VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018

DBMS LABORATORY WITH MINI-PROJECT REPORT

On

HOTEL MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirement
for the award of the degree of

Bachelor of Engineering
in
Information Science & Engineering
by

Chandana S M (1BG201S015)
Ranjana B K (1BG20IS039)
Shweta K (1BG20IS054)

Under the guidance of
Mrs. Huda Mirza Saifuddin
Assistant Professor

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by AICTE

Department of Information Science and Engineering

2022 – 2023

B.N.M. Institute of Technology

An Autonomous Institution under VTU

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the Mini-project entitled **HOTEL MANAGEMENT SYSTEM** is carried out by **Ms. Shweta K[1BG20IS054]**, **Ms. Chandana S M [1BG201S015]** and **Ranjana B K [1BG20IS039]** the bonafide students of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project prescribed for the said Degree.

Mrs. Huda Mirza Saifuddin
Assistant Professor,
Dept. of ISE
BNMIT

Dr. S. Srividhya
Professor & Head,
Dept. of ISE
BNMIT

Name & Signature of the Examiners with date:

1.

2.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The hotel management system is a system that provides us to reserve rooms, check whether the rooms are vacant or not, etc by using online browsing. This system is very useful to all, especially business people.

Business people who don't have sufficient time for these can use these types of online Hotel Management Systems. Through this project, we will reduce the faults in bills of their expenditure and decrease the time of the delay to give the bills to the customers. We can also save the bills of the customer. Through this project, we can also include all the taxes on the bills according to their expenditures. It has a scope to reduce the errors in making the bills. Computerized bills can be printed within a fraction of a second. Online ordering of Booking is possible by using this software.

We are confident that this software package can be readily used by non-programming personnel avoiding the human-handled chance of error. This project is used by online users and administrators(management of the hotel).

1.1 Objective

The main objective of the entire activity is to automate the process of day-to-day activities of the hotel:

- Room activities
- Admission of a new customer
- Assign a room according to the customer's demand
- Checkout of a computer and releasing the room
- Finally compute the bill
- Packages available
- Advance online bookings
- Online cancellation

The project has some more features:

- System connectivity
- No data duplication
- No paperwork is required
- Time efficient
- Cost efficient

- Automatic data validation
- User-friendly environment

These objectives are explained in further sections of the report.

1.2 Scope of the Project

In the present time, there is a great rush in hotels as these have become a necessity for the middle and upper class of society. People travel a lot, stay in hotels, goes to the hotels for functions, meetings and refreshments. Our project is developed keeping in mind the general needs of the customers when he goes to the hotel. The hotel management project provides room booking, staff management and other necessary hotel management features. The system allows the manager to post available rooms in the system. Customers can view and book a room online. Admin can either approve or disapprove the customer's booking request.

1.2 Motivation

We often experience a situation where we have an overload of data. In the setting of a hotel, there are numerous entities and attributes one needs to keep track of to run the hotel in a hassle-free manner. The hotel management project provides room booking, staff management, and other necessary features. The system allows the manager to post available rooms in the system. Customers can view and book rooms online. Admin can either approve or disapprove the customer's booking request.

1.4 Requirement Analysis

1. Channel Manager

You can list property information, including room rent, availability, inventory and descriptions across multiple sales channels, OTAs and GDS. Get an overview of every post’s status and engagement and make quick changes from a dashboard. You can avoid double bookings and other manual errors by automatically adjusting distribution for vacant, occupied or under-maintenance rooms.

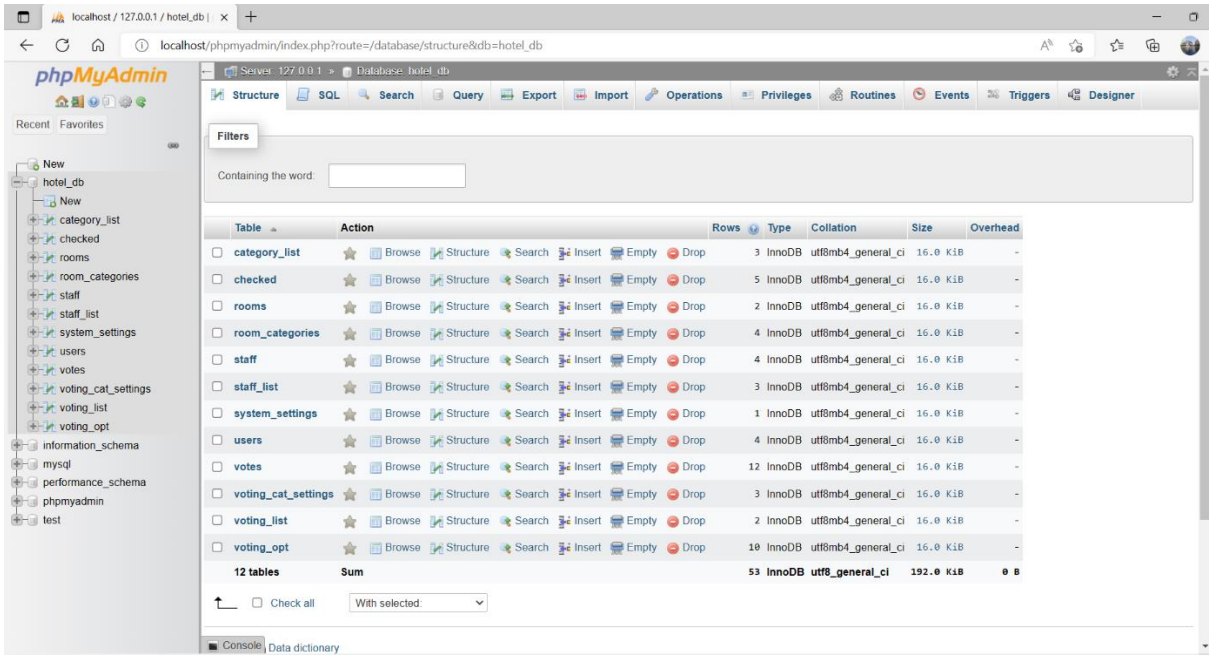

Figure 1.1: Channel manager

- Inventory allocation
- Selective pricing and availability
- Real-time sync and updates
- GDS connections

2. Front Desk Management

This module lets you view and update room bookings, generate invoices, process payments and manage guest check-ins and outs. With an interactive calendar and drag-and-drop tools, you can track reservations (both current and upcoming), schedule housekeeping, monitor room status, and perform night audits and shift audits.

You can connect the system with your bank accounts and preferred payment gateways to create a connected payment and check-out experience for your guests. Some platforms also offer inventory tracking and search tools, enabling your staff to have a definite answer whenever guests request services, food, information or products.

- Invoicing
- Payment processing
- Commission calculator
- Guest profiles
- Integrated calendars
- Customizable checklists
- Digital check-ins and check-outs
- Drag-and-drop reservations
- Front desk dashboard

- Search tools

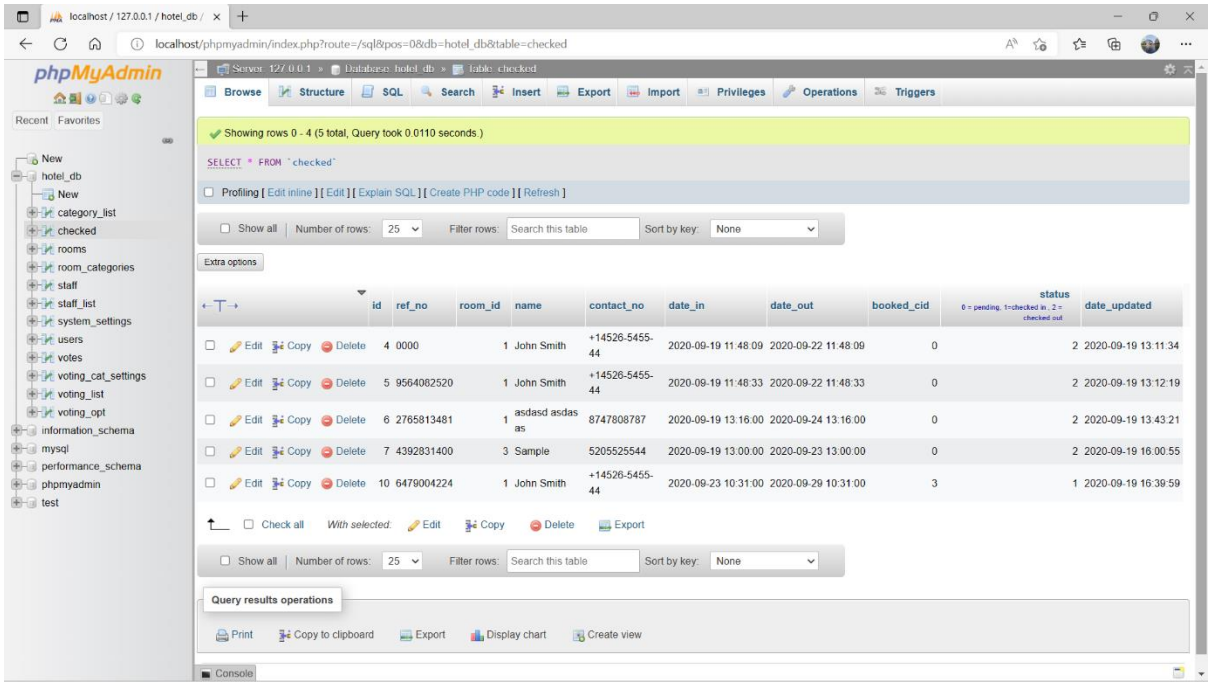

Figure 1.2: Front desk management

3. Groups and Allotments

If you plan on hosting an event or a group of guests at your hotel, this feature set becomes relevant, not to mention crucial. You can block rooms, export room lists and configure billing settings to bill a single account for multiple rooms.

You can get an overview of bookings, equipment, meeting rooms, facilities and event information coupled with guest profiles and history. Advanced platforms can also streamline corporate bookings by directly sending invoices to the parent company and supporting custom pricing for repeat customers.

- Configure cut-off dates
- Import rooming lists
- Dynamic room rates
- Block rooms

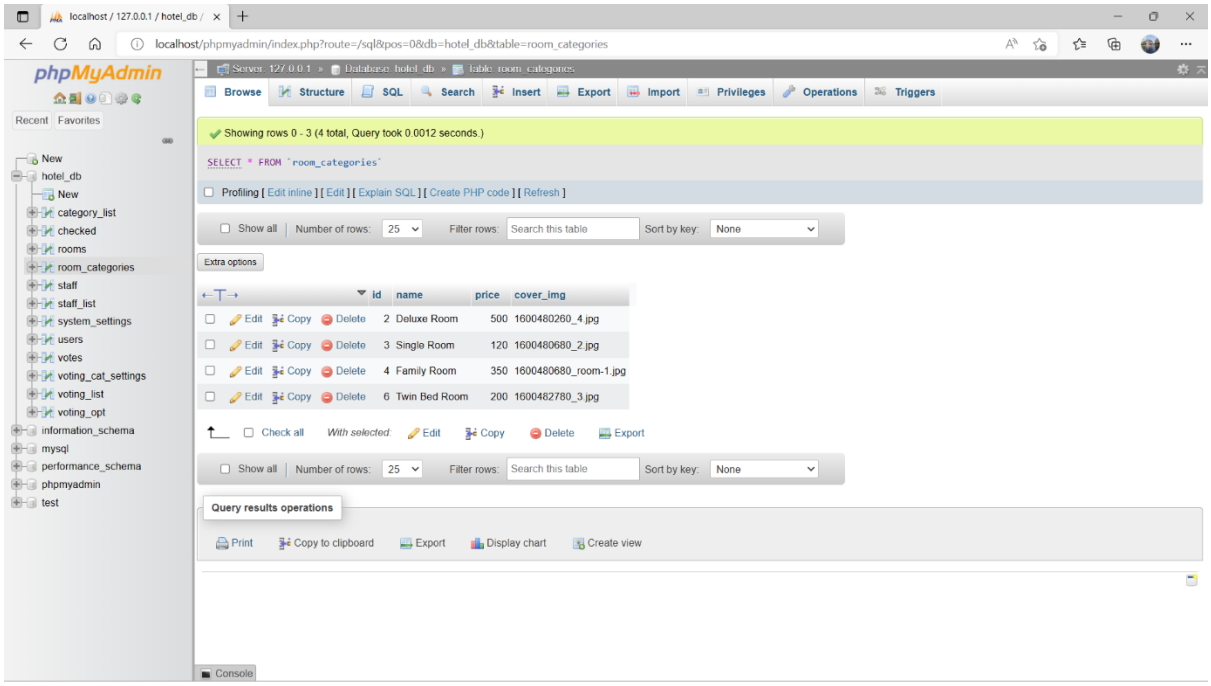

Figure 1.3: Groups and allotments

4. Housekeeping and Maintenance

You can onboard housekeeping teams and assign role-based profiles and user access. They can log in from any device, view a list of assigned tasks and update the status.

