DBMS Project

BY
SASHANK GOPALAN MYTREYAN

2022115051

2022115102

HOTEL MANAGEMENT

OBJECTIVE

Hotel management project provides room booking and other necessary hotel management features. The system allows the manager to post available rooms in the system. Customers can view and book room online. Admin has the power of either approving or disapproving the customer's booking request. Other hotel services can also be viewed by the customers and can book them too. The system is hence useful for both customers and managers to portable manage the hotel activities.

IMPLEMENTATION

In this Section we will do Analysis of Technologies to use for implementing the project.

FRONT END

HTML Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

CSS Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML.CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

JS JavaScript s a high-level, interpreted scripting language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,and major web browsers have a dedicated JavaScript engine to execute it.

BACK END

PL/PGSQL Postgresql modularity promotes code reusability, allowing developers to encapsulate specific operations, such as *room allocation, billing calculations, or guest check-in/check-out, in separate functions*. Transaction management ensures the consistency of the database, crucial for handling concurrent operations like booking modifications. PL/pgSQL also supports robust data validation and error handling, enhancing the system's reliability.

PostgreSQL is used for retrieve the details.It helps to keep track of *workers residents,accounts and generation of reports and also to maintain the day-to-day state of admission/Vacation of Residents, List of Workers , payment details etc*.It will help admin to handle customers information, room allocation details, payment details, billing information

---

PROCEDURES

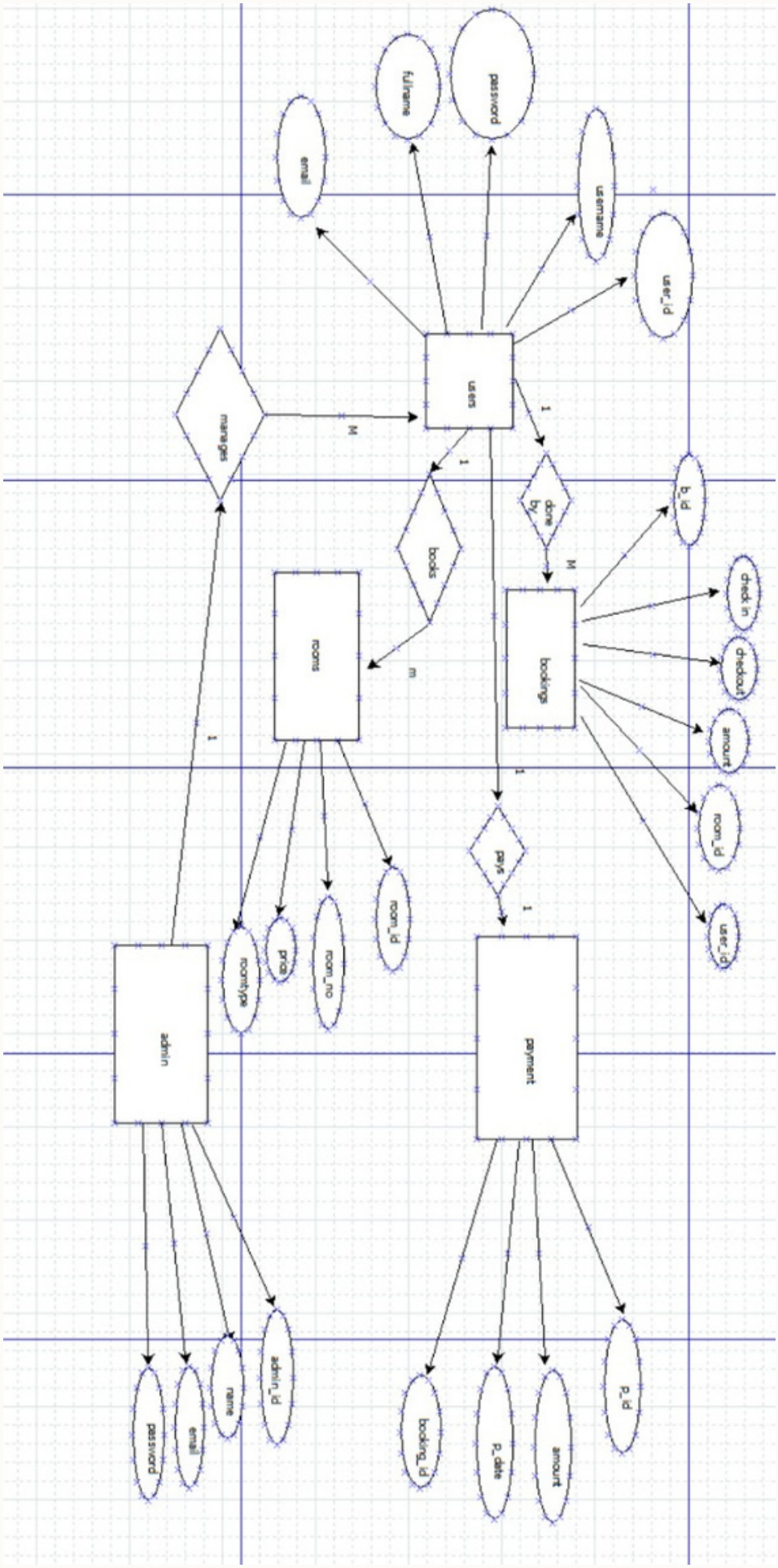
POSSIBLE TABLES:

The database schema encompasses several tables designed to manage and organize data for a Hotel Database Management System (DBMS).

