Topic : **Resource Booking System**

Group no : **MAT_PG.01.01_11**

Campus : **Matara.**

Submission Date : 20/10/2020

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**
IT20172350	M.P.O.M Gomes	0715282784
IT20255510	Y.N Jayasekara	0770377445
IT20163204	Nishan Himanka Manimendra	0718822810
IT20153618	R.Sahan Sandeepa	0786130630
IT20148522	Gihan Madhuhara E.G.	0763335341

Exercise 1:

1)

- Associate any reservation with an account.
- Limit every account to a single user.
- Accept the date and time for the specific time to search available rooms.
- Calculate and view charges for accommodation and other service.
- Cancel Reservations.
- Display and change records of guests.
- In the event of invalid input, users should be sufficiently supported to fill in the mandatory fields.
- System should accept payments via various payment methods.
- Change rooms.
- Guest can modify reservation details before payment.
- Guest can make an inquiry for their needed.

2)

The guest can reservation a room in different types of room for guest's priority. Frist guest must visit the web page of "Redstone hotel" official page. Then Guest must log in to user account with username & password, If Guest haven't a user account, he/she must make an account and register. Then he/she can select the hotel or resort branches from Colombo, Mirissa, Kandy, next guest can select room type. When guest select room type receptionist check room availability and notified and display the room for guest selection. Then guest has the available room for reservation, He/she can select some additional item & services for their reservation if they want. After select the all needed for guest, payment will generate for selected type of room and display the amount to pay. Then he/she can make the payment for hotel booking for selected method (pay later or pay now). Guest can cancel the booking if they want. Guest can also make an inquiry about reserve a wedding hall or conference hall. When customer make an inquiry, system notified the receptionist there is an inquiry. Then receptionist send information for the inquiry for guest. Receptionist also confirmed the check-in and check-out status and manage payment in hotel reservations. Also, guest can give a feedback about hotel and their service. Hotel administer of this hotel also have ability to do receptionist work. It has ability to update hotel information and create report of reservation in hotel.

NOUN	**VERB**
Redundant User – Guest **Attributes** User account Hotel Resort Branches Colombo Mirissa Kandy	**Customer** Booking Cancel Log in Select Register Visit **Receptionist** Manage Notified

Hotel information Username Password **Out of scope** Web page Hotel Administer **Classes** Room Payment Inquires Receptionist Reservation Additional Item & Service Feedback Report	**Hotel Administer** Create Update **System** Check Generate Display

Guest	
Responsibility	Collaboration
Register	
Provide guest information.	
Keep track of reservation.	Reservation
Add new payment.	Payment
Update user profile	
Leave feedbacks	Feedback

Room	
Responsibility	Collaboration
Provide room information.	
Calculate reservation amount.	
Cost	
Update details	

Reservation	
Responsibility	Collaboration
Provide details of reservation.	
Handel process where room booked	Room
Confirm reservation.	

Payment	
Responsibility	Collaboration
Keep a record of all customer payment.	Guest
Calculate and store amount of deposit.	Room
retuned.	

Receptionist	
Responsibility	Collaboration
Keep a record of all reservation.	Reservation.
Keep a record of all guest.	Guest.
Keep track of guest check-in and check-out status.	
Respond to guest inquiries.	Inquires.

Inquires	
Responsibility	Collaboration
Keep a record of all inquiries.	Guest.

Additional Item & Services	
Responsibility	Collaboration
Keep record of all Item & services.	
Manage all the operation of services.	

