Topic: Automated Parking System

Group no: MLB_01.01_08

Campus: Malabe

Submission Date: 15 /10/2021

We declare that this is our own work, and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	**Name**	**Contact Number**
IT21044168	Perera K. H. T	0714074458
IT21036002	Pathirathna S. N	0701254564
IT21015212	Tennekoon V. L. K	0717474477
IT21011566	Perera D. T. S	0717128428
IT21003714	Rimas M. J. M	0775481997

Introduction

Online parking reservation system facilitate people to acquire a parking space in a much more convenient way. Scheduling and reserving allow the user to reserve their desired parking lot without any burden. Furthermore, the system allows the users to get an idea on the status of the location and location routes whether it is reserved or not or have customer required facilities.

The system is designed to check and record the available parking spaces in a certain location with the route address. Also, the system enables the users to check the availability of location. The objective of this system is to design an automated parking reservation system which allows the user to save time and avoid traffic jam and make it more convenient for reservation through online payment methods. Thus, this system allows for reservations without any inconvenience recommend the most facilitated and safest parking places.

User Requirements

1. A user can create an account as a customer or as an admin. When creating the customer account, it should be an email or phone number that is not already in use of an account.
2. Users can login to the system by entering correct user credentials.
3. Unregistered customer can register in the system by entering the details.
4. Customers can access the website, view, and reserve available parking locations.
5. Non-registered customer can only book for hourly parking whereas registered customers have the privilege of booking hourly and monthly parking.
6. Customers can place the reservation through online transactions. Payment can be paid through debit/credit or through PayPal. Registered customers can wish to do their monthly payment at the end month as a monthly subscription.
7. Only registered customers receive discounts for the prepaid monthly reservations.
8. System admin can update/edit information of parking locations and add/remove new parking locations to/from the system.
9. Customers need to contact the admins to resolve any issues related to reservations.
10. Customers can give their feedback and suggestions.
11. System admin can generate reports such as financial reports.
12. Registered customer can edit and manage the account.

Noun/Verb Analysis

1. A user can create an account as a customer or as an admin. When creating the customer account, it should be an email or phone number that is not already in use of an account.
2. Users can login to the system by entering correct user credentials.
3. Unregistered customer can register in the system by entering the relevant customer details.
4. Customers can access the website, view, and reserve available parking locations.
5. Non-registered customer can only book for hourly reservations whereas registered customers have the privilege of booking hourly and monthly reservations.
6. Customers can place the reservation through online transactions. The Payment can be paid through debit/credit card or through PayPal. Registered customers can wish to do their monthly payment at the end month as a monthly subscription.
7. Only registered customers receive discounts for the prepaid monthly reservations.
8. System admin can update/edit information of parking locations and add/remove new parking locations to/from the system.
9. Customers need to contact the admins to resolve any inquires related to reservations.
10. Customers can give their feedback and suggestions.
11. System admin can generate reports such as financial reports.
12. Registered customer can edit and manage the account.

Noun Analysis

1. User
2. Account
3. Customer
4. Admin
5. Customer account
6. Email
7. Phone number
8. Users
9. System
10. User credentials
11. Unregistered customer
12. System
13. Customer details
14. Customers
15. Website
16. Parking locations
17. Non-registered customer
18. Hourly reservations
19. Registered customers
20. Privilege
21. Monthly reservations
22. Customers
23. reservation
24. online transactions
25. Payment
26. debit/credit card
27. PayPal
28. Registered customers
29. Payment
30. Subscription
31. Registered customers
32. Discounts
33. Prepaid monthly reservations.
34. System admin
35. Information
36. Parking locations
37. Parking locations
38. system
39. Customers
40. Admins
41. Inquires
42. Reservations
43. Customers
44. feedback
45. suggestions.
46. System admin
47. reports
48. financial reports
49. Registered customer
50. account

Classes

- Unregistered customer
- Registered customers
- Parking locations
- Hourly Reservation
- Monthly Reservations
- Payment
- Inquires
- Feedback
- Admin
- Reports

Out of scope

- Customer account
- System
- User credentials
- System
- Website
- Privilege
- System admin
- Information
- System admin
- Account

Redundant

- Customer
- Users
- Customers
- Non-registered customer
- Customers
- Registered customers
- Admins
- Parking locations
- Unregistered customer
- Suggestions
- Reservations
- Customers
- Registered customer

Attribute

- Email
- Phone number
- Customer details
- Hourly parking
- Online transactions
- debit/credit card
- Subscription
- Discounts
- Prepaid monthly reservations
- Financial reports

Verb Analysis

Unregistered Customer

- Create
- Register
- Entering
- Access
- View
- Reserve
- Book
- Place
- Paid
- Contact
- Give

Registered Customer

- Login
- Entering
- Access
- View
- Reserve
- Booking
- Place
- Paid
- Wish
- Receive
- Contact
- Edit/Manage

Admin

- Login
- Entering
- Update/edit
- Add/remove
- Resolve
- Generate

CRC Cards

Class: Unregistered Customer	
Responsibility	**Collaborations**
Register Details	
Search Location	
View Location	Parking Location
Reserve Location	Hourly Reservation, Monthly Reservation
Pay Parking fees	Payment
Make Inquiry	Inquiry
Give Feedback	Feedback

Class : Registered Customer	
Responsibility	**Collaborations**
Login to account	
Search Location	
View Location	Parking Location
Reserve Location	Hourly Reservation, Monthly Reservation
Pay Parking fees	Payment
Receive Discount	
Make Inquiry	Inquiry
Give Feedback	Feedback

Class: Parking Locations	
Responsibility	**Collaborations**
Add parking locations	
Remove parking locations	
Update parking locations	

Class: Payment	
Responsibility	**Collaborations**
Store the payment details	
Place the reservation through online transaction methods.	
Will receive discounts	
Display Payment details	

Class Name: Hourly Reservations	
Responsibility	**Collaborations**
Store reservations details	Payment, location
Update reservation details	
Display reservation details	

