Information Systems and Data Modeling – IT1090

Assignment

Title: Automated Parking System	
Batch Number: Y1S2 CSNE 1.1	Group Number: MLB_WD_CSNE_01.01_10

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

Rangana Wijesinghe	……..
Chanika Gayashan	……..
Kavisha Rajapaksha	……..
Gavishka Sahan	……..
T.Ravishka Lakshan	……..

Submitted on: 04/06/2023

Hypothetical Scenario

This automatic parking system is a great opportunity for users to get a high level of service. This makes it very easy to reserve a space for your vehicle. This automatic system is designed with high technology and high security. The system works to allocate suitable space according to the size of your vehicle. Unregistered customers can register an account by providing their username, ID number and other requirements.

One grant advantage is that these parking spaces do not require drivers or vehicle owners to be assigned and are managed by staff members. It also includes separate functions to ensure that the system runs without errors.

Registered customers have a unique ID. Customers can contact us if they have any problems. The administrator is responsible for managing the activities of the entire system.

Requirements Analysis

Automated Parking System is a fully web-based designed system. Our system has requirements and has developed depending on the customer's requirements. Requirement describes services the system will provide to the user and what they can do. These requirements can be divided into three parts:

The main requirement of the System

• Functional Requirements

• Non-Functional Requirements

• External Interface Requirements

Functional Requirements

Functional requirements are the expectations that users expect from a system. The automated parking system must provide specific responses to input and behave accordingly in different situations. Our automated parking system is designed based on functional requirements and includes essential functions. There are several requirements: identify available parking spaces, localize to open spaces, prevent unauthorized access, manage parking occupancy, provide real-time parking information, integrate with payment systems, provide data analytics, and operate reliably with minimal maintenance requirements. includes doing.

This system has several main functional requirements. They are,

- System Admin
- Member
- Guest

System Admin

User requirements

- Admin login to the site by providing valid login credentials.
- The administrator user should be able to view and manage parking occupancy and availability in real-time.
- Admin can manage user accounts and user access.
- Admin can manage user accounts and user access.
- Administrator can manage the database.
- Administrators can access the front and back end of the system.
- Administrators can edit system features as per user requirements.
- The administrator user should be able to access customer support and assistance through the system if needed.

System requirements

- The system should validate the user login credentials.
- The system should store members.
- The system should store details of the specific movies existing and adding.
- The system should delete details of the deleted sources by the user.
- The system should delete details of the customers deleted by the user.
- The system should display a successful or an error message after the update of all information regarding the automated parking system.
- The system should provide real-time data on parking occupancy and availability.
- The system should be able to generate reports on parking usage and revenue.
- The system should be able to manage and update parking rates and fees.

Member

user requirements

- The user should be able to reserve a parking space in advance, if available.
- The user should be able to receive notifications about parking availability and updates on their parking status.
- The user should be able to pay for parking through the system using various payment methods.
- The user should be able to access the system from a mobile app or web portal.
- The user's personal and payment information should be stored securely within the system to ensure privacy and security.
- User needs to verify his payment details.
- Users can edit, manage, delete, and update their own profiles.

System requirements

- The system should allow users to create and manage their accounts within the system.
- The system should allow users to view their parking history and receipts for past parking sessions.
- The system should be able to notify users about parking availability and updates on their parking status.
- The system should be able to accept parking payments through various payment methods.
- The system should be accessible to users of a mobile app or web portal.
- The system should notify a successful or error message after logging in.
- When a customer makes a payment, a verification must be done by the system.
- The system needs to store payment details and check their validity.
- The system should delete the details from the database of a deleted account.

Guest

user requirements

- The unregistered user should be able to view parking availability in real-time
- The unregistered user should be able to see parking rates and fees.
- The unregistered user should be able to contact customer support for assistance.
- The unregistered user should be able to access information about the parking system, such as location and hours of operation.
- The unregistered user should be able to register for an account within the system if they choose to do so.

System requirements

- The system shall provide unregistered users with real-time data on their parking availability.
- The system shall clearly display unregistered users' parking charges and fee lines.
- The system should allow unregistered users to contact the customer support unit for assistance.
- The system should display information such as the parking system, location and operating hours.
- The System provides an easy way for unregistered users to register for an account if they choose to do so.

