Topic : Hotel Reservation System For Special events

Group no : MLB_07.01_12

Campus : Malabe

Submission Date :

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**
IT21158018	R.A.Ahamed	076 8497485
IT21159176	D.G.T.D.Deniyegedara	077 1920470
IT21158568	P.Sindujan	076 0098222
IT21158872	W.M.B.V.Vimukthi	077 3318249
IT21159480	M.I.M.Mufeel	075 7416964

System Requirements

1. Visitors can overview the system by using a web browser and they can check the availability of services.
2. If a visitor wants to make a reservation, they must be they must register with the system by providing details such as Name, Address, Email and contact.
3. register users can log in to the system by using a unique username and password.
4. After that registered user can make a reservation or payment.
5. After reservation, date of reservation and reservation ID is generated.
6. Registered customers must enter their payment details like payment type, card details.
7. After the conformation of payment details, date of payment and payment ID is generated.
8. If visitors and the registrar user want to know more about the system or event, they can resolve it by inquiring and those inquiries will be answered by an admin.
9. Admin must log in to the system by using the admin username and password to answer those inquiries.
10. admin should be able to add event details such as event price and event name to the system.
11. Admin can do lots of work in this system such as generate reports (financial, reservation…etc.), approve the reservations, add new packages, update the packages, and can watch the summarized feedback.
12. If staff wants to make a reservation, they must be login to the system by using their username and password.
13. Staff can get a discount when they make a payment for their reservation by using staff ID.
14. When a registered user makes a reservation, if there is a discount on the relevant event, it will be added at the time of payment.

Identified Classes

1. Visitor
2. register User
3. reservation
4. payment
5. admin
6. event and packages
7. report
8. staff
9. discount
10. inquiries

CRC Cards

Visitor	
responsibility	**collaborators**
Checking the availability	Reservation
Register to the system	

Register user	
responsibility	**collaborators**
Log in to the system	
Checking the availability and make a reservation	Reservation
Make a payment	Payment

Reservation	
responsibility	**collaborators**
Place reservation	
Calculate amount	Payment
Display bill amount	

Payment	
responsibility	**collaborators**
Make a payment	
Conform the payment details	Register User, Reservation
Calculate discount	Discount
Display final bill amount	

Admin	
responsibility	**collaborators**
Log in to the admin account	
Replay to the inquiries	Inquiries
Generate reports	Reports
Add new packages and update the packages	Packages
Approve reservation	Reservation

Event and packages	
Responsibility	**collaborators**
Add event details and update the system	Admin
Add new packages and update under the relevant event	Admin
Display the details	