Class Name: Monthly Reservations	
Responsibility	**Collaborations**
Store reservations details	Payment, Location
Update reservation details	
Display reservation details	

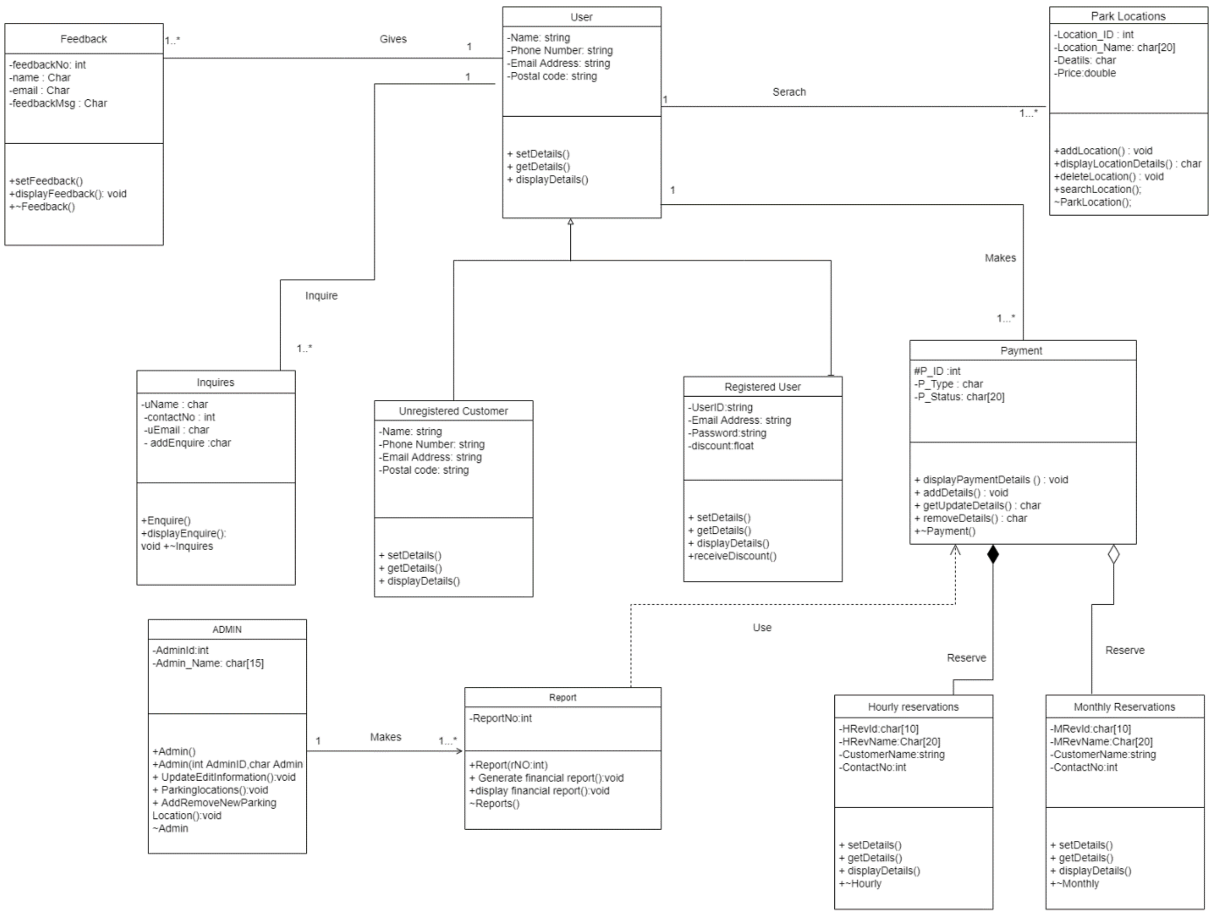
Class: Feedback	
Responsibility	**Collaborations**
Add User Details	
Add Feedback	
Display Feedback	

Class: Inquires	
Responsibility	**Collaborations**
Add User Details	
Add Enquire	
Update Enquire	
Display Enquire	

Class: Admin	
Responsibility	**Collaborations**
Login to system	
Update/edit information	Customer
Parking Locations	
Add/remove new parking location	

Class: Reports	
Responsibility	**Collaborations**
Generate financial report	Customer
Display reports	

Class Diagram



Codes

Customer.h

```
class Customer {
   protected: char name[20];
              char email[30];
              char num[11];
              char code[5];

   public: void setDetails(char const pname, char const pemail,char const
pnum, char const pcode);
           char getDetails();
           Customer() {}
           Customer(char const pname[], char const pemail[], char const
pnum[], char const pcode[]);
           void displayDetails();



};

class Unregistered_customer: public Customer{
    private: char userID[6];

    public:Unregistered_customer( char const pname[], char const pemail[],char
const pnum[],char  const pcode[]);
           void displayunregDetails();
           char getDetails();

};

class Registered_customer: public Customer{
    private: char userID[6];
             char password[8];
             float discount;

    public:Registered_customer(char const ID[],char const pemail[], char const
ppw[], float disc);
             void displayregDetails();
             char getDetails();



};
```

Customer.cpp

```
#include <iostream>
#include<cstring>
#include "customer.h"
using namespace std;

Customer::Customer(char const pname[], char const pemail[], char const pnum[],
char const pcode[]){
  strcpy( name,pname);
  strcpy( email, pemail);
  strcpy ( num, pnum);
  strcpy (code, pcode);
}

void Customer::displayDetails(){
    cout<< name <<endl;
    cout<< email <<endl;
    cout<< num <<endl;
    cout<< code <<endl;
    cout<<endl;
}

Unregistered_customer::Unregistered_customer(char const pname[],char const
pemail[],char const pnum[],char const pcode[]){
  strcpy( name,pname);
  strcpy( email, pemail);
  strcpy ( num, pnum);
  strcpy (code, pcode);
};


void Unregistered_customer::displayunregDetails(){

    cout<<"Name: "<< name <<endl;
    cout<<"Email Address" <<email <<endl;
    cout<<"Phone Number: "<< num <<endl;
    cout<<"Postal Code: "<< code <<endl;
    cout<<endl;

}


Registered_customer::Registered_customer(char const ID[], char const
pemail[],char const ppw[], float disc){

  strcpy( userID,ID);
  strcpy( email, pemail);
  strcpy ( password, ppw);
  discount= disc;

};

void Registered_customer::displayregDetails(){
    cout<<"User ID: "<< userID <<endl;
    cout<<"Email Address: "<< email <<endl;
    cout<<"Password: "<< password <<endl;
    cout<<"Discount Amount: Rs."<< discount <<endl;
    cout<<endl;

}
```