Your front office can maintain constant visibility into room statuses and send messages to housekeepers during unforeseen circumstances. Enterprise platforms let you onboard third-party service providers and define rules for automated work orders.

- Staff onboarding
- Task tracking
- Work orders
- Housekeeping zones
- Room status tracking
- Staff communication portals

5. Rate Management

Manage room rates across every channel, website and reward program. You can access a unified view of your pricing structure and adjust channel-specific rates with a few clicks. For example, offer coupon codes for website bookings and drive up OTA prices to incentivize traffic to book through your native booking engine.

Advanced solutions have algorithms to compare historical data on pricing, competitor rates, market conditions, and local events to offer dynamic rate adjustments and ensure you maintain healthy rent structures.

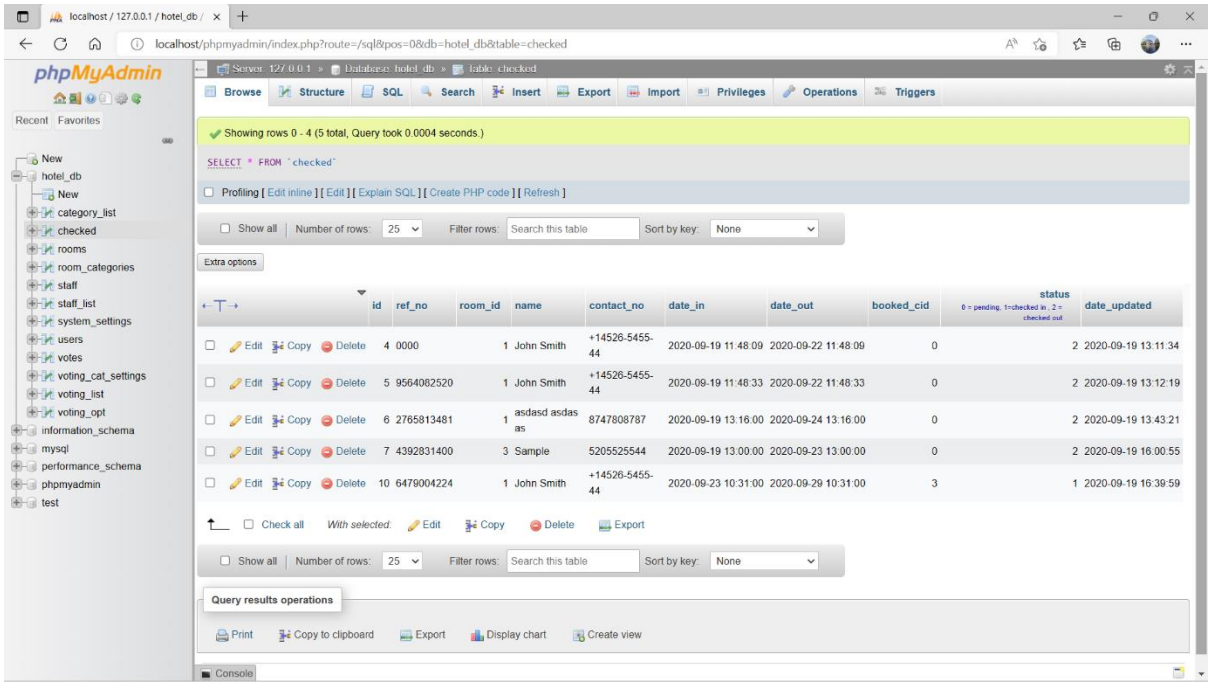

Figure 1.4: Rate management

- Channel-specific pricing
- Coupon codes
- Work orders
- OTA vs. direct booking
- Dynamic rates
- Rate adjustments

6. Reporting

Hotel management software helps you collect guest, staff, and operational data and generate profitability, audit, room and tax, shift audit, departure/arrival, housekeeping, and other reports. You can also specify rules for automated monthly or weekly reporting.

Some platforms offer business intelligence to track ADR, RevPAR, GOPPAR, and a slew of other KPIs to help you analyze data and find opportunities.

- Daily reconciliation
- Report exporting
- Report Scheduling
- Tax reports
- Profitability reports
- Property performance reports
- Revenue forecasting
- Occupancy and revenue goals

7. Guest Experience

Guests are the focal point of any hotel management business, and you need specialized tools to ensure customer satisfaction and repeat visits. Hotel management systems help create guest profiles for tracking their journeys and preferences. You can suggest tour guides, gift cards, customizable reservations, and personalized pricing using the information. Most solutions support connections with third-party CRM software for in-depth customer data management.

- Guest preference
- Gift cards
- Location tours and guides
- Customer-specific pricing
- Additional services

8. Backend Management

Track and manage every asset your business owns (except rooms and properties under the front office's jurisdiction). You can handle staff members, event rooms, inventory, point-of-sale systems, utilities, and other hotel assets.

A few platforms also have review managers that gather customer reviews from every channel and allow you to respond from a centralized dashboard.

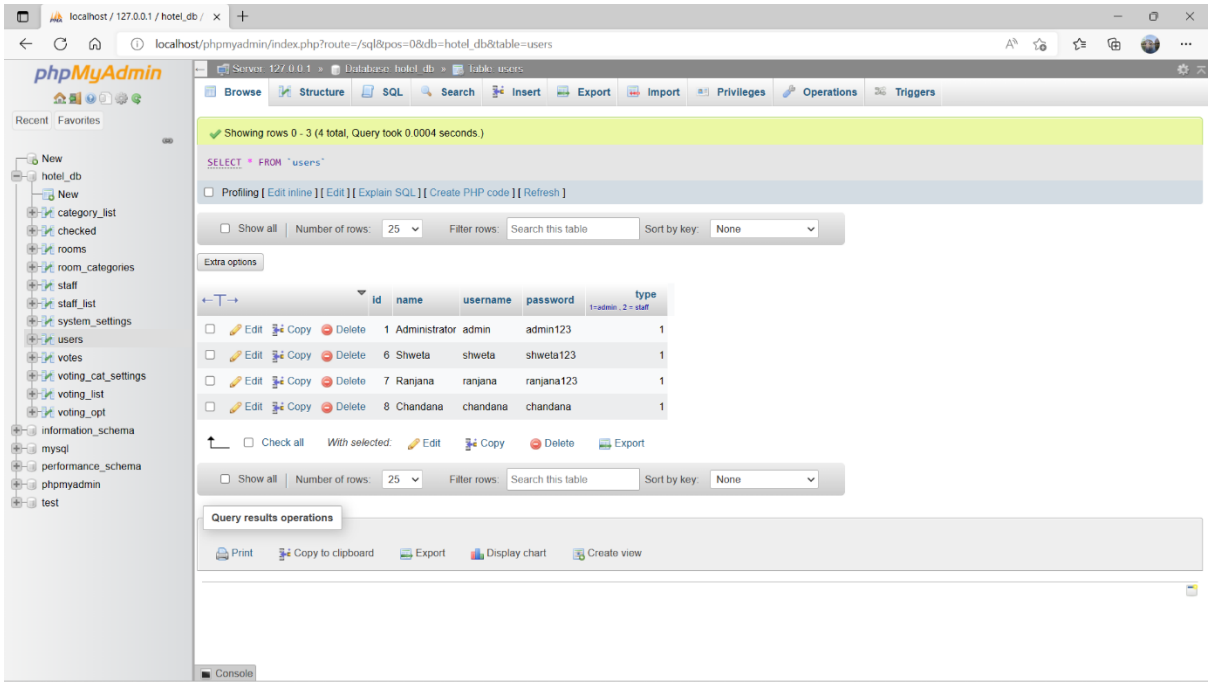

Figure 1.5: Staffing

- Event management
- Staff management
- Staff invoicing
- Utility costs and hotel spending analysis

- Inventory tracking
- Promotional campaigns

9. Other Capabilities

Other than the aforementioned features, hotel management solutions require a few miscellaneous qualities to succeed. Suppose you're managing cross-country hotel chains, or you want to attract international tourists. The right hotel management software will help you translate your websites and listings to targeted local languages, accept multiple currencies and save customers the exchange hassle.

On top of that, the platform should be mobile-optimized and support eSignatures for remote workflows. It should also be PCI-DSS compliant and have SSL certifications to create a secure payment environment and safeguard data transfers.

- Intuitive UI
- Customizable interface
- Electronic signatures
- Mobile-optimized
- Multi-currency

Chapter 2

Methodology

2.1 Entity-Relationship Diagram

An entity–relationship model (ER model) describes interrelated things of interest in a specific domain of knowledge. An ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between instances of those entity types. In software engineering, an ER model is commonly formed to represent things that a business needs to remember to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

The entities of the database are:

- Customers
- Hotel
- Checked
- Rooms
- Room_categories
- Staff
- Votes
- Voting_category

Description of entities are:

• Hotel: This table will store the details pertaining to the hotel such as the id and address of the hotel.

• Customer: This table will store the credentials of all the customers visiting the hotel.

• Checked: This table will store the check-in and check-out details of the customers.

• Room: This table will store the pricing details of the rooms available in the hotel.

• Room_categories: This table will store the various categories of rooms available in the hotel.

• Staff: This table will store the details of staff.

• Votes: This table will store the various categories of votes according rooms available in the hotel.