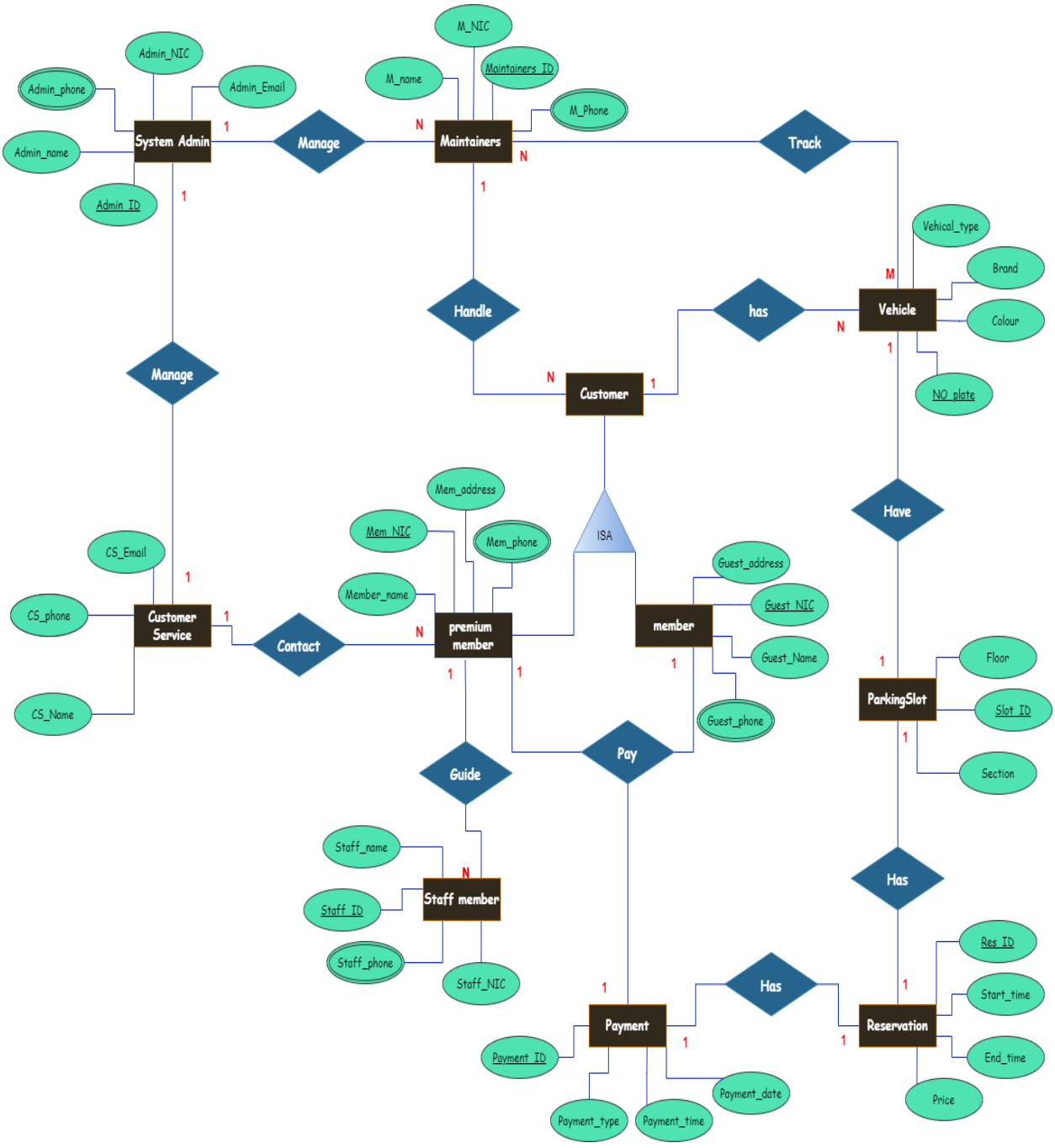
Non-Functional Requirements

- Performance: The system should be able to handle a large number of concurrent users and transactions without significant slowdown or latency.
- Availability: The system should be available 24/7 with minimal downtime or maintenance windows.
- Scalability: The system must scale up or down to manage changes in demand without affecting performance or availability.
- Security: The system must be secure and protect user data and payment information from unauthorised access or breach and other unauthorised activities.
- Usability: The system should be user-friendly and easy to navigate with a clear and intuitive user interface.
- Storage: System must have sufficient storage to store user's details and new user details.