Park_Location.h

```
class ParkLocation {

  private:
    int Location_ID;
    char Location_Name[20];
    char Details[25];
    double Price;

  public:
    void parkLocation(int pLocation_ID, char pLocation_Name[], char
pDetails[], double pPrice);
    void addLocation();
    void displayLocationDetails();
    void deleteLocation();
    void searchLocation();
};
```

Park Location.cpp

```
#include<iostream>
#include<cstring>
#include"ParkLocation.h"
using namespace std;



void ParkLocation::parkLocation(int pLocation_ID, char pLocation_Name[], char
pDetails[], double pPrice)
{

   Location_ID = pLocation_ID;
  strcpy(Location_Name, pLocation_Name);
  strcpy(Details, pDetails);
   Price = pPrice;
}

void ParkLocation::addLocation()
{

}
void ParkLocation::displayLocationDetails()
{
  cout<<"Location ID: "<<Location_ID<<endl;
  cout<<"Location Name: "<<Location_Name<<endl;
  cout<<"Details: "<<Details<<endl;
  cout<<"Price: "<<Price<<endl;
}
```

Payment.h

```
class Payment{

private:
  int p_ID;
  char p_Type[25];
  char p_Status[20];

public:
  void payment(int pP_ID, char pP_Type[], char pP_Status[]);
  void displayPaymentDetails();
  void addDetails();
  void getUpdatePaymentDetails();
  char removeDetails();
};
```

Payment.cpp

```
#include<iostream>
#include<cstring>
#include"payment.h"
using namespace std;


void Payment::payment(int pP_ID, char pP_Type[], char pP_Status[]){

   p_ID = pP_ID;
  strcpy (p_Type ,pP_Type);
  strcpy(p_Status ,pP_Status);
}
void Payment::displayPaymentDetails(){
  cout<<"Payment ID: "<<p_ID<<endl;
  cout<<"Payment Type: "<<p_Type<<endl;
  cout<<"Payment Status: "<<p_Status<<endl;
}

void Payment::getUpdatePaymentDetails()
{

}
```

Hourly.h

```
class Hourly {
private:
       int HRevID;
       char HRevName[40];
       char CustomerName[20];
       char ContactNo[10];

public:
   Hourly();
   void setHourlyDetails(int H_ID,const char hName[],const char cName[],const
char cNo[]);
   void displayHourlyDetails();
   void updateHourlyDeatils();
   ~Hourly(void);

};
```

Hourly.cpp

```
#include "Hourly.h"
#include <iostream>
#include<cstring>
using namespace std;

Hourly::Hourly()
{
  HRevID=0;
  strcpy(HRevName, "");
  strcpy(CustomerName, "");
  strcpy(ContactNo, "");
}



void Hourly::setHourlyDetails(int H_ID,const char hName[],const char cName[],
const char cNo[])
{
       HRevID=H_ID;
       strcpy(HRevName,hName);
       strcpy(CustomerName,cName);
       strcpy(ContactNo,cNo);

}
  void Hourly::displayHourlyDetails()
{
    cout << "Hourly reservation ID: " << HRevID << endl;
    cout << "Resevation Name:  " << HRevName << endl;
    cout << "Customer Name: " <<CustomerName<< endl;
     cout << "Contact No:" <<ContactNo<< endl;
    cout << endl;
}

   void Hourly::updateHourlyDeatils()
   {

   }

   Hourly::~Hourly()
  {

  }
```

Monthly.h

```
class Monthly {
private:
	int MRevID;
	char MRevName[40];
	char MemberName[20];
	char ContactNO[10];


public:
  Monthly();
  void setMonthlyDetails(int MID,const char MName[],const char cName[], const
char cNo[]);
  void displayMonthlyDetails();
  void updateMonthlyDeatils();
  ~Monthly(void);
};
```

Monthly.cpp

```
#include "Monthly.h"
#include <iostream>
#include<cstring>
using namespace std;

Monthly::Monthly()
{
  MRevID=0;
  strcpy(MRevName, "");
  strcpy(MemberName, "");
  strcpy(ContactNO, "");
}


  void Monthly::setMonthlyDetails(int MID,const char MName[],const char
cName[], const char cNo[])
{
	MRevID=MID;
	strcpy(MRevName,MName);
```

```
       strcpy(MemberName,cName);
       strcpy(ContactNO,cNo);

}
  void Monthly::displayMonthlyDetails()
{
    cout << "Hourly reservation ID: " << MRevID << endl;
    cout << "Resevation Name:   " << MRevName << endl;
    cout << "Customer Name: " <<MemberName<< endl;
     cout << "Contact No:" <<ContactNO<< endl;
    cout << endl;
}

   void Monthly::updateMonthlyDeatils()
   {

   }

   Monthly::~Monthly()
  {

  }
```

Feedback.h

```
class Feedback
{
private:
  int feedbackNo;
  char name[25];
  char email[50];
  char feedbackMsg[100];

public:
  void feedback(int fbNo, char userName[], char userEmail[], char
addFeedback[]);
  void displayFeedback();
};
```

