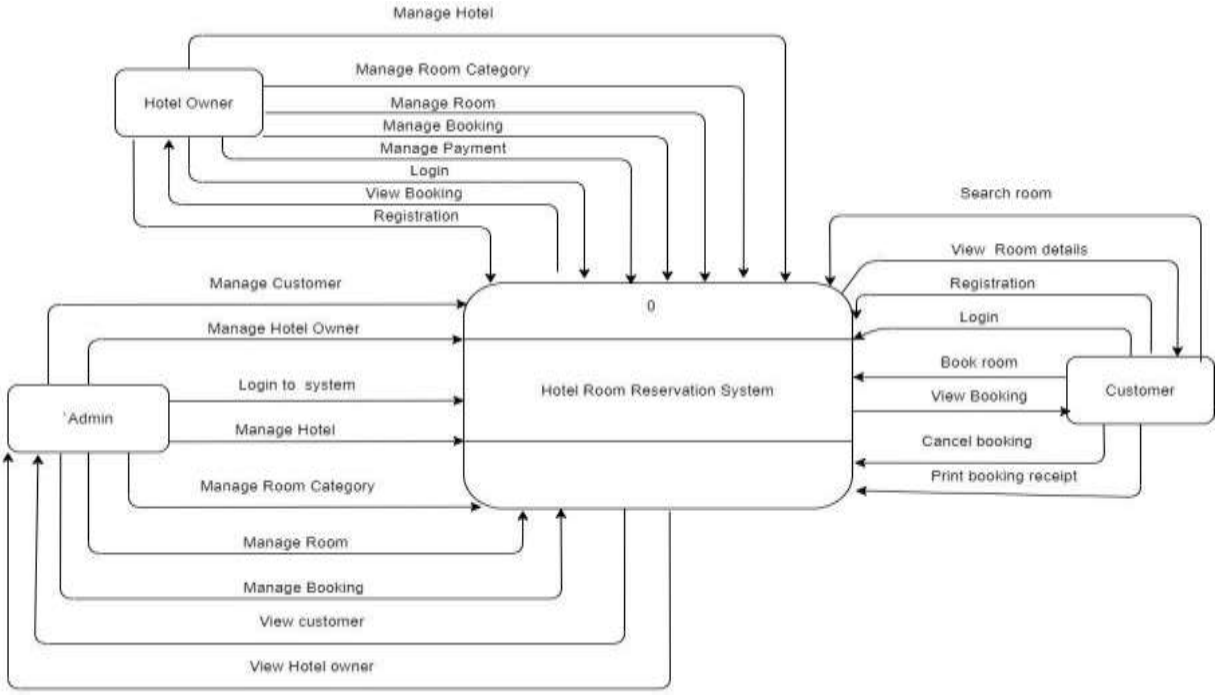


Figure 8. Context Level Diagram of Hotel Room Reservation System

Level 1DFD

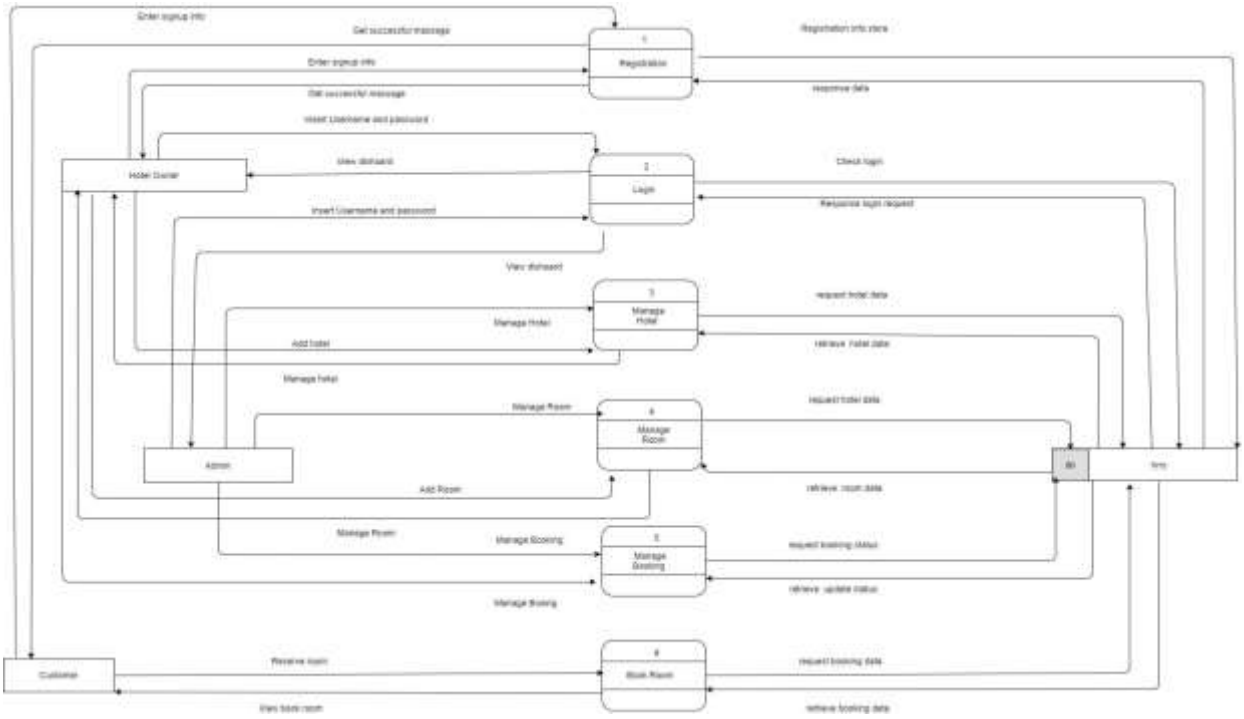


Figure 9. DFD Level-1

Level 2-Process 1 DFD (Registration)

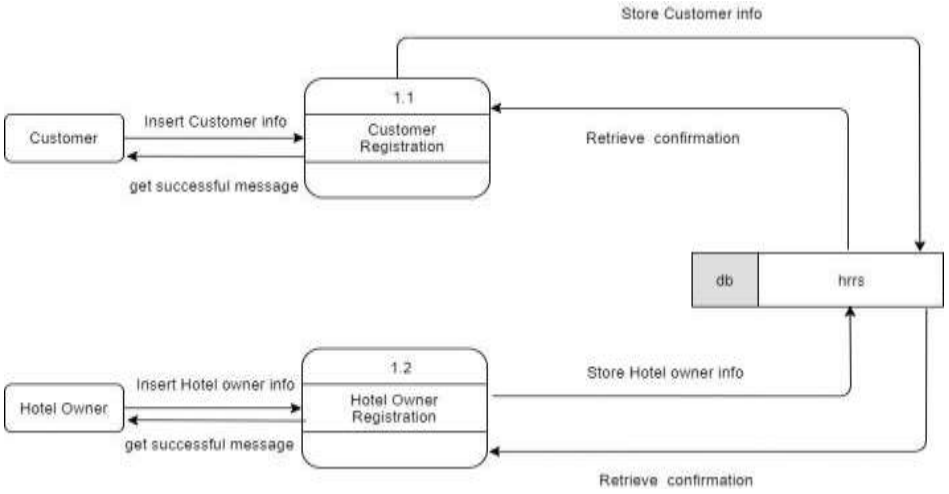


Figure 10. Level-2 Process 1 Registration)

Level 2-Process 2 DFD (Login)

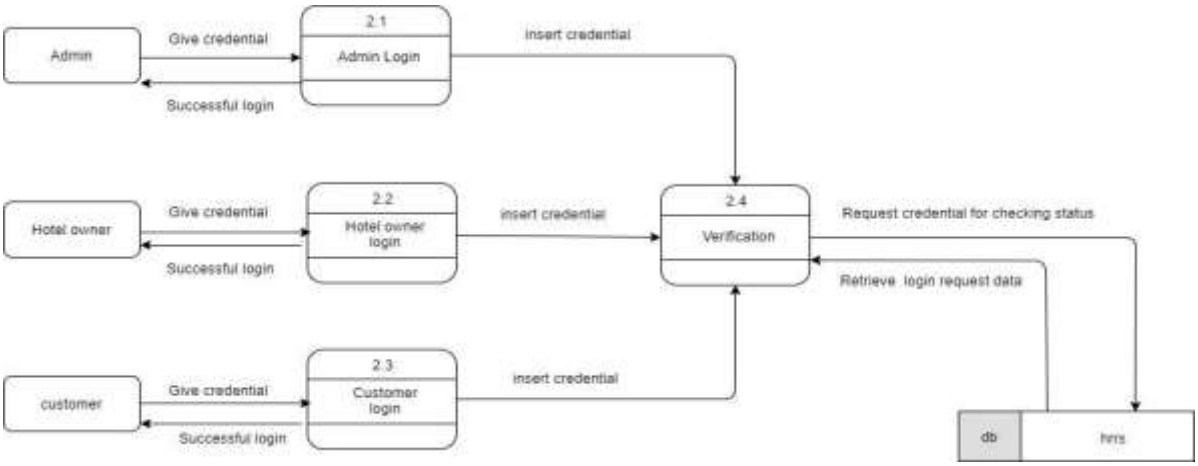


Figure 11. Level 2-Process 2 DFD (Login)

Level 2-Process 3 DFD (Manage Hotel)

