# Sri Lanka Institution of Information Technology

MLB_09.01_07

# Automated Parking System

## Information System and Data Modelling-IT1090

# Information Systems and Data Modeling – IT1090

## Assignment

Title: Automated parking system

Batch Number: MLB_09.1 Group Number: MLB_9.1_07

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:

| | | | |
|---|---|---|---|
| **1** | IT21275398 | Godage D.N.P | [illegible] |
| **2** | IT21173936 | Kodithuwakku K.G.K.M.J | [illegible] |
| **3** | IT21174612 | De Davin G.G.P.N | Prasanga |
| **4** | IT21176210 | Asmitha Thiraviyarasa | [illegible] |
| **5** | IT21176142 | Abiramy Kumaresan | K.Abiramy |

**Submitted on: 29/04/2022**

# Introduction

An automated parking system enables people to acquire a parking space in a much more convenient way. Every day people spend a lot of time searching for a place to park their vehicles. Therefore, they need to search for available places to park their vehicles and it is only a waste of time and fuel and it also increases the traffic congestion. This automated parking system is dedicated for customers to check available parking area and book them for a limited time. We have enhanced the service to provide customers with the solution to lack of parking space in modern cities.

## Hypothetical scenario

The automated parking system is dedicated for customers to check available parking and reserve for a certain period of time. There are two types of users who can access this system: registered and unregistered. The system allows users to both view and use the system for reservations. Users who have registered can gain access to the system by logging in. If unregistered users wish to register, they can do so by signing up for the system.

Users can choose and reserve a preferred parking space ahead of time. The parking fee is calculated based on the locations and time slots chosen by the users. Users can pay using any online transaction method they prefer. To complete the payment, a QR code is generated. Users must contact support service agents to update reservations.

# Requirement Analysis.

## Main Requirements for the System

The online parking reservation system (OPRS) is a secure and comprehensive web-based system that was designed and installed to meet all of the system's needs. Functional requirements and non-functional requirements are the two types of requirements.

### Functional Requirements

The OPRS also has a set of primary functional criteria, which are described below. Unregistered users, registered users, managers, support service agents, developers, and the system administrator can all access the website. The portions of the website that each party can access, on the other hand, differs.

### Registered/ Unregistered User

**User Requirements: -**

**Access:**

• Customers (registered and unregistered) have access to the system's client-side. Customers can go to the website to look at available parking spots and make reservations.

### Registration:

• Unregistered users can create a new user account by entering their first and last names, email address, address, and phone number, as well as generating and confirming a new password.

• Registered users have the ability to manage and amend their user profiles.
• A registered user can access the system by inputting their user credentials correctly.

**Searching for locations:**

• Customers can look for parking spots near their desired location.

• Customers can utilize the filter option to narrow down search results based on distance to the location and parking fees.

• Customers are given thorough information about each place, such as driving instructions, the area around the venue, the amount of parking spaces available, security measures, and parking prices.

**Reservation:**

• Customers can use the web platform to reserve any available parking spots.

• Customers should confirm their parking reservation by entering the date and time slots.

**Payment:**

• Customers can pay by credit card, debit card, or online.

• Customers must provide proper payment information and confirm payment.

• Customers will receive an e-invoice as proof of payment and reservation.

• To gain access to a reserved parking spot, customers are given a unique QR code.

**Feedback and customer support:**

• Customers can rank and comment on parking locations as well as the website's service.

• Customers can contact customer care representatives with questions or concerns about reservations, payments, cancellations, and other services.

**System Requirements**

• Users (unregistered and registered) should be able to view and book parking spaces as well as make payments through the website.

• Users should be able to register and create new accounts through the system.

• When creating a new user account, the system should validate the user's credentials and save the user's information.

• Users should be able to control their own accounts in the system.

• When a registered user attempts to log into the system, the system should validate the user's login credentials.

• The system should present a list of parking locations, each with precise and complete information.

• The system should calculate the parking price correctly based on the location fee and the customer's usage time.

• Only registered users should be able to access reports.

## Manager

**User Requirements**

•Manager is allowed to access both client-side and server-side of the system

•Manager can update/edit information of parking locations and add/remove new parking locations to/from the system database

•Manager can update availability status of parking locations.

•Manager can view all reservation and payment records and history of customers and generate weekly/monthly reports containing said information.

### System Requirements

• The system should allow the manager to access both the client-side and server-side of the website, as well as conduct the operations listed above.

• The system should keep track of all client reservations and payments and accurately display each customer's history.

• The system should create financial reports that include each customer's reservation and payment history.

## Support Service Agent

### User Requirements

• Customer inquiries can be viewed and responded to via e-mail or the web platform forums by a support service agent.

• Assistance service Agents can take live phone calls from customers and answer issues with reservations, payments, or any other service on the internet.

• Customer reservations can be edited, updated, or canceled by support service employees.

### System Requirements

• The system should link clients with available support service agents as soon as possible.

• Support service agents should be able to see, edit/update, or cancel customer reservations using the system.

• The system should allow customer service representatives to view past payment records and reservation histories

## Developer

### User Requirements

• The developer has access to the system's client and server sides.