The attributes of these entities are:

Hotel(id, email, hotel_name, about_content, contact, cover_img)

Staff(id, name, username, password, type)

Customer(id, name, username, password, type)

Checked(date_in, date_out, booked_cid)

Rooms(id, room, category_id, status)

Room_categories(id, name, price, cover_img)

Votes(id, voting_id, category_id, voting_opt_id, user_id)

Vote_category(id, voting_id, category_id, max_selection)

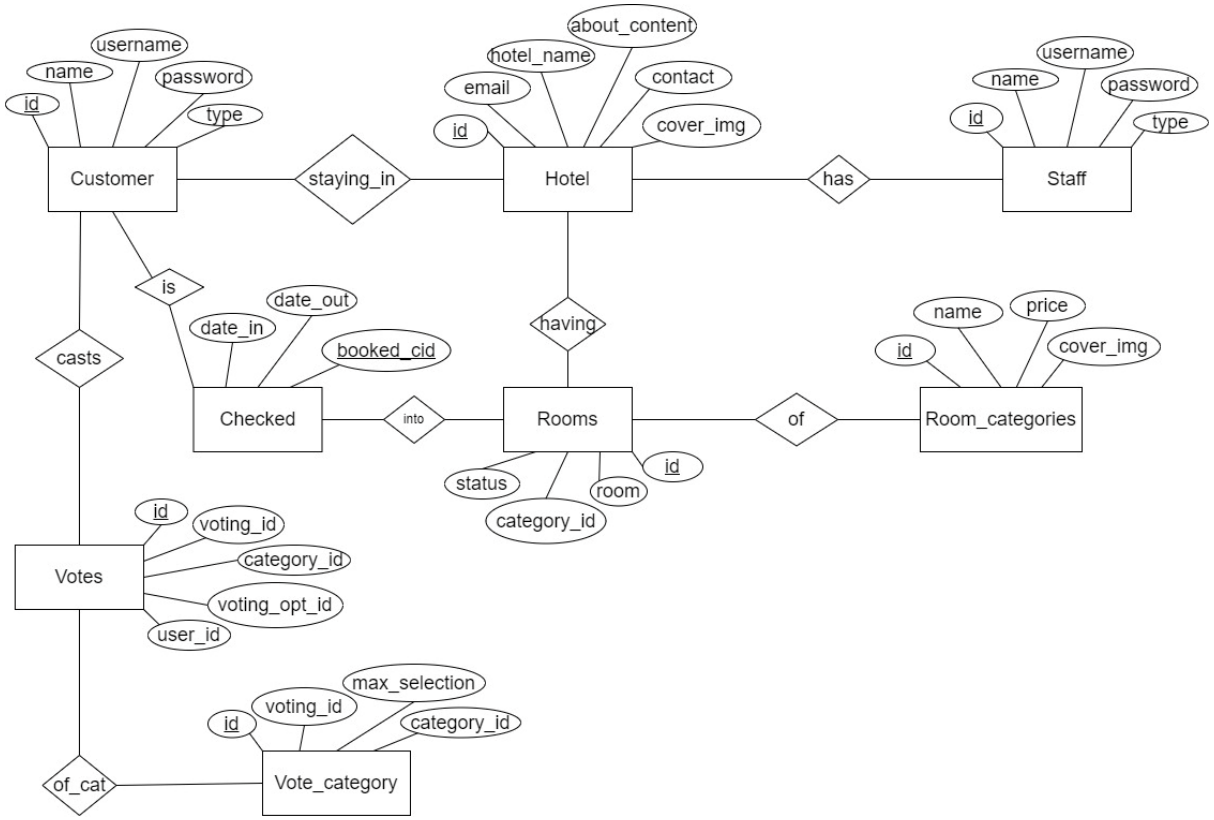

Figure 2.1: Entity Relationship diagram

The above figure shows all the entities and the relationships between each of them. The ER diagram is used to show the entities and their attributes in a graphical representation.

2.2 ER to Relational Mapping

ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types

Step 2: Mapping of Weak Entity Types

Step 3: Mapping of Binary 1:1 Relation

Step 4: Mapping of Binary 1: N Relationship Types

Step 5: Mapping of Binary M: N Relationship Types

Step 6: Mapping of Multivalued attributes

Step 7: Mapping of N-ary Relationship Types

• Step 1: Mapping of Regular Entity Types

1.1: For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.

1.2: Choose one of the key attributes of E as the primary key for R.

1.3: If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

• Step 2: Mapping of Weak Entity Types

2.1: For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.

2.2: Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).

2.3: The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W if any.

• Step 3: Mapping of Binary 1:1 Relation

For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R. There are three possible approaches:

3.1: Foreign Key approach: Choose one of the relations-say S-and to include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

3.2: Merged relation option: An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.

3.3: Cross-reference or relationship relation option: The third alternative is to set up a third relation R to cross-reference the primary keys of the two relations S and T representing the entity types.

• Step 4: Mapping of Binary 1: N Relationship Types

4.1: For each regular binary 1: N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type.

4.2: Include as a foreign key in S the primary key of the relation T that represents the other entity type participating in R.

4.3: Include any simple attributes of the 1: N relation type as attributes of S.

• Step 5: Mapping of Binary M: N Relationship Types

5.1: For each regular binary M: N relationship type R, create a new relation S to represent R.

5.2: Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.

5.3: Also include any simple attributes of the M: N relationship type (or simple components of composite attributes) as attributes of S.

• Step 6: Mapping of Multivalued attributes

6.1: For each multivalued attribute A, create a new relation R.

6.2: This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.

6.3: The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

• Step 7: Mapping of N-ary Relationship Types

7.1: For each n-ary relationship type R, where n>2, create a new relationship S to represent R.

7.2: Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.

7.3: Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Correspondence between ER and Relational Models

- Entity type “Entity” relation
- 1:1 or 1: N relationship type Foreign key (or “relationship” relation)
- M: N relationship type “Relationship” relation and two foreign keys
- n-ary relationship type “Relationship” relation and n foreign keys
- Simple attribute
- Composite attribute Set of simple component attributes
- Multi-valued attribute Relation and foreign key
- Value set Domain
- Key attribute Primary (or secondary) key

2.3 Relational Database Schema

A relational database schema is the tables, columns and relationships that make up a relational database. A relational database schema helps admins and users organize and understand the structure of a database. This is particularly useful when designing a new database, modifying an existing database to support more functionality or building integration between databases. The schema diagram for Restaurant Management System consists of attributes, key attributes such as foreign key and primary key.

Each table has a primary key which is a not null type of data and is fixed and cannot be modified further once entered. A database schema is a basic outline of the database of a database management system that defines the common elements of a database and schema makes it easier to implement the operations efficiently on the relations and other attributes. The Schema for the Restaurant Management System is shown with all the key attributes and their references using rows and arrows.

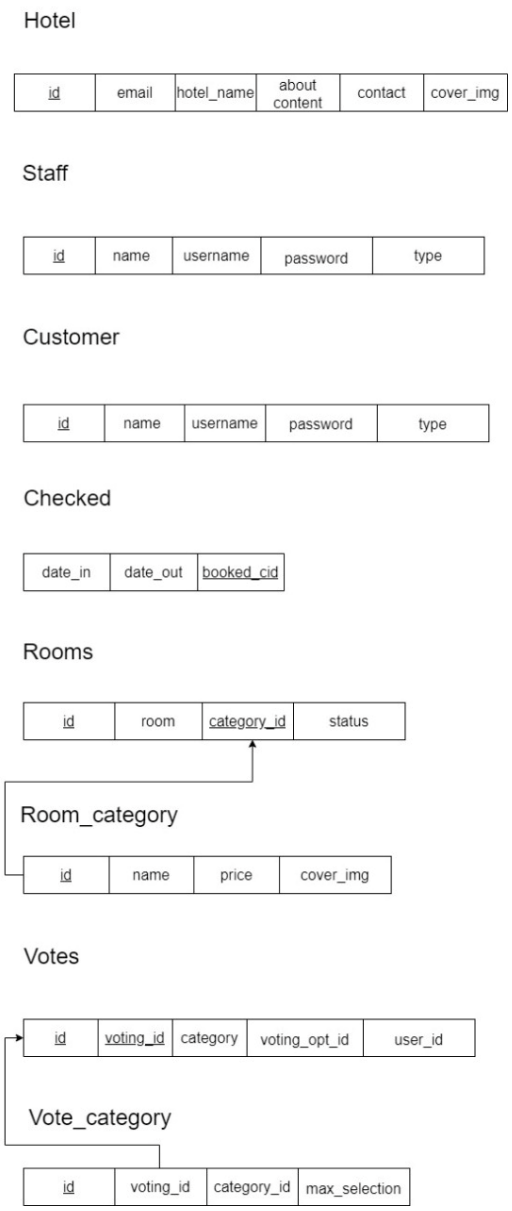

Figure 2.2: Relational Database Schema

The above figure shows the relations (tables) used in Restaurant Management System. The underlined attributes are the primary keys of that table and the references for the foreign keys are shown using arrows.

2.4 Relational Model

The relational model (RM) for database management is an approach to managing data using a structure and language consistent with first-order predicate logic. The Relational model contains all entities and their relations with other entities. It also contains all relations

having (m: n) cardinality. A relational data model is the primary data model, which is used widely around the world for data storage and processing.

This data model shows the foreign key and primary key as PK and FK and the reference table or tuple for the foreign key is specified by arrows. This makes it much easier to understand the structure of the schema than the schema diagram. The relational model (RM) for database management is an approach to managing data using a structure and language consistent with first-order predicate logic.

2.5 Normalized Relational Schema

Normalization involves arranging attributes in relations based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished by applying some formal rules either by a process of synthesis or decomposition. Synthesis creates a normalized database design based on a known set of dependencies. Decomposition takes an existing (insufficiently normalized) database design and improves it based on the known set of dependencies. Normalization involves arranging attributes in relations based on dependencies between attributes, ensuring that the dependencies are properly enforced by database integrity constraints. Normalization is accomplished by applying some formal rules either by a process of synthesis or decomposition.

1 NF - First normal form

The first normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic (indivisible) values, and the value of each attribute contains only a single value from that domain.

The first normal form enforces these criteria:

• Eliminate repeating groups in individual tables.

• Create a separate table for each set of related data.

• Identify each set of related data with a primary key

The attributes which are in 1NF in Restaurant Management System are:

Hotel(<u>id</u>, email, hotel_name, about_content, contact, cover_img)

Staff(<u>id</u>, name, username, password, type)

Customer(<u>id</u>, name, username, password, type)

Checked(date_in, date_out, booked_cid)

Rooms(id, room, category_id, status)

Room_categories(id, name, price, cover_img)

Votes(id, voting_id, category_id, voting_opt_id, user_id)

Vote_category(id, voting_id, category_id, max_selection)

2NF: Second normal form

The second normal form (2NF) is a normal form used in database normalization. A relation that is in the first normal form (1NF) must meet additional criteria if it is to qualify for the second normal form. Specifically: a relation is in 2NF if it is in 1NF and no non-prime attribute is dependent on any proper subset of any candidate key of the relation. A non-prime attribute of a relation is an attribute that is not a part of any candidate key of the relation.

The attributes which are in 2NF in Restaurant Management System are:

Hotel(id, email, hotel_name, about_content, contact, cover_img)

Staff(id, name, username, password, type)

Customer(id, name, username, password, type)

Checked(date_in, date_out, booked_cid)

Rooms(id, room, category_id, status)

Room_categories(id, name, price, cover_img)

Votes(id, voting_id, category_id, voting_opt_id, user_id)

Vote_category(id, voting_id, category_id, max_selection)

3NF: Third normal form

The third normal form(3NF) is a normal form that is used in normalizing a database design to reduce the duplication of data and ensure referential integrity by ensuring that (1) the entity is in the second normal form, and (2) all the attributes in a table are determined only by the candidate keys of that relationship and not by any non-prime attributes. 3NF was designed to improve database processing while minimizing storage costs. 3NF de-modeling was ideal for online transaction processing (OLTP) applications with heavy order entry types of needs. The Mapped relations or tables are a combination of two or more tables that

consists of one or more primary keys of different tables. In this project, the mapped tables mainly consist of the primary keys of one or more tables.

The attributes which are in 3NF in Restaurant Management System are:

Hotel(id, email, hotel_name, about_content, contact, cover_img)

Staff(id, name, username, password, type)

Customer(id, name, username, password, type)

Checked(date_in, date_out, booked_cid)

Rooms(id, room, category_id, status)

Room_categories(id, name, price, cover_img)

Votes(id, voting_id, category_id, voting_opt_id, user_id)

Vote_category(id, voting_id, category_id, max_selection)

2.6 Key Attributes

Table 2.6 Key Attributes

Attribute	Primary Key	Foreign Key
Hotel	id	-
Staff	id	-
Customer	id	-
Checked	booked_cid	id
Rooms	id	-
Room_categories	id	category_id
Votes	id	-
Vote_category	id	voting_id

The above table shows primary keys and foreign keys in the database.

• PRIMARY KEY

A primary key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It cannot accept null, or duplicate values. Only one Candidate Key can be Primary Key. The hotel management database contains the primary keys: id, booked_cid

• FOREIGN KEY

Foreign Key is a field in a database table that is the primary key in another table. It can accept multiple null and duplicate values. The hotel management database contains the foreign keys: id, category_id, voting_id.