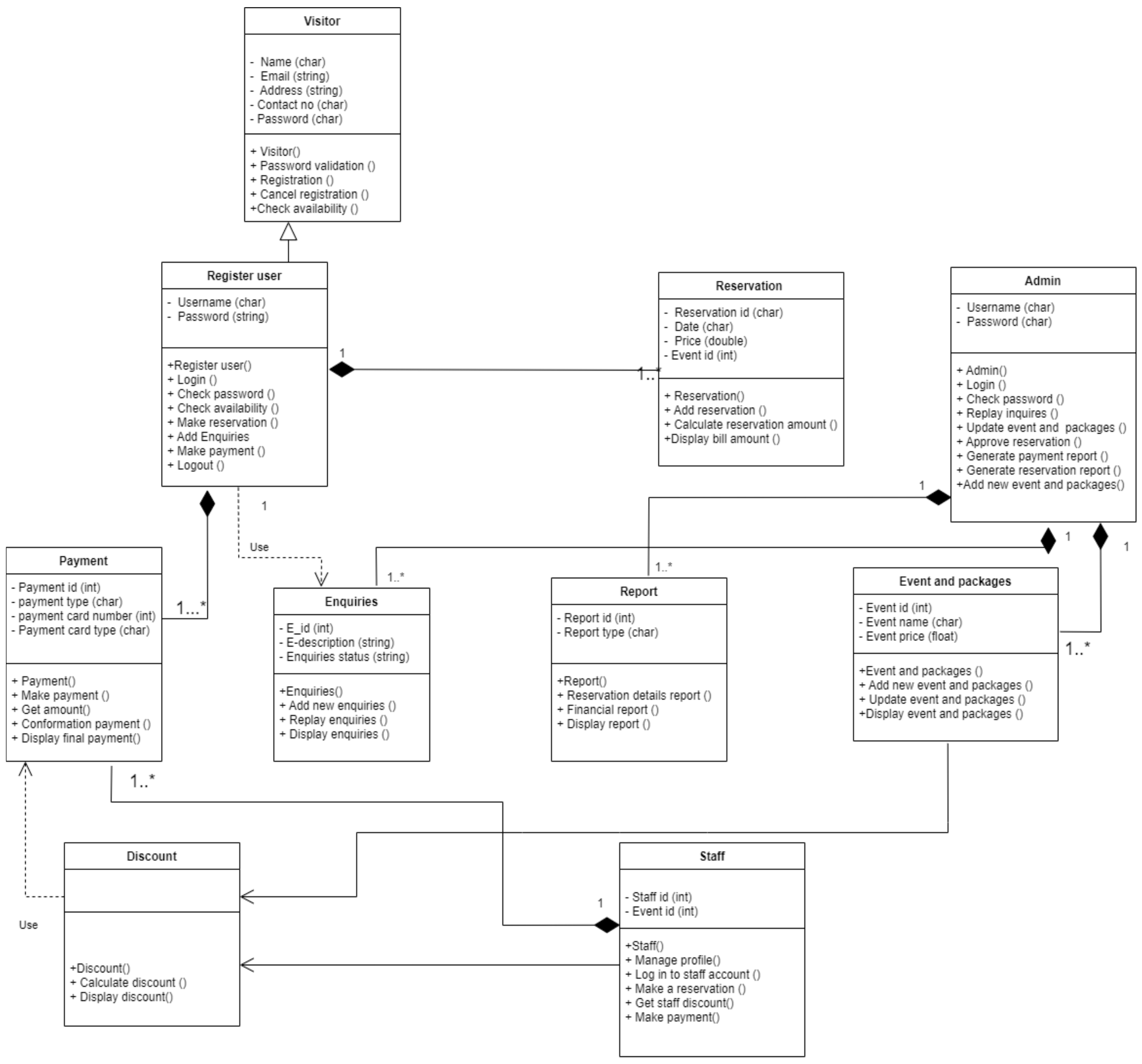
Reports	
Responsibility	**collaborators**
create reservation report	Reservation
Create financial report	Payment
Display the report	

Staff	
Responsibility	**collaborators**
Log in to staff account	
Make a reservation	Reservation
Get staff discount	Discount
Make payment	Payment

Discount	
Responsibility	**collaborators**
Calculate discount	
Display the discount amount	Payment

Inquiries	
Responsibility	**collaborators**
Add new inquiries	
Answering to the inquiries	Admin

Class Diagram



;

Code

Reservation.h

```
class Reservation{
	private:
		char ReservationID[8];
		char Date[10];
		double Price;
		int EventId;
	public:
		Reservation();
		Reservation(char rID[],char date[],double price,int eID);
		void AddReservation();
		int calculateReservationAmount();
		void DisplayBillAmount();
		~Reservation();
};
```

Reservation.cpp

```
#include <iostrean>
#include “Reservation.h”
#include <cstring>

Reservation::Reservation(){
	strcpy(ReservationID,"");
	strcpy(Date,"");
	Price=0;
	EventId=0;
};
```

```
Reservation::Reservation(char rID[],char date[],double price,int eID){
        strcpy(ReservationID,rID);
        strcpy(Date,date);
        Price=price;
        EventId=eID;
 };

int Reservation::calculateReservationAmount(){
};
void Reservation::DisplayBillAmount(){
};
Reservation::~Reservation(){
};
```