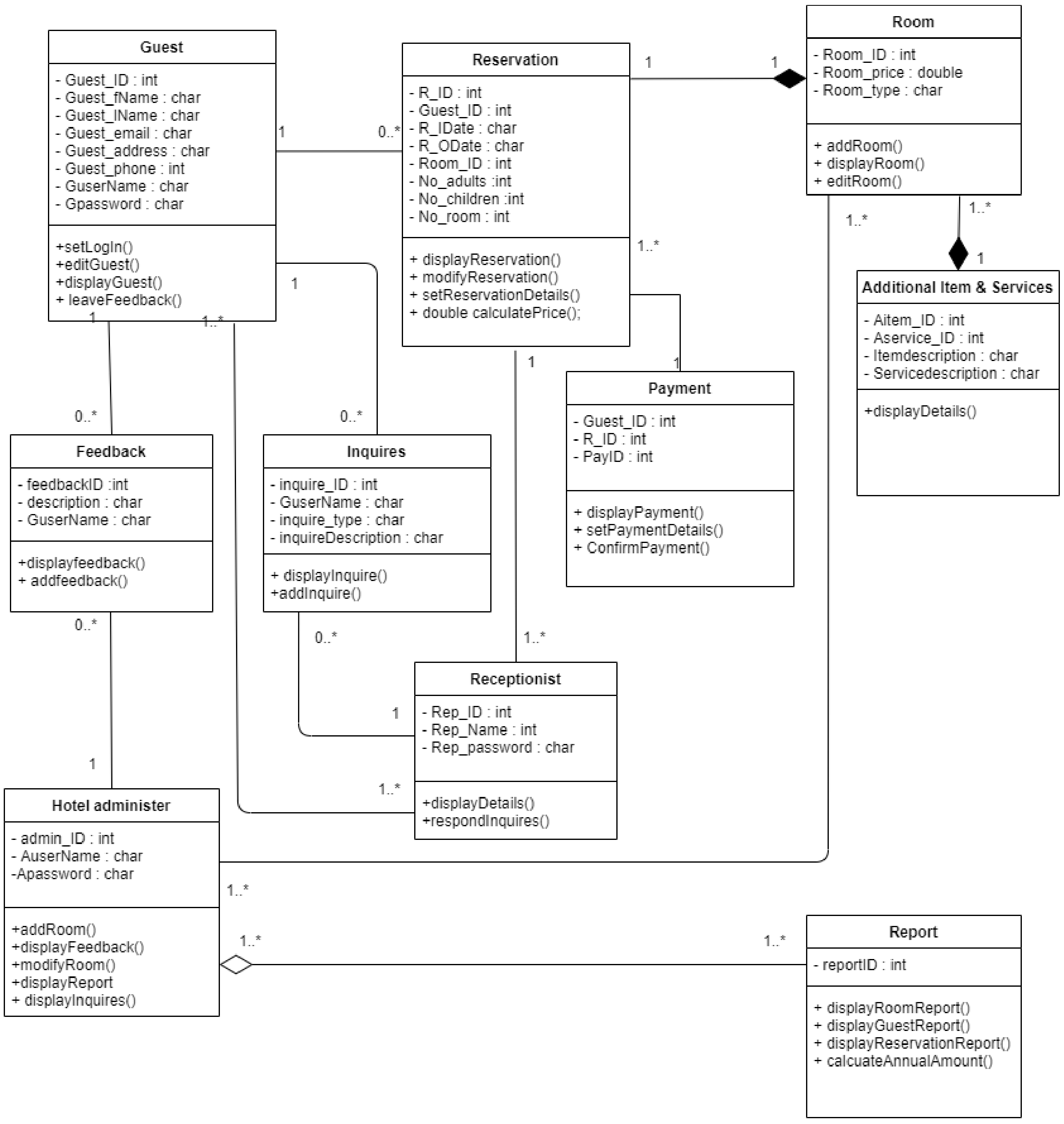
Report	
Responsibility	Collaboration
List of reservation history	Reservation
List of room details	Room

Photo List of guest details	Guest
Annual accounting summary	Payment
Annual sales report	

Feedback	
Responsibility	Collaboration
Show feedbacks	
Store feedbacks description	

Hotel administer	
Responsibility	Collaboration
Add rooms	Room
Create reports	Report
Respond to guest inquires	Inquires

3)



Guest
- Guest_ID : int
- Guest_fName : char
- Guest_lName : char
- Guest_email : char
- Guest_address : char
- Guest_phone : int
- GuserName : char
- Gpassword : char
+setLogIn()
+editGuest()
+displayGuest()
+ leaveFeedback()
Reservation
- R_ID : int
- Guest_ID : int
- R_IDate : char
- R_ODate : char
- Room_ID : int
- No_adults :int
- No_children :int
- No_room : int
+ displayReservation()
+ modifyReservation()
+ setReservationDetails()
+ double calculatePrice();
Room
- Room_ID : int
- Room_price : double
- Room_type : char
+ addRoom()
+ displayRoom()
+ editRoom()
Additional Item & Services
- Aitem_ID : int
- Aservice_ID : int
- Itemdescription : char
- Servicedescription : char
+displayDetails()
Payment
- Guest_ID : int
- R_ID : int
- PayID : int
+ displayPayment()
+ setPaymentDetails()
+ ConfirmPayment()
Feedback
- feedbackID :int
- description : char
- GuserName : char
+displayfeedback()
+ addfeedback()
Inquires
- inquire_ID : int
- GuserName : char
- inquire_type : char
- inquireDescription : char
+ displayInquire()
+addInquire()
Receptionist
- Rep_ID : int
- Rep_Name : int
- Rep_password : char
+displayDetails()
+respondInquires()
Hotel administer
- admin_ID : int
- AuserName : char
-Apassword : char
+addRoom()
+displayFeedback()
+modifyRoom()
+displayReport
+ displayInquires()
Report
- reportID : int
+ displayRoomReport()
+ displayGuestReport()
+ displayReservationReport()
+ calcuateAnnualAmount()
1
0..*
1
1
1..*
1..*
1
1..*
1
1
1
1..*
0..*
1..*
0..*
0..*
1
0..*
1..*
1..*
1
1..*
1..*
1..*

4)