Feedback.cpp

```
#include "Feedback.h"
#include <iostream>
#include <cstring>
using namespace std;

// Assign User Details and Feedback Message

void Feedback::feedback(int fbNo, char userName[], char userEmail[], char
addFeedback[]) {
  int feedbackNo = fbNo;
  strcpy(name, userName);
  strcpy(email, userEmail);
  strcpy(feedbackMsg, addFeedback);
}

// Display User Details and Feedback Message

void Feedback::displayFeedback() {
  cout << "Feedback No : " << feedbackNo << endl;
  cout << "Customer Name : " << name << endl;
  cout << "Email : " << email << endl;
  cout << "Feedback : " << feedbackMsg << endl;
}
```

Inquiry.h

```
class Inquiry
{
private:
  char uName[25];
  int contactNo;
  char uEmail[50];
  char addEnquire[100];

public:
  void enquire(char username[], int cNo, char useremail[], char addEnq[]);
  void displayEnquire();
};
```

Inquiry.cpp

```
#include "Inquiry.h"
#include <iostream>
#include <cstring>
using namespace std;


// Assign User Details and Inquiry Details

void Inquiry::enquire(char username[], int cNo, char useremail[], char
addEnq[]){
  strcpy(uName, username);
  int  contactNo = cNo;
  strcpy(uEmail, useremail);
  strcpy(addEnquire, addEnq);
}

// Display User Details and Inquiry Details

void Inquiry::displayEnquire(){
  cout << "Customer Name : " << uName << endl;
  cout << "Contact Number : " << contactNo << endl;
  cout << "Email : " << uEmail << endl;
  cout << "Inquiry : " << addEnquire << endl;
}
```

Admin.h

```
class Admin
{
private:
	int AdminID;
	char name[25];

public:
	void Admins(int ADNo, char ADName[]);
	void displayAdmin();
};
```

Admin.cpp

```
#include "Admin.h"
#include <iostream>
#include <cstring>
using namespace std;

void Admin::Admins(int ADNo, char ADName[])
{
	int AdminID = ADNo;
	strcpy(name, ADName);
}

void Admin::displayAdmin()
```

```
{
	cout << "ADMIN ID : " << AdminID << endl;
	cout << "ADMIN Name : " << name << endl;
}
```

Report.h

```
class Report
{
private:
	int ReportNo;
	char name[25];

public:
	void Reports(int rpNo, char reportName[]);
	void displayReport();
};
```

Report.cpp

```
#include "Report.h"
#include <iostream>
#include <cstring>
using namespace std;




void Report::Reports(int rpNo, char reportName[])
```

```
{
	int ReportNo = rpNo;

	strcpy(name, reportName);

}


void Report::displayReport()

{

	cout << "Report No : " << ReportNo << endl;

	cout << "Repoet Name : " << name << endl;


}
```

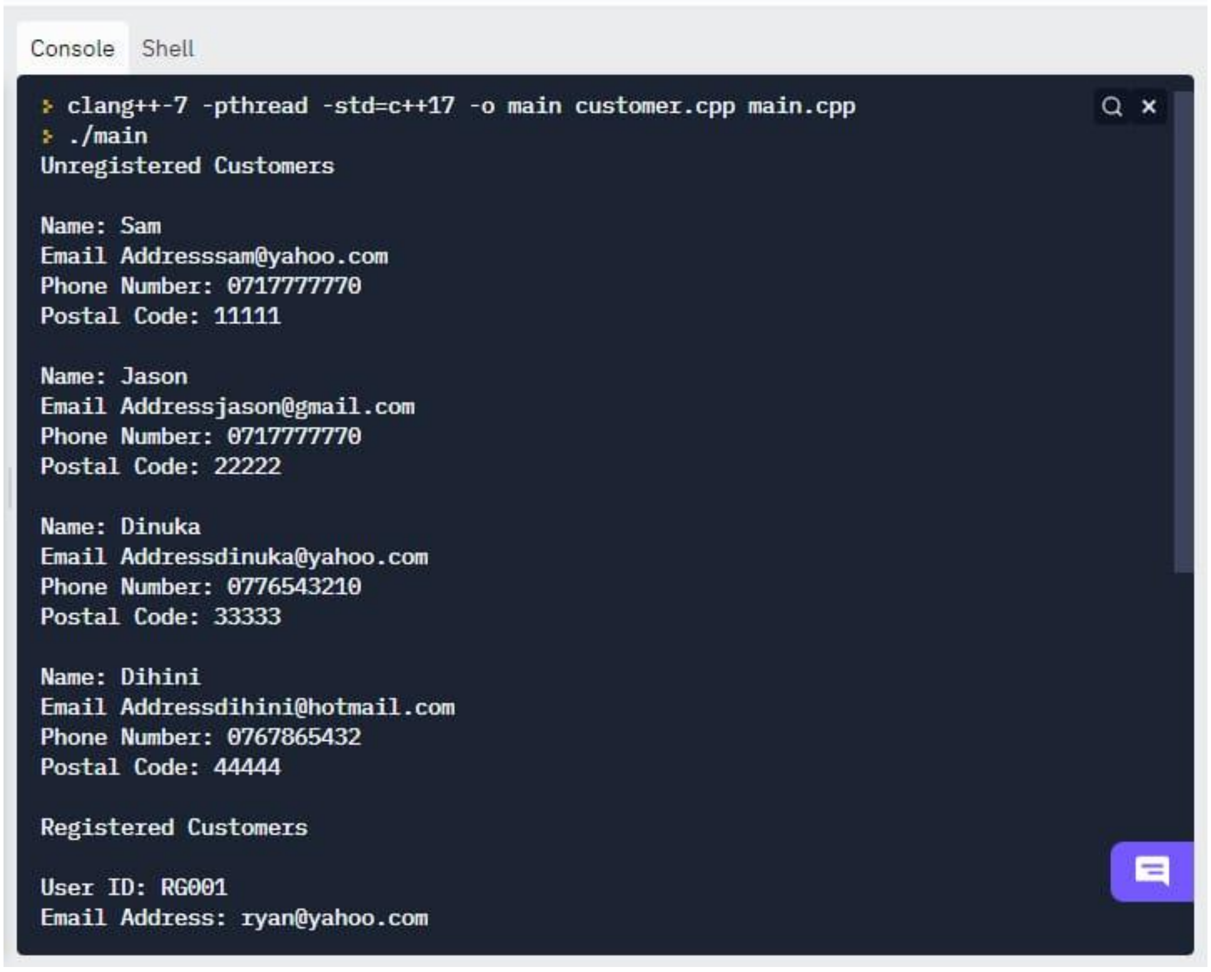
Main.cpp

```
 #include<iostream>
                              #include<cstring>
                              #include "customer.h"
                              #include"payment.h"
                              #include"ParkLocation.h"
                              #include "Hourly.h"
                              #include "Monthly.h"
                              #include "Admin.h"
                              #include "Report.h"
                              #include "Feedback.h"
                              #include "Inquiry.h"

                              using namespace std;

                              int main()
                              {

                              cout<<"Unregistered Customers " <<endl<<endl;
                                Unregistered_customer UC1("Sam", "sam@yahoo.com",
                              "0717777770", "11111");
                                UC1.displayunregDetails();
                                 Unregistered_customer UC2("Jason", "jason@gmail.com",
                              "0717777770", "22222");
                                UC2.displayunregDetails();
                                 Unregistered_customer UC3("Dinuka", "dinuka@yahoo.com",
                              "0776543210", "33333");
                                UC3.displayunregDetails();
                                 Unregistered_customer UC4("Dihini", "dihini@hotmail.com",
                              "0767865432", "44444");
                                UC4.displayunregDetails();
                              cout<<"--------------------------------"<<endl;


                                cout<<"Registered Customers " <<endl<<endl;
                                Registered_customer RC1("RG001", "ryan@yahoo.com", "zxcv8n7",
                              100.00);
                                RC1.displayregDetails();
                                 Registered_customer RC2("RG002", "thilini@gmail.com",
                              "as3h6gf", 150.00 );
                                RC2.displayregDetails();
                                 Registered_customer RC3("RG003", "shenal@gmail.com",
                              "qwe6rty", 100.00);
                                RC3.displayregDetails();
```