• The website is maintained and updated by the developer.

• The developer is in charge of keeping the system's security features up to date.

• The developer has the ability to change, add, or remove parking spots.

### System Requirements

• The system should allow developers to view the website's client-side and server-side code, as well as conduct the operations listed above.

• The system should allow for new changes to be made.

## System Administrator

### User requirements

• The system's administrator has access to both the client and server sides.

• Parking location information can be updated/modified, and new parking sites can be added/removed from the system database.

• The administrator can change the availability of parking spaces based on the reservations made.

• The administrator has the ability to go over and process user-confirmed reservations.

• The system generates a monthly charge for subscription-enabled registered users, which the administrator reviews.

• The administrator evaluates and reacts to customer feedback and reviews.

### System Requirements

• The system should allow the administrator to access the website on both the client and server sides and conduct the operations listed above.

• Customer feedback and reviews should be saved in the system.

## Non-Functional Requirements

The non-functional requirements of OPRS are listed down below.

### Availability

• The system should be available 24 hours a day, seven days a week. All of the website's services should be available at all times.

• The system should work in any browser and on any operating system.

### Usability

• The system should have a simple and user-friendly interface.

• The user should be able to simply navigate to each webpage and complete their responsibilities in a timely manner.

### Reliability

• The system should be able to complete all tasks and provide all services without any faults or failures.

• If the system encounters an error, it should be able to recover quickly and ensure that any impact on the system or user is minimal.

• The system should be able to accurately authenticate and detect any errors in user login and payment credentials.

• When it comes to parking sites, the system should be able to calculate distance and pricing accurately.

• All relevant information about reservations and payments should be stored in the system via backup procedures.

• In general, users should be able to trust the entire website and use the web platform with confidence.

**Speed**

• The system should do all jobs as quickly as possible.

• Even if numerous users are using the website at the same time, the speed should not suffer.

• Given the security features associated with the website, the more crucial operations, such as payment processing, should be completed as quickly as feasible.

**Capacity**

• The system should be able to retain all user account credentials, parking location information, reservation information, payment information, and customer feedback.

• The system should be able to function normally in the event of a system failure while storing the above data on servers.

**Security**

• The system should only allow a user to submit erroneous login credentials up to five times before notifying the account owner via email or text message of failed login attempts.

• Only authorized staff should be able to access the server-side of the website, which should be password protected.

• A configured security system should be used to prevent malicious access requests (such as hacking).

• The Secure Sockets Layer (SSL) protocol should be used to encrypt all sensitive information, such as user card/account data and passwords.

# Data Requirements

## Login

- Login ID (Log_ID)
- Username(Username)
- Password(User_password)

## Registered member

- Member first name (f_name)
- Member last name(l_name)
- Member ID (M_ID)
- Email (e_mail)
- Contact number (con_number)
- Member's address(M_address)
- Member login Id(Log_ID)

## Park Location

- Location number(location_ID)
- Location name (location_name)
- Address(details)
- Price (price)

## Payment

- Customer ID(Customer _ID)
- Customer name(M_name)
- Payment ID(P_ID)
- Amount (Amount)
- Contact number(c_number)
- Reservation type(r_type)
- Payment type (p_type)
- Payment status (p_status

**Reservation**

- Reservation no (Reserv_ID)
- Location number(location_ID)
- Customer ID(Customer_ID)
- Duration(duration)
- Date(date)