Visitor.h

```
class visitor {
        Protected:
     Char name;
     Char email;
     Char address;
     int contact_no;
     Char password;

        Public:
     visitor ();
     visitor ( Char R_name,Char umail, Char adr,int c_no,Char psw);
     void PasswordValidation();
            Void registration ();
            Void CancelRegistration ();
            Void CheckAvailability ();
            ~visitor ();
};
```

Visitor.cpp

```
#include <iostrean>
#include “visitor.h”
#include <cstring>

visitor::visitor (){
      Strcpy(name,"");
      Strcpy(email,"");
      Strcpy(address,"");
      contact_no = 0;
      Strcpy(password,"")
};
visitor::visitor(Char R_name,Char umail, Char adr,int c_no,Char psw) {

      Strcpy(name,R_name);
      Strcpy(email,umail);
      Strcpy(address,adr);
      contact_no = c_no;
      Strcpy(password,psw);
};
void visitor:: PasswordValidation(){
};
```

```
Void visitor::registration (){
};
Void visitor::CancelRegistration (){
};
Void visitor::CheckAvailability (){
};
visitor::~visitor (){
};
```

<u>RegisterUser.h</u>

```
#include "visitor.h"
#include "Enquiries.h"
#include "Reservation.h"
#include "payment.h"
#define size 2
Class RegisterUser : Public visitor{

        Protected:
                Reservation *r1[size];
                payment *p1[size];
                Char username;
                Char password;
        Public:
                RegisterUser();
                RegisterUser(Reservation *r[],payment *p[], Char uname,Char psw);
                Void login ();
                Void CheckPassword ();
                Void CheckAvailability ();
                Void AddEnquire( Enquire *e1);
                Void MakeReservation();
                Void MakePayment ();
                Void logout ();
                ~RegisterUser ();
        }
```

RegisterUser.cpp

```
#include <iostrean>
#include “RegisterUser.h”
#include <cstring>



RegisterUser::RegisterUser(){
    for ( int i=0; i<size; i++){
        r1[i] = 0;
    };
    for ( int a=0; i<size; i++){
        p1[i] = 0;
    };
    Strcpy(username,"");
    Strcpy(password,"");
};
```

```
RegisterUser::RegisterUser(Reservation *r[],payment *p[],Char uname,Char
psw){
        for ( int i=0; i<size; i++){
                r1[i] = r[i];
        };
        for ( int a=0; a<size; a++){
                p1[a] = p[a];
        };
        Strcpy(username,uname);
        Strcpy(password,psw);
};

Void RegisterUser::login(){
};
Void RegisterUser::CheckPassword(){
};
Void RegisterUser::CheckAvailability(){
};
VoidRegisterUser::AddEnquire(Enquire *e1){
};
Void RegisterUser::MakeReservation(){
};
Void RegisterUser::MakePayment(){
};
Void RegisterUser::logout(){
};
RegisterUser::~RegisterUser(){
        for ( int i=0; i<size; i++){
                delete r1[i] ;
        };
        for ( int a=0; a<size; a++){
                delete p1[a] ;
        };
};
```

Payment.h

```
class payment // make payment class
{
//declare propties
private :
int payment_id;
char payment_type[20];
int payment_card_number;
char payment_card_type[20];

//declare function
public :

payment();//default constractor

payment(int p_id,char p_type[20] , int p_c_number ,char p_c_type [20] ); //over loding
constructor(constructor with parameters)

void MakePayment();

int Getamount();

void ComfomationPayment();

void DispayFinalPayment();

~payment();//destructor constructor
};
```