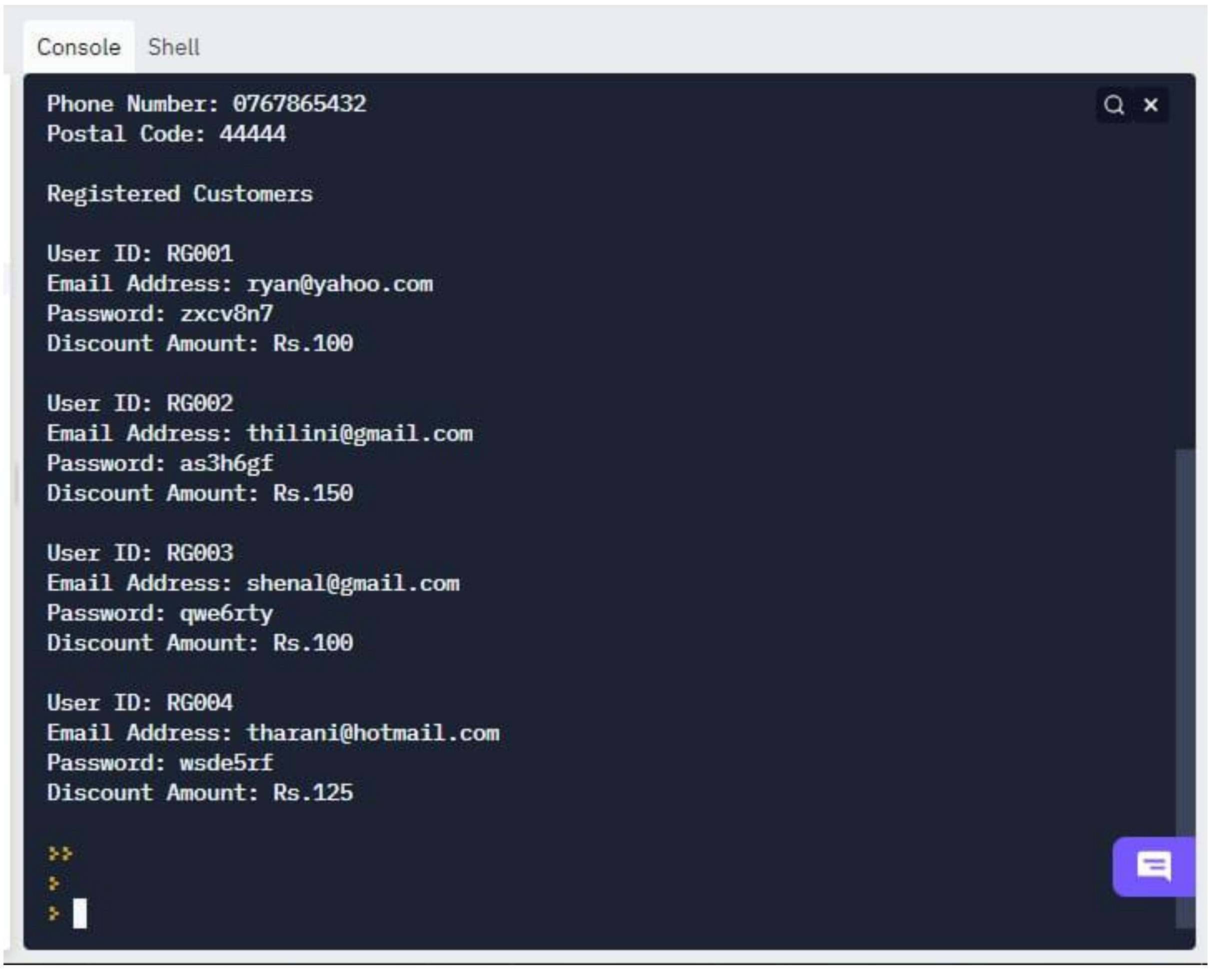
```
   Registered_customer RC4("RG004", "tharani@hotmail.com",
"wsde5rf", 125.00);
  RC4.displayregDetails();
  cout<<"--------------------------------"<<endl;

  Payment pay;
  pay.payment(12,(char*)"PayPal",(char*)"Done");
  pay.displayPaymentDetails();

  cout<<"--------------------------------"<<endl;

  ParkLocation parkL;
  parkL.parkLocation(0001,(char*)"NugegodaPark",
(char*)"Nugegoda Highlevel Rd, Nugegoda", 500.00);
  parkL.displayLocationDetails();
  cout<<"--------------------------------"<<endl;

  Hourly h1,h2;



    h1.setHourlyDetails(1001, "Nugegoda Park","Dihini",
"0767765432");
    h2.setHourlyDetails(1002, "TownHall",
"Dinuka","0776543210");

    h1.displayHourlyDetails();
    h2.displayHourlyDetails();
cout<<"--------------------------------"<<endl;

    Monthly m1,m2;



    m1.setMonthlyDetails(1011, "Nugegoda Park","Ann",
"076765389");
    m2.setMonthlyDetails(1022, "Mall Center
Nugegoda","Janithi","0712543890");

    m1.displayMonthlyDetails();
    m2.displayMonthlyDetails();
    cout<<"--------------------------------"<<endl;

  Report rpNo;
      rpNo.Reports(4197392, (char*)"CUSTOMR REPORTS");
      rpNo.displayReport();

      cout << "\n" << endl;
```

```
        cout << "--------------------------------" << endl;

        Admin ADNo;
        ADNo.Admins(4, (char*)"Nimal");
        ADNo.displayAdmin();
        cout << "\n" << endl;
        cout << "--------------------------------" << endl;


 Feedback fb;
  fb.feedback(17, (char*)"Saman Perera",
(char*)"samanPerera@gmail.com", (char*)"Great Solution");
//Create Feedback object
  fb.displayFeedback(); //Display Feedback

  cout << "\n" << endl;

  Inquiry inq;
  inq.enquire((char*)"Dasun Nethsara", 717128328,
(char*)"Nethsara@gmail.com", (char*)"Change Password");
//Create Inquiry object
  inq.displayEnquire(); //Display Inquiry


    char ch;
    cin>>ch;


  return 0;

}
```