1. Users Table:
   - Fields: id, name, address, phoneNum, email
   - Primary Key: id
2. HotelManager Table:
   - Fields: id, userName, password
   - Primary Key: id
   - Foreign Key: id references "User"(id)
   - Unique Constraint: userName
3. *Payment* Table:
   - Fields: id
   - Primary Key: id
   - Foreign Key: id references "User"(id)
4. HotelRoom Table:
   Fields: id, hotelID, roomNum, type, status
   Primary Key: id
5. Booking Table:
   Fields: id, hotelID, consumerID, start, end, PIN
   Primary Key: id
   Foreign Keys: userid references Users.

This comprehensive schema allows for the management of users, hotel owners, managers, consumers, hotels, room types, availability, discount rates, bookings, and booking records, facilitating a robust Hotel DBMS.

ER DIAGRAM


fullname
password
email
username
user_id
users
manages
M
1
1
done by
b_id
books
M
check in
rooms
bookings
checkout
m
amount
1
room_id
pays
1
user_id
roomtype
price
room_no
room_id
admin
payment
password
email
name
admin_id
booking_id
p_date
amount
p_id

3RD NORMAL FORM

Users Table:

- It is already in 1NF and 2NF since the primary key (id) uniquely identifies each record, and there are no partial dependencies.

HotelManager Table:

- It is already in 1NF and 2NF with a primary key (id) and a unique constraint on userName.

Payment Table:

- There is only one field (id) which is the primary key. It is already in 1NF and 2NF.

HotelRoom Table:

- It has a primary key (id) and does not contain any transitive dependencies.

Booking Table:

- It has a primary key (id) and does not contain any transitive dependencies.

Therefore, all the given tables satisfy the conditions of 3NF as they are already in 1NF and 2NF, and there are no transitive dependencies present.

SQL QUERY

CREATION:

```
-- Users Table
CREATE TABLE Users (
    id INT PRIMARY KEY,
    name VARCHAR(255),
    address VARCHAR(255),
    phoneNum VARCHAR(15),
    email VARCHAR(255)
);

-- HotelManager Table
CREATE TABLE HotelManager (
    id INT PRIMARY KEY,
    userName VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    FOREIGN KEY (id) REFERENCES Users(id)
);

-- Payment Table
CREATE TABLE Payment (
    id INT PRIMARY KEY,
    FOREIGN KEY (id) REFERENCES Users(id)
);

-- HotelRoom Table
CREATE TABLE HotelRoom (
    id INT PRIMARY KEY,
    hotelID INT,
    roomNum INT,
    type VARCHAR(50),
    status VARCHAR(50)
);

-- Booking Table
CREATE TABLE Booking (
    id INT PRIMARY KEY,
    hotelID INT,
    consumerID INT,
    start DATE,
    end DATE,
    PIN VARCHAR(6),
    FOREIGN KEY (consumerID) REFERENCES Users(id)
);
```

SQL QUERY

FUNCTIONS:

```
-- Function to calculate the total booked nights for a given booking
DELIMITER //

CREATE FUNCTION CalculateBookedNights(start_date DATE, end_date DATE)
RETURNS INT
BEGIN
    DECLARE total_nights INT;
    SET total_nights = DATEDIFF(end_date, start_date);
    RETURN total_nights;
END //

DELIMITER ;

-- Trigger to update HotelRoom status to "Booked" upon new booking
CREATE TRIGGER UpdateRoomStatusAfterBooking
AFTER INSERT ON Booking
FOR EACH ROW
BEGIN
    UPDATE HotelRoom
    SET status = 'Booked'
    WHERE id = NEW.hotelID AND roomNum = NEW.roomNum;
END;

-- Function to calculate the total amount paid by a consumer for a booking
DELIMITER //

CREATE FUNCTION CalculateTotalAmountPaid(consumer_id INT) RETURNS
DECIMAL(10,2)
BEGIN
    DECLARE total_amount DECIMAL(10,2);
    SELECT SUM(Payment.id) INTO total_amount
    FROM Payment
    WHERE Payment.id = consumer_id;
    RETURN total_amount;
END //

DELIMITER ;

-- Trigger to update Payment table upon new booking
CREATE TRIGGER UpdatePaymentAfterBooking
AFTER INSERT ON Booking
FOR EACH ROW
BEGIN
    INSERT INTO Payment (id)
    VALUES (NEW.consumerID);
END;
```

```
-- Function to get the number of available rooms in a hotel
DELIMITER //

CREATE FUNCTION GetAvailableRoomsCount(hotel_id INT) RETURNS INT
BEGIN
    DECLARE available_rooms INT;
    SELECT COUNT(*)
    INTO available_rooms
    FROM HotelRoom
    WHERE hotelID = hotel_id AND status = 'Available';
    RETURN available_rooms;
END //

DELIMITER ;

-- Trigger to update HotelRoom status to "Available" upon booking cancellation
CREATE TRIGGER UpdateRoomStatusAfterCancellation
AFTER DELETE ON Booking
FOR EACH ROW
BEGIN
    UPDATE HotelRoom
    SET status = 'Available'
    WHERE id = OLD.hotelID AND roomNum = OLD.roomNum;
END;
```

CONCLUSION

In conclusion, the designed Hotel Database Management System (DBMS) presents a comprehensive and structured solution for efficiently managing various aspects of a hotel's operations. The project encompasses multiple interconnected tables, each serving a specific purpose and contributing to the overall functionality of the system.