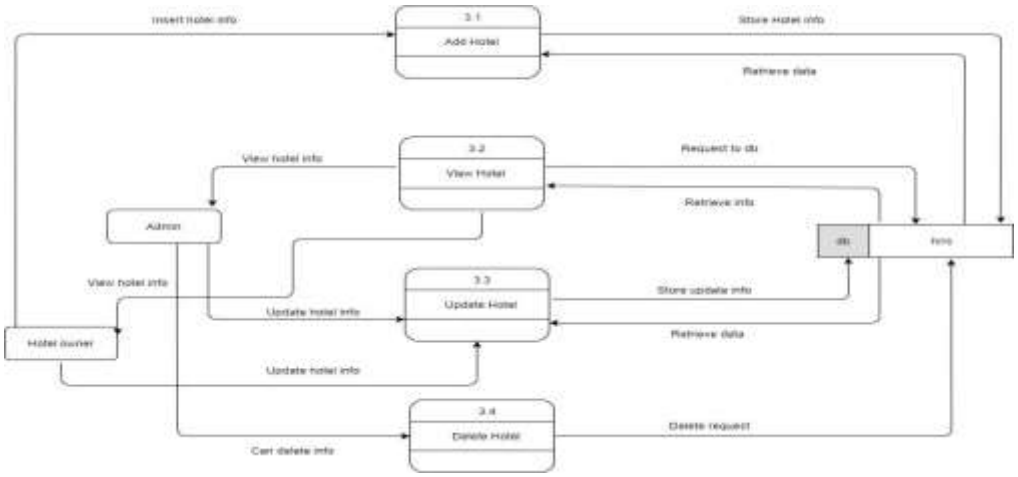


Figure 12. Level 2-Process 3 DFD (Manage Hotel)

Level 2-Process 4 DFD (Manage Room)

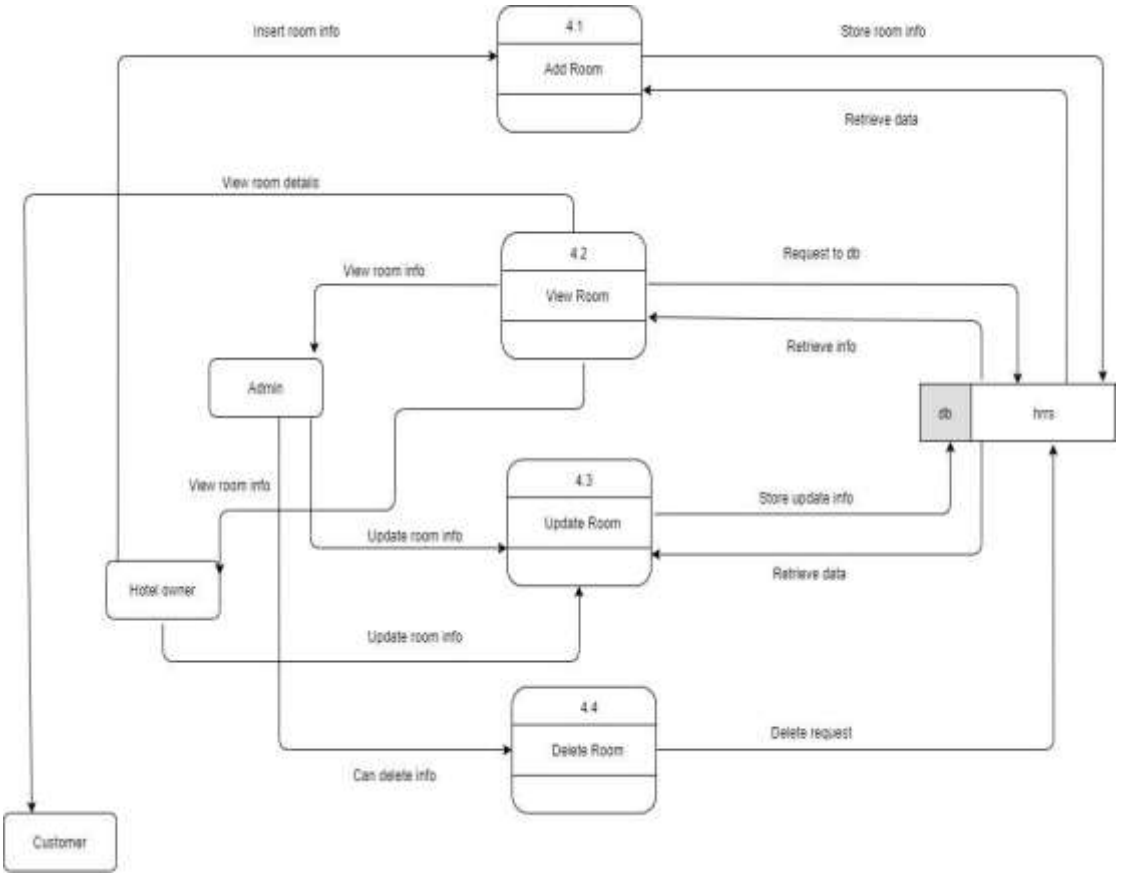


Figure 13. Level 2-Process 4 DFD (Manage Room)

Level 2-Process 5 DFD (Manage Booking)

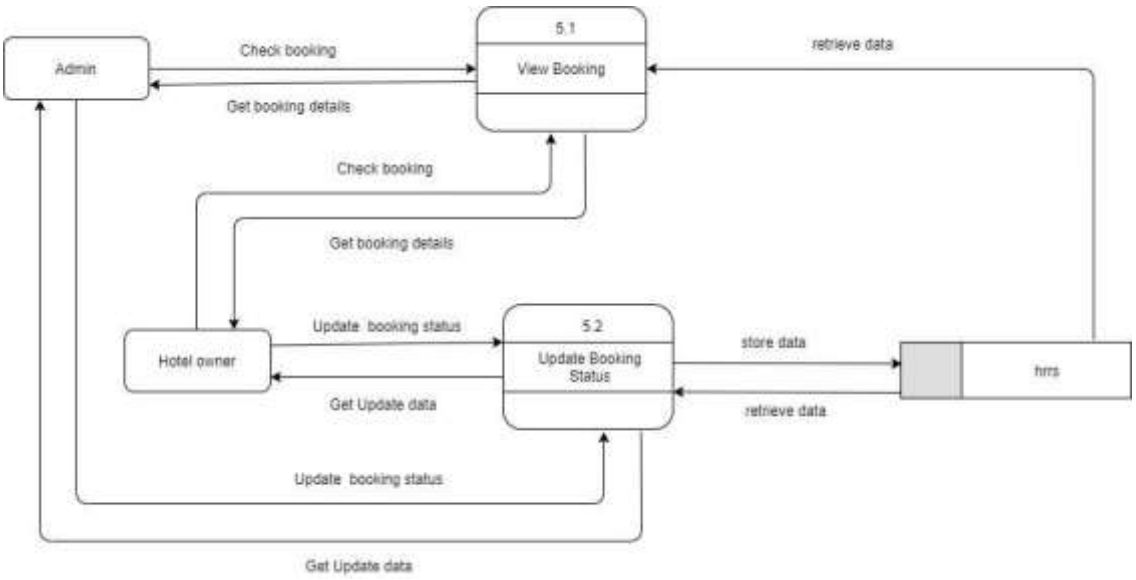


Figure 14. Level 2-Process 5 DFD (Manage Booking)

Level 2-Process 6 DFD (Book Room)

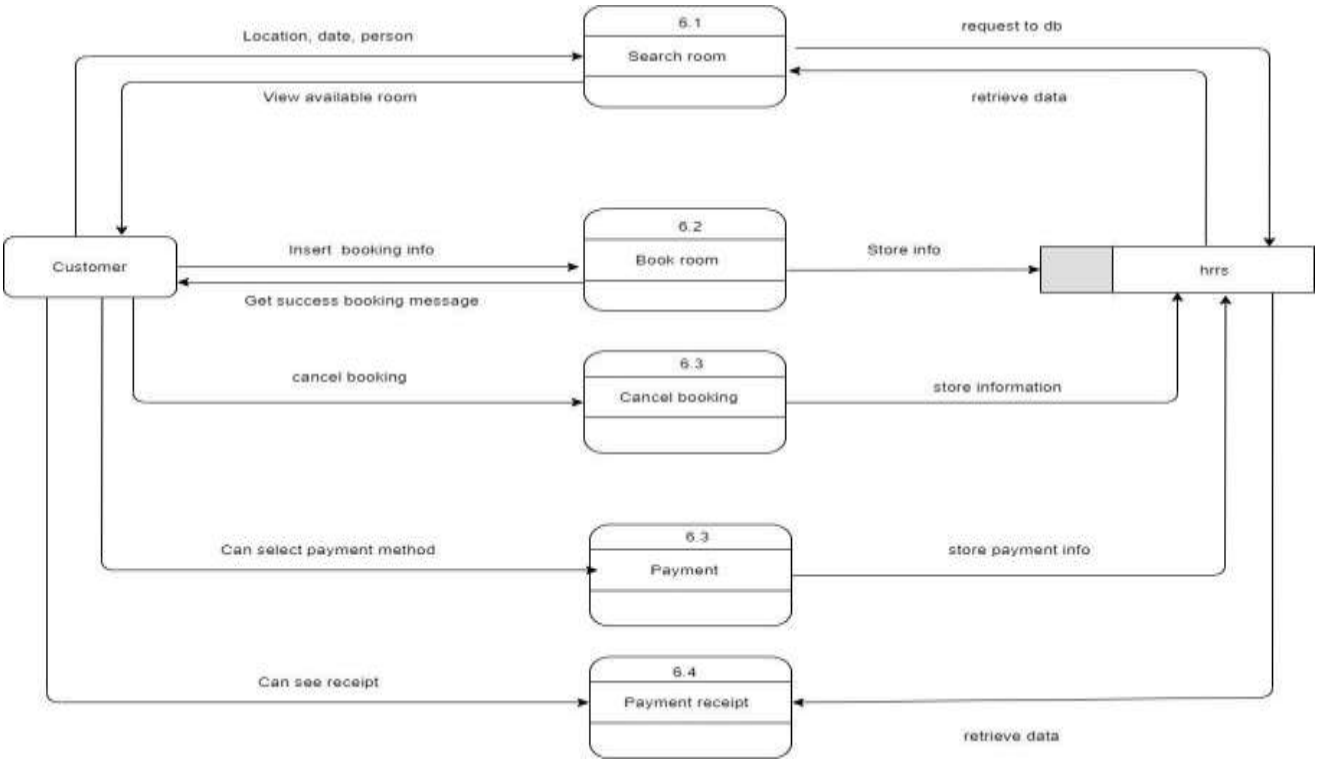


Figure 15. Level 2-Process 6 DFD (Book Room)