External Interface Requirements

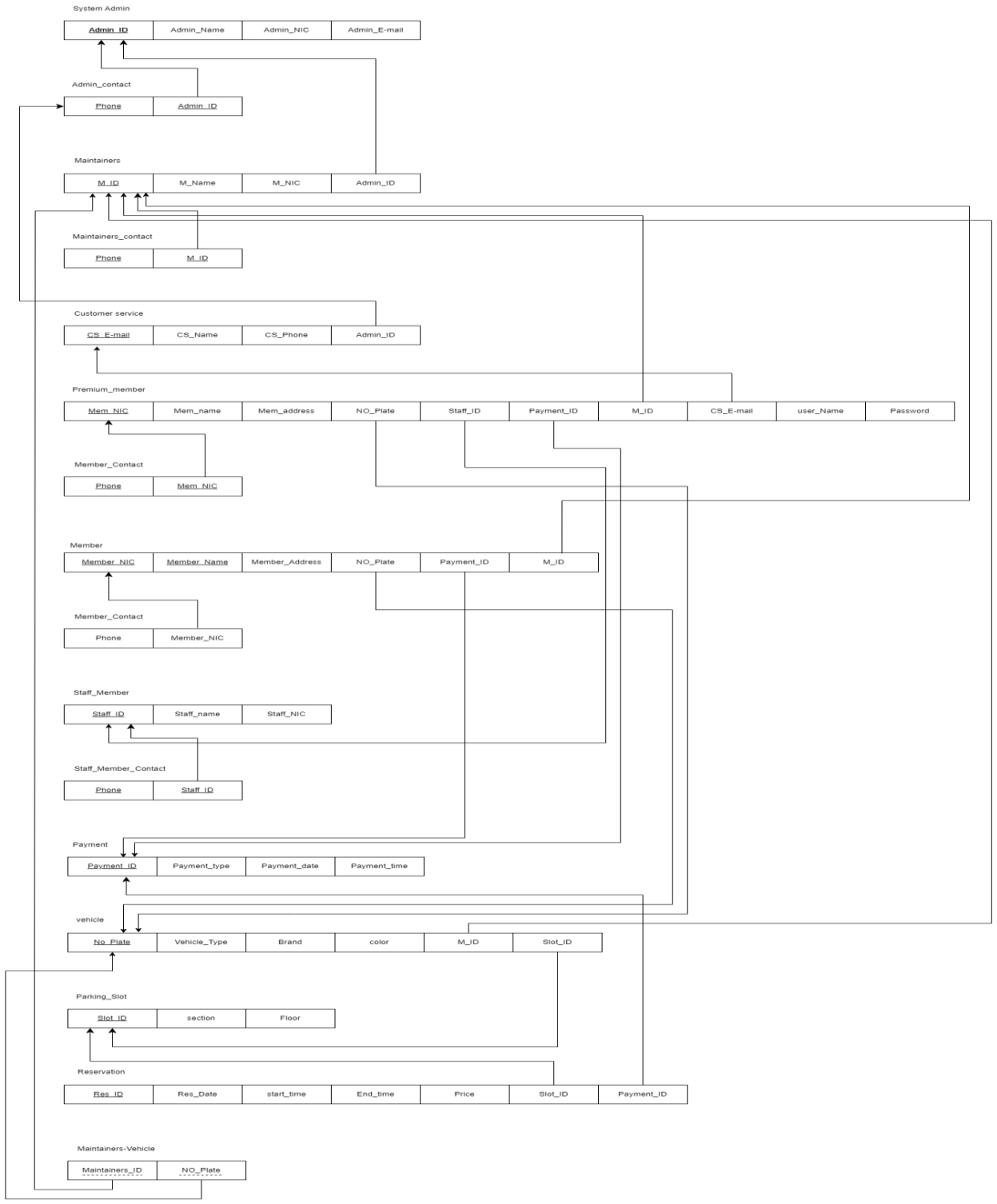
- Payment Gateway: The system should be integrated with a payment gateway to allow users to pay for parking using various payment methods.
- Mobile Application: The system should have a mobile application interface for users to book and pay for parking ahead of time, view parking availability and receive notifications.
- Web Portal: The system should also have a web portal interface for users to reserve and pay for parking, view parking availability, and receive notifications.
- Hardware Interfaces: The system should have hardware interfaces to interact with parking gates, sensors, and other components.
- Third-Party Integrations: The system should integrate with third-party systems such as security cameras, license plate recognition systems and other parking management systems.