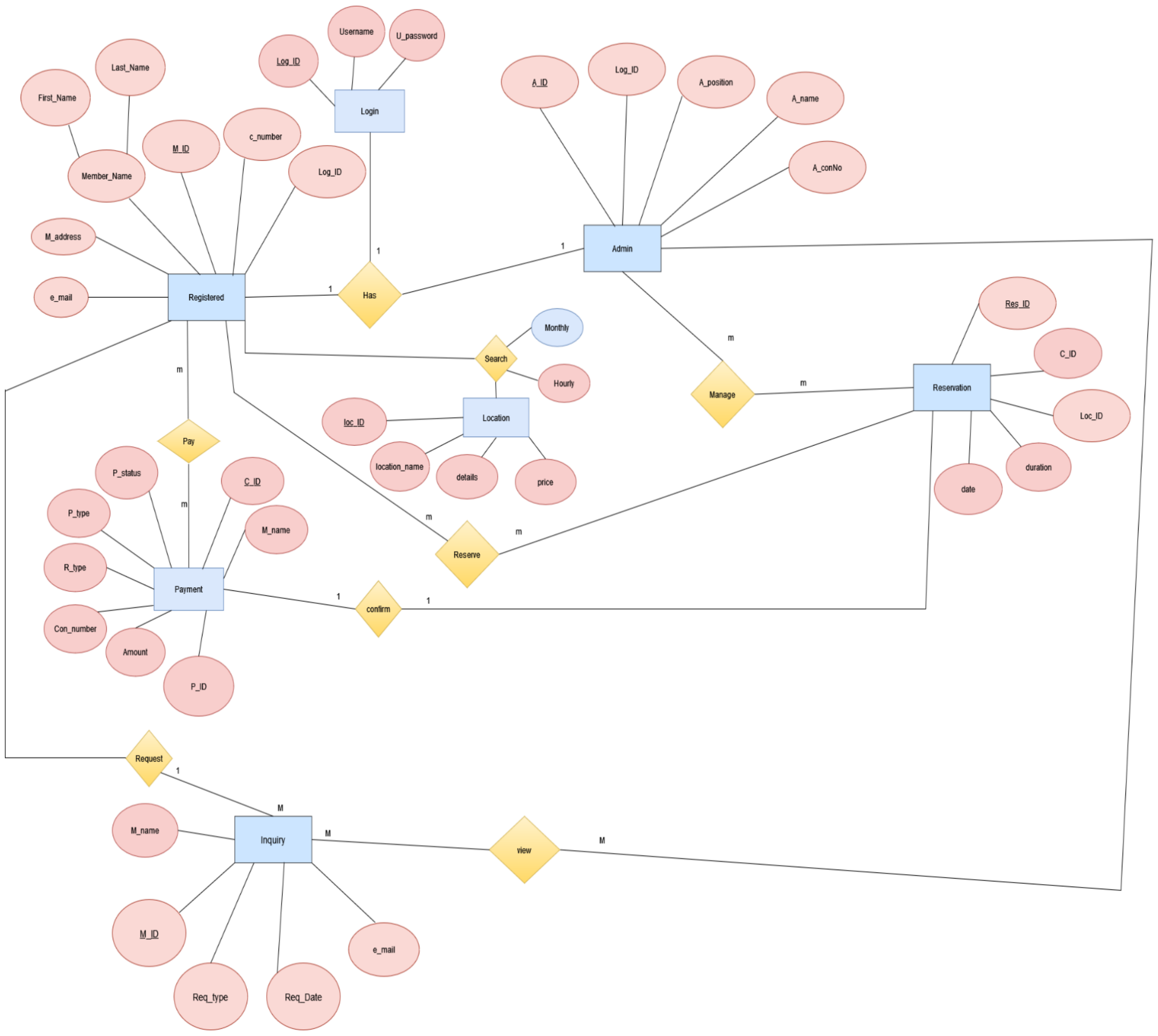
**Admin/Staff**

- Admin ID(A_ID)
- Login ID (Log_ID)
- Admin name(A_name)
- Admin position(A_position)
- Contact no(A_conNo.)