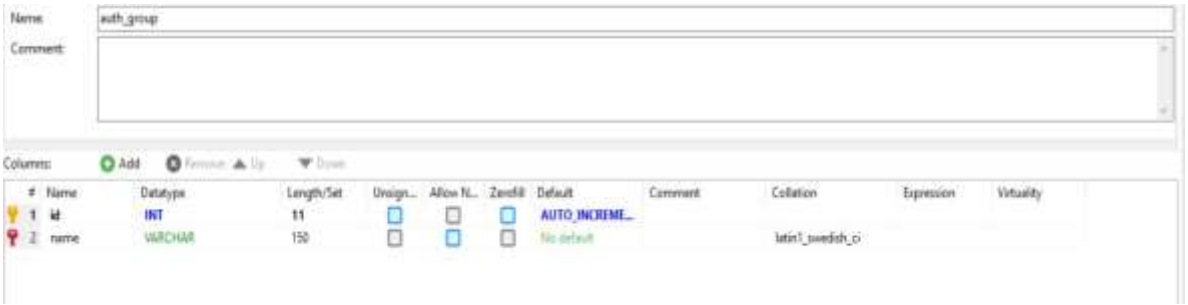
System Design

Requirement analysis provides the software designer with a representation of information, function, and behavior that can be translated into data, architectural, interface, and component-level designs. For our project, the requirement analysis was performed in the following task phases.

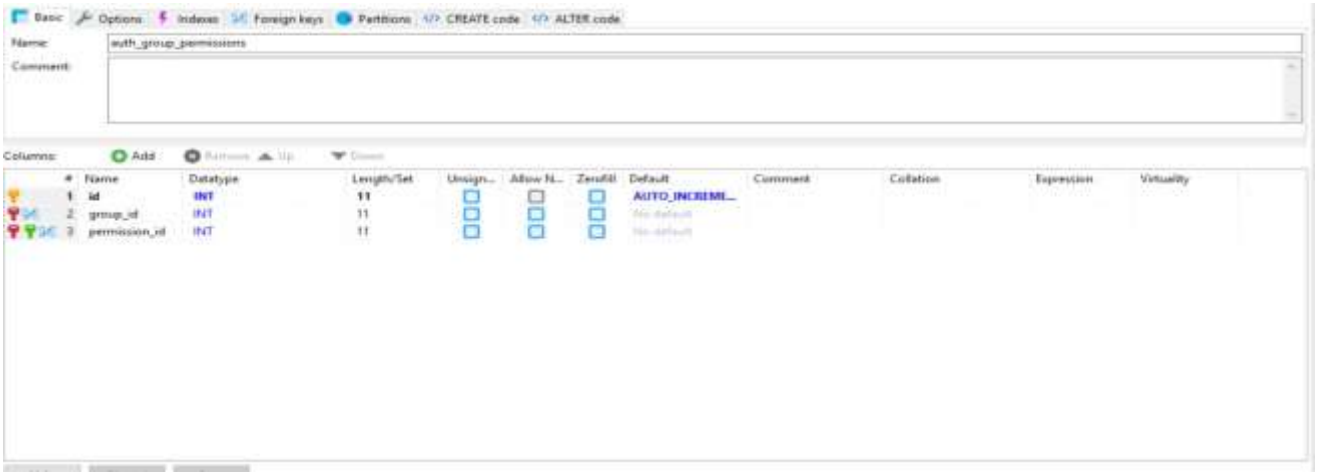
Database field design:

Hours:

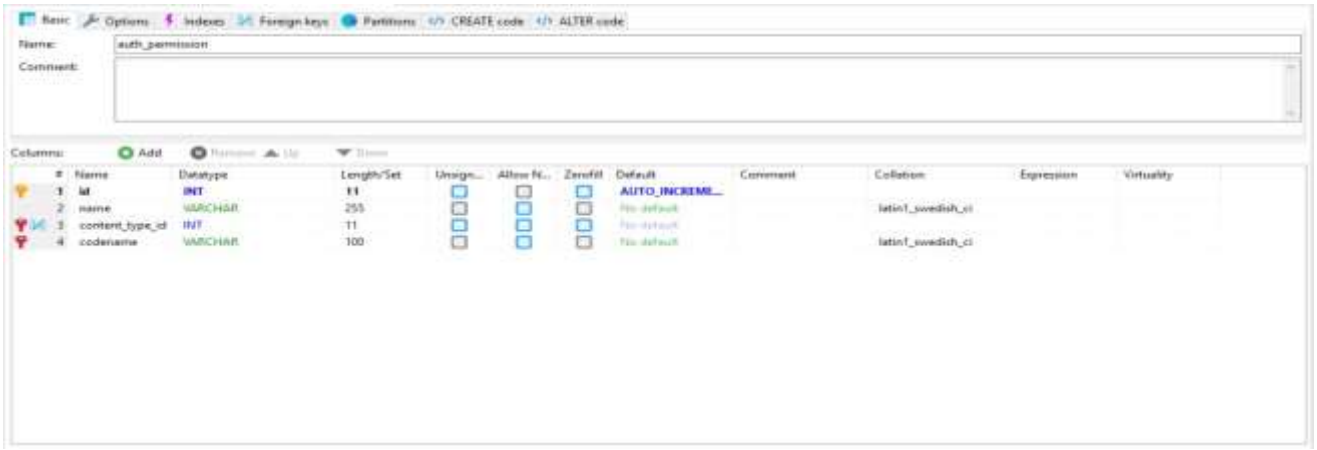
Auth_group:



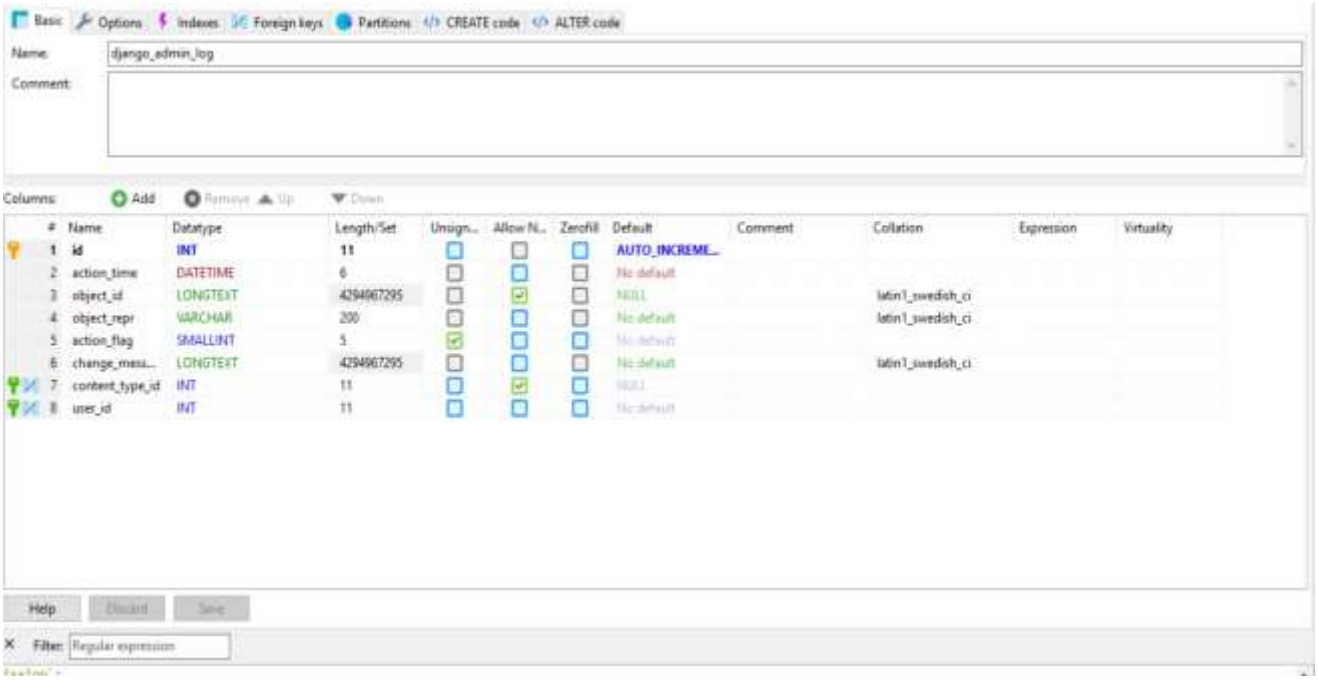
Auth_group_permissions



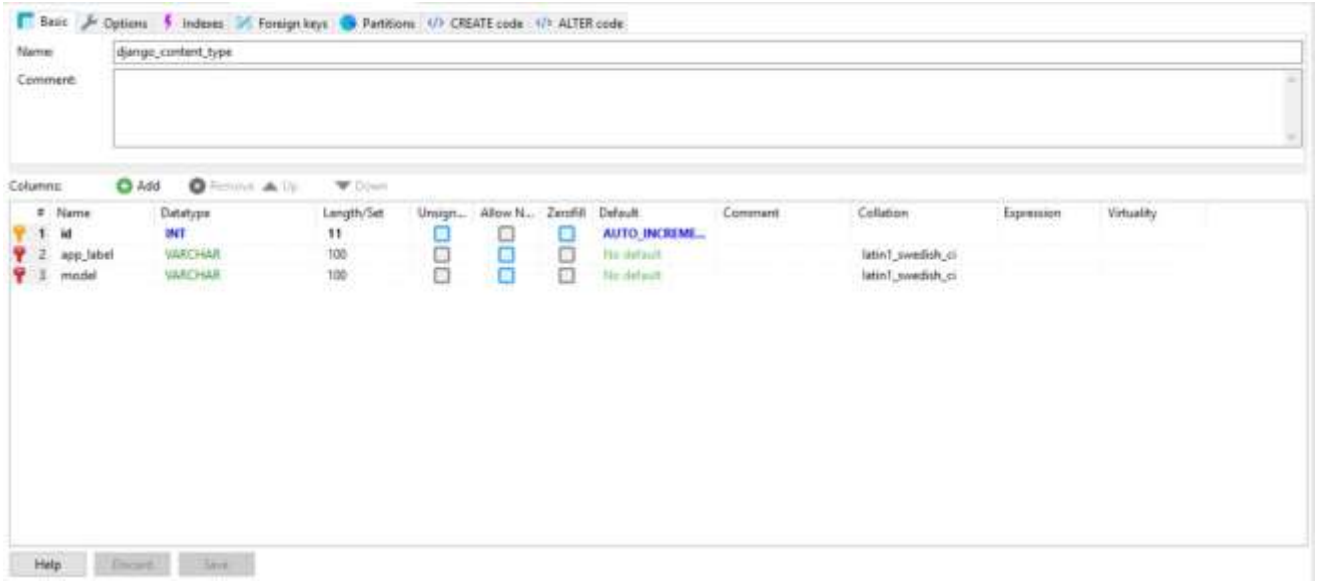
Auth_permission



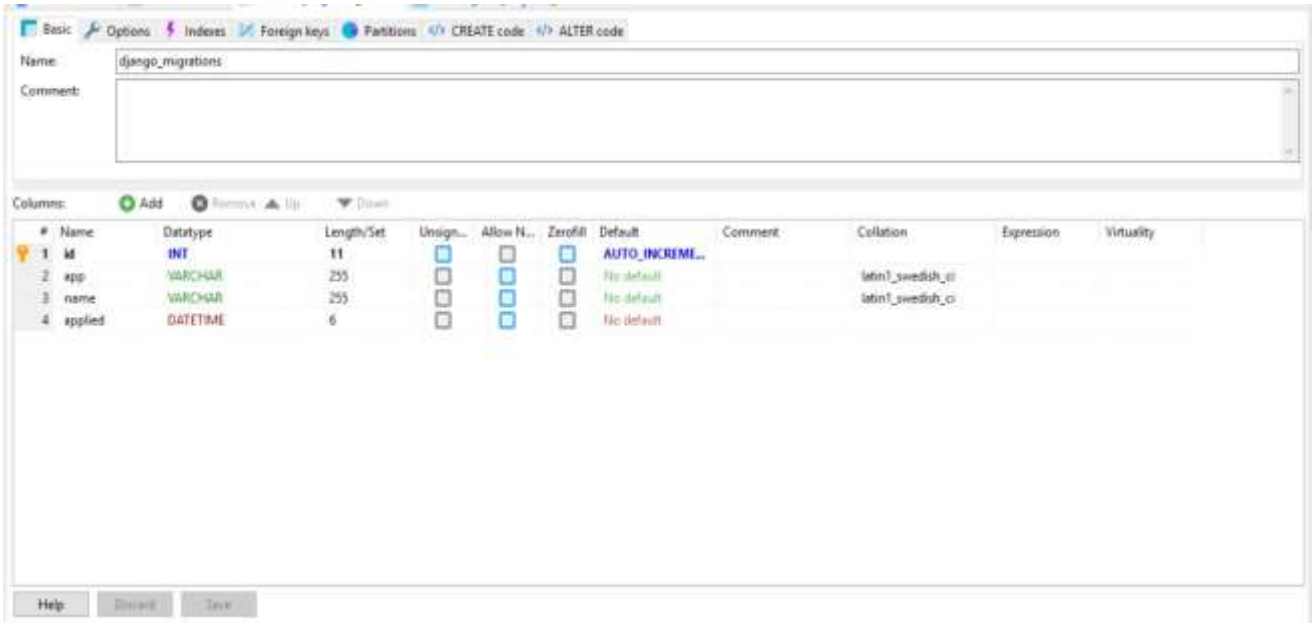
Django_admin_log



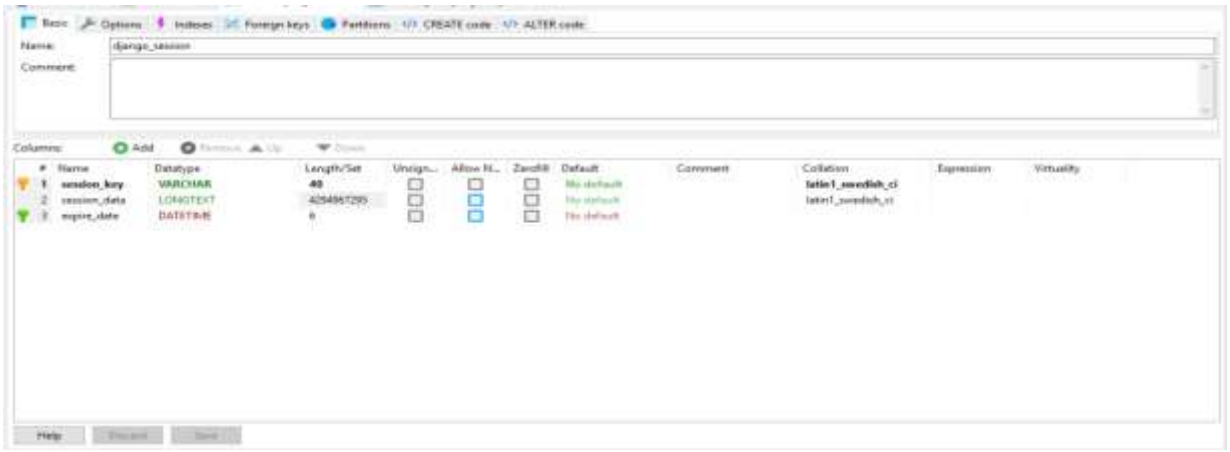
Django_content_type



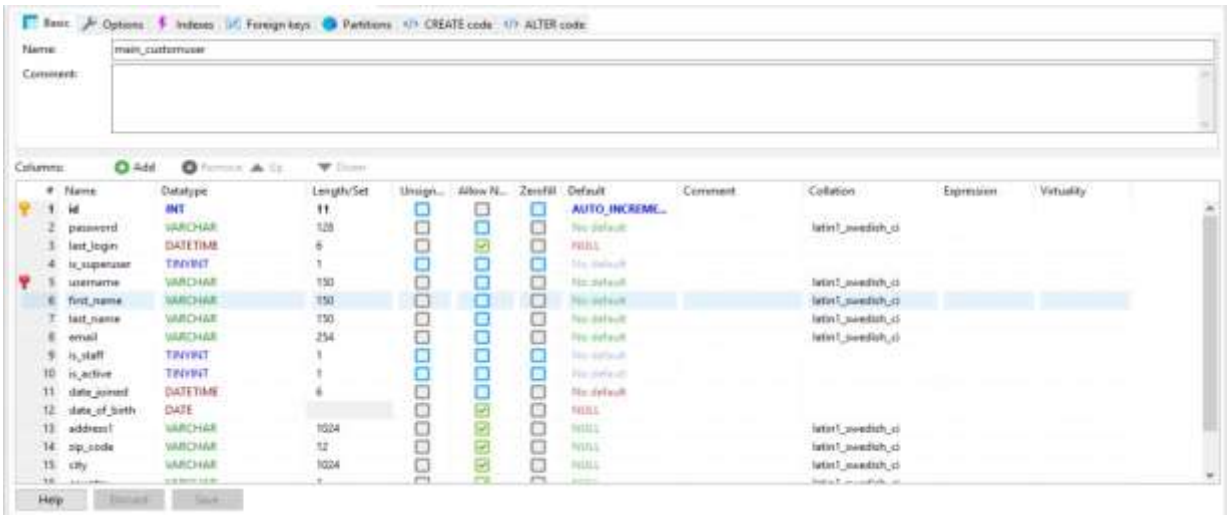
Django_migrations



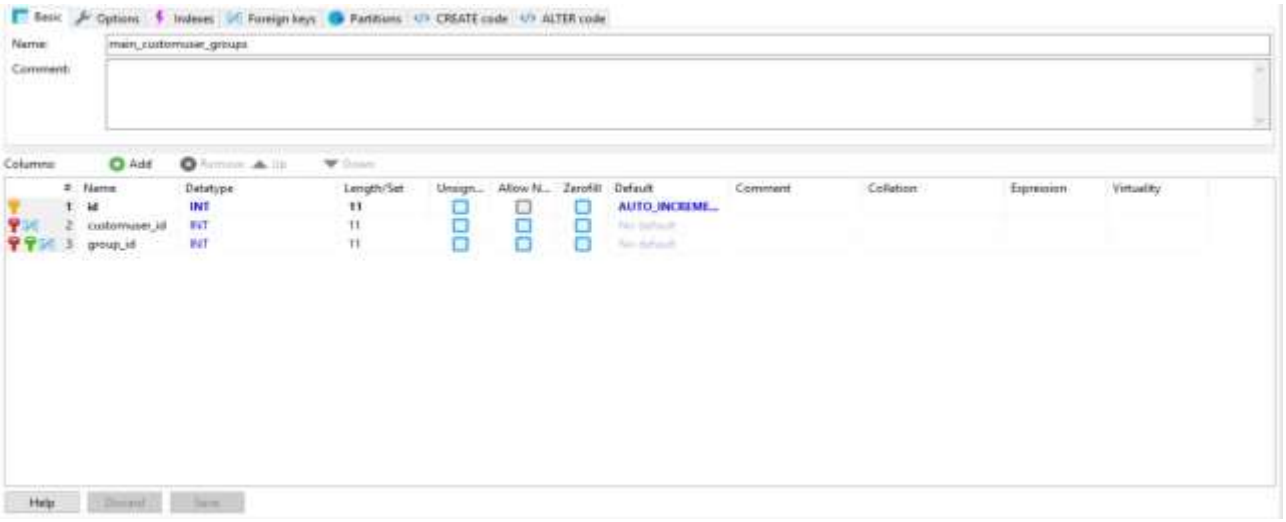
Django_session



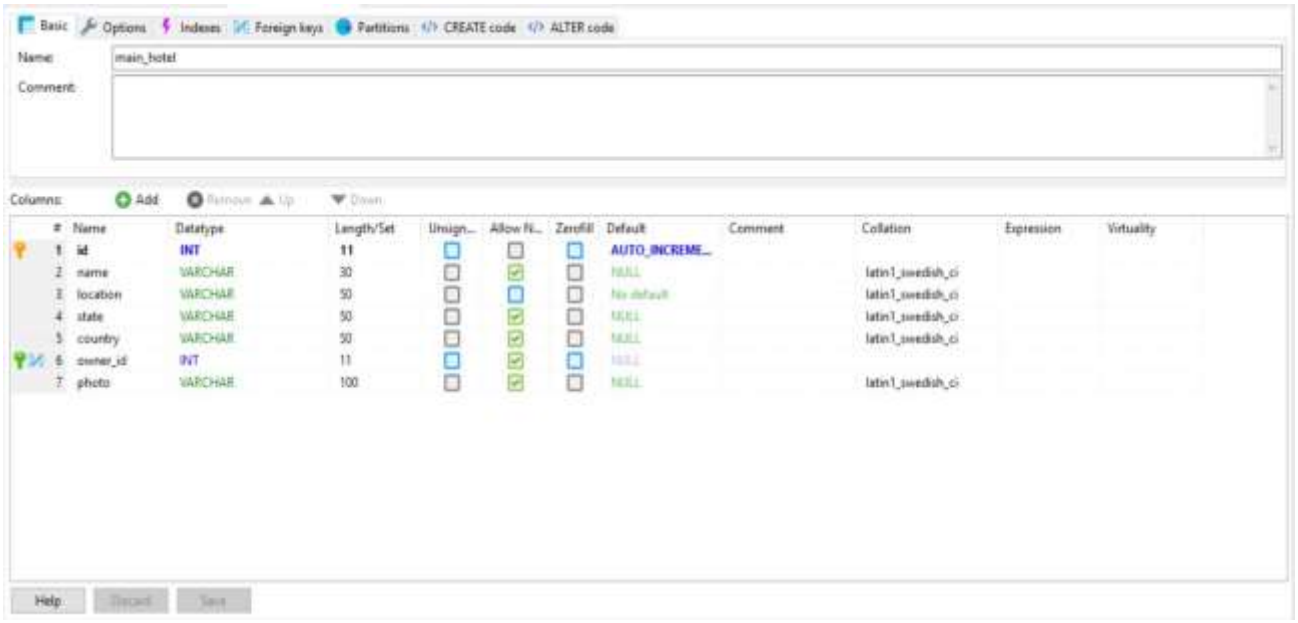
Main_customuser



