

Carmen Municipal College

A.Y. 2014-2025

**HOTELBOOKER**

CC105 – Information Management

Midterm Project

CALAMBA, LLOYD

CANDEL, KYLE

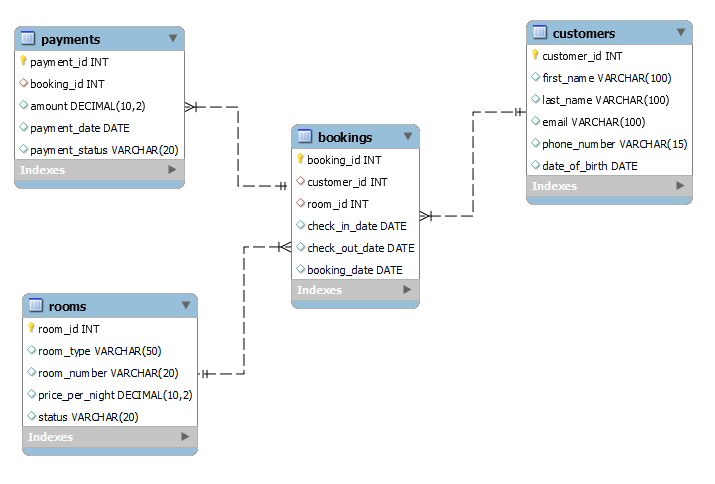
**I. Introduction**

**HotelBooker** is a multi-platform mobile and web application designed to streamline the hotel booking process. The system is tailored for travelers who wish to book rooms in hotels with ease and efficiency. HotelBooker targets users who are looking for a quick, reliable, and user-friendly way to secure accommodations for their trips.

Users can create and manage multiple bookings, view available rooms, and choose from a variety of room types that match their preferences. The system also allows users to filter their searches based on criteria such as room type, price, and availability, making it easier for them to find the best options. Additionally, HotelBooker enables users to modify or cancel bookings as necessary and track their payments in real time.

The app supports a seamless booking experience, with features such as automatic status updates (e.g., available, booked, maintenance) for rooms, real-time payment processing, and the ability to store customer data securely. HotelBooker aims to offer a convenient and flexible solution to travelers who need a hassle-free way to manage their hotel reservations.

**II. Entity Relationship Diagram**



**III. Database Tables**

**Table Name: Customers**

This table contains the personal information of the customer.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| customer\_id | INT | Unique identifier of the customer (Primary Key). |
| first\_name | VARCHAR(100) | First name of the customer. |
| last\_name | VARCHAR(100) | Last name of the customer. |
| email | VARCHAR(100) | Email address of the customer. |
| phone\_number | VARCHAR(15) | Phone number of the customer. |
| date\_of\_birth | DATE | Date of birth of the customer. |

**Table Name: Rooms**

This table contains details about the available rooms in the hotel.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| room\_id | INT | Unique identifier of the room (Primary Key). |
| room\_type | VARCHAR(50) | Type of room (e.g., Single, Double, Suite). |
| room\_number | VARCHAR(20) | Unique room number (must be unique). |
| price\_per\_night | DECIMAL(10,2) | Price per night for the room. |
| status | VARCHAR(20) | Status of the room (e.g., Available, Booked). |

**Table Name: Bookings**

This table stores information about the bookings made by customers.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| booking\_id | INT | Unique identifier for the booking (Primary Key). |
| customer\_id | INT | Foreign key referring to the customer. |
| room\_id | INT | Foreign key referring to the room. |
| check\_in\_date | DATE | Date when the customer checks in. |
| check\_out\_date | DATE | Date when the customer checks out. |
| booking\_date | DATE | Date when the booking was made. |

**Table Name: Payments**

This table stores payment details associated with bookings.

|  |  |  |
| --- | --- | --- |
| **Fieldname** | **Type (size)** | **Description** |
| payment\_id | INT | Unique identifier for the payment (Primary Key). |
| booking\_id | INT | Foreign key referring to the booking. |
| amount | DECIMAL(10,2) | Amount paid for the booking. |
| payment\_date | DATE | Date when the payment was made. |
| payment\_status | VARCHAR(20) | Status of the payment (e.g., Pending, Completed). |

**IV. Functionalities**

**Creating Tables:**

This code shows the creation of the necessary tables for the hotel booking system.

