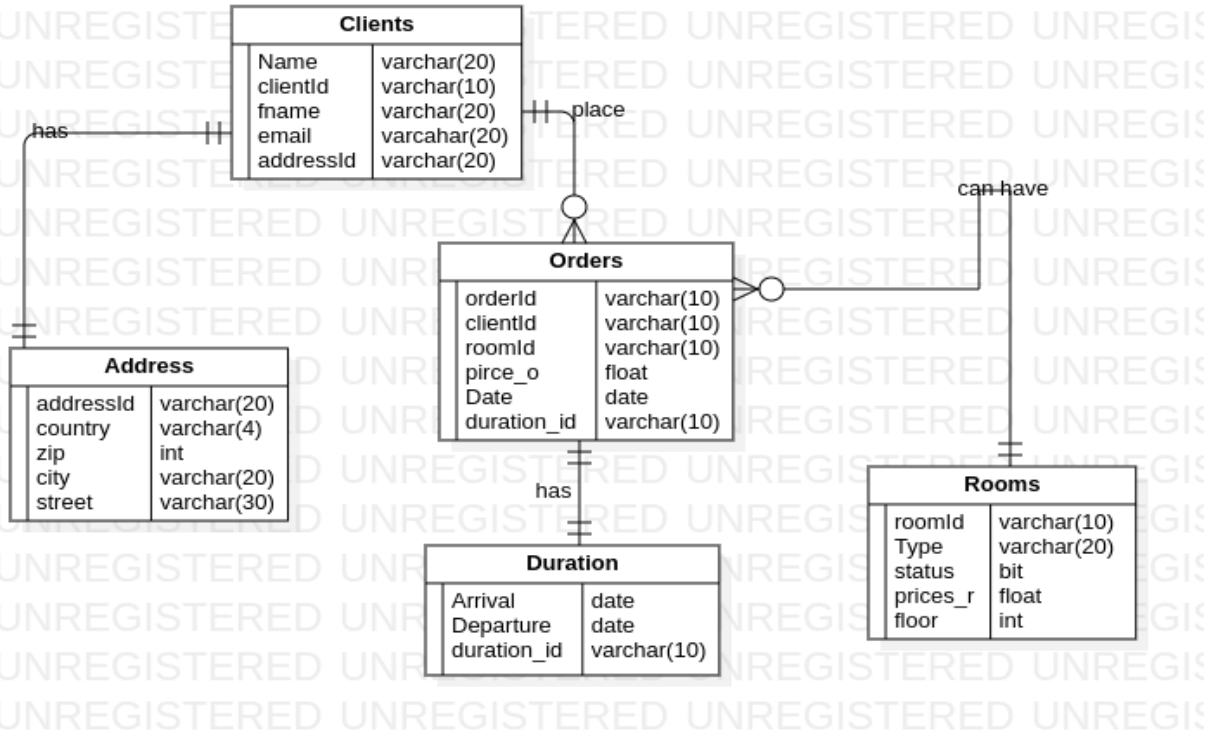
Hotel Management System



1. Hotel Management System

1.1.Database Planning

We can Divide our Hotel Management System Into Following Sections.

1.1.1 Mission Statement

"The purpose of Hotel Management System is to help out Stack-Holders in Manage them room bookings and optimize their profit. The system will keep track of the rooms and generate information related to finance, room booking, and Generate financial report."

1.1.2 Mission Objective of Database

Mission Objective of the database is

- To maintain (Enter, Update and Delete) data on rooms
- To maintain (Enter, Update and Delete) data on Customer
- To maintain (Enter, Update and Delete) data on Managers
- To maintain (Enter, Update and Delete) data on Customer Activities
- To maintain (Enter, Update and Delete) data on Bookings
- To maintain (Enter, Update and Delete) data on Finance
- To maintain (Enter, Update and Delete) data on Free Rooms
- To maintain (Enter, Update and Delete) data on Booked Rooms
- To maintain (Enter, Update and Delete) data on Room Cancellation
- To maintain (Enter, Update and Delete) data on Branch
- To maintain (Enter, Update and Delete) data on Employees
- To maintain (Enter, Update and Delete) data on Wages

- To maintain (Enter, Update and Delete) data on Viewing

- To Search on rooms
- To Search on Customer
- To Search on Finance
- To Search on Branch
- To Search on Employees

- To track on Rooms
- To track on Client
- To track on Free Rooms
- To track on Booked Rooms

- To report on rooms
- To report data on Customer
- To report on Bookings
- To report on Finance
- To report on Branch
- To report on Employees

1.1.3 CONCEPTUAL DESIGN

1.1.3.1 Identify entity types

Orders, Clients, Rooms And Time Duration, Address

1.1.3.2 Identify relationship types

Orders are placed by Client

Rooms are Booked by Client

Clients place the orders

Orders has valid Duration

Client has Address

1.1.3.3 Identifying and associating attributes

Orders:orderId, clientId, roomId, durationId,price_o, date

Clients: clientId, composite(name, fName), email, addressId

Rooms : roomId, type, status, prices_r, floor

Time Duration: durationId, arrival, departure
Address: addressId, country, zip, city, street

1.1.3.4 Attribute domains

orderId, has domain of int with auto increment

clientId, has domain of int with auto increment

roomId, has domain of int with auto increment

durationId, has domain int with auto increment

addressId, has domain of int with auto increment

prices_r, price_o, has domain of float numbers

arrival, departure, date, has domain of all possible date

city, type,country, composite(name, fName), has domain of varchar(20)

street, email, has domain of varchar(30)

status has bit domain with possible values of 1 or 0

zip,floor, has domain of int

1.1.3.5 Candidate, primary, and alternate key attributes

Primary keys:

orderId, is primary key for Orders table

clientId, is primary key for Clients table

roomId, is primary key for Rooms Table

durationId, is primary key for Duration table

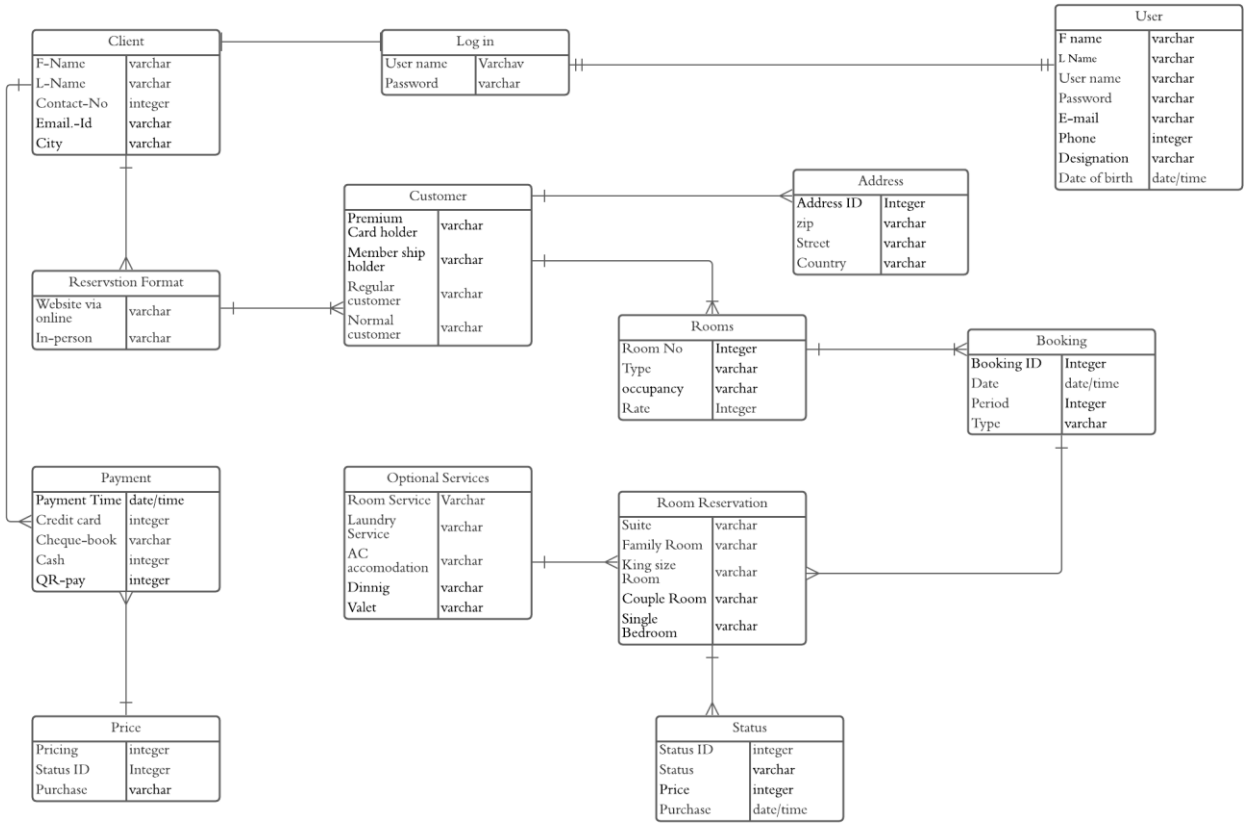
addressId, is primary key for Address table

Rest of attributes are alternate keys.