ER Diagram



Picture 1

Relational schema



Used entities and attributes.

System Admin

Admin_ID

Admin_name

Admin_username

Admin_password

Admin_email

Maintainers

Maintainers_ID

Maintainers_name

Maintainers_NIC

Maintainers_phone

Customer Service

Customer Service_email

Customer Service_phone

Customer Service_Name

Customer

Customer_ID

Customer_name

Customer_phone

Premium Member

Member_ID

Member_username

Member_password

Member_name

Member_NIC

Member_phone

Member_address

Member

Guest_name

Guest_NIC

Guest_address

Guest_phone

Staff Member

Staff member_ID

Staff member_name

Staff member_phone

Staff member_NIC

Payment

Payment_ID

Payment_type

Payment_time

Payment_date

Reservation

Reservation_ID

Reservation_Start_time

Reservation_End_time

Reservation_price

ParkingSlot

ParkingSlot_ID

ParkingSlot_floor

ParkingSlot_section

Vehicle

Vehicle_type

Vehicle_brand

Vehicle_colour

Vehicle_No Plate

SQL Queries to create the Database

```
create table System_Admin(
    Admin_ID char(4) not null,
    Admin_Name varchar(50) not null,
    Admin_NIC varchar(12) not null,
    Admin_EMail varchar(30),
    constraint Admin_pk primary key(Admin_ID),
    constraint CHK_system_Admin check (Admin_ID like '[A-a][0-9][0-9][0-9]'),   /*A and 3 numbers*/
    constraint CHK1_system_admin check (Admin_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][V-v]' or Admin_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
)

INSERT INTO System_Admin values ('A001','MathewJoshep','934783154v','MathewJ@gmail.com');
INSERT INTO System_Admin values ('A002','James Harden','934768521v','jamesh@gmail.com');
INSERT INTO System_Admin values ('A003','Robert North','200156384462','robertn@gmail.com');
INSERT INTO System_Admin values ('A004','John Wick','963584236v','johnww@gmail.com');
INSERT INTO System_Admin values ('A005','Michael Jones','200068452315','michaeljon@gmail.com');
```

```
create table Admin_contact(
    Admin_ID char(4) not null,
    phone integer not null,
    constraint Admin_pkA primary key(phone),
    constraint Admin_FKA foreign key(Admin_ID) references System_Admin(Admin_ID)
)

INSERT INTO Admin_contact values ('A001',0713564129);
INSERT INTO Admin_contact values ('A001',0759587412);
INSERT INTO Admin_contact values ('A002',0702156842);
INSERT INTO Admin_contact values ('A003',0782136954);
INSERT INTO Admin_contact values ('A004',0753695482);
INSERT INTO Admin_contact values ('A004',0785648213);
INSERT INTO Admin_contact values ('A005',0723115489);



create table maintainers(
    M_ID char(4) not null,
    M_Name varchar(50) not null,
    M_NIC varchar(12) not null,
    Admin_ID char(4),
    constraint maintainers_PK primary key(M_ID),
    constraint Admin_FKM foreign key(Admin_ID) references System_Admin(Admin_ID),
```

```
    constraint CHK_maintainers check (M_ID like '[M-m][0-9][0-9][0-9]'),    /*M and 3 numbers*/
    constraint CHK1_maintainers check (M_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][V-v]' or M_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
)

INSERT INTO maintainers values ('M001','Rechel','988125469v','A002');
INSERT INTO maintainers values ('M002','David','921659845v','A001');
INSERT INTO maintainers values ('M003','William','200095846854','A001');
INSERT INTO maintainers values ('M004','Richard','859623156v','A003');
INSERT INTO maintainers values ('M005','Joseph','748591245v','A005');
INSERT INTO maintainers values ('M006','Thomes','852958475V','A004');

create table maintainers_contact(
    M_ID char(4) not null,
    phone integer not null,
    constraint maintainers_contact_PK primary key(phone),
    constraint maintainers_FKM foreign key(M_ID) references maintainers(M_ID)
)

INSERT INTO maintainers_contact values ('M001',0778956412);
INSERT INTO maintainers_contact values ('M002',0775615236);
INSERT INTO maintainers_contact values ('M003',0778954213);