• UNIQUE KEY

A unique key is a set of one or more fields/columns of a table that uniquely identify a record in a database table. It is like a primary key but it can accept only one null value and it cannot have duplicate values.

• NOT NULL

The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that insertion of a new record, or updating of a record without adding a value to this field cannot be done. A field that has unique values is, essentially, a key. However, a key is used to uniquely identify a row in a table, while an index is used to sort, or group, the rows in the table. A key should not change once it has been initially set, as it might be referenced elsewhere in the database. The hotel management database contains NOT NULL attributes: id, booked_cid, category_id, voting_id.

• THE DEFAULT SQL CONSTRAINT

At times, to insert a value into a column when no other value is provided can be done automatically. That's where the DEFAULT SQL constraint comes in. The constraint can be used to define a column's default value. This can be a handy way to add the current date and time to a column or to avoid having to use NULL values. The hotel management database contains default constraint attributes.

• CHECK CONSTRAINT

A check constraint is a type of integrity constraint in SQL that specifies a requirement that must be met by each row in a database table. The constraint must be a predicate. It can refer to a single column, or multiple columns of the table.

Chapter 3

System Requirements and Specifications

3.1 User Requirements

Every user:

• Should be comfortable with basic working of the computer.

• Must have been knowledgeable in English.

• Must carry a login ID and password used for authentication.

• Login ID and password used for identification of administrator.

There is no facility for a guest login.

3.2 Software requirements

1. Software is designed to run on any platform above Microsoft Windows 7 (32bit).

2. Microsoft .NET Frameworks 4.0 or above.

3. Microsoft SQL Server Management Studio Express 2010.

3.3 Hardware requirements

1. Operating System supports all known operating systems, such as Windows, Linux, Mac

2. Computer 512MB + RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse.

3. Hard drives should be in an NTFS file system formatted with a minimum of 10 GB of free space. A Laser printer will need to be used to print these reports and notes.

3.4 Security Requirements

There is a need for an encrypted login authentication for admin and customer since sensitive information and inventory should be protected. Information transmission should be securely transmitted without any changes in information to avoid disturbances in orders and billing.

3.5 Tools and Languages used

3.5.1 Tools

The following lists the various software, frameworks, and editors that will be used for the project.

• MySQL Workbench

MySQL Workbench is a visual database design tool that integrates SQL development, administration, database design, creation and maintenance into a single integrated development environment for the MySQL database system. It is the successor to DB Designer 4 from fabFORCE.net, and replaces the previous package of software, MySQL GUI Tools Bundle.

• XAMPP

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible. XAMPP's ease of deployment means WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer, with the advantage that common add-in applications such as Word Press and Joomla! can also be installed with similar ease using Bitnami. Though it is a heavy app for most operating systems even when owing to its less size it takes a load on the processor speed.

3.5.2 Languages

- **PHP**

PHP is a general-purpose scripting language especially suited to web development. It was created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group. PHP originally stood for Personal Home Page, but it now stands for the recursive initialism PHP: Hypertext Pre-processor.

PHP code is usually processed on a web server by a PHP interpreter implemented as a module, a daemon, or as a Common Gateway Interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via a command-line interface (CLI).

There are a lot more things to do with PHP:

o User can generate dynamic pages and files.

o User can create, open, read, write and close files on the server.

o Users can collect data from the web form such as user information, email, credit card info and much more.

➢ MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

- MySQL is released under an open-source license, which makes it free to use.
- MySQL is a very powerful program. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, and so on.

➢ HTML

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating webpages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web.HTML elements are the building blocks of HTML pages. The definition of HTML is Hypertext Mark-up Language:

• Hypertext is the method by which you move around on the web by clicking on special text called hyperlinks which bring you to the next page. The fact that it is hyper just means it is not Restaurant Management System Dept. of ISE, BNMIT 2019-2020 Page 17 linear-i.e., you can go to any place on the Internet whenever you want by clicking on links.

• Mark-up is what HTML tags do to the text inside them. They mark it as a certain type of text (italicized text, for instance).

• HTML is a Language, as it has code words and syntax like any language. With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

• HTML elements are delineated by tags, written using angle brackets. and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page. HTML can embed programs written in a scripted language such as JavaScript which affects the behavior and content of the web page.

➢ CSS

Cascading Style Sheets • Inclusion of CSS defines the look and layout of content. HTML consists of a series Sports Club Management System Dept. of ISE, BNMIT 2021-22 Page 17 of short codes typed into text-file by the site author-these are the tags. The text is then saved as an HTML file, and viewed through a browser, like Internet Explorer or Netscape Navigator. This browser reads the file and translates the text into a visible form.

Chapter 4

System Design and Development

4.1 Architectural Design

The requirements of the software should be transformed into an architecture that describes the software's top-level structure and identifies its components. This is accomplished through architectural design (also called system design), which acts as a preliminary blueprint from which software can be developed. IEEE architectural design is the process of defining a collection of hardware and software components and their interface to establish the framework for the development of a computer system.

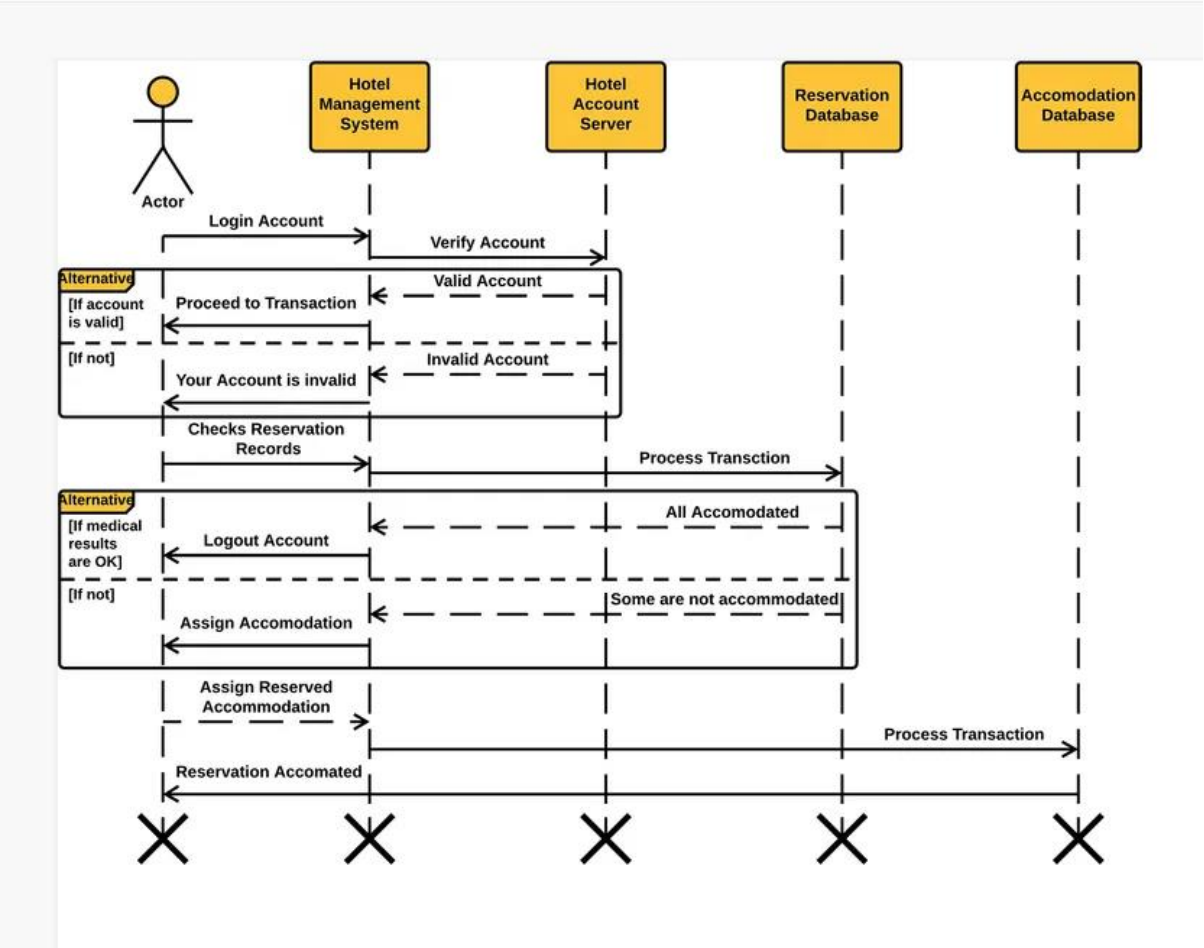

Figure 4.1: Data flow between the application and the end user

This framework is established by examining the software requirement document and designing a model for providing implementation details. These details are used to specify the components of the system along with their inputs, outputs, functions, and the interaction between them. An architectural design performs several functions.

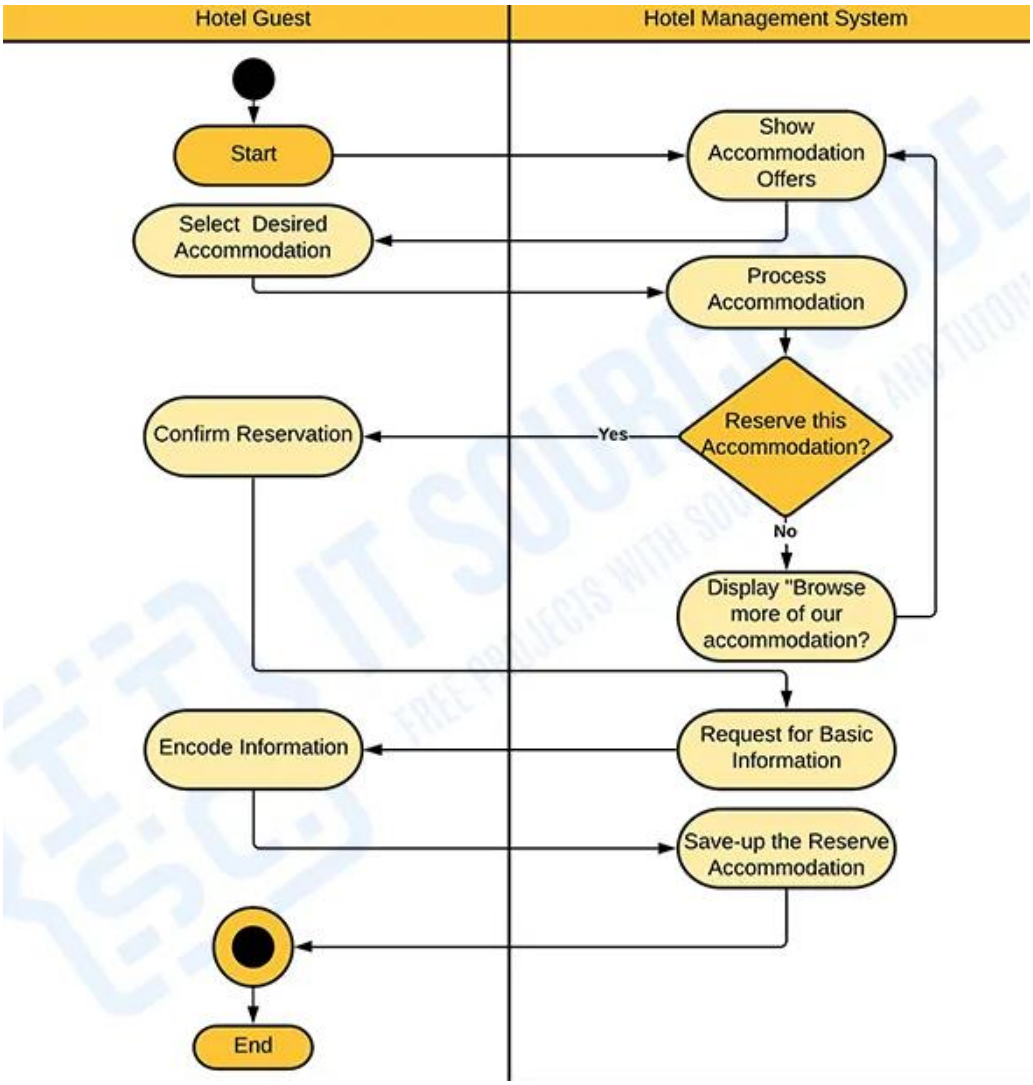

Figure 4.2: Architecture of the database

Chapter 5

Implementation

Back end

From the requirements collected and analyzed, the conceptual schema for the database is created using the entity-relationship (ER) diagram. The Back End is used to create the table and insert the values in the respective tables in the database management system. The car rental system back end focuses on application data access, database administration, data transformation and backup methods.