1.1.3.6 Model redundancy

All the relations are either one to one or one to many. The is not additional redundancy has been observe.

1.1.3.7



1.1.1.1 Sql Queries

```
#SARDAR BADAR, MUHAMMAD AWIAS, AND HASSAN ALI JADOON
CREATE DATABASE hotma;
use hotma;
CREATE TABLE Clients (
 clientId int IDENTITY(1,1) PRIMARY KEY,
 name varchar(20) not Null,
 fName varchar(20) Not Null,
 email varchar(20) unique not NULL,
);

CREATE TABLE [Rooms] (
 [roomId] int not null  PRIMARY KEY ,
 [type] varchar(20) not null,
```

```
  [status] bit not null,
  [price_r] float not null,
  [floor] int not null

);



CREATE TABLE [Address] (
  [addressId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [clientId] int unique NOT NULL FOREIGN KEY REFERENCES Clients(clientId),
  [country] varchar(4) not null,
  [zip] int not null,
  [city] varchar(20) not null,
  [street] varchar(30) not null

);

CREATE TABLE [Orders] (
  [orderId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [clientId] int NOT NULL FOREIGN KEY REFERENCES  Clients(clientId),
  [roomId] int  NOT NULL FOREIGN KEY REFERENCES Rooms(roomId),
  [price_o] float not null,
  [date_o] date not null

);

CREATE TABLE [Duration] (
  [durationId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [arrival] date not null,
  [departure] date not null,
```

```
 [orderId] int unique NOT NULL FOREIGN KEY REFERENCES Duration(orderId)



);









--INSERTION QUERIES FOR
use hotma;
INSERT into Clients (
 name,
 fName,
 email
) values
('Adil','Nawaz', 'adilna@gmail.com'),
('Adl','Nawaz', 'adilw@gmail.com'),
('Ad','awaz', 'adil@gmail.com');



INSERT into[Address] (
 [clientId] ,
 [country] ,
 [zip],
 [city] ,
 [street]


) values (3, 'pk',22340,'london' ,'london streets');
```

```
--INSERT DATA IN ROOMS
INSERT into[Rooms] (
 [roomId],
 [type],
 [status] ,
 [price_r],
 [floor]

)Values(2,'First Class', 0, 100.4, 2),
(1,'First Class', 0, 100022.4, 2),
(3,'First Class', 0, 10030.4, 34);

--Insert Order
INSERT into [Orders] (
 [clientId],
 [roomId] ,
 [price_o] ,
 [date_o]
)SELECT clientId , roomId, price_r, '2009-11-11' FROM  Clients c, Rooms r  where clientId =2 AND roomId=3
AND status = 0;

--Change Status
UPDATE Rooms
SET status=1 where roomId =1;

--GET FINANCE QUERY WTITH AWAIS, BADAR, AND HASSAN
SELECT  DISTINCT a.roomId as roomNO , b.maxPrice as highestPrice, r.price_r as currentPrice,
b.revenu as revenu,
b.countr as booking
FROM  ((Orders a
     INNER JOIN
```

```
(
  SELECT
  MAX(price_o) as maxPrice,
  sum(price_o) as revenu,
  COUNT(o2.roomId) as countr,
  o2.roomId
  FROM Orders o2
  GROUP BY o2.roomId
) b ON a.roomId = b.roomId AND
    a.price_o = b.maxPrice)
INNER JOIN Rooms r on  r.roomId = a.roomId) WHERE a.date_o ='2009-11-11';
```

1.1.4 Views

There is only one view in our Systems

1.2.Task Assign to Members

1.2.1 Abdul Haseeb Khan And Khanzada Haider Ali:

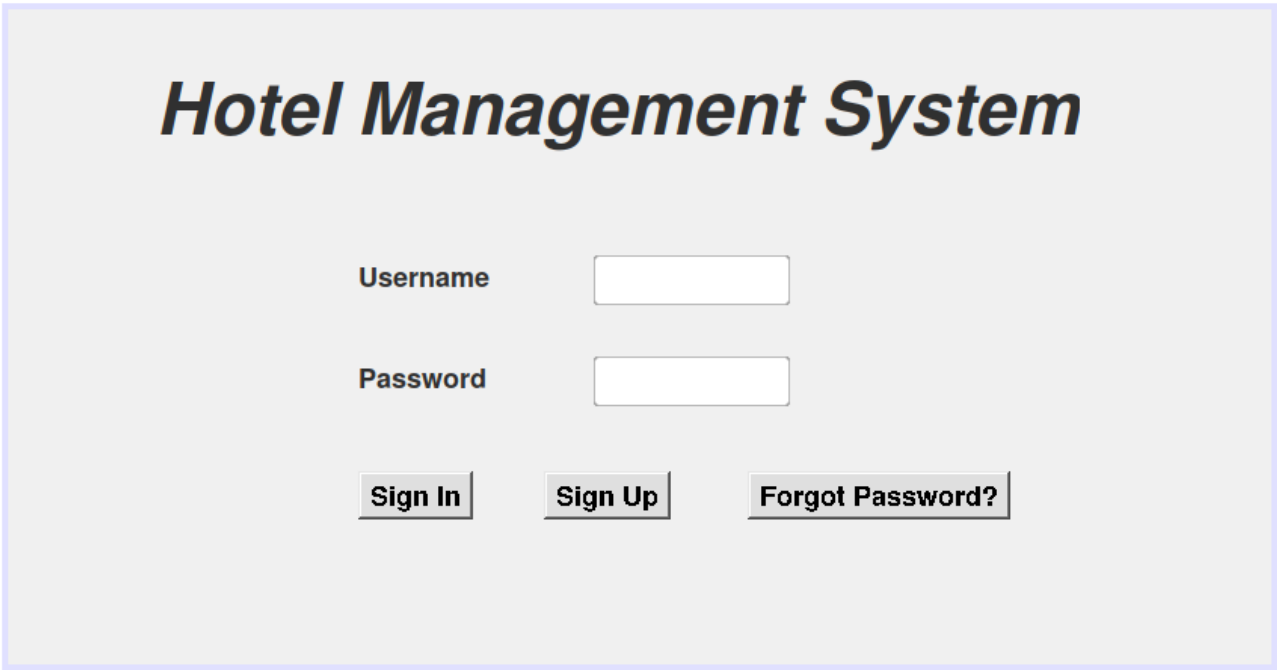
Abdul Haseeb Khan and Khanzada Hadier will design and develop the Login Functionalites in Our Hotel Managemnt System by interacting with database to validate the user. This task does not include any java program or moduling. It's purely SQL related. We will be able to finish our tasks in the project as soon as we learn the queries required to do our task. Our task includes the handling of making of new Ids

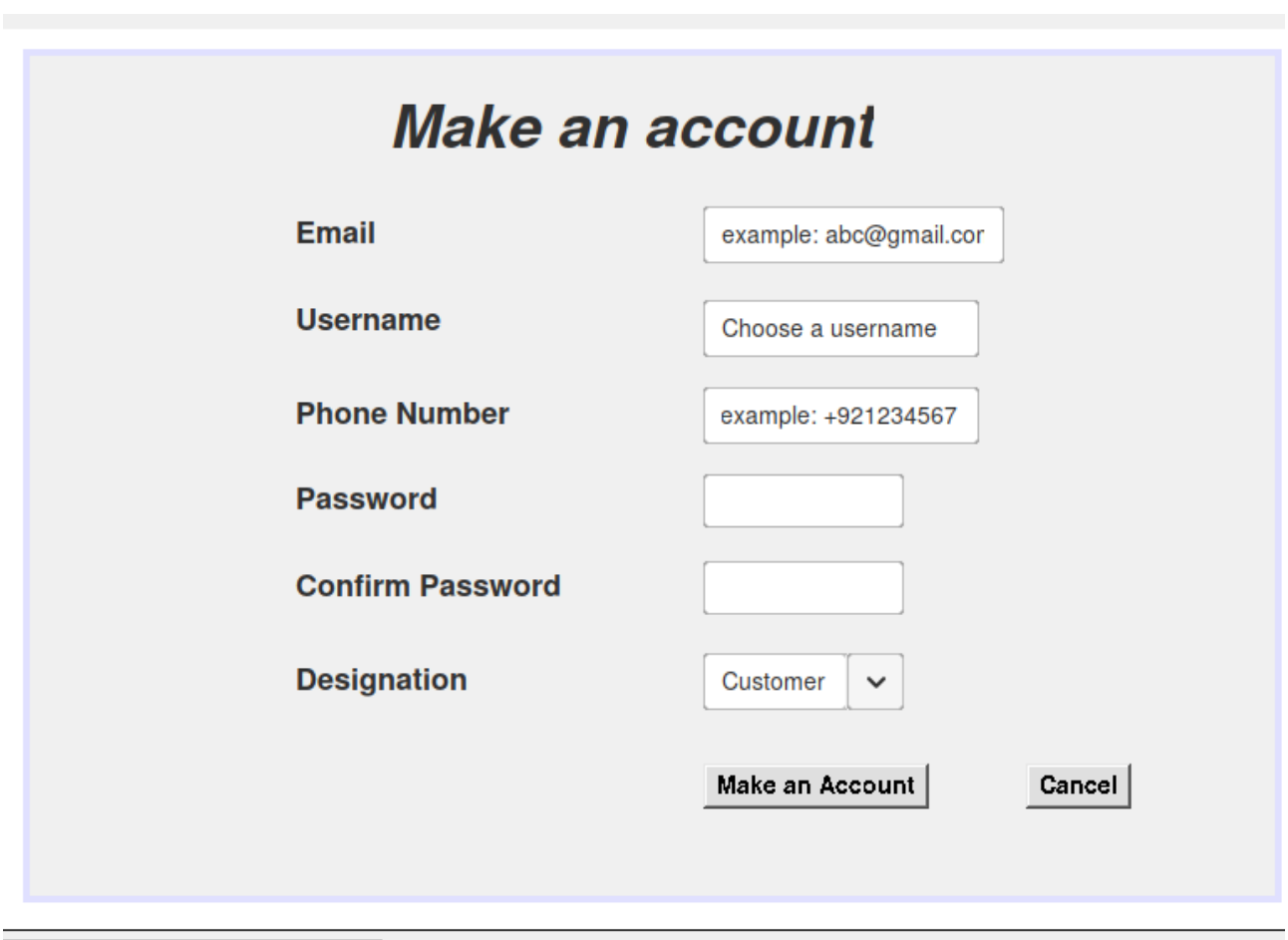
by the users. This includes first time making of a new id which includes unique usernames as provided by the user and the password. After making of the new ids the users can log in through the data they provided. This will include cross checking of the usernames and passwords across the ids of the users. Our work will take almost a week if we know the certain queries which we will learn as the classes go by.