INSERT INTO maintainers_contact values ('M004',0746498754);
```

```
INSERT INTO maintainers_contact values ('M005',0712378965);

INSERT INTO maintainers_contact values ('M005',0775800125);



create table customer_service(
    CS_EMail varchar(30) not null,
    CS_Name varchar(25) not null,
    CS_Phone integer not null,
    Admin_ID char(4),
    constraint customer_service_PK primary key(CS_EMail),
    constraint Admin_FKC foreign key(Admin_ID) references System_Admin(Admin_ID)
)


INSERT INTO customer_service values ('JakeSally@Gmail.com','Jake
Sully',0756984585,'A001');

INSERT INTO customer_service values ('ThomasShelby@Gmail.com','Thomas
Shelby',0729748996,'A001');

INSERT INTO customer_service values ('ChristopherSammy@Gmail.com','Christopher
Sammy',0703050680,'A005');

INSERT INTO customer_service values ('CharlesDickens@Gmail.com','Charles
Dickens',0704512389,'A003');

INSERT INTO customer_service values ('DanielKelly@Gmail.com','Daniel
Kelly',0782569840,'A004');

INSERT INTO customer_service values ('AdamHunt@yahoo.com','Adam
Hunt',0759840321,'A002');
```

```
create table staff_Member(
    staff_ID char(4) not null,
    staff_Name varchar(50) not null,
    staff_NIC varchar(12) not null,
    constraint staff_Member_PK primary key(staff_ID),
    constraint chk_staff_Member check (staff_ID like '[S-s][0-9][0-9][0-9]'),   /*S and 3 numbers*/
    constraint chk1_staff_Member check (staff_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][V-v]' or staff_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
)

INSERT INTO staff_Member values ('S001','John Kanell','200025486874');
INSERT INTO staff_Member values ('S002','Matthew Sam','987562314v');
INSERT INTO staff_Member values ('S003','Anthony John','658941233v');
INSERT INTO staff_Member values ('S004','Mark Adam','200112364859');
INSERT INTO staff_Member values ('S005','Mary Aniston','752169961v');

create table staff_Member_Contact(
    staff_ID char(4) not null,
    Phone integer not null,
    constraint staff_Member_Contact_PK primary key(Phone),
    constraint staff_Member_FKS foreign key(staff_ID) references staff_Member(staff_ID)
```

```
)


INSERT INTO  staff_Member_Contact values ('S001',0785658548);

INSERT INTO  staff_Member_Contact values ('S002',0716245568);

INSERT INTO  staff_Member_Contact values ('S003',0778415203);

INSERT INTO  staff_Member_Contact values ('S004',0701233559);

INSERT INTO  staff_Member_Contact values ('S005',0729156820);




create table Payment(

    Payment_ID char(10) not null,

    Payment_Type varchar(10) not null,

    Payment_Date varchar(15) not null,

    Payment_Time varchar(15) not null,

    constraint Payment_PK primary key(Payment_ID),

    constraint chk_Payment check (Payment_ID like '[P-p][0-9][0-9][0-1][0-9][0-9][0-9][0-9][0-
9][0-9]'), /*P, year, month, 5 numbers*/

    constraint chk1_Payment check (Payment_Date like '[0-3][0-9]-[0-1][0-9]-[2][0][0-9][0-9]'),
/*date-month-year*/

    constraint chk2_Payment check (Payment_Time like '[0-2][0-9].[0-5][0-9].[0-5][0-9]')
/*hour-minutes-seconds*/

)


INSERT INTO Payment values  ('P230509556','Online','28-05-2023','18.36.23');

INSERT INTO Payment values  ('P210815623','Card','12-08-2021','08.30.12');
```

```
INSERT INTO Payment values  ('P220145692','Cash','15-01-2022','20.06.05');
INSERT INTO Payment values  ('P230408215','Online','30-04-2023','23.10.00');
INSERT INTO Payment values  ('P230610230','Cash','03-06-2023','05.51.52');
INSERT INTO Payment values  ('P230101245','Cash','12-01-2023','12.25.10');
INSERT INTO Payment values  ('P230114258','Card','01-01-2023','08.26.31');
INSERT INTO Payment values  ('P220145789','Cash','04-01-2022','14.56.45');
INSERT INTO Payment values  ('P220178952','Online','16-01-2022','16.13.31');
INSERT INTO Payment values  ('P230114279','Online','02-01-2023','09.35.00');
INSERT INTO Payment values  ('P230612345','cash','05-06-2023','05.15.23');