5.1 List of Modules

Hotel(id, email, hotel_name, about_content, contact, cover_img)

Staff(id, name, username, password, type)

Customer(id, name, username, password, type)

Checked(date_in, date_out, booked_cid)

Rooms(id, room, category_id, status)

Room_categories(id, name, price, cover_img)

Votes(id, voting_id, category_id, voting_opt_id, user_id)

Vote_category(id, voting_id, category_id, max_selection)

5.2 Module Description

Mentioned are all the queries used to perform various tasks in MySQL such as insert, delete, update. A short description of the query is also provided.

Create statements

Description: This query is used to create tables.

Syntax: CREATE TABLE <tablename>(<attributes> <type>, <attributes> <type>[Primary Key]);

Table structure for table `checked`

```
CREATE TABLE `checked` (
  `id` int(30) NOT NULL,
  `ref_no` varchar(100) NOT NULL,
  `room_id` int(30) NOT NULL,
  `name` text NOT NULL,
  `contact_no` varchar(20) NOT NULL,
  `date_in` datetime NOT NULL,
  `date_out` datetime NOT NULL,
  `booked_cid` int(30) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0 = pending, 1=checked in , 2 = checked out',
  `date_updated` datetime NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure for table `rooms`

```
CREATE TABLE `rooms` (
  `id` int(30) NOT NULL,
  `room` varchar(30) NOT NULL,
  `category_id` int(30) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT 0 COMMENT '0 = Available , 1= Unvailables'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure for table `room_categories`

```
CREATE TABLE `room_categories` (
  `id` int(30) NOT NULL,
  `name` text NOT NULL,
  `price` float NOT NULL,
  `cover_img` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure for table `system_settings`

```
CREATE TABLE `system_settings` (
  `id` int(30) NOT NULL,
  `hotel_name` text NOT NULL,
  `email` varchar(200) NOT NULL,
  `contact` varchar(20) NOT NULL,
  `cover_img` text NOT NULL,
  `about_content` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Table structure for table `users`

```
CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `name` varchar(200) NOT NULL,
  `username` varchar(100) NOT NULL,
  `password` varchar(200) NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 2 COMMENT '1=admin , 2 = staff'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Updating all the primary key attributes for all tables after inclusion of ‘id’.

Indexes for table `checked`
ALTER TABLE `checked`
  ADD PRIMARY KEY (`id`);

Indexes for table `rooms`
ALTER TABLE `rooms`
  ADD PRIMARY KEY (`id`);

Indexes for table `room_categories`
ALTER TABLE `room_categories`
  ADD PRIMARY KEY (`id`);

Indexes for table `system_settings`
ALTER TABLE `system_settings`
  ADD PRIMARY KEY (`id`);

Indexes for table `users`
ALTER TABLE `users`
  ADD PRIMARY KEY (`id`);

Using AUTO_INCREMENT for dumped tables

AUTO_INCREMENT for table `checked`
ALTER TABLE `checked`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

AUTO_INCREMENT for table `rooms`
ALTER TABLE `rooms`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

AUTO_INCREMENT for table `room_categories`
ALTER TABLE `room_categories`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;

AUTO_INCREMENT for table `system_settings`
ALTER TABLE `system_settings`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

AUTO_INCREMENT for table `users`
ALTER TABLE `users`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

Front End

The first page of the front end is the home page which has options like login, signup, view cars, about us and contacts us and admin login. When the person is a new user, he/she has to signup first where he has filled in all user's user personal details like name, email, contact number. After signup user is direct the ed to login page.

- **HTML**

- **Home page [Menu/Dashboard]**

```
<?php
include 'db_connect.php';
$folder_parent = isset($_GET['fid'])? $_GET['fid'] : 0;
$folders = $conn->query("SELECT * FROM folders where parent_id = $folder_parent and user_id = '".$_SESSION['login_id']."'  order by name asc");


$files = $conn->query("SELECT * FROM files where folder_id = $folder_parent and user_id = '".$_SESSION['login_id']."'  order by name asc");

?>
<style>
   .folder-item{
      cursor: pointer;
   }
   .folder-item:hover{
      background: #eaeaea;
      color: black;
      box-shadow: 3px #0000000f;
   }
   .custom-menu {
      z-index: 1000;
      position: absolute;
      background-color: #ffffff;
      border: 1px solid #0000001c;
      border-radius: 5px;
      padding: 8px;
      min-width: 13vw;
}
a.custom-menu-list {
   width: 100%;
   display: flex;
```

```
    color: #4c4b4b;
    font-weight: 600;
    font-size: 1em;
    padding: 1px 11px;
}
.file-item{
    cursor: pointer;
}
a.custom-menu-list:hover,.file-item:hover,.file-item. Active {
    background: #80808024;
}
table th,td{
    /*border-left:1px solid gray;*/
}
a.custom-menu-list span.icon{
        width:1em;
        margin-right: 5px
}

</style>
<div class="container-fluid">
    <div class="col-lg-12">
        <div class="row">
            <div class="card col-lg-12">
                <div class="card-body" id="paths">
                <!-- <a href="index.php?page=files" class="">..</a>/ -->
                <?php
                $id=$folder_parent;
                while($id > 0){

                    $path = $conn->query("SELECT * FROM folders where id = $id  order by name
asc")->fetch_array();
                    echo '<script>
                        $("#paths").prepend("<a
href=\"index.php?page=files&fid='.$path['id'].'\">'.$path['name'].'</a>/")
                    </script>';
                    $id = $path['parent_id'];

                }
                echo '<script>
                        $("#paths").prepend("<a href=\"index.php?page=files\">..</a>/")
                    </script>';
                ?>
```

```
            </div>
          </div>
        </div>

        <div class="row">
          <button class="btn btn-primary btn-sm" id="new_folder"><i class="fa fa-plus"></i>
New Folder</button>
          <button class="btn btn-primary btn-sm ml-4" id="new_file"><i class="fa fa-
upload"></i> Upload File</button>
        </div>
        <hr>
        <div class="row">
          <div class="col-lg-12">
          <div class="col-md-4 input-group offset-4">

            <input type="text" class="form-control" id="search" aria-label="Small" aria-
describedby="inputGroup-sizing-sm">
            <div class="input-group-append">
              <span class="input-group-text" id="inputGroup-sizing-sm"><i class="fa fa-
search"></i></span>
            </div>
          </div>
          </div>
        </div>
        <div class="row">
          <div class="col-md-12"><h4><b>Folders</b></h4></div>
        </div>
        <hr>
        <div class="row">
          <?php
          while($row=$folders->fetch_assoc()):
          ?>
            <div class="card col-md-3 mt-2 ml-2 mr-2 mb-2 folder-item" data-id="<?php echo
$row['id'] ?>">
              <div class="card-body">
                  <large><span><i class="fa fa-folder"></i></span><b class="to_folder">
<?php echo $row['name'] ?></b></large>
              </div>
            </div>
          <?php endwhile; ?>
        </div>
        <hr>
        <div class="row">
          <div class="card col-md-12">
```

```
            <div class="card-body">
               <table width="100%">
                  <tr>
                     <th width="40%" class="">Filename</th>
                     <th width="20%" class="">Date</th>
                     <th width="40%" class="">Description</th>
                  </tr>
                  <?php
               while($row=$files->fetch_assoc()):
                  $name = explode(' ||',$row['name']);
                  $name = isset($name[1]) ? $name[0] ." (".$name[1].").".$row['file_type'] :
$name[0] .".".$row['file_type'];
                  $img_arr = array('png','jpg','jpeg','gif','psd','tif');
                  $doc_arr =array('doc','docx');
                  $pdf_arr =array('pdf','ps','eps','prn');
                  $icon ='fa-file';
                  if(in_array(strtolower($row['file_type']),$img_arr))
                     $icon ='fa-image';
                  if(in_array(strtolower($row['file_type']),$doc_arr))
                     $icon ='fa-file-word';
                  if(in_array(strtolower($row['file_type']),$pdf_arr))
                     $icon ='fa-file-pdf';

if(in_array(strtolower($row['file_type']),['xlsx','xls','xlsm','xlsb','xltm','xlt','xla','xlr']))
                     $icon ='fa-file-excel';
                  if(in_array(strtolower($row['file_type']),['zip','rar','tar']))
                     $icon ='fa-file-archive';

               ?>
                  <tr class='file-item' data-id="<?php echo $row['id'] ?>" data-name="<?php
echo $name ?>">
                     <td><large><span><i class="fa <?php echo $icon ?>"></i></span><b
class="to_file"> <?php echo $name ?></b></large>
                     <input type="text" class="rename_file" value="<?php echo $row['name']
?>" data-id="<?php echo $row['id'] ?>" data-type="<?php echo $row['file_type'] ?>"
style="display: none">

                     </td>
                     <td><i class="to_file"><?php echo date('Y/m/d h:i
A',strtotime($row['date_updated'])) ?></i></td>
                     <td><i class="to_file"><?php echo $row['description'] ?></i></td>
                  </tr>

               <?php endwhile; ?>
```

```
                </table>

            </div>
        </div>

    </div>
  </div>
</div>
<div id="menu-folder-clone" style="display: none;">
  <a href="javascript:void(0)" class="custom-menu-list file-option edit">Rename</a>
  <a href="javascript:void(0)" class="custom-menu-list file-option delete">Delete</a>
</div>
<div id="menu-file-clone" style="display: none;">
  <a href="javascript:void(0)" class="custom-menu-list file-option edit"><span><i class="fa
fa-edit"></i> </span>Rename</a>
  <a href="javascript:void(0)" class="custom-menu-list file-option download"><span><i
class="fa fa-download"></i> </span>Download</a>
  <a href="javascript:void(0)" class="custom-menu-list file-option delete"><span><i
class="fa fa-trash"></i> </span>Delete</a>
</div>

<script>

  $('#new_folder').click(function(){
    uni_modal('','manage_folder.php?fid=<?php echo $folder_parent ?>')
  })
  $('#new_file').click(function(){
    uni_modal('','manage_files.php?fid=<?php echo $folder_parent ?>')
  })
  $('.folder-item').dblclick(function(){
    location.href = 'index.php?page=files&fid='+$(this).attr('data-id')
  })
  $('.folder-item').bind("contextmenu", function(event) {
  event.preventDefault();
  $("div.custom-menu").hide();
  var custom =$("<div class='custom-menu'></div>")
    custom.append($('#menu-folder-clone').html())
    custom.find('.edit').attr('data-id',$(this).attr('data-id'))
    custom.find('.delete').attr('data-id',$(this).attr('data-id'))
  custom.appendTo("body")
  custom.css({top: event.pageY + "px", left: event.pageX + "px"});

  $("div.custom-menu .edit").click(function(e){
    e.preventDefault()
```

```
        uni_modal('Rename Folder','manage_folder.php?fid=<?php echo $folder_parent
?>&id='+$(this).attr('data-id') )
    })
    $("div.custom-menu .delete").click(function(e){
        e.preventDefault()
        _conf("Are you sure to delete this Folder?",'delete_folder',[$(this).attr('data-id')])
    })
})