Sample Output

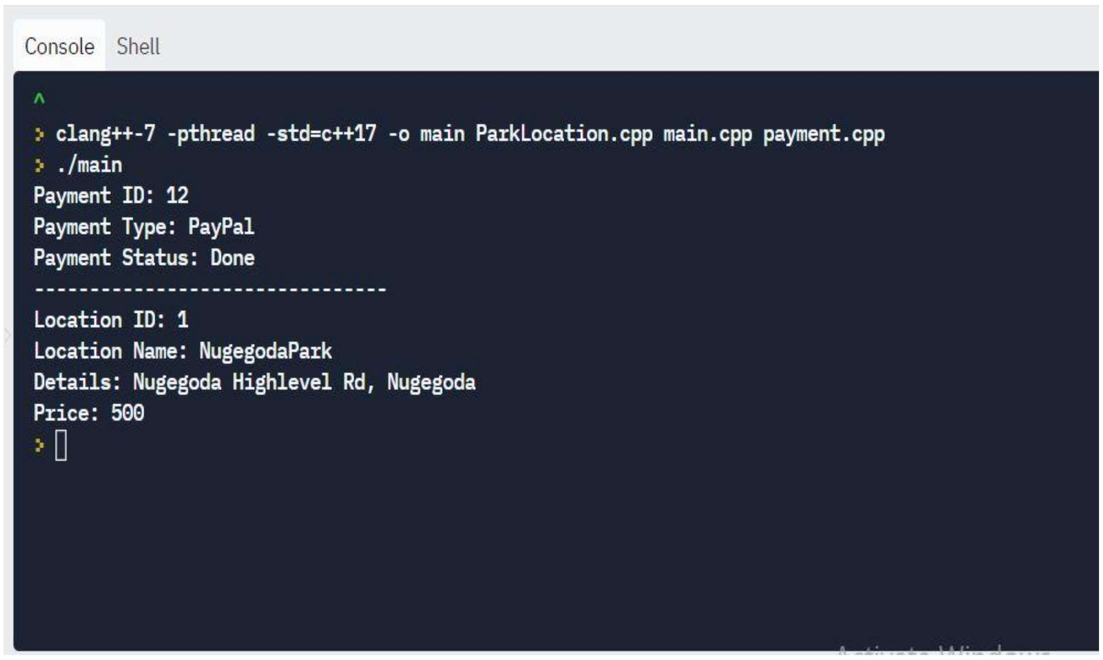


```
> clang++-7 -pthread -std=c++17 -o main customer.cpp main.cpp
> ./main
Unregistered Customers

Name: Sam
Email Addresssam@yahoo.com
Phone Number: 0717777770
Postal Code: 11111

Name: Jason
Email Addressjason@gmail.com
Phone Number: 0717777770
Postal Code: 22222

Name: Dinuka
Email Addressdinuka@yahoo.com
Phone Number: 0776543210
Postal Code: 33333

Name: Dihini
Email Addressdihini@hotmail.com
Phone Number: 0767865432
Postal Code: 44444

Registered Customers

User ID: RG001
Email Address: ryan@yahoo.com
```