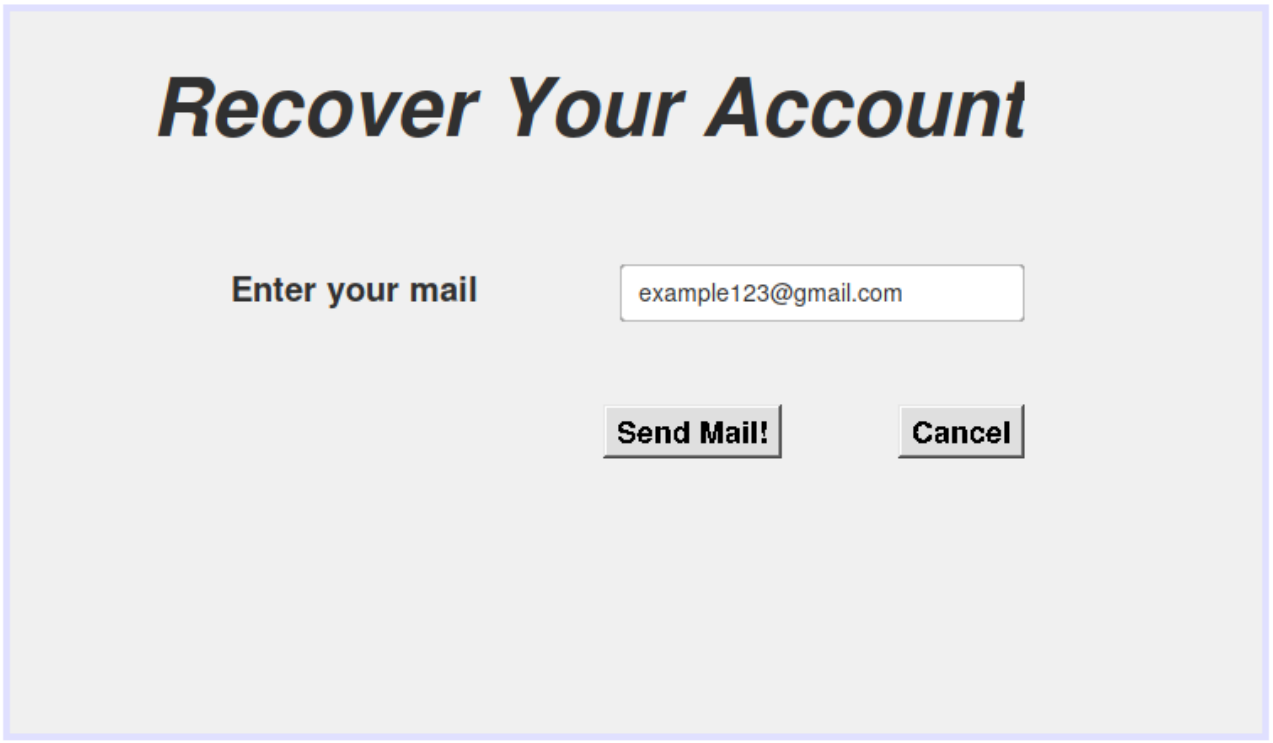
Mission Objectives:

- To maintain (Enter, Update and Delete) data on Customers
- To maintain (Enter, Update and Delete) data on Managers
- To maintain (Enter, Update and Delete) data on Employees

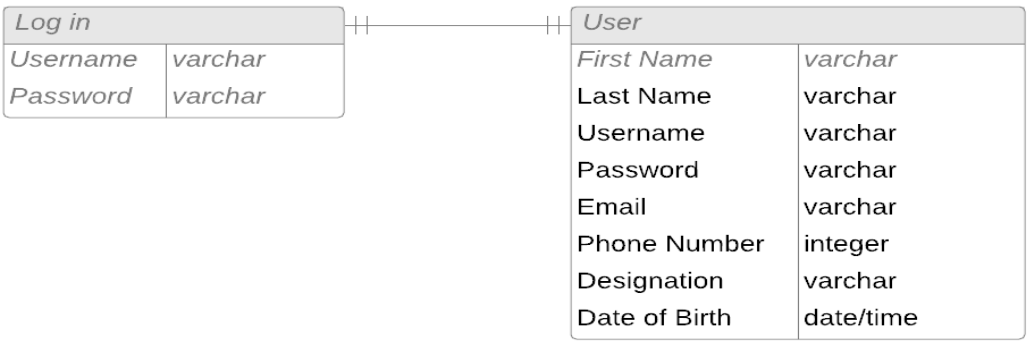
MAJOR USER VIEW (UI PROTOTYPES)




Make an account
Email
example: abc@gmail.cor
Username
Choose a username
Phone Number
example: +921234567
Password
Confirm Password
Designation
Customer
Make an Account
Cancel


Recover Your Account
Enter your mail
example123@gmail.com
Send Mail!
Cancel

ER DIAGRAM:


Log in
Username varchar
Password varchar
User
First Name varchar
Last Name varchar
Username varchar
Password varchar
Email varchar
Phone Number integer
Designation varchar
Date of Birth date/time

1.2.2 Ashfaq Rahim And Hasssan Shah Nawaz

Mission objective modules of our project:

Develop and design the dashboard functionalities after user will login the system.

SCOPE:

We want to choose the metrics that matter. We want to keep it visual. We want to make it interactive for collaboration. We want to pull data from all sources to get the full picture and keep the dashboards up to date or refreshed. And, finally, we want to make it simple to access and easy to use.

TIME AND COMPLEXITY:

The time taken for developing and designing dashboard is 2 months. Building an effective dashboard according to best practices for dashboard design is the culmination of a comprehensive process that would usually include gathering requirements, defining, and creating a data model. However, the importance of proper dashboard design should not be understated. Poorly designed dashboards could fail to convey useful information and insights and even make the data less comprehensible than it was originally.

MISSION STATEMENT:

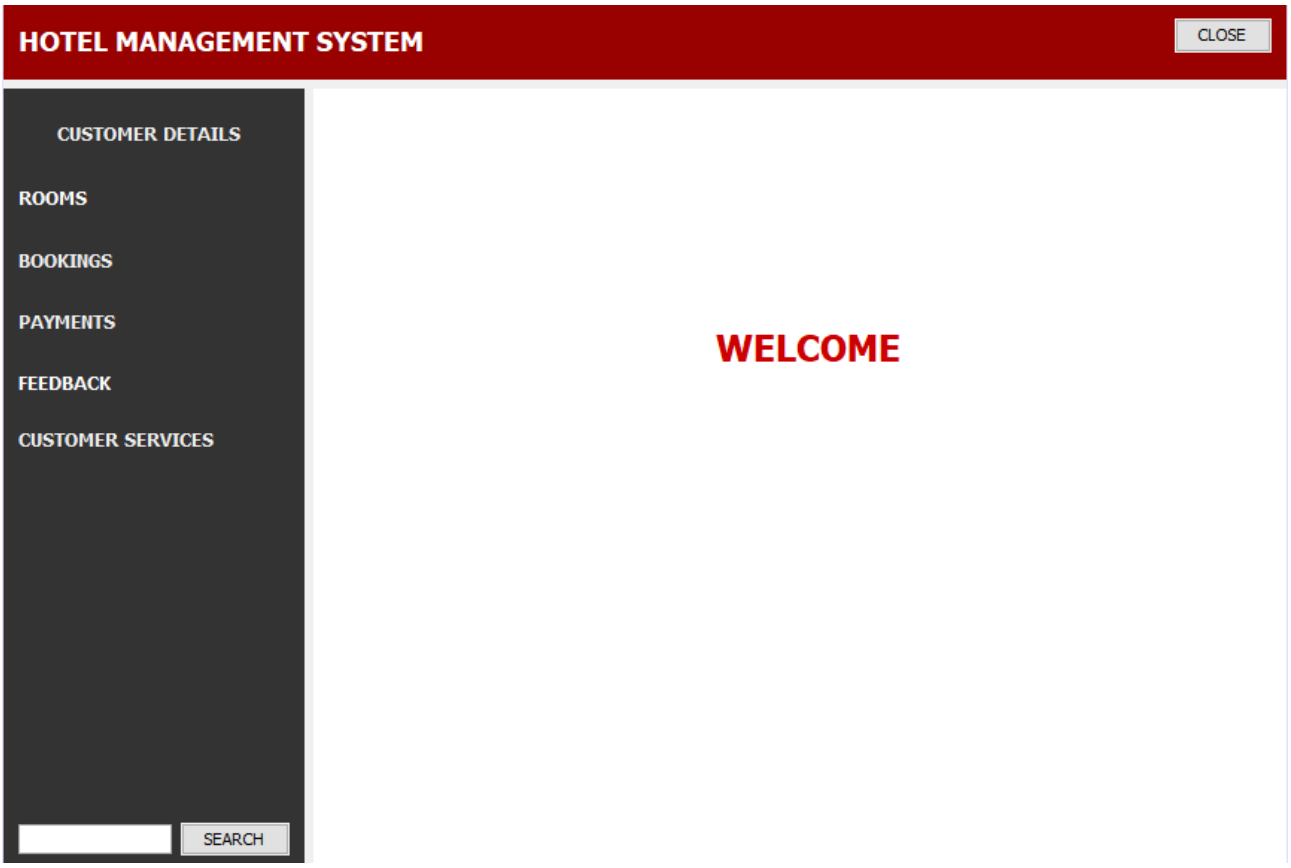
The mission of our hotel is to provide outstanding lodging facilities and services to our guests. Our hotel focuses on individual business and leisure travel, as well as travel associated with groups meetings. we emphasize high quality standards in our rooms and food and beverage divisions. We provide a fair return on investment for our owners and recognize that this cannot be done without well trained , motivated and enthusiastic employees.

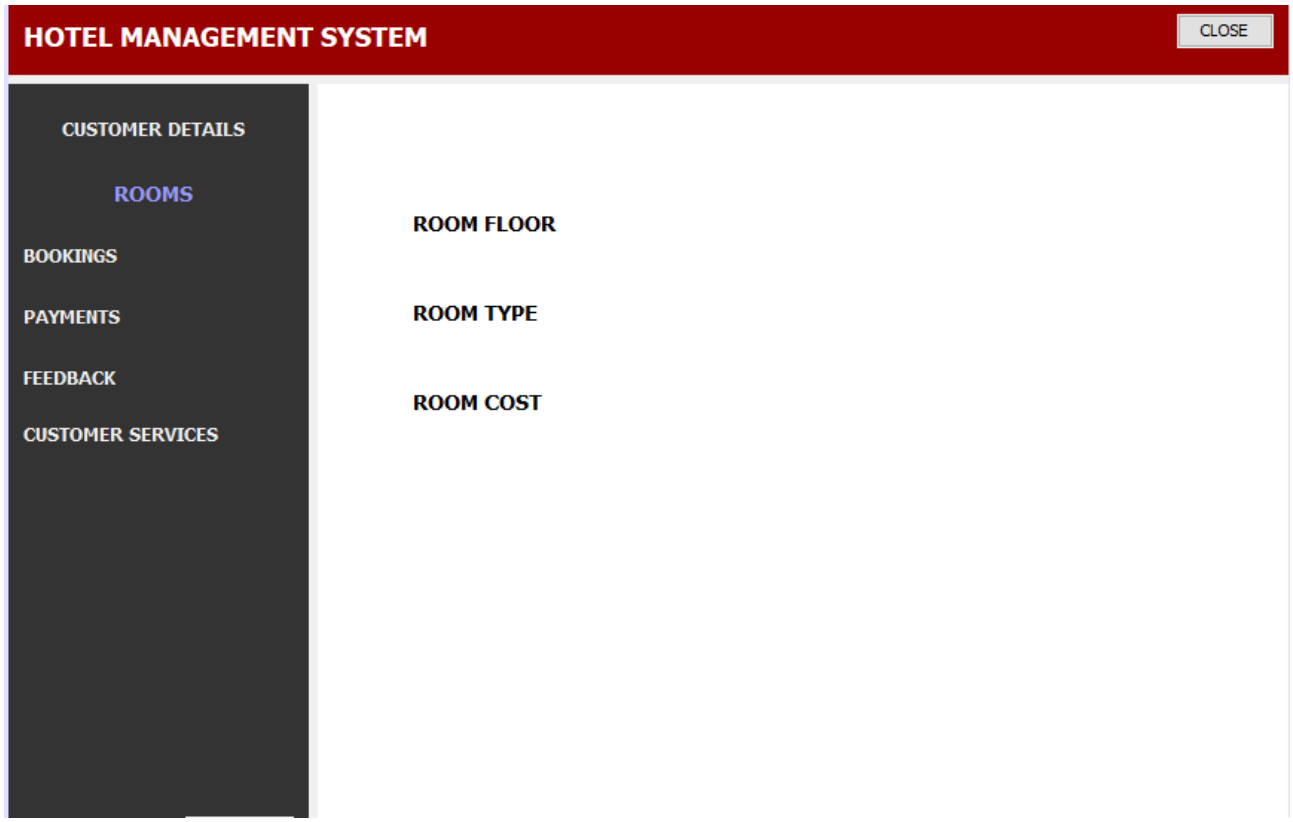
MISSION OBJECTIVE:

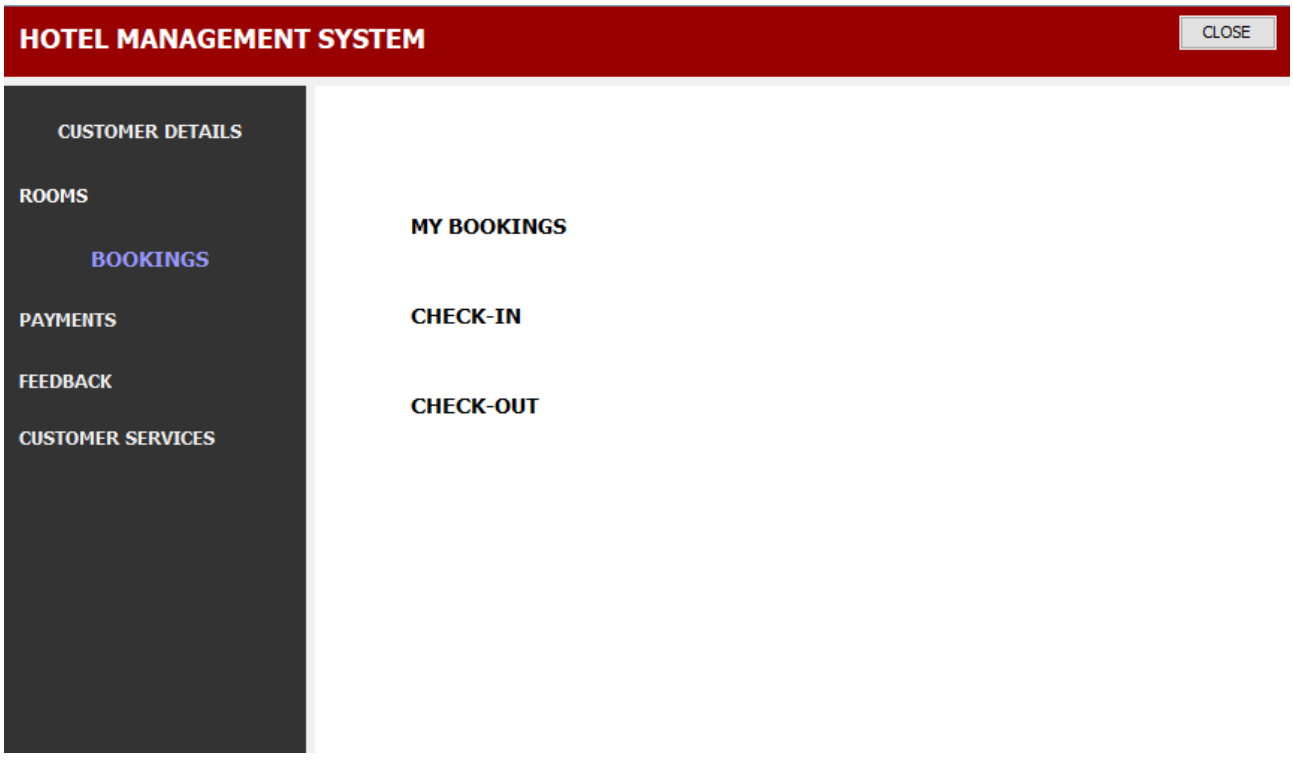
- To maintain (Enter, Update and Delete) data on Rooms
- To maintain (Enter, Update and Delete) data on Bookings
- To maintain (Enter, Update and Delete) data on Payment