    //FILE
    $('.file-item').bind("contextmenu", function(event) {
    event.preventDefault();

    $('.file-item').removeClass('active')
    $(this).addClass('active')
    $("div.custom-menu").hide();
    var custom =$("<div class='custom-menu file'></div>")
        custom.append($('#menu-file-clone').html())
        custom.find('.edit').attr('data-id',$(this).attr('data-id'))
        custom.find('.delete').attr('data-id',$(this).attr('data-id'))
        custom.find('.download').attr('data-id',$(this).attr('data-id'))
    custom.appendTo("body")
    custom.css({top: event.pageY + "px", left: event.pageX + "px"});

    $("div.file.custom-menu .edit").click(function(e){
        e.preventDefault()
        $('.rename_file[data-id="'+$(this).attr('data-id')+'"]').siblings('large').hide();
        $('.rename_file[data-id="'+$(this).attr('data-id')+'"]').show();
    })
    $("div.file.custom-menu .delete").click(function(e){
        e.preventDefault()
        _conf("Are you sure to delete this file?",'delete_file',[$(this).attr('data-id')])
    })
    $("div.file.custom-menu .download").click(function(e){
        e.preventDefault()
        window.open('download.php?id='+$(this).attr('data-id'))
    })

    $('.rename_file').keypress(function(e){
        var _this = $(this)
        if(e.which == 13){
            start_load()
            $.ajax({
                url:'ajax.php?action=file_rename',
```

```
            method:'POST',
            data:{id:$(this).attr('data-id'),name:$(this).val(),type:$(this).attr('data-
type'),folder_id:'<?php echo $folder_parent ?>'},
            success:function(resp){
               if(typeof resp != undefined){
                  resp = JSON.parse(resp);
                  if(resp.status== 1){
                        _this.siblings('large').find('b').html(resp.new_name);
                        end_load();
                        _this.hide()
                        _this.siblings('large').show()
                  }
               }
            }
         })
      }
   })

})
//FILE


   $('.file-item').click(function(){
      if($(this).find('input.rename_file').is(':visible') == true)
      return false;
      uni_modal($(this).attr('data-name'),'manage_files.php?<?php echo $folder_parent
?>&id='+$(this).attr('data-id'))
   })
   $(document).bind("click", function(event) {
   $("div.custom-menu").hide();
   $('#file-item').removeClass('active')

});
   $(document).keyup(function(e){

   if(e.keyCode === 27){
      $("div.custom-menu").hide();
   $('#file-item').removeClass('active')

   }

});
   $(document).ready(function(){
      $('#search').keyup(function(){
```

```
        var _f = $(this).val().toLowerCase()
        $('.to_folder').each(function(){
            var val  = $(this).text().toLowerCase()
            if(val.includes(_f))
                $(this).closest('.card').toggle(true);
                else
                $(this).closest('.card').toggle(false);


        })
        $('.to_file').each(function(){
            var val  = $(this).text().toLowerCase()
            if(val.includes(_f))
                $(this).closest('tr').toggle(true);
                else
                $(this).closest('tr').toggle(false);


        })
    })
})
function delete_folder($id){
    start_load();
    $.ajax({
        url:'ajax.php?action=delete_folder',
        method:'POST',
        data:{id:$id},
        success:function(resp){
            if(resp == 1){
                alert_toast("Folder successfully deleted.",'success')
                    setTimeout(function(){
                        location.reload()
                    },1500)
            }
        }
    })
}
function delete_file($id){
    start_load();
    $.ajax({
        url:'ajax.php?action=delete_file',
        method:'POST',
        data:{id:$id},
        success:function(resp){
```

```
            if(resp == 1){
                alert_toast("Folder successfully deleted.",'success')
                    setTimeout(function(){
                        location.reload()
                    },1500)
            }
        }
    })
}

</script>
```

- **User access page**

```
<?php include('db_connect.php');
$cat = $conn->query("SELECT * FROM room_categories");
$cat_arr = array();
while($row = $cat->fetch_assoc()){
    $cat_arr[$row['id']] = $row;
}
$room = $conn->query("SELECT * FROM rooms");
$room_arr = array();
while($row = $room->fetch_assoc()){
    $room_arr[$row['id']] = $row;
}
?>
<div class="container-fluid">
    <div class="col-lg-12">
        <div class="row mt-3">
            <div class="col-md-12">
                <div class="card">
                    <div class="card-body">
                        <table class="table table-bordered">
                            <thead>
                                <th>#</th>
                                <th>Category</th>
                                <th>Reference</th>
                                <th>Status</th>
                                <th>Action</th>
                            </thead>
                            <tbody>
                                <?php
                                $i = 1;
```

```
                        $checked = $conn->query("SELECT * FROM checked where status = 0
order by status desc, id asc");
                        while($row=$checked->fetch_assoc()):
                        ?>
                        <tr>
                            <td class="text-center"><?php echo $i++ ?></td>
                            <td class="text-center"><?php echo
$cat_arr[$row['booked_cid']]['name'] ?></td>
                            <td class=""><?php echo $row['ref_no'] ?></td>
                               <td class="text-center"><span class="badge badge-
warning">Booked</span></td>
                            <td class="text-center">
                                  <button class="btn btn-sm btn-primary check_out" type="button"
data-id="<?php echo $row['id'] ?>">View</button>
                            </td>
                        </tr>
                    <?php endwhile; ?>
                    </tbody>
                </table>
            </div>
        </div>
    </div>
  </div>
</div>

<script>
  $('table').dataTable()
  $('.check_out').click(function(){
    uni_modal("Check Out","manage_check_out.php?checkout=1&id="+$(this).attr("data-
id"))
  })
  $('#filter').submit(function(e){
    e.preventDefault()

location.replace('index.php?page=check_in&category_id='+$(this).find('[name="category_id
"]').val()+'&status='+$(this).find('[name="status"]').val())
  })
</script>
```

- **User Login**

```
<?php
```

```
include('db_connect.php');
if(isset($_GET['id'])){
$user = $conn->query("SELECT * FROM users where id =".$_GET['id']);
foreach($user->fetch_array() as $k =>$v){
   $meta[$k] = $v;
}
}
?>
<div class="container-fluid">

   <form action="" id="manage-user">
      <input type="hidden" name="id" value="<?php echo isset($meta['id']) ? $meta['id']: "
?>">
      <div class="form-group">
         <label for="name">Name</label>
         <input type="text" name="name" id="name" class="form-control" value="<?php
echo isset($meta['name']) ? $meta['name']: " ?>" required>
      </div>
      <div class="form-group">
         <label for="username">Username</label>
         <input type="text" name="username" id="username" class="form-control"
value="<?php echo isset($meta['username']) ? $meta['username']: " ?>" required>
      </div>
      <div class="form-group">
         <label for="password">Password</label>
         <input type="password" name="password" id="password" class="form-control"
value="<?php echo isset($meta['password']) ? $meta['id']: " ?>" required>
      </div>
      <div class="form-group">
         <label for="type">User Type</label>
         <select name="type" id="type" class="custom-select">
            <option value="1" <?php echo isset($meta['type']) && $meta['type'] == 1 ?
'selected': " ?>>Admin</option>
            <option value="2" <?php echo isset($meta['type']) && $meta['type'] == 2 ?
'selected': " ?>>User</option>
         </select>
      </div>
   </form>
</div>
<script>
   $('#manage-user').submit(function(e){
      e.preventDefault();
      start_load()
      $.ajax({
```

```
        url:'ajax.php?action=save_user',
        method:'POST',
        data:$(this).serialize(),
        success:function(resp){
            if(resp ==1){
                alert_toast("Data successfully saved",'success')
                setTimeout(function(){
                    location.reload()
                },1500)
            }
        }
    })
  })
</script>
```

- **JavaScript**

Admin Page

```
<?php
session_start();
Class Action {
   private $db;

   public function __construct() {
      ob_start();
   include 'db_connect.php';

   $this->db = $conn;
   }
   function __destruct() {
      $this->db->close();
      ob_end_flush();
   }

   function login(){
      extract($_POST);
      $qry = $this->db->query("SELECT * FROM users where username = '".$username."'
and password = '".$password."' ");
      if($qry->num_rows > 0){
         foreach ($qry->fetch_array() as $key => $value) {
            if($key != 'passwors' && !is_numeric($key))
```

```
                $_SESSION['login_'.$key] = $value;
            }
            if($_SESSION['login_type'] == 1)
                return 1;
            else
                return 2;
        }else{
            return 3;
        }
    }
    function logout(){
        session_destroy();
        foreach ($_SESSION as $key => $value) {
            unset($_SESSION[$key]);
        }
        header("location:login.php");
    }

    function save_user(){
        extract($_POST);
        $data = " name = '$name' ";
        $data .= ", username = '$username' ";
        $data .= ", password = '$password' ";
        $data .= ", type = '$type' ";
        if(empty($id)){
            $save = $this->db->query("INSERT INTO users set ".$data);
        }else{
            $save = $this->db->query("UPDATE users set ".$data." where id = ".$id);
        }
        if($save){
            return 1;
        }
    }

    function save_settings(){
        extract($_POST);
        $data = " hotel_name = '$name' ";
        $data .= ", email = '$email' ";
        $data .= ", contact = '$contact' ";
        $data .= ", about_content = '".htmlentities(str_replace("'","&#x2019;",$about))."' ";
        if($_FILES['img']['tmp_name'] != ''){
                    $fname = strtotime(date('y-m-d H:i')).'_'.$_FILES['img']['name'];
                    $move = move_uploaded_file($_FILES['img']['tmp_name'],'../assets/img/'.
$fname);
```

```
                $data .= ", cover_img = '$fname' ";

        }

        // echo "INSERT INTO system_settings set ".$data;
        $chk = $this->db->query("SELECT * FROM system_settings");
        if($chk->num_rows > 0){
            $save = $this->db->query("UPDATE system_settings set ".$data." where id =".$chk-
>fetch_array()['id']);
        }else{
            $save = $this->db->query("INSERT INTO system_settings set ".$data);
        }
        if($save){
        $query = $this->db->query("SELECT * FROM system_settings limit 1")-
>fetch_array();
        foreach ($query as $key => $value) {
            if(!is_numeric($key))
                $_SESSION['setting_'.$key] = $value;
        }

            return 1;
                }
    }

    function save_category(){
        extract($_POST);
        $data = " name = '$name' ";
        $data .= ", price = '$price' ";
        if($_FILES['img']['tmp_name'] != ''){
                    $fname = strtotime(date('y-m-d H:i')).'_'.$_FILES['img']['name'];
                    $move = move_uploaded_file($_FILES['img']['tmp_name'],'../assets/img/'.
$fname);
                $data .= ", cover_img = '$fname' ";

        }
        if(empty($id)){
            $save = $this->db->query("INSERT INTO room_categories set ".$data);
        }else{
            $save = $this->db->query("UPDATE room_categories set ".$data." where id=".$id);
        }
        if($save)
            return 1;
    }
    function delete_category(){
```

```
        extract($_POST);
        $delete = $this->db->query("DELETE FROM room_categories where id = ".$id);
        if($delete)
            return 1;
    }
    function save_room(){
        extract($_POST);
        $data = " room = '$room' ";
        $data .= ", category_id = '$category_id' ";
        $data .= ", status = '$status' ";
        if(empty($id)){
            $save = $this->db->query("INSERT INTO rooms set ".$data);
        }else{
            $save = $this->db->query("UPDATE rooms set ".$data." where id=".$id);
        }
        if($save)
            return 1;
    }

    function delete_room(){
        extract($_POST);
        $delete = $this->db->query("DELETE FROM rooms where id = ".$id);
        if($delete)
            return 1;
    }

    function save_check_in(){
        extract($_POST);
        $data = " room_id = '$rid' ";
        $data .= ", name = '$name' ";
        $data .= ", contact_no = '$contact' ";
        $data .= ", status = 1 ";