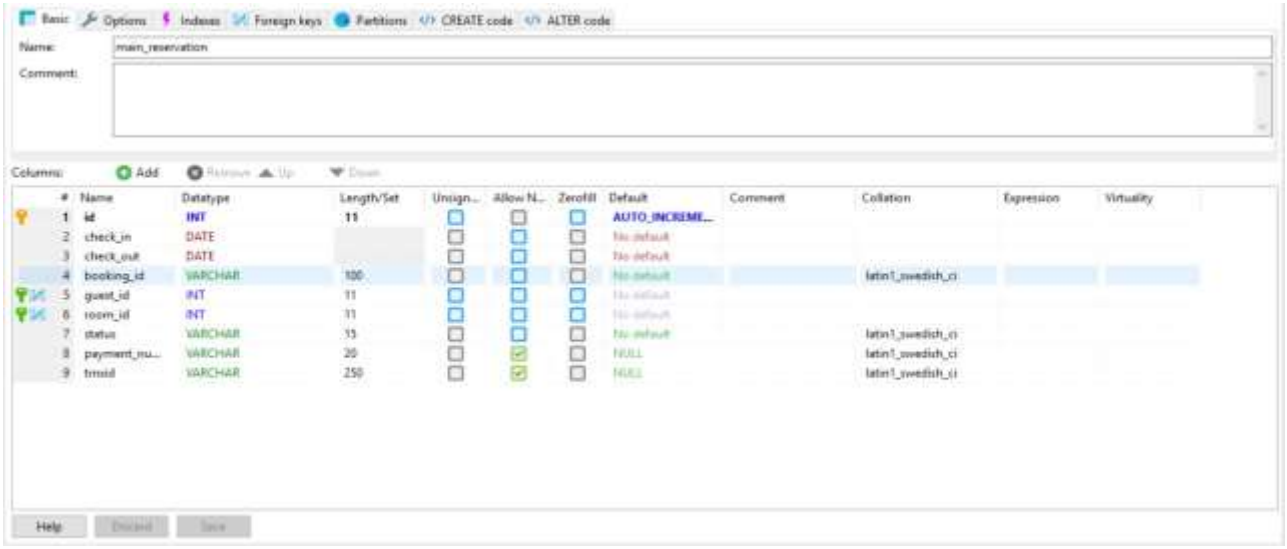
Main_customuser_groups



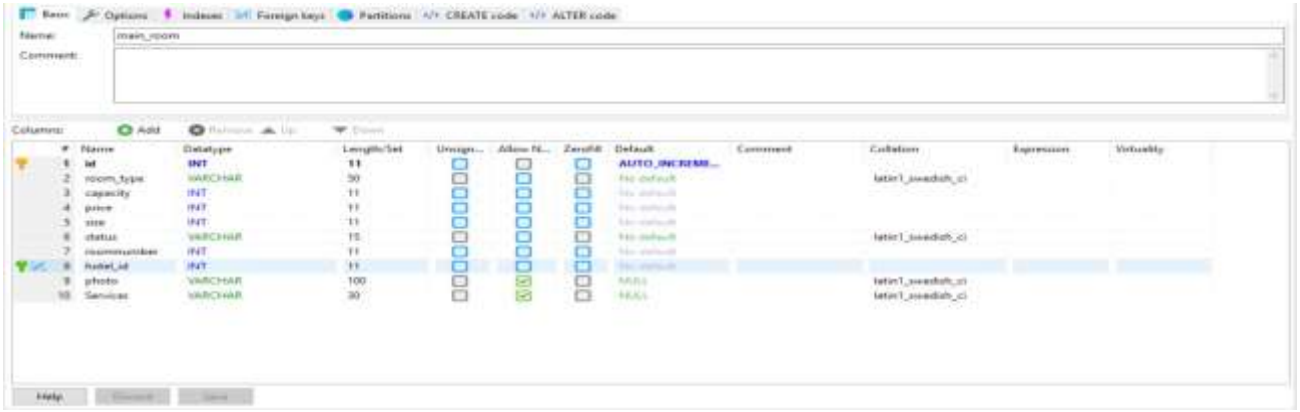
Main_hotel



Main_reservation

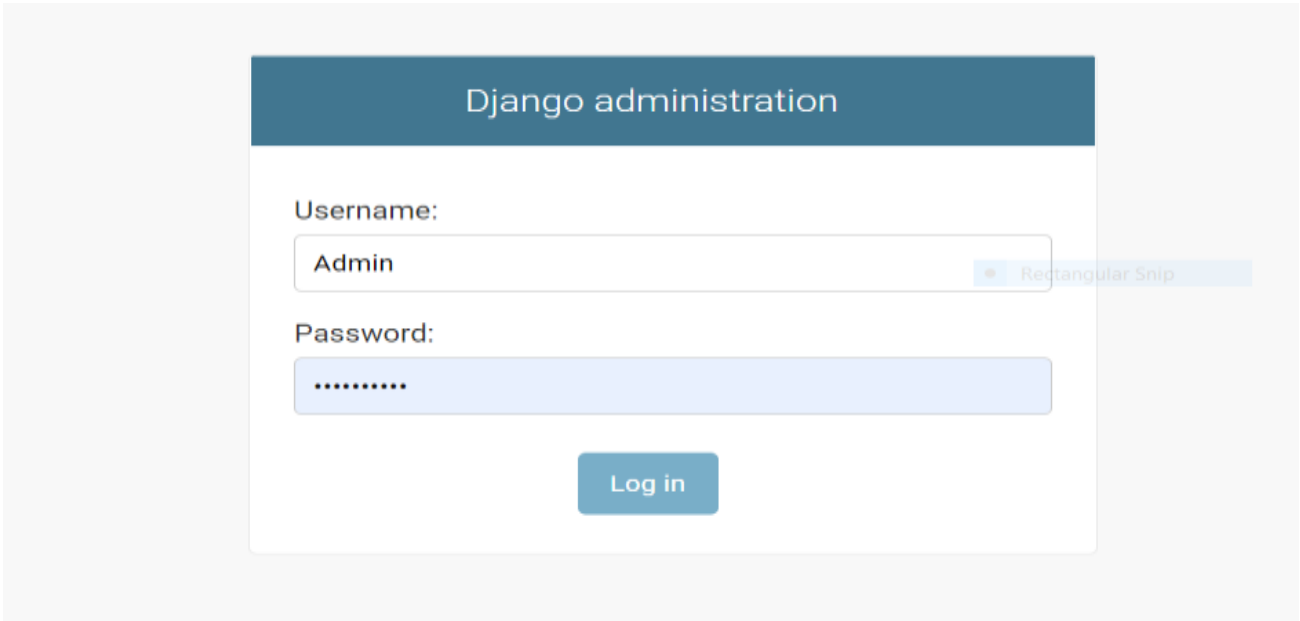


Main_room



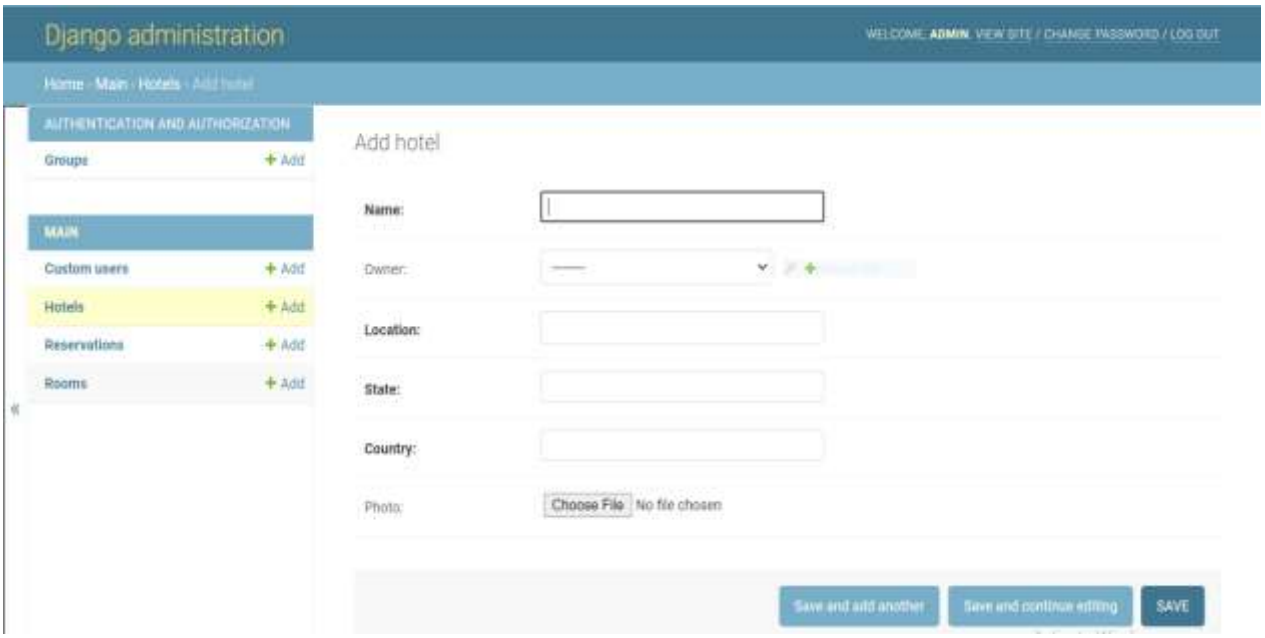
UI design of Admin Panel

Admin Login

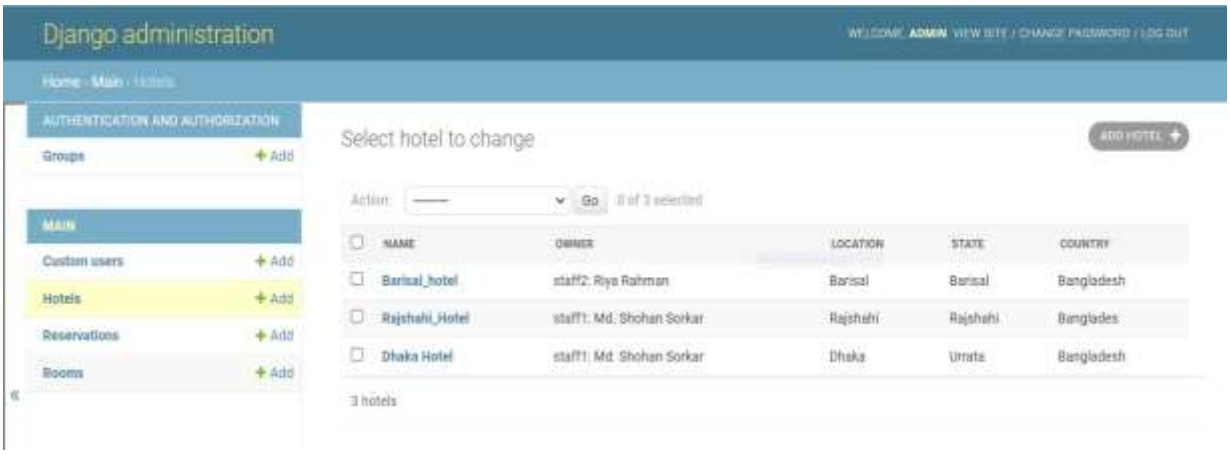


Admin dashboard

Add hotels

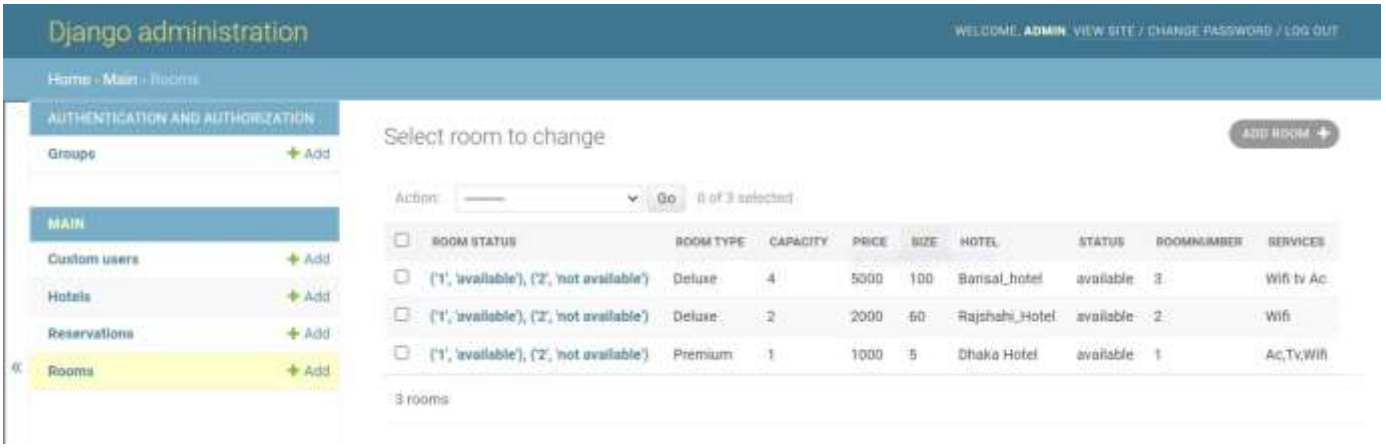


Hotel List

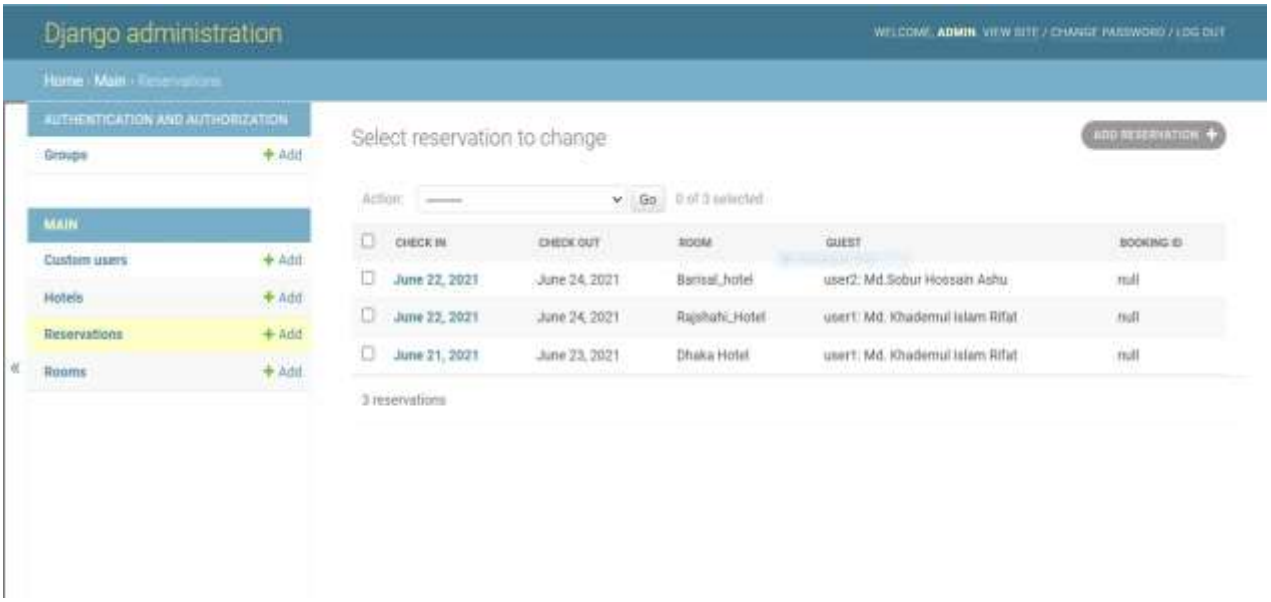


Add Rooms

Room List