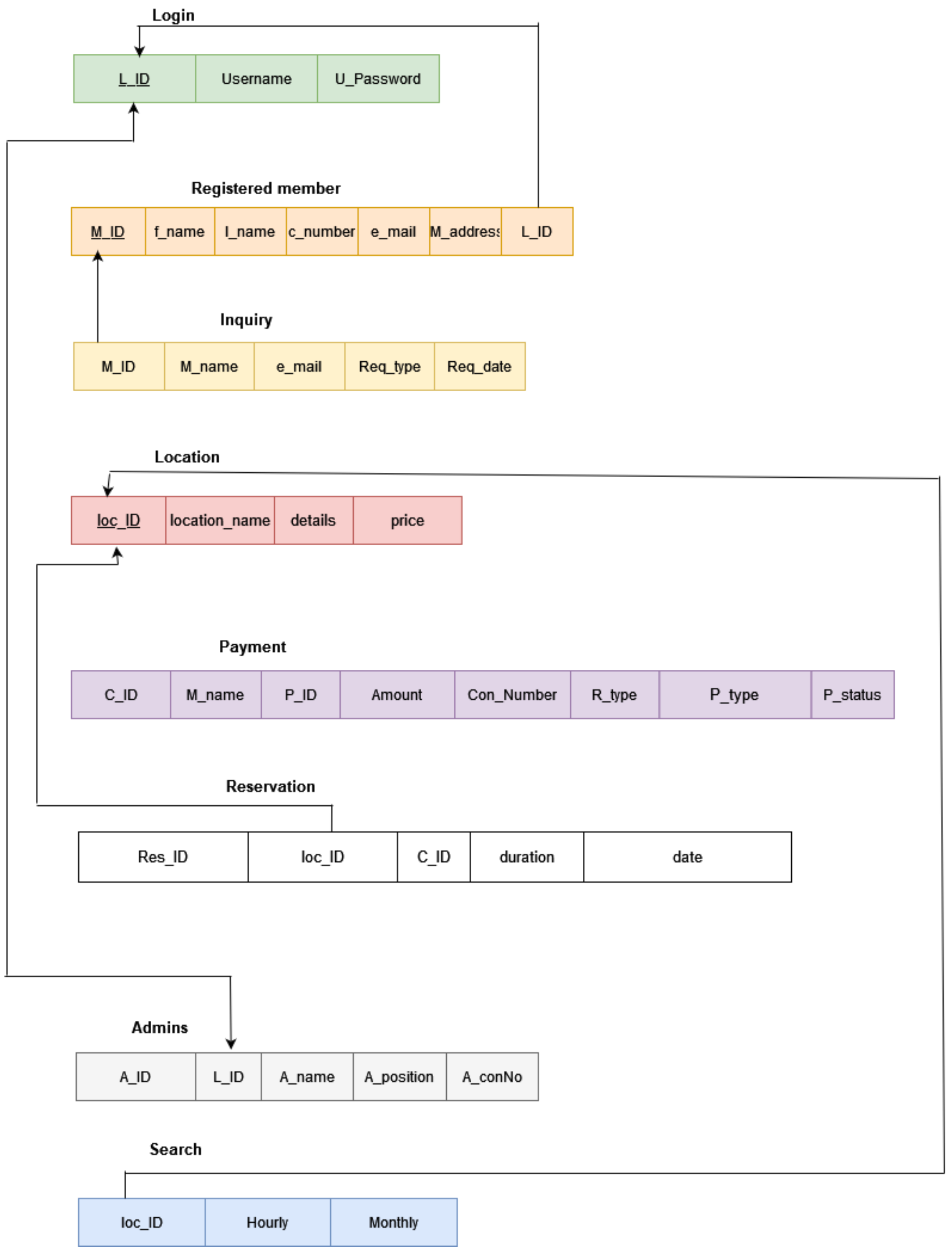
**Inquiry**

- Member name(M_name)
- Customer ID(M_ID)
- Email(e_mail)
- Request type(Req_type)
- Request date(Req_date)