        $data .= ", date_in = '".$date_in.' '.$date_in_time."' ";
        $out= date("Y-m-d H:i",strtotime($date_in.' '.$date_in_time.' +'.$days.' days'));
        $data .= ", date_out = '$out' ";
        $i = 1;
        while($i== 1){
            $ref  = sprintf("%'.04d\n",mt_rand(1,9999999999));
            if($this->db->query("SELECT * FROM checked where ref_no ='$ref'")->num_rows
<= 0)
                $i=0;
        }
        $data .= ", ref_no = '$ref' ";
```

```
        if(empty($id)){
            $save = $this->db->query("INSERT INTO checked set ".$data);
            $id=$this->db->insert_id;
        }else{
            $save = $this->db->query("UPDATE checked set ".$data." where id=".$id);
        }
        if($save){

            $this->db->query("UPDATE rooms set status = 1 where id=".$rid);
                return $id;
        }
    }
    function save_checkout(){
        extract($_POST);
            $save = $this->db->query("UPDATE checked set status = 2 where id=".$id);
            if($save){

                $this->db->query("UPDATE rooms set status = 0 where id=".$rid);
                    return 1;
            }

    }
    function save_book(){
        extract($_POST);
        $data = " booked_cid = '$cid' ";
        $data .= ", name = '$name' ";
        $data .= ", contact_no = '$contact' ";
        $data .= ", status = 0 ";

        $data .= ", date_in = '".$date_in.' '.$date_in_time."' ";
        $out= date("Y-m-d H:i",strtotime($date_in.' '.$date_in_time.' +'.$days.' days'));
        $data .= ", date_out = '$out' ";
        $i = 1;
        while($i== 1){
            $ref  = sprintf("%'.04d\n",mt_rand(1,9999999999));
            if($this->db->query("SELECT * FROM checked where ref_no ='$ref'")->num_rows
<= 0)
                $i=0;
        }
        $data .= ", ref_no = '$ref' ";

            $save = $this->db->query("INSERT INTO checked set ".$data);
            $id=$this->db->insert_id;
```

```
        if($save){
                return $id;
        }
    }

}
<?php
ob_start();
$action = $_GET['action'];
include 'admin_class.php';
$crud = new Action();

if($action == 'login'){
    $login = $crud->login();
    if($login)
        echo $login;
}
if($action == 'logout'){
    $logout = $crud->logout();
    if($logout)
        echo $logout;
}
if($action == 'save_user'){
    $save = $crud->save_user();
    if($save)
        echo $save;
}
if($action == "save_settings"){
    $save = $crud->save_settings();
    if($save)
        echo $save;
}
if($action == "save_category"){
    $save = $crud->save_category();
    if($save)
        echo $save;
}
if($action == "delete_category"){
    $save = $crud->delete_category();
    if($save)
        echo $save;
}
if($action == "save_room"){
```

```
    $save = $crud->save_room();
    if($save)
        echo $save;
}
if($action == "delete_room"){
    $save = $crud->delete_room();
    if($save)
        echo $save;
}
if($action == "save_check-in"){
    $save = $crud->save_check_in();
    if($save)
        echo $save;
}
if($action == "save_checkout"){
    $save = $crud->save_checkout();
    if($save)
        echo $save;
}
if($action == "save_book"){
    $save = $crud->save_book();
    if($save)
        echo $save;
}
```

5.3 Queries

Triggers

- Trigger

```
<body id="page-top">
        <!-- Navigation-->
        <div class="toast" id="alert_toast" role="alert" aria-live="assertive"
aria-atomic="true">
        <div class="toast-body text-white">
        </div>
      </div>
        <nav class="navbar navbar-expand-lg navbar-light fixed-top py-3"
id="mainNav">
            <div class="container">
                <a class="navbar-brand js-scroll-trigger" href="./"><?php echo
$_SESSION['setting_hotel_name'] ?></a>
                <button class="navbar-toggler navbar-toggler-right"
type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-
```

```
controls="navbarResponsive" aria-expanded="false" aria-label="Toggle
navigation"><span class="navbar-toggler-icon"></span></button>
                <div class="collapse navbar-collapse" id="navbarResponsive">
                    <ul class="navbar-nav ml-auto my-2 my-lg-0">
                        <li class="nav-item"><a class="nav-link js-scroll-
trigger" href="index.php?page=home">Home</a></li>
                        <li class="nav-item"><a class="nav-link js-scroll-
trigger" href="index.php?page=list">Rooms</a></li>
                        <li class="nav-item"><a class="nav-link js-scroll-
trigger" href="index.php?page=about">About</a></li>
                    </ul>
                </div>
            </div>
        </nav>