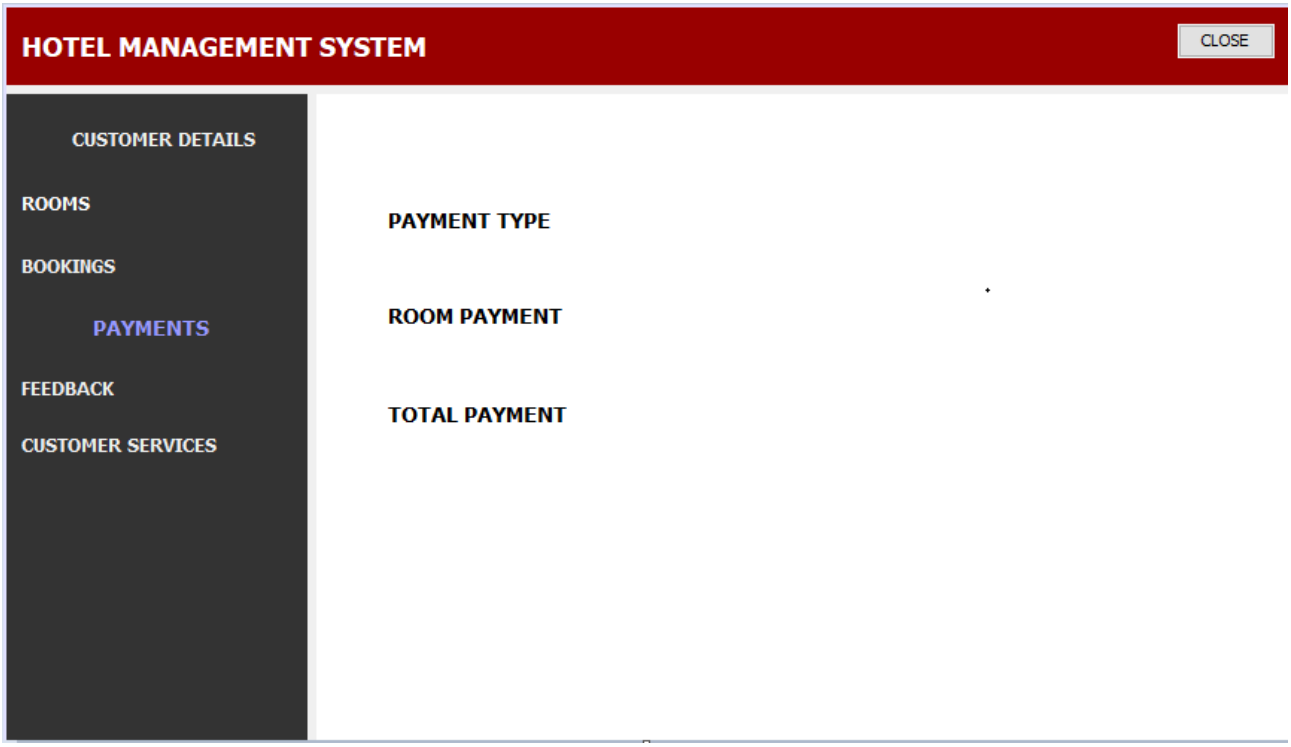
- To Search on rooms
- To Search on Bookings
- To Search on Payment
- To track on Rooms
- To track on Bookings
- To track on Payments
- To report on rooms
- To report on Bookings
- To report on Payments

MAJOR USER VIEW (UI PROTOTYPE):




HOTEL MANAGEMENT SYSTEM
CLOSE
CUSTOMER DETAILS
ROOMS
BOOKINGS
PAYMENTS
FEEDBACK
CUSTOMER SERVICES
ROOM FLOOR
ROOM TYPE
ROOM COST


HOTEL MANAGEMENT SYSTEM
CLOSE
CUSTOMER DETAILS
ROOMS
BOOKINGS
PAYMENTS
FEEDBACK
CUSTOMER SERVICES
MY BOOKINGS
CHECK-IN
CHECK-OUT



CONCEPTUAL DESIGN

Identify entity types

customers details, booking, rooms, payment and feedback

Identify relationship types

bookings are placed by customers

Rooms are booked by customers

customers place the booking

customers have done payment

customers have feedback

Identifying and associating attributes

Customers:customerid,name,mobile,email,age,gender

Booking:bookingid,date,period,type

Rooms:room_no.type,occupancy,fee

Payment:bill_no.,paymentid,paymentdate,paymentmode

Feedback:feedbackid,yes,no,date

Attribute domains

Customerid has domain of int with auto increment

Name has domain of varchar

Mobile has domain of int

Email has domain of varchar

Age has domain of int

gender has domain of varchar

bookingid has domain of int with auto increment

date has domain of date/time

period has domain of int

type has domain of int varchar

room_no. has domain of int

type has domain of varhar

occupancy has domain of varchar

fee has domain of int

bill_no. has domain of int with auto increment

paymentid has domain of int with auto increment

paymentdate has domain of date/time

paymentmode has domain of varchar

feedbackid has domain of int with auto increment

yes has domain of varchar

no has domain of varchar

date has domain of date/time

Candidate, primary, and alternate key attributes

Primary keys:

customerid is primary key for Orders table

room_no. is primary key for Clients table

feedbackid is primary key for Rooms Table

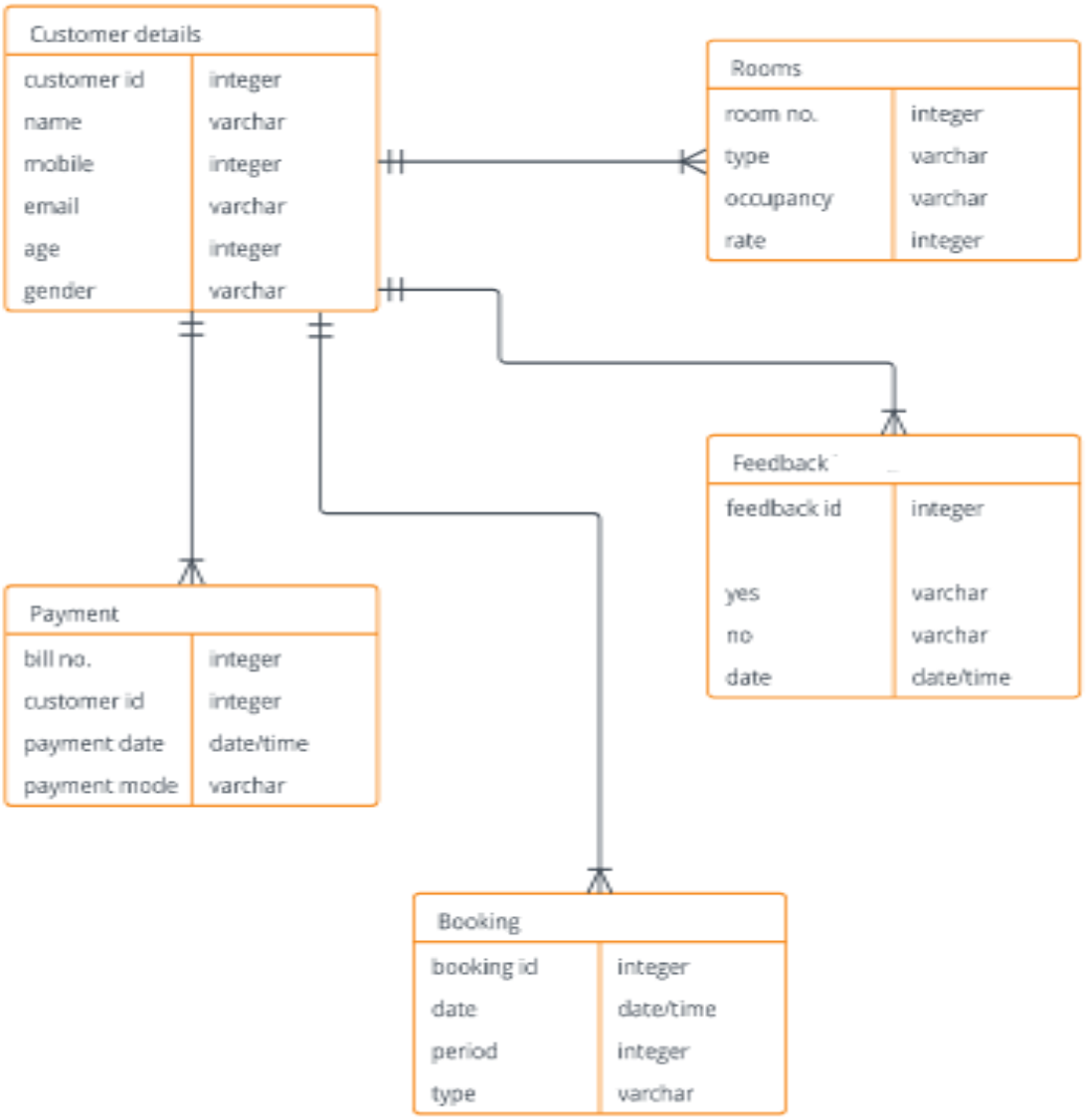
bill_no. is primary key for Duration table

Rest of attributes are alternate keys.

Model redundancy

We have change data type of the PK so it will be easy to work with less redundancy. So All the relations are either one to one or one to many. The is not additional redundancy has been observe.