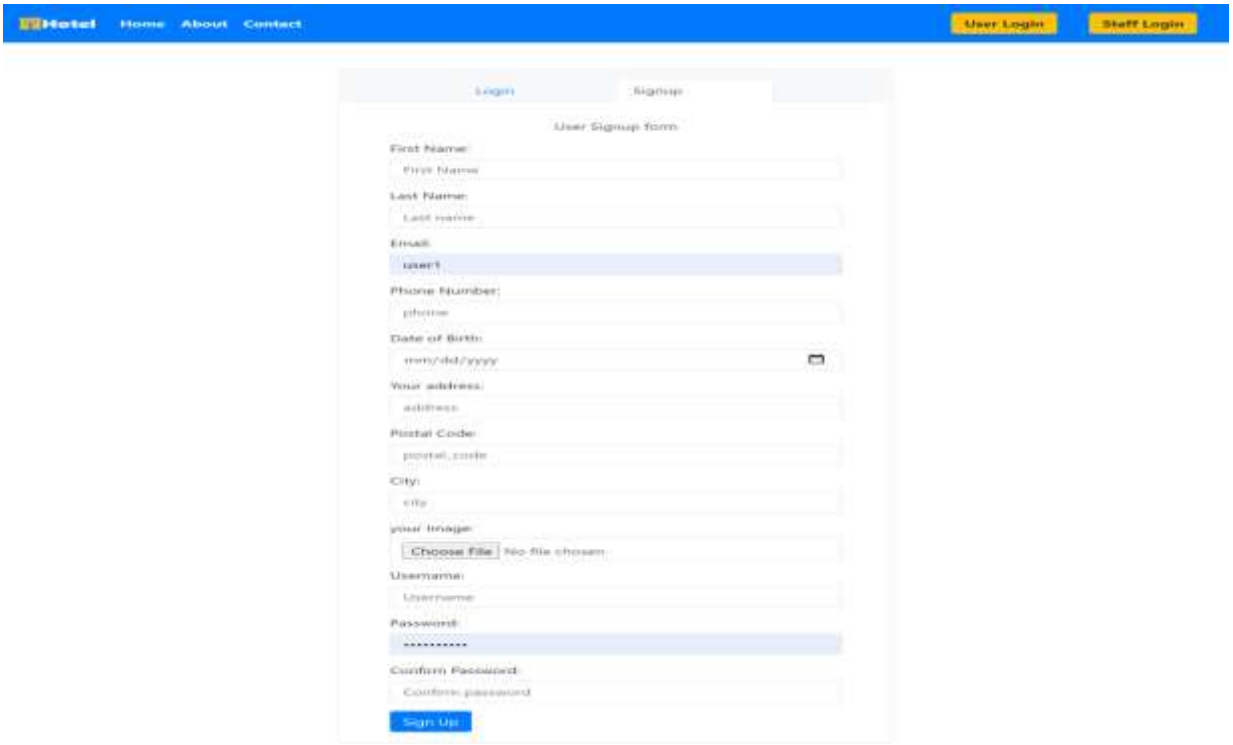
Book List



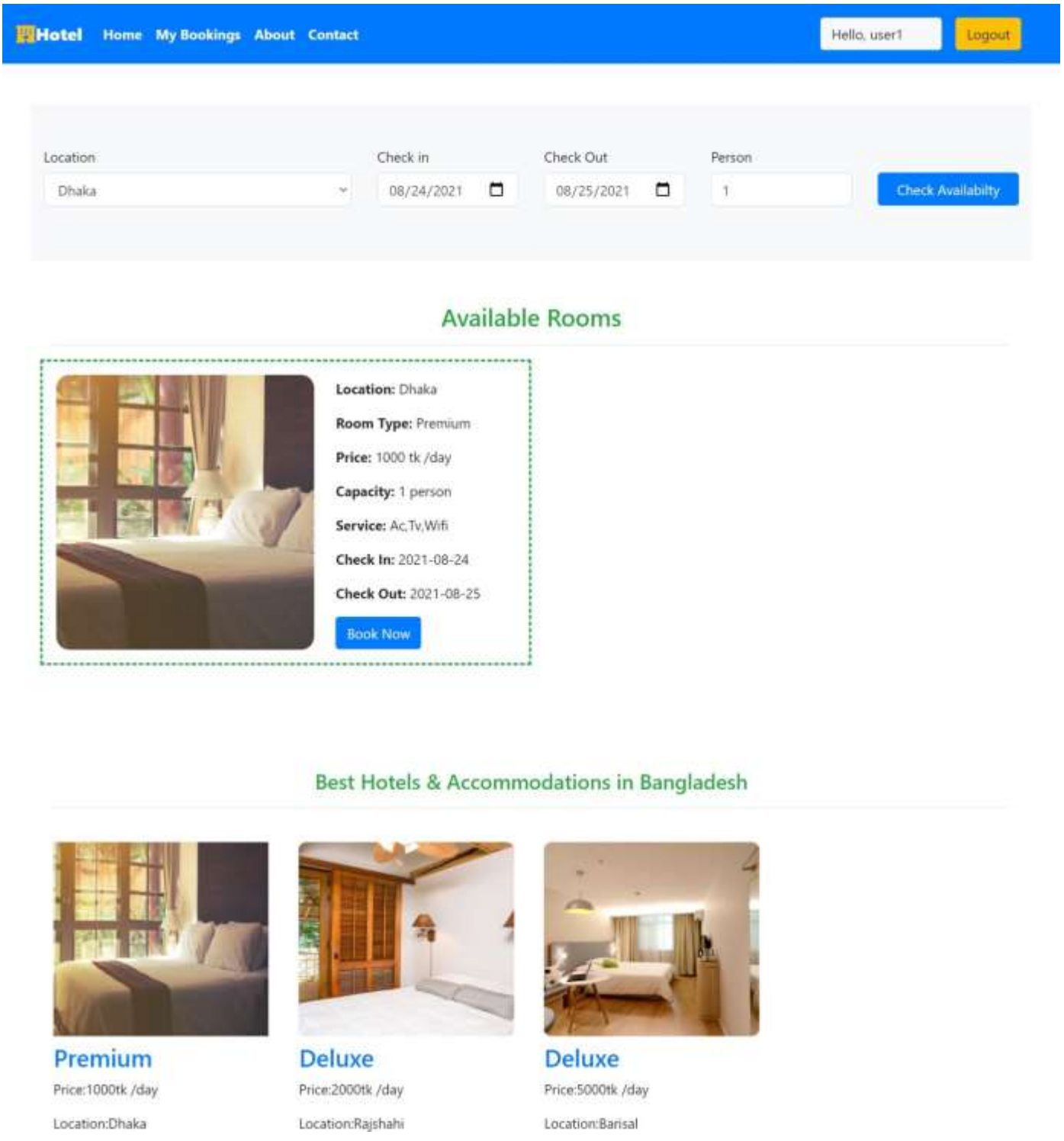
Booking details

User Login

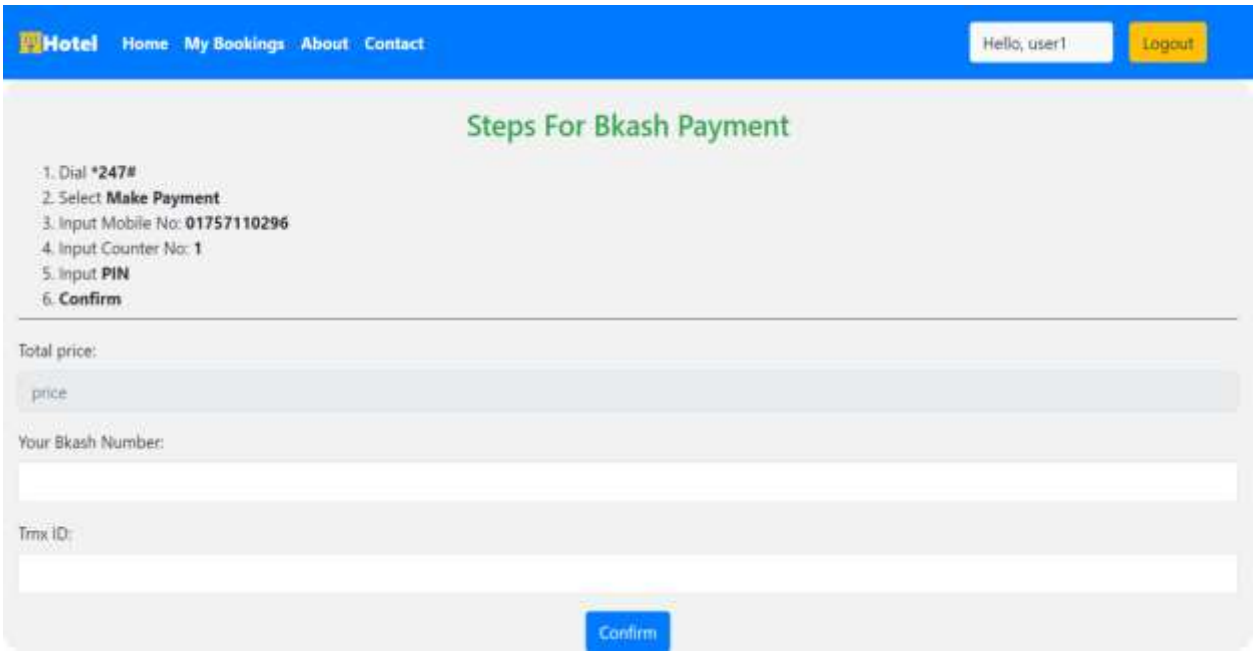
User Registration



User Room Booking



User booking Payment



User booking List

System Testing

System testing is a critical stage in the development of any software. It is the process in which different modules or elements of the Python hotel management system are verified and tested for their proper functioning in the integrated form. The main objective is detecting and removing any defects in the system's working before it goes live. Every aspect of the Python hotel management system is tested, such as the user interface, backend logic, database, and security gates or tests. The objective is to determine that the Python hotel management system meets its functional requirements and works abundantly and reliably.