        <?php
        $page = isset($_GET['page']) ?$_GET['page'] : "home";
        include $page.'.php';
        ?>
```

Chapter 6

RESULTS AND DISCUSSIONS

The project is compiled and executed on Microsoft Edge. Some screenshots are present here to show the working of the application.

- **User side**

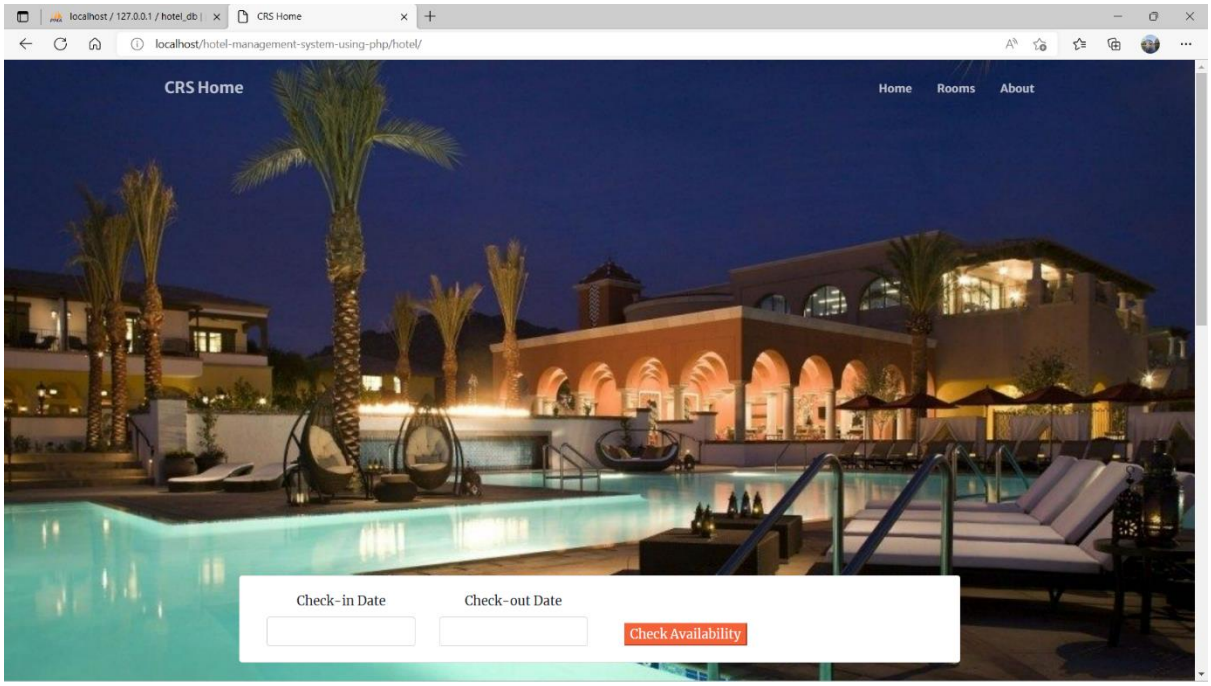

Figure 6.1 : Home page

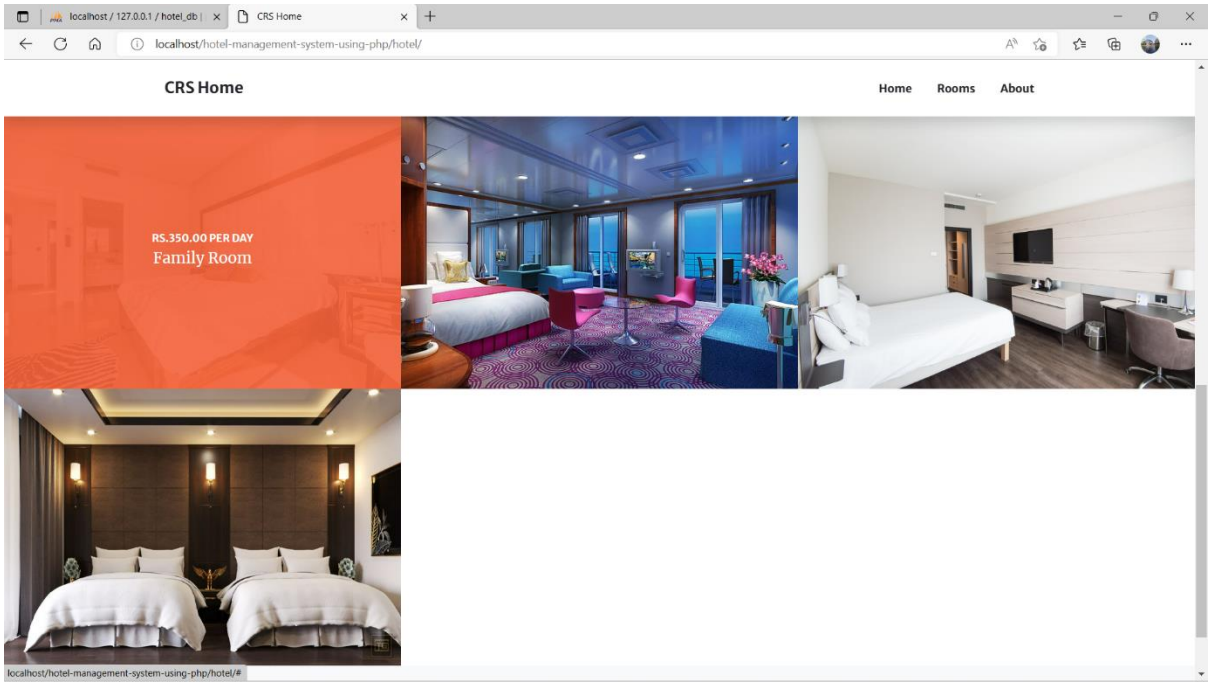

Figure 6.2: Rooms

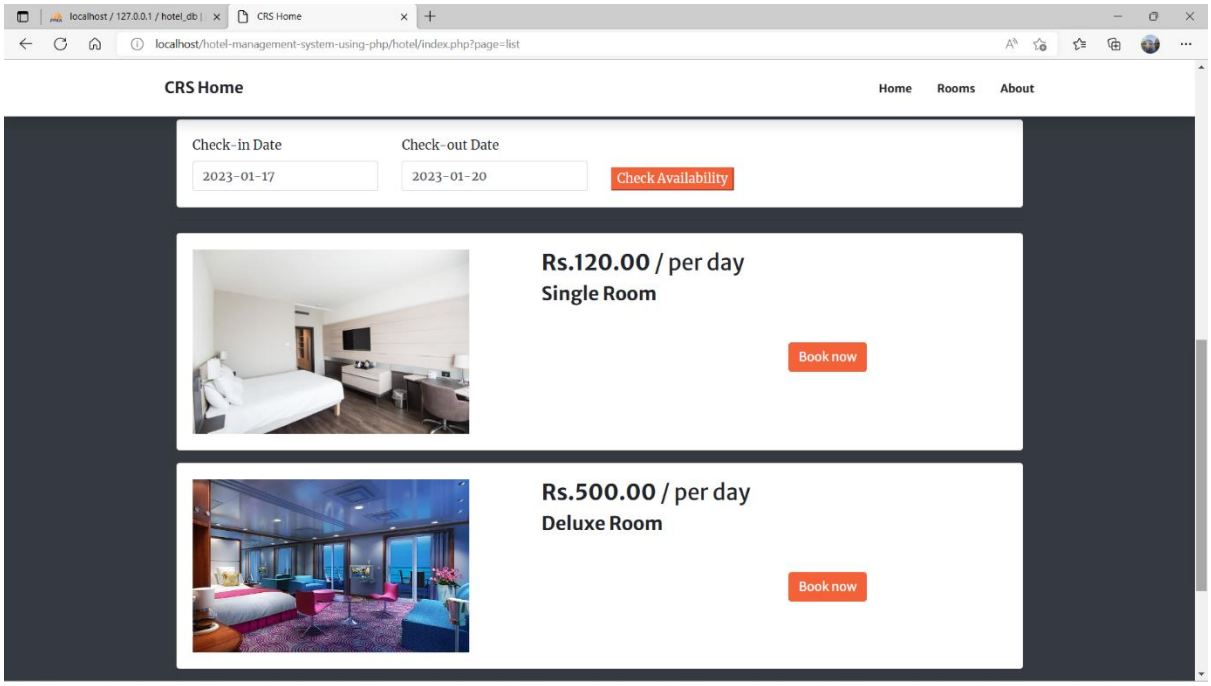

Figure 6.3: Checking the availability of the rooms

➢ Admin side

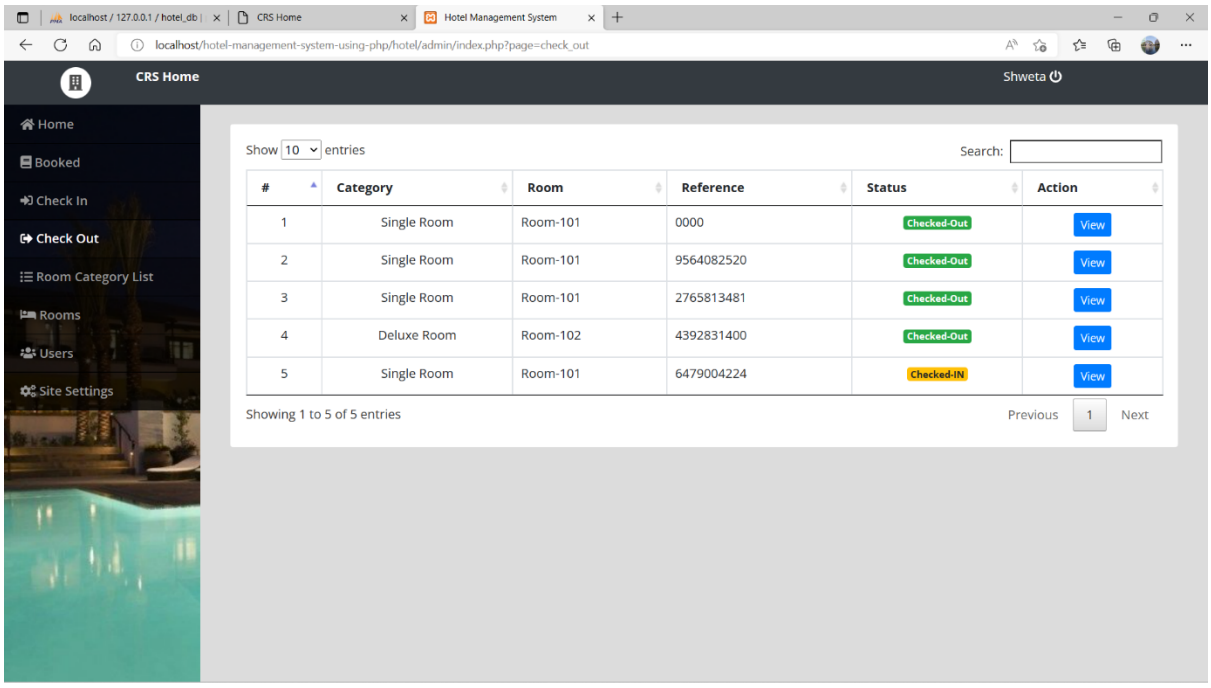

Figure 6.4: Status of a customer's stay

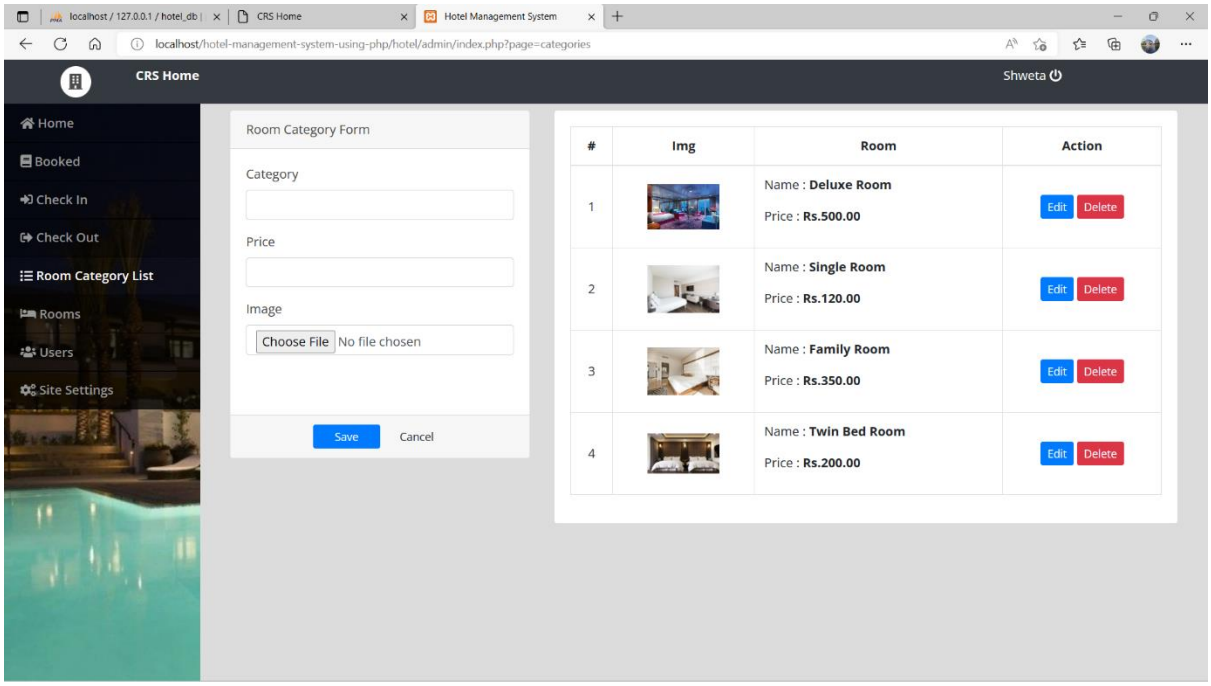

Figure 6.5: Details of rooms

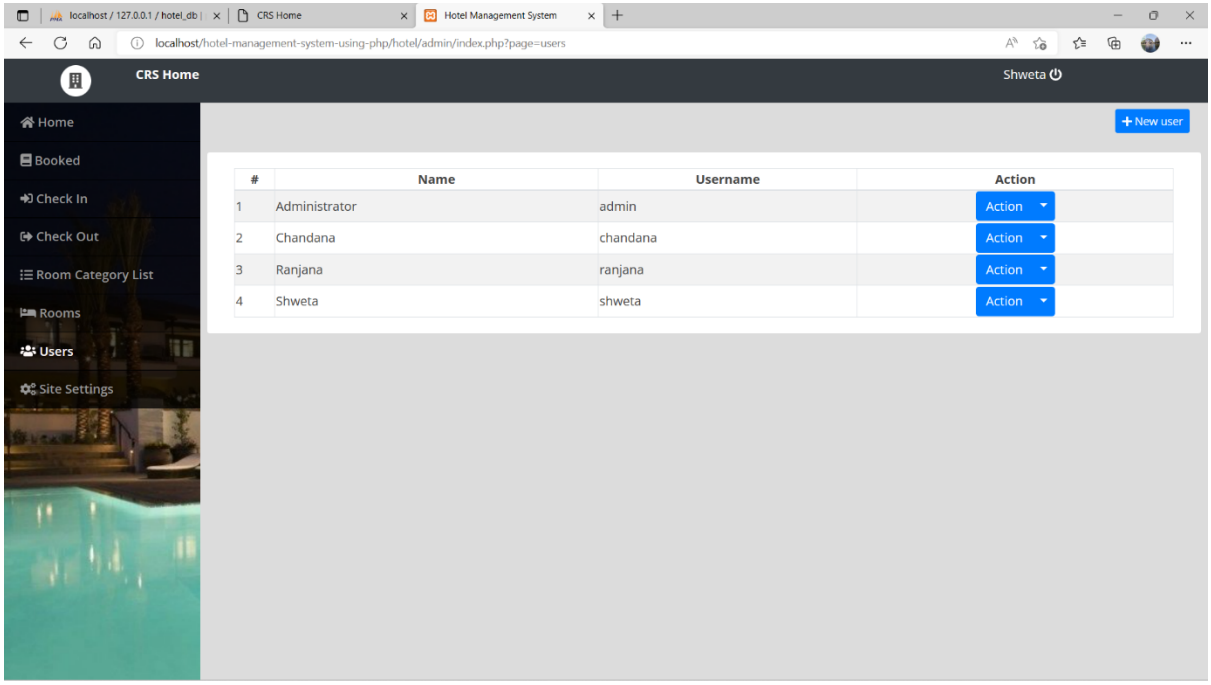

Figure 6.6: Admin in action

Chapter 7

Conclusion

Planned approach toward working: The maintenance of the hotel will be well-planned and organized. The data will be stored efficiently with optimal disk space consumption in data stores which will help in the retrieval of information as well as its storage under resource constraints.

Accuracy: The level of accuracy in the proposed system will be higher. All operations would conform to integrity constraints and correctness and it will be ensured that whatever information is received at or sent from the center is accurate.

Reliability: The reliability of the proposed system will be high due to the above-mentioned reasons. This comes from the fact that only the data which conforms accuracy clause would be allowed to commit back to the disk. Other properties like transaction management and rollback during system or power failure etc get automatically taken care of by the SQL systems, which is undoubtedly an excellent choice of the DBMS system. Properties of atomicity, consistency, isolation and data security are intrinsically maintained.

No redundancy: In the proposed system it will be ensured that no repetition of information occurs; neither on physical storage nor on a logical implementation level. This economizes on resource utilization in terms of storage space. Also, even in the case of concurrent access no anomalies occur and consistency is maintained. In addition to all this, principles of normalization have been endeavoured to be followed.

Immediate retrieval of information: The main objective of the proposed system is to provide a quick and efficient platform for retrieval of information. Queries allowed by the database.

References :

[1] Oracle Database MySQL PL/SQL 101- Christopher Allen (Oracle Press)
[2] Fundamentals of Database Systems, seventh edition, Elamasri Navathe.
[3]https://www.sourcecodester.com/php/14458/hotel-management-system-project-using-phpmysql.html
[4] https://code-projects.org/?s=hotel+management
[5] https://stackoverflow.com/questions/26236028/xampp-connect-to-SQL-server
[6] php - How to fix "Warning: mysqli::__construct(): (HY000/1045): Access denied for user 'user'@'localhost' (using password: YES)" on MacOS - Stack Overflow
[7] Research and Design of Hotel Management System Model | Atlantis Press (atlantis-press.com)
[8] Measuring Hotel Guest Satisfaction by Using an Online Quality Management System: Journal of Hospitality Marketing & Management: Vol 23, No 4 (tandfonline.com)
[9] https://www.tutorialspoint.com
[10] www.youtube.com
[10.1] https://youtu.be/F8J0sWFZNFQ
[10.2] https://youtu.be/Ed7wI7DF0pc
[10.3] https://youtu.be/xFEhNiSNeAg
[11] www.geeksforgeeks.com