INSERT INTO Payment values  ('P230545021','cash','25-05-2023','23.12.52');
INSERT INTO Payment values  ('P230410200','card','25-04-2023','21.23.10');
INSERT INTO Payment values  ('P221212000','card','12-12-2022','01.24.52');



create table vehicle(
    NO_Plate varchar(10) not null,
    vehicle_Type varchar(20) not null,
    Brand varchar(15),
    color varchar(20),
    M_ID char(4),
    Slot_ID varchar(4),
    constraint vehicle_PK primary key(NO_Plate),
    constraint maintainers_FKv foreign key(M_ID) references maintainers(M_ID),
```

```
   constraint CHK_Vehicle check (NO_Plate like '[A-Z][A-Z]-[0-9][0-9][0-9][0-9]' /*2 letters-4numbers*/ or NO_Plate like '[A-Z][A-Z][A-Z]-[0-9][0-9][0-9][0-9]') /*3 letters-4numbers*/

)


INSERT INTO vehicle values  ('AFG-3476','car','Toyota','red','M003','A205');

INSERT INTO vehicle values  ('ABD-9032','car','Nissan','sky blue','M001','B305');

INSERT INTO vehicle values  ('ACD-3525','van','Benz','black','M002','C203');

INSERT INTO vehicle values  ('FK-1230','jeep','Toyota','pearl white','M004','A120');

INSERT INTO vehicle values  ('WK-1235','bike','Honda','cherry red','M005','D102');

INSERT INTO vehicle values  ('FE-5698','car','Toyota','white','M005','A121');

INSERT INTO vehicle values  ('HT-1230','van','Mitsubishi','white','M001','B306');

INSERT INTO vehicle values  ('ACD-9521','bike','Honda','black','M002','A200');

INSERT INTO vehicle values  ('ACA-5200','cab','Ford','blue','M005','C206');

INSERT INTO vehicle values  ('AAA-2555','lorry','Toyota','yellow','M003','D202');

INSERT INTO vehicle values  ('DW-4598','car','Benz','red','M005','A201');

INSERT INTO vehicle values  ('DA-8222','bike','BMW','brown','M001','B300');

INSERT INTO vehicle values  ('ND-1254','bus','Laylend','blue','M002','C201');

INSERT INTO vehicle values  ('ACE-3120','van','Mitsubishi','white','M003','C205');

INSERT INTO vehicle values  ('AEE-8333','cab','Toyota','gray','M002','B101');



create table Member(

   member_NIC varchar(12) not null,

   member_Name varchar(50) not null,
```

```
    member_Address varchar(100) not null,
    NO_Plate varchar(10) not null,
    Payment_ID char(10),
    M_ID char(4),
    constraint member_PK primary key(member_NIC),
    constraint vehicle_FK3 foreign key(NO_Plate) references vehicle(NO_Plate),
    constraint maintainers_FKG foreign key(M_ID) references maintainers(M_ID),
    constraint CHK1_member check (member_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][V-v]' or member_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')
)

INSERT INTO member values  ('863549210v','Sam Fisher','12a, 8th lane','FE-5698','P230500045','M002');
INSERT INTO member values  ('862130904v','George Bush','Berguvsvägen 8','AAA-2555','P230610230','M001');
INSERT INTO member values  ('200030604895','Timothy Jem','Forsterstr. 57','ACA-5200','P210815623','M001');
INSERT INTO member values  ('200130604512','Dorothy Pual','C/ Araquil, 67','ACD-9521','P230498546','M003');
INSERT INTO member values  ('561230156v','Sharon Chanika','23 Tsawassen Blvd','HT-1230','P230408215','M005');

create table member_Contact(
    member_NIC varchar(12) not null,
    phone integer,
```

```
    constraint member_Contact_PK primary key(phone),
    constraint member_FKG foreign key(member_NIC) references member(member_NIC)
)

INSERT INTO member_Contact values  ('863549210v',0119853157);
INSERT INTO member_Contact values  ('862130904v',0706023591);
INSERT INTO member_Contact values  ('200030604895',0745630012);
INSERT INTO member_Contact values  ('200130604512',0723059874);
INSERT INTO member_Contact values  ('561230156v',0702356920);
INSERT INTO member_Contact values  ('561230156v',0332258469);



create table Premium_Member(
    Mem_NIC varchar(12) not null,
    Mem_Name varchar(50) not null,
    Mem_Address varchar(100) not null,
    NO_Plate varchar(10) not null,
    staff_ID char(4),
    Payment_ID char(10),
    M_ID char(4),
    CS_EMail varchar(30),