Testing goals

The main goals of the system testing include:

- Determining that all the functional requirements are met, such as room booking, guest registration, payment processing, etc.
- Determining non-functional requirements, such as security, Performance, and scalability

, are fully implemented.

- Any fault or inconsistency in the system's living is discovered and treated.
- Checking that the system runs as specified in regular and unique circumstances.

Testing Types

Several types of Testing were used in the system testing of the hotel management system to determine that it can function reliably, and the related tests can be repeated with the same results:

- Unit testing.
- Integration testing.
- Regression testing.
- Stress testing.

Unit Testing

This type of Testing is the process in which individual parts or components of the system are tested or verified. The parts are checked on a one-to-one basis in isolation to exclude any mistakes or faults. Following are the main parts or components of the Python hotel management system that are tested:

User Module: In the system, the tests of different aspects of user registration, login, and setup are carried out. Here, the security of the login method is also verified. A test is carried out to verify if login is possible with the wrong password or not. A test is conducted to check if a login is possible without registration. Finally, a test is made on the login to determine if the system can provide the facility to more than one user for registration and login at the same time.

Room Management Module: Here, a test is carried out to ascertain the number of rooms added to the system, as well as the number of additions of room types and categories. A test is also carried out to verify the procedure for room addition and the system's capability to protect rooms of the same type.

Booking Module: In this module, testing ensured users could search rooms, make reservations, and check the status of their Bookings. The system's response to invalid Bookings, for example, when there are no available rooms, was also checked.

Table 26. Unit Testing Process (1)

Testing scenario, No:1	
Scenario	Customer Registration
Input's	Name, Email, phone, password
Desired Output's	When enter all basic info correctly, new Customers will be registered in the system.
Actual Output's	For new customer registration, my system work correctly
Verdict	Getting Results from Desired Outputs and Actual Outputs decided this system is successful for new customers' registration.

Table 26 represents a unit testing flow, where individual system components are isolated and tested. A test case is conducted with specific input values, and the output is noted. The system is then compared with the expected outcomes and adjusted in case of any variance before moving to the next component.

Integration Testing

Integration Testing primarily ensures that the different modules and components of the hotel management system work together. Integration testing is vital since these modules are multiple and interact with one another.

Database Integration: Testing was conducted to ensure that data is correctly transferred between the hotel management system's different front-end interfaces and the backend database. When a customer books a room, the booking modules should be able to execute the transaction and update the database with the booked rooms' data.

Payment Gateway Integration: Integrating the different booking modules and the payment gateway was also tested to ensure that online payments are processed securely. Additionally, the booking status was ensured to be updated accordingly.

Admin and Hotel Owner Dashboard: When testing the booking process of customers into the pages of the admin and hotel owner dashboard interfaces, it was found that the changes of an admin or hotel owner clicking on a particular booking to approve/disapprove the transaction does not change the appropriate column of booking status for a selected booking only.

Table 27. Shows the system integration testing.

Testing scenario, No:2	
Scenario	Customer Login testing scenario of my system
Input's	Email, password of User for Login
Desired Output's	When you enter your Email, password then get access level defines.
Actual Output's	For login, my system works correctly
Verdict	Getting results from Desired Outputs and Actual Outputs decided this system is successful for login.

Functional Testing

Functional Testing aims to verify that the system's core features work as the design provides. It is designed to test basic and advanced functionalities from an end-user perspective. As part of the functionality testing process, we tested the room booking feature, different authentications and their respective privileges, and payment processing.

Room Booking Functionality: We tested the system to confirm that a customer can search for rooms and proceed to book. The system was also tested to verify its capacity to handle alternative booking situations, such as when all rooms are already booked. We also tested the alternation user requirements following a case of a sudden cancellation. After the test, we confirmed that the system was working as it should.

User Authentication and Role-Based Access: We also tested the system to confirm that users, admins, and hoteliers can log in and access the various parts of the system as required. Here, we confirmed that the users could perform different system privileges, such as managing user accounts and room types for the admins. For the hoteliers, we confirmed that the system could allow them to manage their bookings and effectively show their availability.

Payment Processing: We tested the payment feature to confirm that customers can pay for the booked rooms. We also had to confirm that the system can securely accept the various methods of payment and systematically deal with payment failures. The system was also tested to confirm whether it can effectively show account statements to both parties and later update the booking details after payment.

Table 28. Functional Testing Process

Testing scenario, No:3	
Scenario	Admin can add, view, update, and delete categories, types, facilities, and rooms.
Input's	Request to manage categories, types, facilities, and rooms.
Desired Output's	Show the information to the Admin
Actual Output's	My system works correctly in showing all records.
Verdict	The process is worked correctly and successfully.

Table 28 shows the functional testing process, which involves simulating user adoption of the core features of Testing. All the user functions were formalized, and various system requirements were used to test these features.

User Acceptance Testing

User Acceptance Testing (UAT) is the final phase of Testing, during which the system is tested from the end-users' perspective. The goal is to ensure that the system meets the needs of its intended users and is ready for deployment.

Customer Testing: End users How? (customers) were asked to test the room booking process – from searching for available rooms to making a payment and receiving a booking successfully. The usability of the customer interface and the clarity of error messages were tested. Admin and Hotel Owner Testing: Admins and hotel owners test their dashboards to ensure they can efficiently manage rooms, bookings, and users. In addition, the ability to generate and access reports and summaries to provide time-critical information for decision-making was tested.

Table 29. User Acceptance Testing Cycle 4

	Testing scenario, No:4
Scenario	Admin can add, view, and cancel Booking for users
Input's	Request to Manage Booking
Desired Output's	Show information to the Admin
Actual Output's	My system works correctly when showing all booking lists and details.
Verdict	The process is worked correctly and successfully.

Table 29 refers to the UAT cycle, during which users interact with the system to confirm and validate that all functionalities are user-friendly, intuitive, and work. User inputs are noted, and final change requests are implemented and checked for the last time before the system goes live.

Security Testing

Security Testing: This is to verify that the system's data and user information are secure from potential vulnerabilities and threats. The hotel management system is tested to ensure that no data breaches, unauthorized accesses, or hacking attacks can occur.

Authentication and Authorization Testing: This ensured that only authorized people could access selected functionality and data based on their roles. Strong password policies and account lockouts were enforced to prevent unauthorized access.

CONCLUSIONS

The goal of this study is to create an automated hotel management system using python to solve the operational inefficiencies within the hospitality industry involving rooms reservations, payment processing, and administrative management. The research addressed the scientific problem of how to combine advanced technology into a scalable, real-time, and user-friendly solution for streamlining hotel operations. These results indicates that the system, as presented, achieves its goals, key features being: real-time booking updates, role-based access control, and secure payment gateways. The system was thoroughly tested with unit, integration, functional, and security tests ensuring reliable operation, scalability and usability and therefore a practical tool to manage hotel operations. A modular system further guarantees ease of maintenance and adaptation for any future progress in the field. This research provides a decomposing of classical hotel management modes with a transition on technological trends. The work lays the theoretical groundwork for ensuring robust authentication, real-time data processing, and secure transactions in hotel management systems. From a management perspective, this system allows hotel managers to optimize their operations, boost customer satisfaction, and make better-informed decisions thanks to real-time data analysis. While the study achieves worthy goals, it has limitations. The system's testing was performed outside, under controlled conditions, and would need to be validated further in a variety of real-world conditions. Moreover, the existing version of the app addresses only essential functionalities, with many opportunities for improvements, including advanced analytics, integration with third-party reservation platforms, and AI-driven personalization features. Future work could explore expanding the system's capabilities to be used with dynamic pricing algorithms, multilingual front-ends, or IoT-enabled intelligent rooms. Research can also examine the efficacy and adaptability of the system at large-scale implementations or different geographical regions. These enhancements would make the system even more a complete package for modern hotel management.

Author Contributions: Conceptualization, M.A., M.S.R., M.A.H.R., M.B.U. and M.M.H.S.; Methodology, M.A.; Software, M.A.; Validation, M.A.; Formal Analysis, M.A.; Investigation, M.A.; Resources, M.A.; Data Curation, M.A.; Writing – Original Draft Preparation, M.A., M.S.R., M.A.H.R., M.B.U. and M.M.H.S.; Writing – Review & Editing, M.A., M.S.R., M.A.H.R., M.B.U. and M.M.H.S.; Visualization, M.A.; Supervision, M.A.; Project Administration, M.A.; Funding Acquisition, M.A., M.S.R., M.A.H.R., M.B.U. and M.M.H.S. Authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the fact that the research does not deal with vulnerable groups or sensitive issues.

Funding: The authors received no direct funding for this research.

Acknowledgement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to restrictions.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

Delizo, R., & Esguerra, E. A. (2013). Functionality survey of hotel online reservation system in the Philippines: Issues and recommendations. *International Journal of Information Technology and Business Management, 10*, 24-34.

Delizo, G. A., & Esguerra, M. A. (2013). Online hotel reservation and management system for the College of International Tourism & Hospitality Management. *International Journal of Computers & Technology, 10*(1), 1201-1229.

Hu, L., & Gu, Y. (2013). The impact of online reservation systems on the consumer's decision. *Web*.

Hu, P., & Gu, D. X. (2013). Development and implementation of web-based online hotel reservation systems. *Applied Mechanics and Materials, 347*, 2947-2951.

Kim, W. G., & Kim, D. J. (2004). Factors affecting online hotel reservation intention between online and non-online customers. *International Journal of Hospitality Management, 23*(4), 381-395.

Kim, Y., & Kim, K. S. (2004). The effects of structural and infrastructural interactivity on initial trust in a web retailer. *Information and Management, 41*(4), 377-397.