DASHBOARD ER DIAMGRAM:



SQL QUERIES

```
CREATE DATABASE hotel;
use hotel;
CREATE TABLE coustomersdetails(
  customerId int IDENTITY(1,1) PRIMARY KEY,
  name varchar(20) not Null,
  mobile int not null,
  email varchar(20) unique not NULL,
  age int not null,
  gender varchar(20), not null
);

CREATE TABLE [Rooms] (
  [room_no.] int not null    PRIMARY KEY ,
  [type] varchar(20) not null,
  [occupany] varchar not null,
  [fee] integer not null,
```

```
);



CREATE TABLE [feedback] (
  [feedbackId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [date] date/time NOT NULL,
  [yes] varchar(20),
  [no] varchar(20)

);

CREATE TABLE [booking] (
  [bookingId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [period] int NOT NULl,
  [type] varchar(20) not null,
  [date] date/time not null

);

CREATE TABLE [payment] (
  [bill_no.] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [paymentid] date not null,
  [paymentmode] date not null,
  [paymentdate] int unique NOT NULL


);




--INSERTION QUERIES FOR
use hotel;
INSERT into customersdetails (
  name,
  mobile,
  email,
  age,
  gender
) values
('shah',12663822, 'shah@gmail.com',21,'male'),
('farooq',12223822, 'farooq@gmail.com',22,'male'),
('karim',12553822, 'karim@gmail.com',23,'male');


INSERT into[payment] (
  [bill_no.] ,
  [paymentid] ,
  [paymentdate],
  [paymentmode]

) values
('gfg128', '2020-2-2','credit card'),
('gfg123', '2020-3-3','credit card'),
('gfg125', '2020-4-4','credit card');


INSERT into[Rooms] (
```

```
 [room_no.],
  [type],
  [occupany] ,
  [fee]

)Values
(212,'First Class','first floor', 10453),
(121,'First Class','third floor', 104343),
(312,'First Class','second floor', 1003443);


INSERT into [booking] (
  [bookingId],
  [date] ,
  [period] ,
  [type]
) value

('2020-2-2',22,'first class'),
('2020-3-3',11,'first class'),
('2020-4-4',4,'first class');

INSERT into [feedback] (
  [feedbackId],
  [yes],
  [no],
  [date]
) value
('yes','no','2020-2-2'),
('yes','no','2020-3-3'),
('yes','no','2020-4-4');



SELECT cutomerId , room_no., '2020-4-4' FROM  customersdetails, Rooms  where customerId = 3
AND room_no.= 1;

--Change Status
UPDATE Rooms
SET date='2020-5-5' where room_no. = 3;
```

1.2.3 Fahad Afzal & Imad Ahmed

Task:

Imad and Fahad must design and develop the room status and selection that will include available rooms, new client's insertion part, available and booked rooms and the price of rooms and client information related part.

Module:

Clients Residential module.

Scope:

We will create client`s residential system. This module will help the client to make his/her choice of reservation easy. It will provide the client with the necessary information about suite and rooms which he/she needs to know.

Development Time:

This module will take about three (1) and a half month (1/2) to complete.

Mission Statement:

The purpose of Hotel Management System is to help stack holders in manage them room bookings and optimize their profit. The system will keep track of the rooms and generate information related to finance, room bookings and generate financial report.

Mission Objectives:

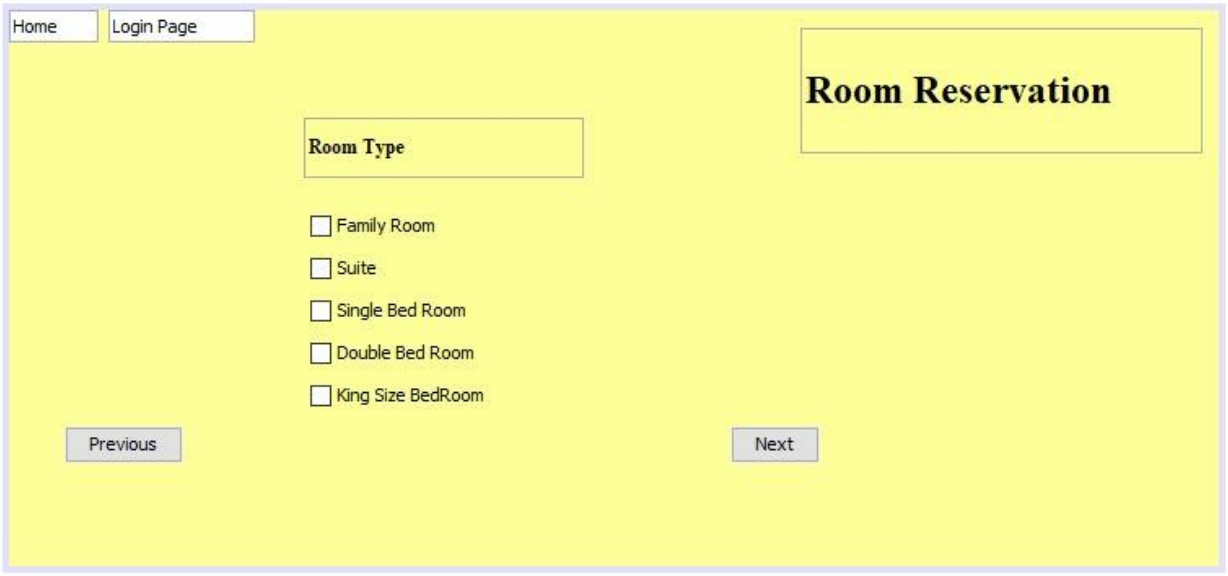
- To maintain (Enter, Update and Delete) data on rooms.
- To maintain (Enter, Update and Delete) data on Customer activities.
- To maintain (Enter, Update and Delete) data on Bookings
- To maintain (Enter, Update and Delete) data on Free Rooms
- To maintain (Enter, Update and Delete) data on Booked Rooms
- To maintain (Enter, Update and Delete) data on Room Cancellation
- To Search on rooms
- To Search on Customer
- To track on Rooms

- To track on Client
- To track on Free Rooms
- To track on Booked Rooms
- To report on rooms
- To report on Bookings

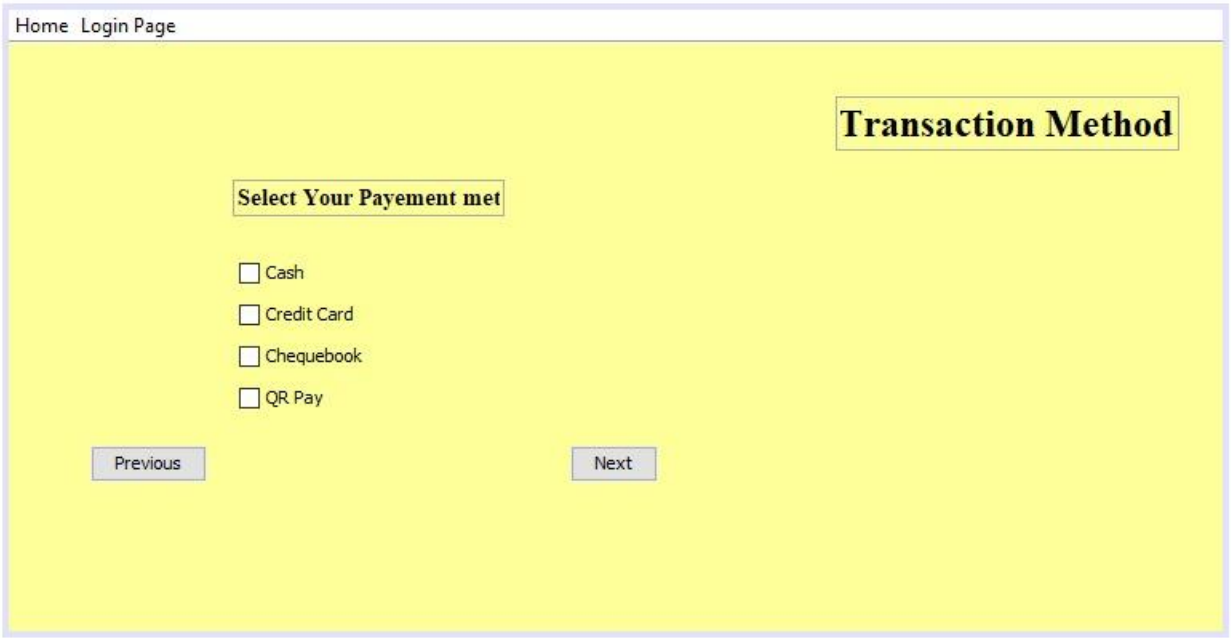
Major User Views (UI Prototypes):