# Entity-Relationship



## Relational Schema



# SQL Database creation

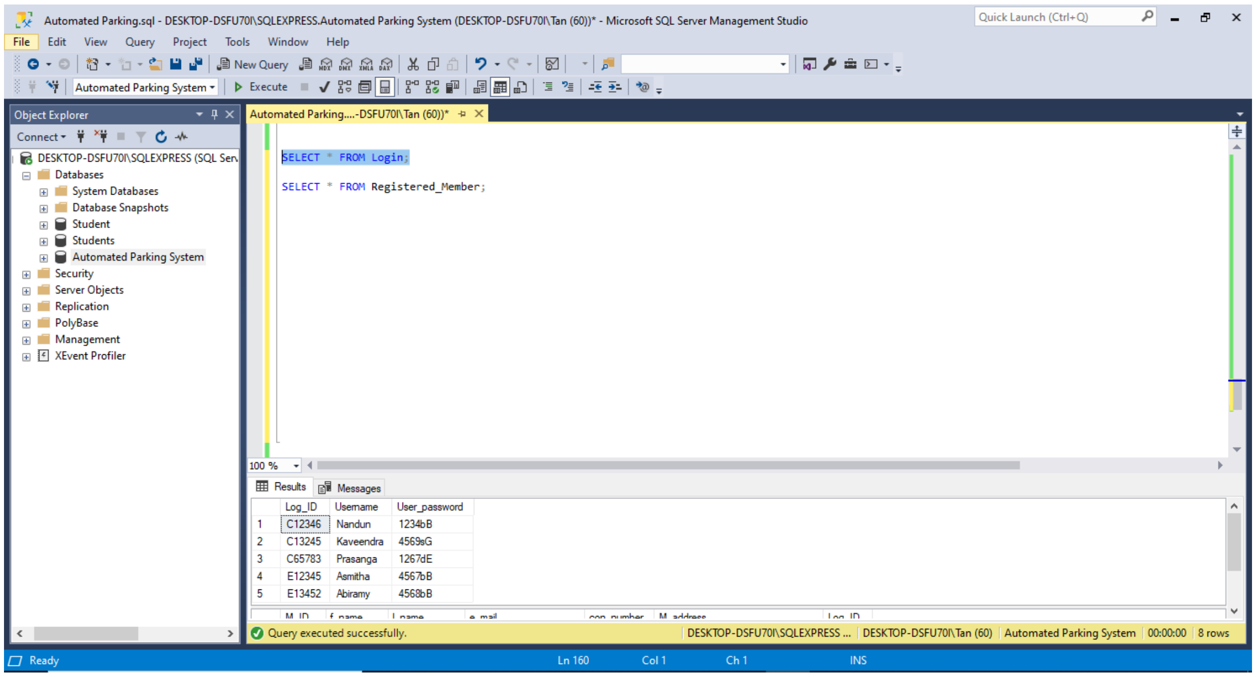
```
--Login Table
CREATE TABLE Login(
Log_ID char(6) not null,
Username char(20),
User_password char(6),
constraint Login_PK PRIMARY KEY(Log_ID),
constraint Log_ck check (Log_ID LIKE'[C\E][0-9][0-9][0-9][0-9][0-9]'), constraint userid check (User_password
LIKE'[0-9][0-9][0-9][0-9][a-z][A-Z]')
);


---Registered Member Table
CREATE TABLE Registered_Member(
M_ID char(6) not null,
f_name varchar(20),
l_name varchar(20),
e_mail varchar(40),
con_number char(10),
M_address char(50),
Log_ID char(6) not null,

constraint C_PK PRIMARY KEY(M_ID),
constraint CHK_MID check (M_ID LIKE '[m\M][0-9][0-9][0-9][0-9]'),

constraint CHK_cont check (con_number LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),
constraint C_FK FOREIGN KEY (Log_ID) references Login(Log_ID));


---Inquiry Table
CREATE table Inquiry(
M_ID char(6) not null,
M_name varchar(20),
E_Mail   char (20),
Req_Type varchar(50),
Req_Date date

Constraint PK_Inquiry PRIMARY KEY (M_ID),
Constraint check_Inquiry_E_Mail check (E_Mail LIKE '%@%.%' )

);
```

```
---Park Location Table
CREATE table Park_Location(
Location_ID char(6) not null,
Location_name     char(30),
Details char (60),
Price int

constraint Location_PK PRIMARY KEY(Location_ID),
constraint loc_ck check (Location_ID LIKE'[L\l][0-9][0-9][0-9][0-9]')
);

---Payment Table
CREATE table     Payment(
P_ID int not null,
Customer_ID char(6) not null,
M_name char(20),
P_type varchar(20),
Amount float,
P_status varchar(10),
Rtype varchar(10),
c_number char(16),

constraint PK_Payment PRIMARY KEY (Customer_ID),
constraint Customer_ID check (Customer_ID LIKE '[m\c][0-9][0-9][0-9][0-9]'), constraint CHK_contNO check
(c_number LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-
9][0-9][0-9]'),
);


--Reservation Table
CREATE table Reservation(
Reserv_ID char(6) not null,
Location_ID char(6) not null,
Customer_ID char(6) not null,
Duration char(20),
Resev_Date date

constraint Reservation_PK PRIMARY KEY(Reserv_ID),
constraint rev_ck check (Reserv_ID LIKE'[H\S][0-9][0-9][0-9][0-9]'), constraint Reservation_FK1 FOREIGN KEY
(Location_ID) references
Park_Location(Location_ID),
constraint Reservation_FK2 FOREIGN KEY (Customer_ID) references Payment(Customer_ID),
);

--Admin Table
CREATE table Admin_s (
A_ID int,
Log_ID char(6),
A_Name varchar(20),
A_Position varchar(20),
A_ConNo char(10),

constraint PKADMIN PRIMARY KEY (A_ID),
constraint FKADMIN FOREIGN KEY (Log_ID) references Login (Log_ID),
);
```

```
--Sample Data
INSERT into Login values ('C12346','Nandun','1234bB');
INSERT into Login values ('C13245','Kaveendra','4569sG');
INSERT into Login values ('E12345','Asmitha','4567bB');
INSERT into Login values ('E13452','Abiramy','4568bB');
INSERT into Login values ('C65783','Prasanga','1267dE');

--Member Info
INSERT into Registered_Member
values('M0001','Nandun','Godage','donnandun@gmail.com','0715589299','Sirisevana,Pahalagama, Wellawaya','C12346');
INSERT into Registered_Member
values('M0002','Kaveendra','Kodithuwakku','kaveendrak@gmail.com','0771231231','100/1, Matale','C13245');
INSERT into Registered_Member
values('M0003','Prasanga','De Davin','prasanga@hotmail.com','0764564560','20/2, Kandy','C65783');
INSERT into Registered_Member
values('M0004','Asmitha','Thiraviyarasa','asmithat@yahoo.com','0726767676','50/1, Jaffna','C98711');
INSERT into Registered_Member
values('M0005','Abiramy','Kumaresan','abiramyk@yahoo.com','0718989898','20/2, Jaffna','C98754');


--Location Info
INSERT into Park_Location values('L0001','Malabe','Kandy Rd,Malabe',500);
INSERT into Park_Location values('L0002','Majestic City',' Station Rd, Colombo',600);
INSERT into Park_Location values('L0003','Liberty Plaza','R. A. De Mel Mawatha, Colombo',450);
INSERT into Park_Location values('l0004','Kandy City Center','High Rd,Kandy',550);
INSERT into Park_Location values('l0005','CarSafe','Kurunegala',650);
```

```
--Payment Info
INSERT into payment values('0022','M0001','Nandun','QR',4500,'Done','Hourly','0715589299');
INSERT into payment values('0033','M0002','Kaveendra','Debit Card ',500,'Done','Hourly','0771231231');
INSERT into payment values('0024','M0003','Asmitha','Debit Card',4400,'Done','Monthly','0726767676');
INSERT into payment values('0067','M0004','Abiramy','Credit Card',800,'Done','Hourly','0718989898');
INSERT into payment values('0078','M0005','prasanga','Paypal',3500,'Done','Monthly','0764564560');


--Reservation Info
INSERT into Reservation values('S0111','L0002','m0001','Hourly','28-04-22');
INSERT into Reservation values('H0002','L0001','m0004','Hourly','28-04-22');
INSERT into Reservation values('H0003','L0002','c2278','Monthly','28-04-22');
INSERT into Reservation values('S0044','L0004','m0005','Hourly','28-04-22');
INSERT into Reservation values('H0005','L0003','c2234','Monthly','28-04-22');



--Admin Details
INSERT into Admin_s values('0001','E12345','Kumar','Operational Manager','0111231231');
INSERT into Admin_s values('0002','E13452','Clark','Security Expert','0114564564');
INSERT into Admin_s values('0003','E12345','Arun','Assistant Manager','0117897897');
INSERT into Admin_s values('0004','E13452','Angela','Web Developer','0712462462');
INSERT into Admin_s values('0005','E65783','Perera','Support Agent','0773693693');


--Inquiry Info
INSERT into INQUIRY values('m0011','Kamal','kamal123@gmail.com','Payment issuse','28-04-22');
INSERT into INQUIRY values('m0090','Amal','amal123@gmail.com','Location issuse','28-04-22');
INSERT into INQUIRY values('m0033','Ravin','ravin123@gmail.com','Parking place issuse','28-04-22');
INSERT into INQUIRY values('m0024','Kavin','kavin123@gmail.com','Reservation issuse','28-04-22');
INSERT into INQUIRY values('m0085','Afra','afra123@gmail.com','Parking
place issuse','28-04-22');
```