Payment.cpp

```
#include<iostream>

#include <cstring>

#include"Payment.h"

using namespace std;

payment::payment()//default constructor

{

	payment_id=0;//payment id initalizing to zero

	strcpy(payment_type,""); //set payment type to blank

	payment_card_number=0;// payment card number initalizing to zero

	strcpy(payment_card_type,""); //set payment card type to blank

};


payment::payment(int p_id,char p_type[20] , int p_c_number ,char p_c_type[20]) //overloaded
constructor

{

	payment_id = p_id;

	strcpy(payment_type,p_type);

	payment_card_number=p_c_number;
```

```
        strcpy(payment_card_type,p_c_type);

};

void payment::MakePayment()// makepayment function implementataion

{



};

int payment::Getamount()//getammout function implementataion

{

};

void payment::ComfomationPayment()//conformation function implementataion

{

};

void payment::DispayFinalPayment()//displayfinalpayment function implementataion

{

};

payment::~payment()// destructor

{

};
```

Inquiries.h

```
using namespace std;
class enquiries // make inquiries class
```

```
{
        private :
                int  enquiries_id;
                char enquiries_description[50];
                char enquiries_status[50];

        public :

                enquiries();//default constractor

                enquiries(int e_id, char e_description[50], char e_status[50]); //over
loading constractor

                void Add_new_enquiries();

                void Replay_enquiries();

                void Display_enquiries();

                ~enquiries();//overloading constructor
};
```

Enquiries.cpp

```
#include<iostreem>
#include" enquiries.h"
#include <cstring>
enquiries::enquiries()
```

```
{
        enquiries_id=0;  //inquiries id initalizing to zero

        strcpy(enquiries_description,""); // set inquiries description to blank

        strcpy(enquiries_status,""); //set inquiries description to blank

};


enquiries::enquiries(int e_id, char e_description[50] , char e_status[50])
//overloaded constructor
{

        enquiries_id=e_id; //set inquiries_id=i_id

        strcpy(enquiries_description,e_description); //set inquiries_description to
i_description

        strcpy(enquiries_status,e_status);        //set inquiries_status to i_status
};
```

<u>Staff.h</u>

```
#include"Payment.h"
#define size 2

using namespace std;
```

```
class Staff {
protected:
        int staff_id;
        int eventId;
        Payment *pm [size]
public:
        Staff();
        Staff(Payment *mm [],int sID);
        void manageProfile();
        void makeReservation();
        int getStaffDiscount();
        void makePayment();
        ~Staff();
};
```

Staff.cpp

```
#include<iostream>
#include <cstring>
#include" Staff.h"
Staff::Staff() {
        staff_id = 0;
        for(int i=0;i<size;i++){
                pm[i]=0;
        }
}
```

```
Staff::Staff(Payment *mm [],int sID){
        staff_id = sID;
        for(int i=0;i<size;i++){
                pm[i]=mm[i];
        }
}
void Staff::manageProfile() {}

void Staff::makeReservation() {}

int Staff::getStaffDiscount() {}

void Staff::makePayment() {}

Staff :: ~staff(){
                for(int i=0;i<size;i++){
                        delete pm[i];
        }

}
```

Admin.h

```
#include"Enquries.h"
#include"Report.h"
#include"event_and_packages.h"
#define size 5
using namespace std;
class admin{
        private:
                Enquries *Enq[size];
                Report *Rp[size];
                event_and_packages *Eap[size];
                char username[20];
                char password[15];
        public:
            admin();
            admin(Enquries *En[],Report *Rip[],event_and_packages *Ep[],char
u_name[],char pass[]);
                void loging();
                void checkpassword();
                void reply_enquries();
                void  update_event_and_packages();
                void  approve_recervation();
                void  genarate_payment_report();
                void genarate_recervation_report();
                void  add_new_event_and_packages();
                ~admin();

};
```