SCREEN # 01

SCREEN # 02

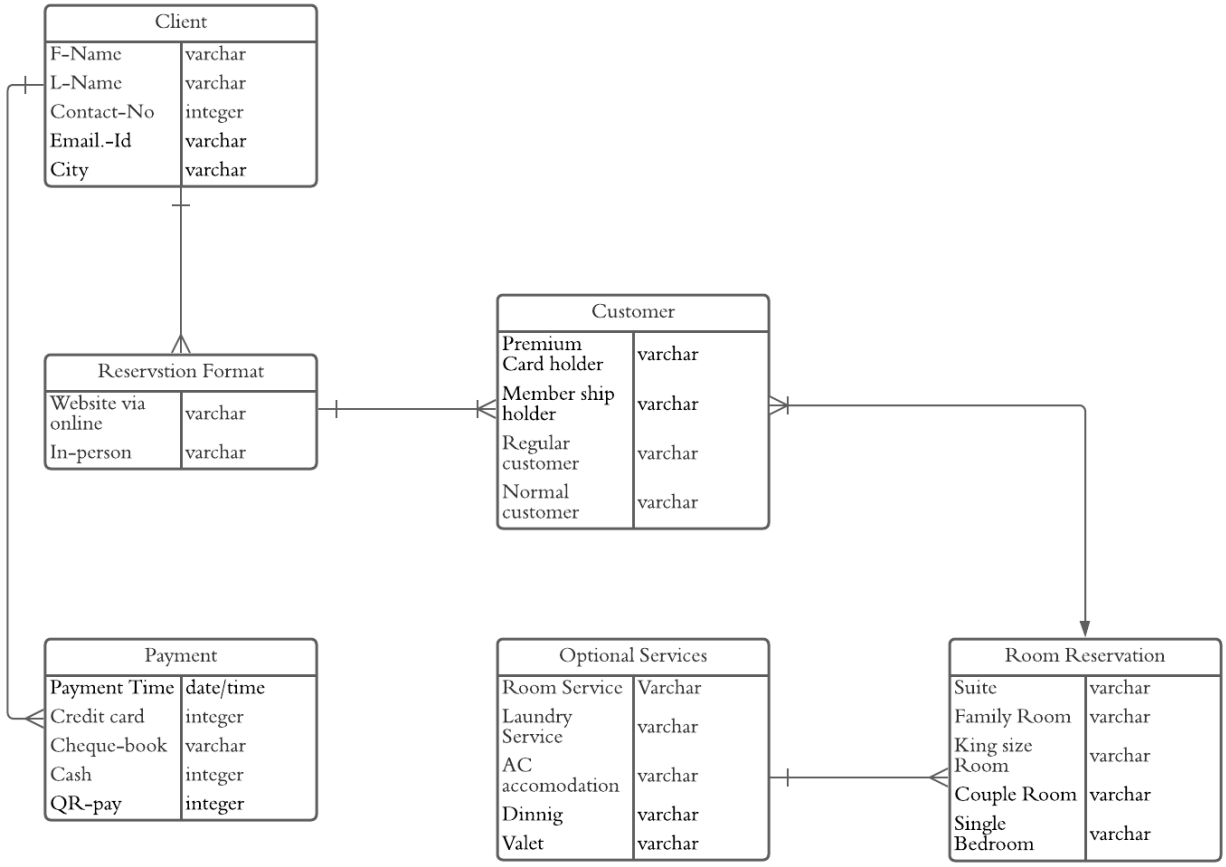


SCREEN # 03



https://drive.google.com/drive/folders/1cusMrL4aN9Ka1rHQJHrPJGaGrRKy1nZS?usp=sharing

ER-Diagram:



…….xxxxxxxxxxTHE-ENDxxxxxxxxxx…….

1.2.4 Hassan Ali Jadoon, Awais And Sardar Badar

1.2.4.1 Scope:

We will create financial module of the app. This module will generate weekly, monthly and annual Financial report of the hotel. It will also calculate the revenue generated by the rooms. In the end we will also integrate all modules of hotel management system.

1.2.4.2 Development Time:

This module requires about 1 months to be completed

1.2.4.3 Complexity:

We will use Swing Framework of Java as assigned by the instructor. Beside that we have to learn Object Oriented concepts of Java. The main issue with the swing framework that it does not have wide community thus we will have to face many issues during the learning phase, also it does not have extensive tutorials.

1.2.4.4 Mission Statement

".The system will keep track of the rooms and generate information related to finance, Generate financial report."

1.2.4.5 Mission Objective of Database:

Mission Objective of the database is

1.2.4.5.1 Badar Mission Objectives:

"By Sardar Badar Saghir"

- To maintain (Enter, Update and Delete) data on Free Rooms
- To maintain (Enter, Update and Delete) data on Booked Rooms
- To maintain (Enter, Update and Delete) data on Room Cancellation
- To report on Annual Finance

1.2.4.5.2 Awais Mission Objectives

"By Muhammad Awais"

- To maintain (Enter, Update and Delete) data on Customer
- To maintain (Enter, Update and Delete) data on Orders
- To maintain (Enter, Update and Delete) data on Wages

- To report on Annual Finance

1.2.4.5.3 Hasssan Mission Objectives:

By Hassan Ali Jadoon.

- To maintain (Enter, Update and Delete) data on rooms
- To track on Rooms
- To track on Free Rooms
- To report on Annual Finance

1.2.4.6 Major Views:

For the sake of simplicity, we are dealing with one major view.

1.2.4.7 Conclusion:

Although we deal with data financial module of app but it will take about 1 months due to wide range of complexity we will face learning framework like swing and OOP concepts as well

1.2.4.8 CONCEPTUAL DESIGN

1.2.4.8.1 Identify entity types

"By Sardar Badar Saghir"

Orders, Clients, Rooms and Time Duration, Address

1.2.4.8.2 Identify relationship types

"By Hassan Ali Jadoon"

Orders are placed by Client

Rooms are Booked by Client

Clients place the orders

Orders has valid Duration

Client has Address

1.2.4.8.3 Identifying and associating attributes

“By Muhammad Awais”

Orders:orderId, clientId, roomId, durationId,price_o, date

Clients: clientId, composite(name, fName), email, addressId

Rooms : roomId, type, status, prices_r, floor

Time Duration: durationId, arrival, departure
Address: addressId, country, zip, city, street

1.2.4.8.4 Attribute domains

“By Sardar Badar Saghir”

orderId, has domain of int with auto increment

clientId, has domain of int with auto increment

roomId, has domain of int with auto increment

durationId, has domain of int with auto increment

addressId, has domain of int with auto increment

prices_r, price_o, has domain of float numbers

arrival, departure, date, has domain of all possible date

city, type,country, composite(name, fName), has domain of varchar(20)

street, email, has domain of varchar(30)

status has bit domain with possible values of 1 or 0

zip,floor, has domain of int

1.2.4.8.5 Candidate, primary, and alternate key attributes

“By Muhammad Awais”

Primary keys:

orderId, is primary key for Orders table

clientId, is primary key for Clients table

roomId, is primary key for Rooms Table

durationId, is primary key for Duration table

addressId, is primary key for Address table

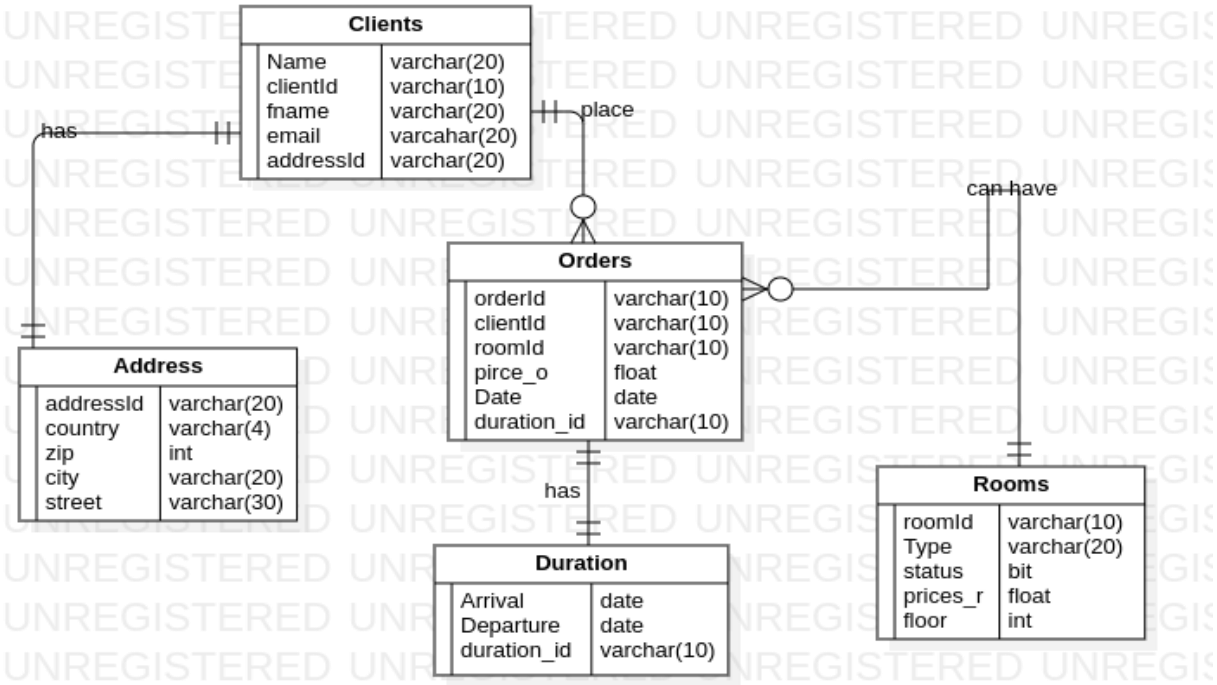
Rest of attributes are alternate keys.