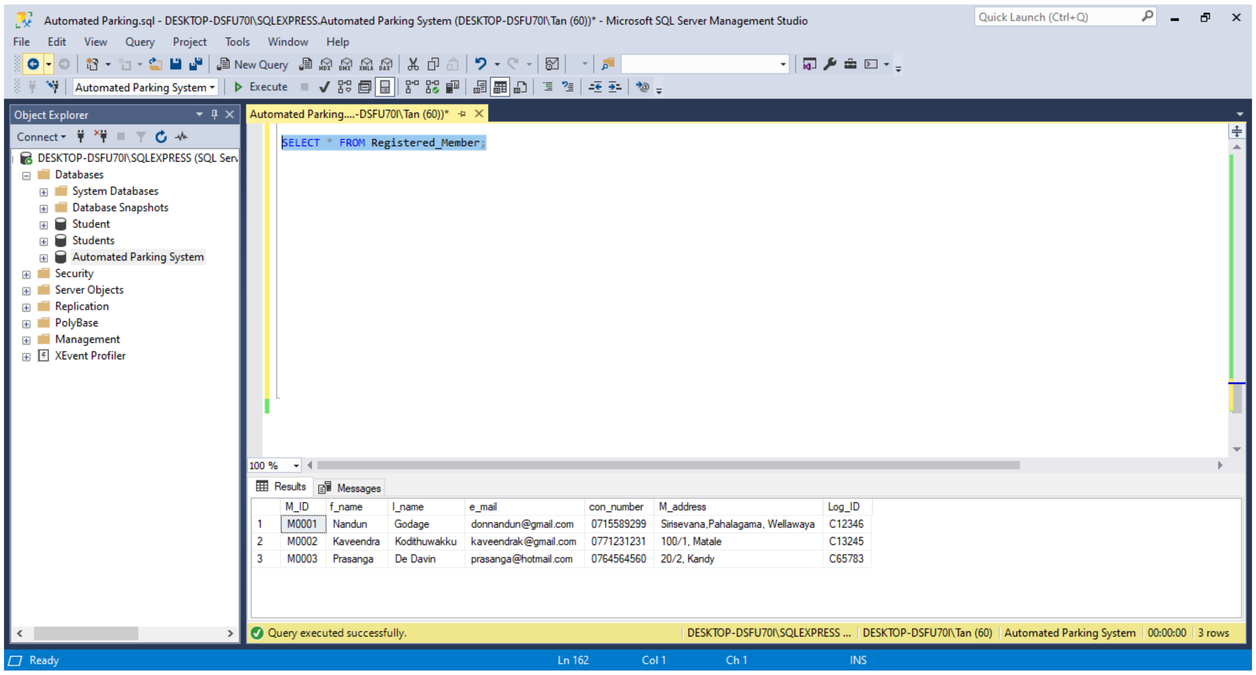
## Examples



```
SELECT * FROM Login;

SELECT * FROM Registered_Member;
```

|  | Log_ID | Username | User_password |
|---|---|---|---|
| 1 | C12346 | Nandun | 1234bB |
| 2 | C13245 | Kaveendra | 4569sG |
| 3 | C65783 | Prasanga | 1267dE |
| 4 | E12345 | Asmitha | 4567bB |
| 5 | E13452 | Abiramy | 4568bB |