-- Create Customers Table

CREATE TABLE Customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(100),

last\_name VARCHAR(100),

email VARCHAR(100),

phone\_number VARCHAR(15),

date\_of\_birth DATE

) ENGINE=InnoDB;

-- Create Rooms Table

CREATE TABLE Rooms (

room\_id INT AUTO\_INCREMENT PRIMARY KEY,

room\_type VARCHAR(50),

room\_number VARCHAR(20) UNIQUE,

price\_per\_night DECIMAL(10, 2),

status VARCHAR(20) DEFAULT 'Available'

) ENGINE=InnoDB;

-- Create Bookings Table

CREATE TABLE Bookings (

booking\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

room\_id INT,

check\_in\_date DATE,

check\_out\_date DATE,

booking\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id) ON DELETE CASCADE,

FOREIGN KEY (room\_id) REFERENCES Rooms(room\_id) ON DELETE CASCADE

) ENGINE=InnoDB;

-- Create Payments Table

CREATE TABLE Payments (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

booking\_id INT,

amount DECIMAL(10, 2),

payment\_date DATE,

payment\_status VARCHAR(20) DEFAULT 'Pending',

FOREIGN KEY (booking\_id) REFERENCES Bookings(booking\_id) ON DELETE CASCADE

) ENGINE=InnoDB;

**Inserting Customer Information:**

This code represents inserting customer information into the **Customers** table.

INSERT INTO Customers (first\_name, last\_name, email, phone\_number, date\_of\_birth)

VALUES ('John', 'Doe', 'john.doe@example.com', '123-456-7890', '1990-05-15');

INSERT INTO Customers (first\_name, last\_name, email, phone\_number, date\_of\_birth)

VALUES ('Jane', 'Smith', 'jane.smith@example.com', '987-654-3210', '1985-10-25');

**Inserting Room Information:**

This code represents inserting information about rooms into the **Rooms** table.

INSERT INTO Rooms (room\_type, room\_number, price\_per\_night, status)

VALUES ('Single', '101', 100.00, 'Available');

INSERT INTO Rooms (room\_type, room\_number, price\_per\_night, status)

VALUES ('Double', '102', 150.00, 'Available');

INSERT INTO Rooms (room\_type, room\_number, price\_per\_night, status)

VALUES ('Suite', '103', 250.00, 'Booked');

**Inserting Booking Information:**

This code represents inserting booking information into the **Bookings** table.

INSERT INTO Bookings (customer\_id, room\_id, check\_in\_date, check\_out\_date, booking\_date)

VALUES (1, 2, '2025-03-10', '2025-03-15', '2025-03-01');

INSERT INTO Bookings (customer\_id, room\_id, check\_in\_date, check\_out\_date, booking\_date)

VALUES (2, 1, '2025-03-20', '2025-03-25', '2025-03-05');

**Inserting Payment Information:**

This code represents inserting payment information into the **Payments** table.

INSERT INTO Payments (booking\_id, amount, payment\_date, payment\_status)

VALUES (1, 750.00, '2025-03-01', 'Completed');

INSERT INTO Payments (booking\_id, amount, payment\_date, payment\_status)

VALUES (2, 500.00, '2025-03-05', 'Pending');

**Displaying Information:**

This code represents displaying filtered information based on different queries.

-- Display all customer information

SELECT \* FROM Customers;

-- Display all rooms that are 'Available'

SELECT \* FROM Rooms WHERE status = 'Available';

-- Display all rooms with a price per night greater than 150

SELECT \* FROM Rooms WHERE price\_per\_night > 150;

-- Display all bookings for customer with ID 1

SELECT \* FROM Bookings WHERE customer\_id = 1;

-- Display booking details along with customer name and room type

SELECT B.booking\_id, C.first\_name, C.last\_name, R.room\_type, B.check\_in\_date, B.check\_out\_date

FROM Bookings B

JOIN Customers C ON B.customer\_id = C.customer\_id

JOIN Rooms R ON B.room\_id = R.room\_id;

-- Display all payments made by customer with ID 1

SELECT \* FROM Payments WHERE booking\_id IN (SELECT booking\_id FROM Bookings WHERE customer\_id = 1);

-- Display total amount paid for bookings by customer with ID 1

SELECT SUM(amount) AS total\_paid FROM Payments WHERE booking\_id IN (SELECT booking\_id FROM Bookings WHERE customer\_id = 1);