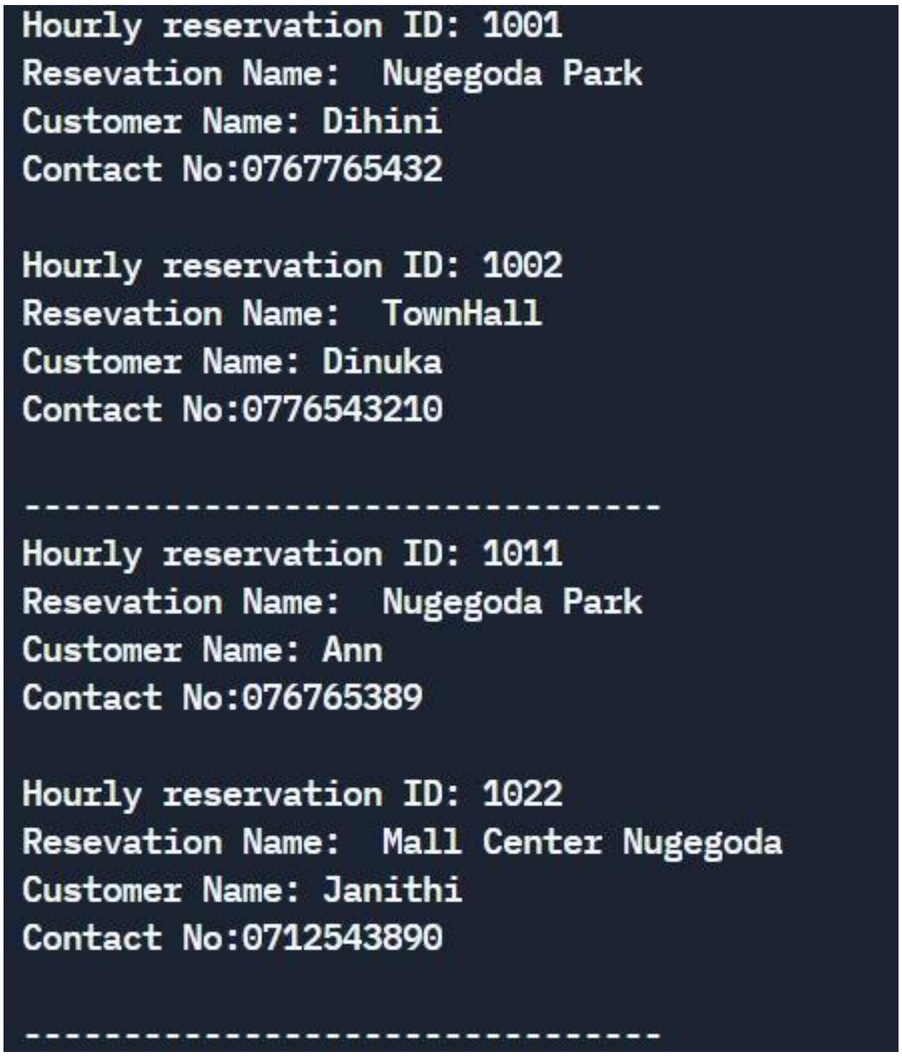
Console Shell

```
Phone Number: 0767865432
Postal Code: 44444

Registered Customers

User ID: RG001
Email Address: ryan@yahoo.com
Password: zxcv8n7
Discount Amount: Rs.100

User ID: RG002
Email Address: thilini@gmail.com
Password: as3h6gf
Discount Amount: Rs.150

User ID: RG003
Email Address: shenal@gmail.com
Password: qwe6rty
Discount Amount: Rs.100

User ID: RG004
Email Address: tharani@hotmail.com
Password: wsde5rf
Discount Amount: Rs.125

>>
>
>
```