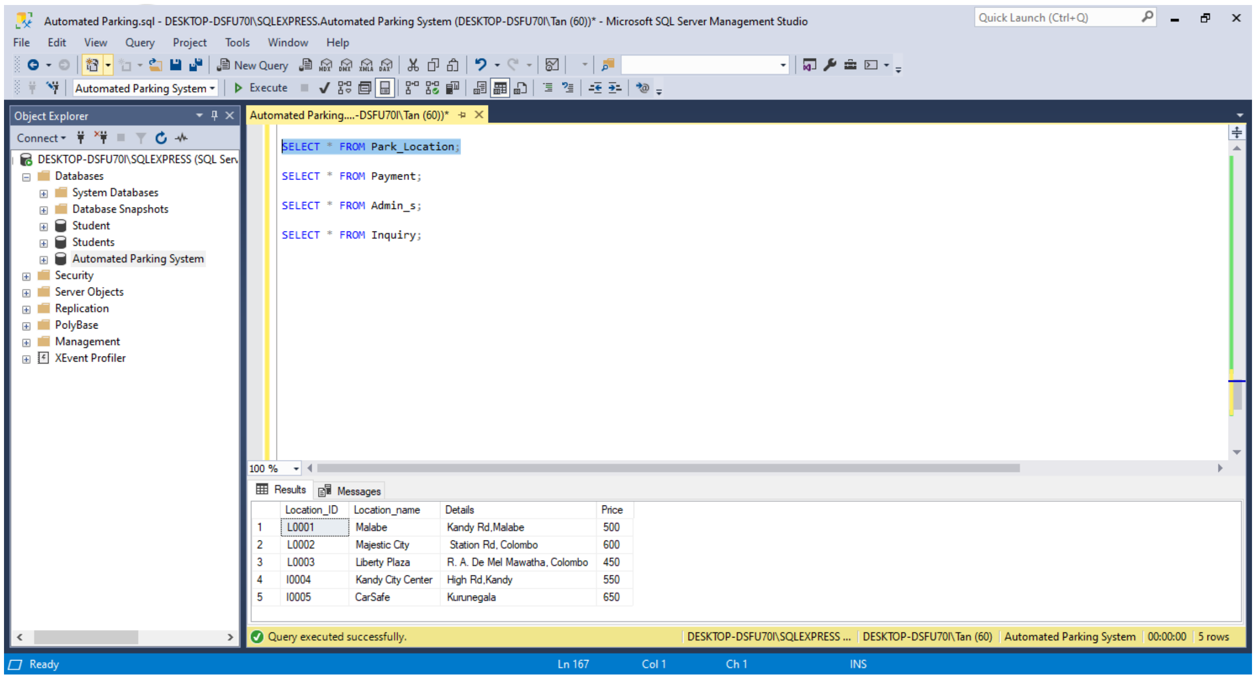
Query executed successfully. DESKTOP-DSFU70I\SQLEXPRESS ... DESKTOP-DSFU70I\Tan (60) Automated Parking System 00:00:00 8 rows



```
SELECT * FROM Registered_Member;
```

|  | M_ID | f_name | l_name | e_mail | con_number | M_address | Log_ID |
|---|---|---|---|---|---|---|---|
| 1 | M0001 | Nandun | Godage | donnandun@gmail.com | 0715589299 | Sirisevana,Pahalagama, Wellawaya | C12346 |
| 2 | M0002 | Kaveendra | Kodithuwakku | kaveendrak@gmail.com | 0771231231 | 100/1, Matale | C13245 |
| 3 | M0003 | Prasanga | De Davin | prasanga@hotmail.com | 0764564560 | 20/2, Kandy | C65783 |

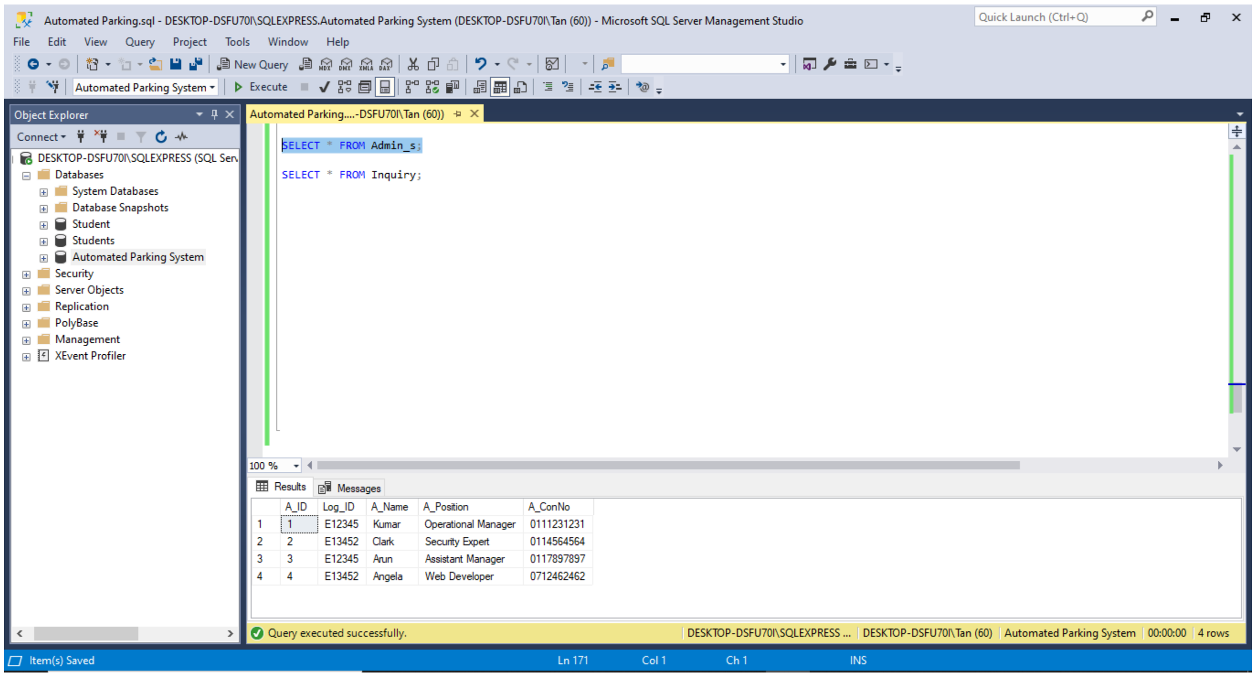
Query executed successfully. DESKTOP-DSFU70I\SQLEXPRESS ... DESKTOP-DSFU70I\Tan (60) Automated Parking System 00:00:00 3 rows