    User_Name varchar(35) not null,
    password varchar(50) not null,
    constraint Premium_Member_PK primary key(Mem_NIC),
```

```
    constraint vehicle_FK1 foreign key(NO_Plate) references vehicle(NO_Plate),

    constraint staff_FK1 foreign key(staff_ID) references staff_Member(staff_ID),

    constraint payment_FKM foreign key(Payment_ID) references Payment(Payment_ID),

    constraint maintainers_FkM2 foreign key(M_ID) references maintainers(M_ID),

    constraint customer_service_FKM foreign key(CS_EMail) references
customer_service(CS_EMail),

    constraint CHK_member check (Mem_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-
9][V-v]' or Mem_NIC like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

)


INSERT INTO Premium_Member values  ('200154867594','Jane Austen','456/f/4, 4th
lane','DW-4598','s002','P230101245','M003','JakeSally@Gmail.com','Jane_A','AustenJ_456/f/4');

INSERT INTO Premium_Member values  ('895623157v','Patricia Marry','Obere Str. 57','AEE-
8333','s003','P230114258','M002','ThomasShelby@Gmail.com','PatriciaMarry','Patricia57');

INSERT INTO Premium_Member values  ('200012364587','Donald Jake','Avda. de la
ConstituciOn 2222','ACE-
3120','s001','P220145789','M001','CharlesDickens@Gmail.com','Donald_Jake','Jake#4321');

INSERT INTO Premium_Member values  ('195620304506','Kevin Hart','Mataderos 2312','ND-
1254','s004','P220178952','M004','AdamHunt@yahoo.com','HARTKEV.ME','KevinH120$');

INSERT INTO Premium_Member values  ('200154868594','Brian Oak','120 Hanover Sq.','DA-
8222','s005','P230114279','M005','DanielKelly@Gmail.com','Brian-Oak','Brian-O45');
```

```
create table Premium_Member_contact(
    Mem_NIC varchar(12) not null,
    phone integer not null,
    constraint member_contact_PK1 primary key(phone),
    constraint member_FKMC foreign key(Mem_NIC) references Premium_member(Mem_NIC)
)

INSERT INTO Premium_Member_contact values  ('200154867594',0777531595);
INSERT INTO Premium_Member_contact values  ('895623157v',0721236592);
INSERT INTO Premium_Member_contact values  ('895623157v',0715820004);
INSERT INTO Premium_Member_contact values  ('895623157v',0785987156);
INSERT INTO Premium_Member_contact values  ('200012364587',0700123560);
INSERT INTO Premium_Member_contact values  ('200012364587',0748254697);
INSERT INTO Premium_Member_contact values  ('195620304506',0755699562);
INSERT INTO Premium_Member_contact values  ('195620304506',0715879584);
INSERT INTO Premium_Member_contact values  ('200154867594',0703045630);
INSERT INTO Premium_Member_contact values  ('200154867594',0759842568);



create table parking_Slot(
    Slot_ID varchar(4) not null,    /*section,floor,slot*/
    Section char(1) not null,       /*1 letter*/
    Floor char(1) not null,         /*1 to 3 num*/
    constraint parking_Slot_PK primary key(Slot_ID),
```

```
    constraint CHK_parking_Slot check (Slot_ID like '[A-Z][0-9][0-9][0-9]'),
    constraint CHK1_parking_Slot check (Section like '[A-Z]'),
    constraint CHK2_parking_Slot check (Floor like '[0-3]')
)

INSERT INTO parking_Slot values ('A205','A','2');
INSERT INTO parking_Slot values ('B305','B','3');
INSERT INTO parking_Slot values ('C203','C','2');
INSERT INTO parking_Slot values ('A120','A','1');
INSERT INTO parking_Slot values ('D102','D','1');
INSERT INTO parking_Slot values ('B202','B','2');
INSERT INTO parking_Slot values ('C105','C','1');
INSERT INTO parking_Slot values ('D202','D','2');
INSERT INTO parking_Slot values ('A121','A','1');
INSERT INTO parking_Slot values ('B306','B','3');
INSERT INTO parking_Slot values ('A200','A','2');
INSERT INTO parking_Slot values ('C206','C','2');
INSERT INTO parking_Slot values ('A201','A','2');
INSERT INTO parking_Slot values ('B300','B','3');
INSERT INTO parking_Slot values ('C201','C','2');
INSERT INTO parking_Slot values ('C205','C','2');
INSERT INTO parking_Slot values ('B101','D','2');
```

```
create table reservation(