<u>Admin.cpp</u>

```
#include<iostream>
#include" Admin.h"
#include <cstring>
```

```
  admin::admin(){



        for (int i=0;i<size;i++){
                    Enq[i]=0;
          }
        for (int p=0;i<size;i++){
                    Rp[p]=0;


          }
        for (int c=0;i<size;i++){
                   Eap[c]=0;
          }


        strcpy(username,"");
        sctrcpy(password,"");
  }
  admin::admin(Enquries *En[],Report *Rip[],event_and_packages *Ep[],char
u_name[],char pass[]){


        for (int i=0;i<size;i++){
                    Enq[i]=En[i];
          }
        for (int p=0;i<size;i++){
                    Rp[p]=Rip[p];
```

```
            }
        for (int c=0;i<size;i++){
                    Eap[c]=Ep[c];
        }

        strcpy(username,u_name);
        strcpy(password,pass);
    }
    void admin::loging(){

    }
    void admin::checkpassword(){
    }
    void admin::reply_enquries(){
    }
    void admin::update_event_and_packages(){
    }
    void admin::approve_recervation(){
    }
    void admin::genarate_payment_report(){
    }
    void admin::genarate_recervation_report(){
    }
    void admin::add_new_event_and_packages(){
    }
    admin::~admin(){
```

```
	for (int i=0;i<size;i++){
		delete Enq[i];
	  }
	for (int p=0;i<size;i++){
		 delete Rp[p];


	  }
	for (int c=0;i<size;i++){
		delete Eap[c];
	}
	}
```

event_and_packages.h

```
#include "Discount.h"
using namespace std;
class event_and_packages{
	protected:
		Discount *D1;
		int eventid;
		char eventname[20];
		float eventprice;

	public:
		event_and_packages();
		event_and_packages(Discount *D,int e_id,char e_name[],float e_price);
		void add_new_event_and_packages();
		void update_event_and_packages();
```

```
                    void display_event_and_packages();
                    ~event_and_packages();

};
```

event_and_packages.cpp

```
#include<iostream>
#include <cstring>
#include "event_and_packages.h"

event_and_packages::event_and_packages(){
          D1 = new Discount(0);
          eventid=0;
          strcpy(eventname,"");
          eventprice=0;


          };
          event_and_packages::event_and_packages(Discount *D,int e_id,char e_name[],float
e_price){
          D1 = new Discount(D);
          eventid=e_id;
          strcpy(eventname,e_name);
          eventprice=e_price;
          };

   event_and_packages::add_new_event_and_packages(){


  };
  void event_and_packages::update_event_and_packages(){
  } ;
  void event_and_packages::display_event_and_packages(){
  } ;

   event_and_packages::~event_and_packages(){

                    delete D1;
  };
Report.h
#include <iostream>
#include <cstring>
using namespace std;
class Report{
        private:
                int ReportID;
                char ReportType[10];
        public:
                Report();
                Report(int rID,char rType[]);
                void ReservationDetailsReport();
                void financialReport();
                void displayReport();
                ~Report();
};

Report::Report(){
        ReportID=0;
        strcpy(ReportType,"");
        };
```

```
Report::Report(int rID,char rType[]){
        ReportID=rID;
        strcpy(ReportType,rType);

};
```

Report.cpp

```
void ReservationDetailsReport(){

};
void Report::financialReport(){
};
void Report::displayReport(){
};
Report::~Report(){
};
```

Main.cpp

```
#include <iostream>
#include “Admin.h ”
#include “Discount.h ”
#include “ Enquiries.h”
#include “ Even and packages.h”
#include “ Payment.h”
#include “Recervation.h ”
#include “ RegisterUser.h”
#include “Report.h ”
#include “ Staff.h”
#include “ visitor.h”
int main ()

{

        Reservation *re ;

        Report *r;

        visitor *v1;

        RegisterUser *ru1;

        payment *p1;

        inquiries *i1;
        admin *ad;

        Staff *st;

        event_and_packages *ev;

        discount*do;




        delete  re, r, v1, ru1, p1, i1, ad, st, ev, do;
}
```