Automated Parking.sql - DESKTOP-DSFU70I\SQLEXPRESS.Automated Parking System (DESKTOP-DSFU70I\Tan (60))* - Microsoft SQL Server Management Studio
Object Explorer
DESKTOP-DSFU70I\SQLEXPRESS (SQL Serv
Databases
System Databases
Database Snapshots
Student
Students
Automated Parking System
Security
Server Objects
Replication
PolyBase
Management
XEvent Profiler
Automated Parking....-DSFU70I\Tan (60))*
Results
Messages

```
SELECT * FROM Park_Location;

SELECT * FROM Payment;

SELECT * FROM Admin_s;

SELECT * FROM Inquiry;
```

| | Location_ID | Location_name | Details | Price |
|---|---|---|---|---|
| 1 | L0001 | Malabe | Kandy Rd,Malabe | 500 |
| 2 | L0002 | Majestic City | Station Rd, Colombo | 600 |
| 3 | L0003 | Liberty Plaza | R. A. De Mel Mawatha, Colombo | 450 |
| 4 | l0004 | Kandy City Center | High Rd,Kandy | 550 |
| 5 | l0005 | CarSafe | Kurunegala | 650 |

```
Query executed successfully.   DESKTOP-DSFU70I\SQLEXPRESS ...   DESKTOP-DSFU70I\Tan (60)   Automated Parking System   00:00:00   5 rows
Ready   Ln 167   Col 1   Ch 1   INS
```


Automated Parking.sql - DESKTOP-DSFU70I\SQLEXPRESS.Automated Parking System (DESKTOP-DSFU70I\Tan (60)) - Microsoft SQL Server Management Studio
Object Explorer
Automated Parking....-DSFU70I\Tan (60))
Results
Messages

```
SELECT * FROM Admin_s;

SELECT * FROM Inquiry;
```

| | A_ID | Log_ID | A_Name | A_Position | A_ConNo |
|---|---|---|---|---|---|
| 1 | 1 | E12345 | Kumar | Operational Manager | 0111231231 |
| 2 | 2 | E13452 | Clark | Security Expert | 0114564564 |
| 3 | 3 | E12345 | Arun | Assistant Manager | 0117897897 |
| 4 | 4 | E13452 | Angela | Web Developer | 0712462462 |

```
Query executed successfully.   DESKTOP-DSFU70I\SQLEXPRESS ...   DESKTOP-DSFU70I\Tan (60)   Automated Parking System   00:00:00   4 rows
Item(s) Saved   Ln 171   Col 1   Ch 1   INS
```

## Performance Considerations

- Any user can search for and save any desired stopping location.
- A registered customer can save an area for the duration of a month.
- Registered customers and administrators must be able to log in to the system as many times as they want.
- The System must be operational 24 hours a day so that a client can access it at any time, and the User must be able to access the services without interruption.
- Users can provide feedback on the system almost immediately.
- Management can keep an eye on the finer details of the customer and the staff.
- The system must respond to the issues as soon as possible.
- Management can keep an eye on the location's finer points and reservation details.
- Customer service representatives must resolve reservation issues.

## Security Requirements

- Visitors should be able to see stopping points through the system.
- Only an enrolled member should be able to reserve a parking space.
- A mail account can only be created by one User account.
- The User account password must be a strong combination of capitalized, lowercase, numerals, and special characters.
- Payment details must be substantial through the bank in order to obtain a stopping ticket.
- When including a modern payment option, ensure that it is accepted by the bank.
- An authorized user, such as an administrator, has the ability to modify system data.

## Contributions

**Godage D.N.P - IT21175398**

- Writing the introduction
- Describing the Hypothetical Scenario
- Supporting to make the requirement analysis
- Supporting to gather Data requirements
- Supporting to model the ER diagram & implement the Relational Schema
- Supporting to make Special security requirements
- Supporting  to Identify Special performance considerations
- Writing SQL commands to create the database tables
- Implementing the database
- Creating, checking document errors, and guiding the group members towards the completion of the assignment

**Kodithuwakku K.G.K.M.J  - IT21173936**

- Making requirement analysis
- Identify special security requirements
- Supporting to write SQL commands
- Supporting to model the ER diagram & implement the Relational Schema
- Supporting to implement the database

**De Davin G.G.P.N - IT21174612**

- Supporting to write SQL commands
- Drawing ER diagram
- Implement the Relational Schema
- Supporting to implement the database

**Asmitha Thiraviyarasa - IT21176210**

- Supporting to write the Hypothetical Scenario
- Identifying special performance considerations
- Supporting to write SQL commands
- Drawing ER diagram
- Implementing the Relational Schema
- Supporting to implement the database

**Abiramy Kumaresan - IT21176142**

- Identifying the Data requirements
- Supporting to write SQL commands
- Drawing ER diagram
- Implementing the Relational Schema
- Supporting to implement the database