    res_ID char(9) not null,
    res_date varchar(10) not null,
    Start_Time varchar(10) not null,
    End_Time varchar(10) not null,
    Price varchar(10) not null,
    Slot_ID varchar(4),
    Payment_ID char(10),
    constraint reservation_PK primary key(res_ID),
    constraint parking_Slot_FKR foreign key(Slot_ID) references parking_Slot(Slot_ID),
    constraint payment_FKR foreign key(Payment_ID) references Payment(Payment_ID),
    constraint CHK_reservation check (res_ID like '[R-r][0-9][0-9][0-3][0-9][0-9][0-9][0-9][0-
9]'), /*R, month,date,4 num*/
    constraint CHK1_reservation check (res_date like '[0-3][0-9]-[0-1][0-9]-[2][0][0-9][0-9]'),
/*date-month-year*/
    constraint CHK2_reservation check (Start_Time like '[0-2][0-9].[0-5][0-9].[0-5][0-9]'),
/*hour-minutes-seconds*/
    constraint CHK3_reservation check (End_Time like '[0-2][0-9].[0-5][0-9].[0-5][0-9]')
/*hour-minutes-seconds*/
)


INSERT INTO reservation values  ('R06150981','15-06-
2023','09.00.00','15.00.00','3000','C105','P230509556');
INSERT INTO reservation values  ('R04140120','14-04-
2023','02.20.00','10.00.00','2650','A205','P210815623');
INSERT INTO reservation values  ('R02101564','02-10-
2023','07.00.00','20.00.00','1200','B305','P220145692');
```

```
INSERT INTO reservation values  ('R07252256','25-07-
2023','06.15.00','09.00.00','2850','A120','P230408215');

INSERT INTO reservation values  ('R06115560','11-06-
2023','09.00.00','06.45.00','3500','D102','P230610230');




create table maintainers_vehicle(
    M_ID char(4) not null,
    NO_Plate varchar(10),
    constraint maintainers_FKMV foreign key(M_ID) references maintainers(M_ID),
    constraint vehicle_FKMV foreign key(NO_Plate) references vehicle(NO_Plate)
)


INSERT INTO maintainers_vehicle values  ('M004','ABD-9032');
INSERT INTO maintainers_vehicle values  ('M001','FK-1230');
INSERT INTO maintainers_vehicle values  ('M005','ABD-9032');
INSERT INTO maintainers_vehicle values  ('M003','ACD-3525');
INSERT INTO maintainers_vehicle values  ('M002','WK-1235');
INSERT INTO maintainers_vehicle values  ('M002','FK-1230');
INSERT INTO maintainers_vehicle values  ('M001','AFG-3476');
```

Performance Considerations

• A vast number of users can use our system at the same time.

• Users can access the system anytime from any device and make bookings, payments etc.

• When customers register in the system, customer information is verified within seconds by the system.

• When customers check slot availability

• The system will verify login information within seconds.

• The system account verification link is emailed to the registered user system.

• The system verifies card payment details within seconds.

• Database is ready after updating within 2 seconds.

• System response time is as short as less than 2 seconds.

• Search results should be displayed within acceptable time limits.

• The system provides a user-friendly interface.

• A success email is sent with the payment report after every payment booking system.

Security Requirements

• User must confirm all relevant information during registration.

• When registering into the system, the user must enter a strong password and use a combination of uppercase letters, lowercase letters, numbers and symbols.

• Only registered users should be allowed to enter the system.

• The system shall bear all security and liability for financial transactions through the system and shall take high measures to prevent unauthorised access to such data.

• Anyone cannot access the admin dashboard or admin page. If someone tries to access the admin page, he/she gets redirected to the home page.

• A database should back up all the data in the system.

• Administrator can change system data.

• The system must be protected from viruses and other malicious software and ensure that the user accessing the system is not affected.

• The privacy of the customer's questions should be protected.