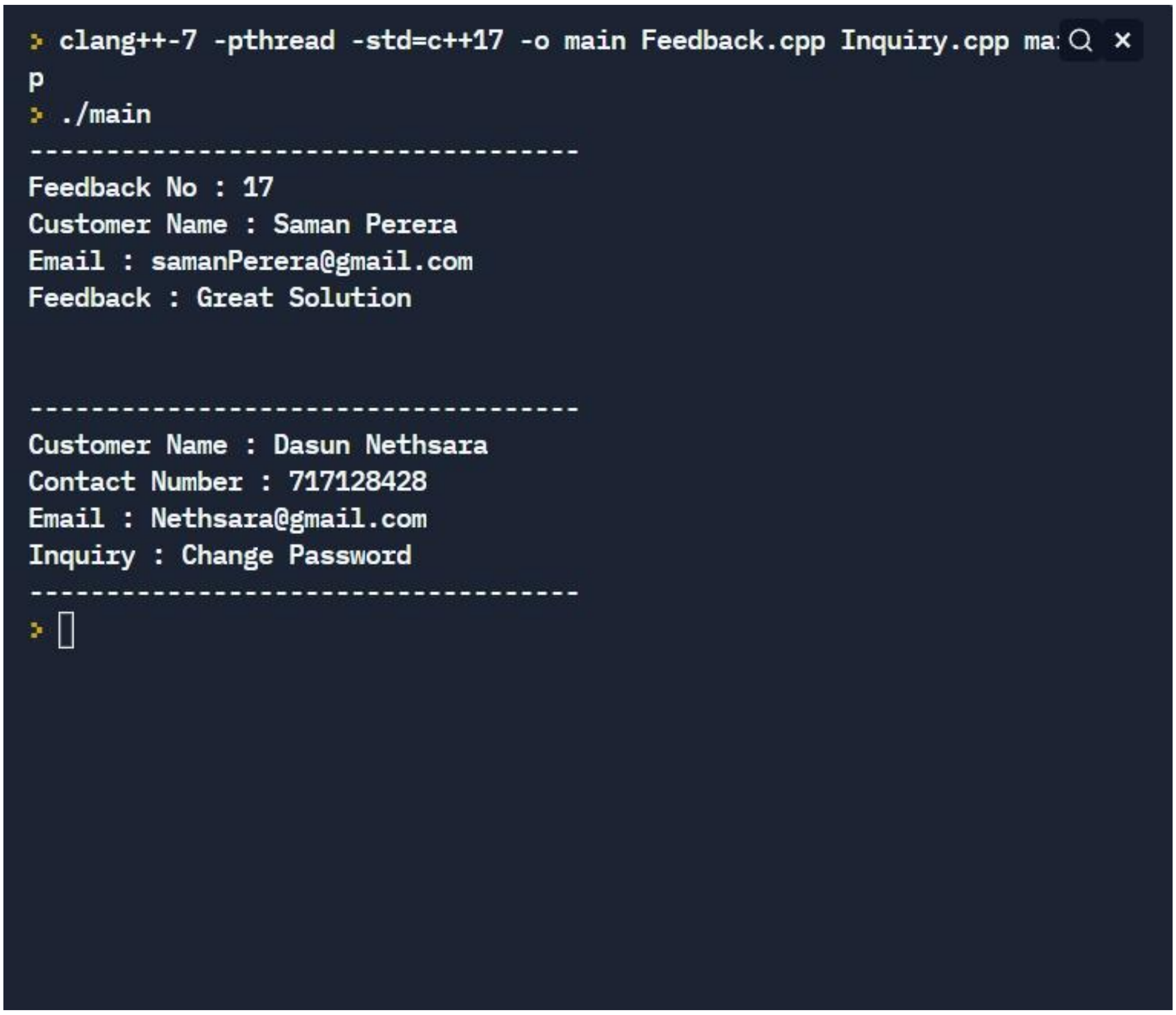


Console Shell

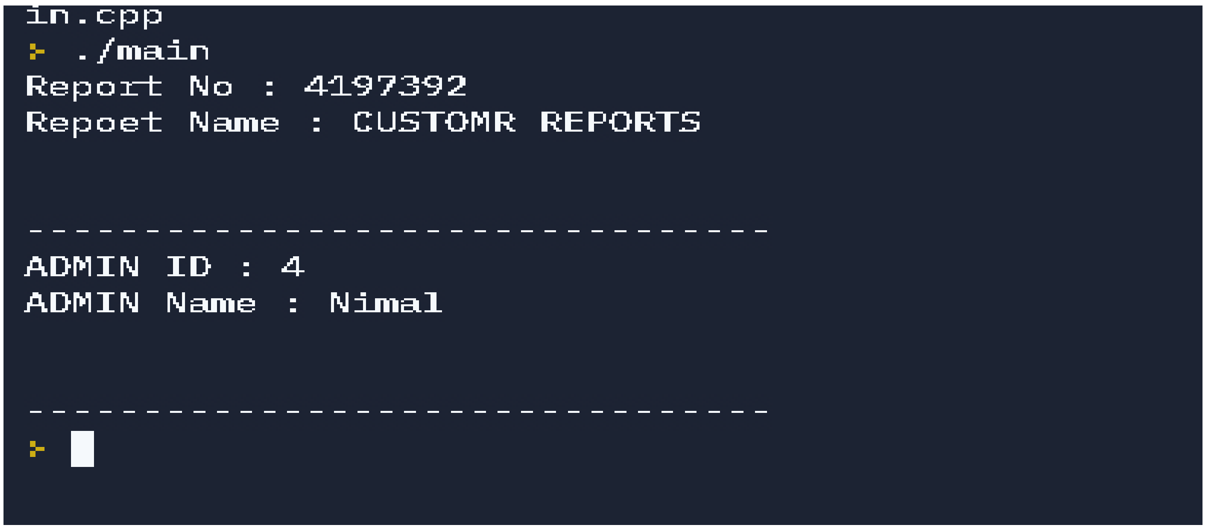
```
^
> clang++-7 -pthread -std=c++17 -o main ParkLocation.cpp main.cpp payment.cpp
> ./main
Payment ID: 12
Payment Type: PayPal
Payment Status: Done
--------------------------------
Location ID: 1
Location Name: NugegodaPark
Details: Nugegoda Highlevel Rd, Nugegoda
Price: 500
>
```



```
Hourly reservation ID: 1001
Resevation Name:  Nugegoda Park
Customer Name: Dihini
Contact No:0767765432

Hourly reservation ID: 1002
Resevation Name:  TownHall
Customer Name: Dinuka
Contact No:0776543210


--------------------------------
Hourly reservation ID: 1011
Resevation Name:  Nugegoda Park
Customer Name: Ann
Contact No:076765389

Hourly reservation ID: 1022
Resevation Name:  Mall Center Nugegoda
Customer Name: Janithi
Contact No:0712543890


--------------------------------
```



```
> clang++-7 -pthread -std=c++17 -o main Feedback.cpp Inquiry.cpp ma
p
> ./main
-----------------------------------
Feedback No : 17
Customer Name : Saman Perera
Email : samanPerera@gmail.com
Feedback : Great Solution


-----------------------------------
Customer Name : Dasun Nethsara
Contact Number : 717128428
Email : Nethsara@gmail.com
Inquiry : Change Password
-----------------------------------
> 
```



```
in.cpp
> ./main
Report No : 4197392
Repoet Name : CUSTOMR REPORTS


-------------------------------
ADMIN ID : 4
ADMIN Name : Nimal


-------------------------------
> 
```

Individual Contributions

IT21044168 -Perera K.H.T

- Created the CRC Card for Payment and Park Location classes.
- Created the class diagram for Payment and Park Location.
- Implemented the coding for the Payment class and the Park Location class.

IT21036002 -S.N.Pathirathna

- Created and documented the user requirements and noun verb analysis.
- Created the class diagram for hourly and monthly reservations.
- Created the CRC card for hourly and monthly reservations
- Created the classes and coded the hourly and monthly reservations.

IT21015212 – Tennakoon V.L.K

- CRC cards for unregistered customer and registered customer
- Identifying the relationships between unregistered and registered customer and other classes
- Drawing of class diagram for unregistered customer and registered customer with relationships to other classes
- Coding of unregistered and registered customer classes

IT21011566 – Perera D.T.S

- I have drawn the class diagram for the Feedback and Inquiry.
- Designed the CRC Cards for the Inquiry class and the Feedback class.
- I have implemented the codes for the Feedback and Inquiry.

IT21003714 - M.J.M.RIMAS

•Created the class diagram for admin and report
•Created the CRC card for admin and report
•Create the classes and code the admin and report