1.2.4.8.6 Model redundancy

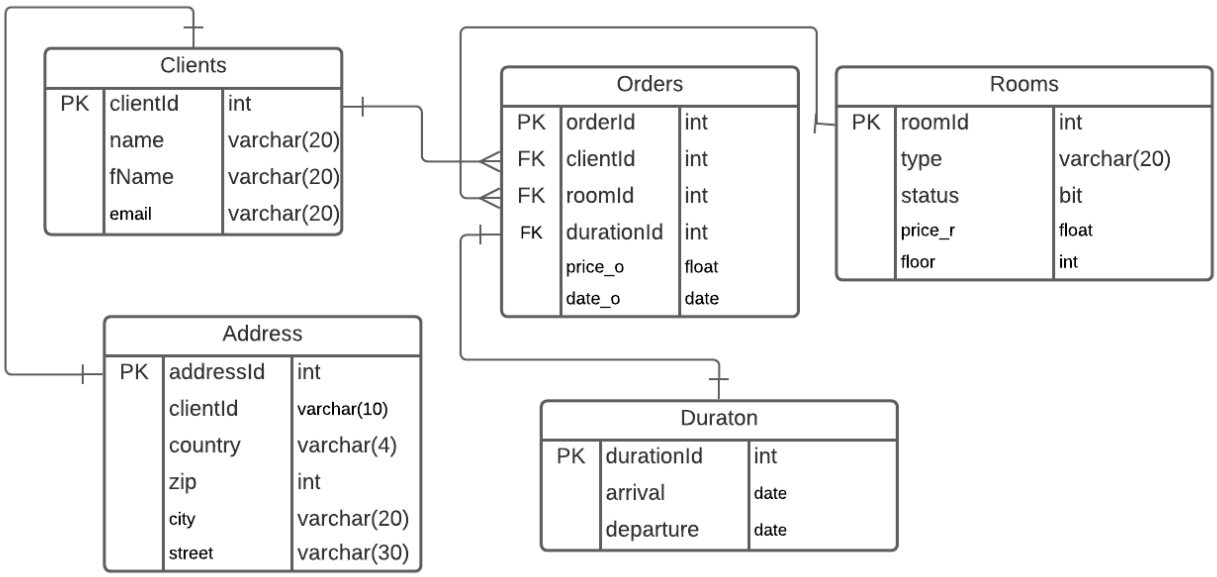
“By Hassan Ali Jadoon”

We have change data type of the PK so it will be easy to work with less redundancy. So All the relations are either one to one or one to many. The is not additional redundancy has been observe.

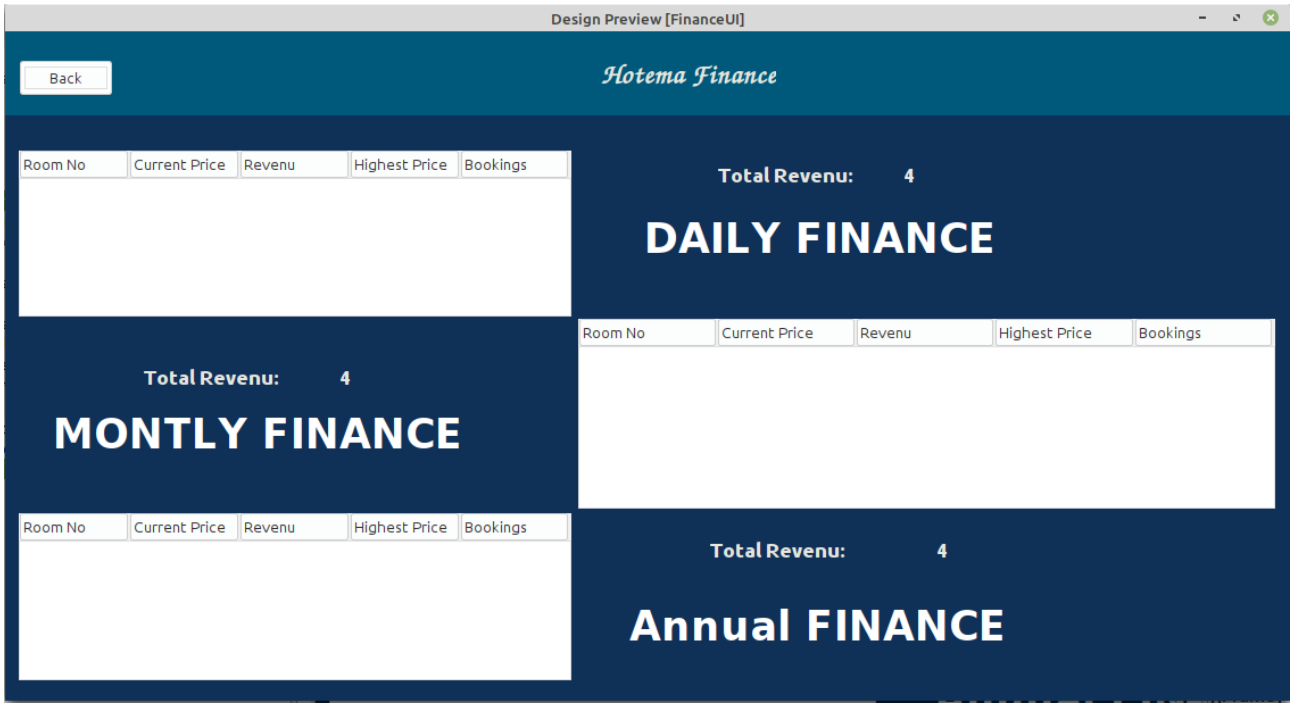
1.2.4.8.7 ER-Digram (By Muhammad Awais & Hassan Ali Jadoon)



FIX ANOMALIES IN ER DIGRAM **(By Sardar Badar Saghir)**



1.2.4.9 Prototype



1.1.1.2 SQL QUERIES

- Create Queries are made by Muhammad Awais
- Sardar Badar created select queries
- Hassan Ali Jadoon created insert queries

```
use master;
DROP Database hotma;
CREATE DATABASE hotma;
use hotma;
CREATE TABLE Clients (
  clientId int IDENTITY(1,1) PRIMARY KEY,
  name varchar(20) not Null,
```

```
  fName varchar(20) Not Null,
  email varchar(20) unique not NULL,
);

CREATE TABLE [Rooms] (
  [roomId] int not null    PRIMARY KEY ,
  [type] varchar(20) not null,
  [status] bit not null,
  [price_r] float not null,
  [floor] int not null

);



CREATE TABLE [Address] (
  [addressId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [clientId] int unique NOT NULL FOREIGN KEY REFERENCES Clients(clientId),
  [country] varchar(4) not null,
  [zip] int not null,
  [city] varchar(20) not null,
  [street] varchar(30) not null

);

CREATE TABLE [Orders] (
  [orderId] int not null IDENTITY(1,1)   PRIMARY KEY ,
```

```
  [clientId] int NOT NULL FOREIGN KEY REFERENCES  Clients(clientId),
  [roomId] int  NOT NULL FOREIGN KEY REFERENCES Rooms(roomId),
  [price_o] float not null,
  [date_o] date not null

);

CREATE TABLE [Duration] (
  [durationId] int not null IDENTITY(1,1)   PRIMARY KEY ,
  [arrival] date not null,
  [departure] date not null,
  [orderId] int unique NOT NULL FOREIGN KEY REFERENCES Duration(orderId)


);
```

```
--INSERTION QUERIES FOR
use hotma;
INSERT into Clients (
 name,
 fName,
 email
) values
('Adil','Nawaz', 'adilna@gmail.com'),
```

('Adl','Nawaz', 'adilw@gmail.com'),

('Ad','awaz', 'adil@gmail.com');

INSERT into[Address] (

[clientId] ,

[country] ,

[zip],

[city] ,

[street]

) values (3, 'pk',22340,'london' ,'london streets');

--INSERT DATA IN ROOMS

INSERT into[Rooms] (

[roomId],

[type],

[status] ,

[price_r],

[floor]

)Values(2,'First Class', 0, 100.4, 2),

(1,'First Class', 0, 100022.4, 2),

(3,'First Class', 0, 10030.4, 34);

--Insert Order

INSERT into [Orders] (

[clientId],

[roomId] ,

[price_o] ,

[date_o]

```
)SELECT clientId , roomId, price_r, '2009-11-11' FROM  Clients c, Rooms r  where clientId =2 AND roomId=3 AND status = 0;


--Change Status
UPDATE Rooms
SET status=1 where roomId =1;



--GET FINANCE QUERY WTITH  BADAR
SELECT  DISTINCT a.roomId as roomNO , b.maxPrice as highestPrice, r.price_r as currentPrice,
b.revenu as revenu,
b.countr as booking
FROM  ((Orders a
    INNER JOIN
    (
      SELECT
      MAX(price_o) as maxPrice,
      sum(price_o) as revenu,
      COUNT(o2.roomId) as countr,
      o2.roomId
      FROM Orders o2
      GROUP BY o2.roomId
    ) b ON a.roomId = b.roomId AND
        a.price_o = b.maxPrice)
  INNER JOIN Rooms r on  r.roomId = a.roomId) WHERE a.date_o ='2009-11-11';
```