4.1)

```
class Guest
  {
          private :
                    int Guest_ID;
                    char Guest_fName[20];
                    char Guest_lName[20];
                    char Guest_email[30];
                    char Guest_address[50];
                    int Guest_phone;
                    char GuserName[10];
                    char Gpassword[10];
          public :
                     Guest();
                     Guest(char GfName[], char GlName[], char Gemail[], char Gaddress[], int
Gphone);
                     void setLogIn(char GUName[], char GPassword[]);
                     void editGuest();
                     void  displayGuest();
                     void leaveFeedback();
 };
 Guest:: Guest() {
        Guest_ID = 0;
        strcpy_s(Guest_fName,”“);
        strcpy_s(Guest_lName,”“);
        strcpy_s(Guest_email,”“);
        strcpy_s(Guest_address,”“);
        Guest_phone = 0;
        strcpy_s(GuserName,”“);
        strcpy_s(Gpassword,”“);
 }
Guest:: Guest(char GfName[], char GlName[], char Gemail[], char Gaddress[], int
Gphone)  {
        strcpy_s(Guest_fName, GfName);
        strcpy_s(Guest_lName, GlName);
        strcpy_s(Guest_email, Gemail);
        strcpy_s(Guest_address, Gaddress);
        Guest_phone = Gphone;
 }
 void Guest::setLogIn(char GUName[], char GPassword[])
{
}
 void Guest:: editGuest()
 {
 }
 void Guest:: displayGuest()
 {
                    }
                   void Guest:: leaveFeedback() { }


                    class Reservation
                 {
                    private :
                              int R_ID;
                              int Guest_ID;
                              char R_iDate[10];
                              char R_oDate[20];
                              int Room_ID;
                              int No_adults;
                              int No_children;
                              int No_room;
                              Room *Rm;
                    public :
                            Reservation ();
                            Reservation (char checkin[], char checkout[], int NoRoom, int NoAdults, int
NoChildren);
                             void modifyReservation();
                             double calculatePrice();
                             void  displayReservation();
            };
```

```
  Reservation:: Reservation ()   {

      R_ID = 0;

      Guest_ID = 0;

      strcpy_s(R_iDate, “”);

      strcpy_s(R_oDate, “”);

      Room_ID = 0;

      No_adults = 0;

      No_children = 0;

      No_room = 0;

}

Reservation:: Reservation(char checkin[], char checkout[], int NoRoom, int NoAdults, int NoChildren)
{

     strcpy_s(R_iDate, checkin);

     strcpy_s(R_oDate, checkout);

     No_room = NoRoom;

     No_adults = NoAdults;

     No_children = NoChildren;

}

void Reservation:: modifyReservation()  { }

double Reservation::calculatePrice()  { }

void  Reservation::displayReservation()  { }
```

```
class Room
  {
        private:
                int Room_ID;
                double Room_price;
                char Room_type[10];
                 Addi_Item_service *Ad;

        public:
                 Room ();
                 Room (int R_ID, double R_price, char R_type[]);
                 void editRoom ();
                 viod displayRoom ();

};
```

```
    Room:: Room()  {

        Room_ID = 0;

        Room_price = 0.0;

        strcpy_s(Room_type , ””);

 }

    Room:: Room(int R_ID, double R_price, char R_type[])   {

        Room_ID = R_ID;

        Room_price = R_price;

        strcpy_s(Room_type , R_type);

}

   void Room:: editRoom ()  {   }

   void Room::displayRoom () {  }
```

```
class Addi_Item_service

   {

          private:

                  int Aitem_Id;

                  int Aservice_Id;

                  char Itemdescription[200];

                  char Servicedescription[200];

          public:

                  Addi_Item_service();

                  Addi_Item_service(int Ai_ID , char itemdesc[], int As_ID, char servicedesc[]);

                  void displayDetails();

              };
```

```
        Addi_Item_service:: Addi_Item_service()   {

                Aitem_Id = 0;

                Aservice_Id = 0;

                strcpy(Itemdescription, “”);

                strcpy(Servicedescription, “”);

    }

       Addi_Item_service:: Addi_Item_service(int Ai_ID , char itemdesc[], int As_ID, char servicedesc[])
{

               Aitem_Id = Ai_ID;

               strcpy(Itemdescription, itemdesc);

               strcpy(Servicedescription, servicedesc);

               Aservice_Id = As_ID;

}

      void Addi_Item_service::displayDetails()  {  }
      class Payment
        {
              private:
                    int Guest_ID;
                    int R_ID;
                     int PayID;
                     Reservation *Res;
              public:
                     Payment ();
                     payment (int G_ID,int RES_ID, int PID);
                      void confirmpayment();
                     void displayPayment ();
   };
```

```
Payment:: Payment()   {

          Guest_ID = 0;

          R_ID = 0;

          int PayID = 0;

}

Payment:: Payment(int G_ID,int RES_ID, int PID)   {

           Guest_ID = G_ID;

          R_ID = RES_ID;

          int PayID = PID;

}

void Payment:: confirmpayment()  {  }

void Payment:: displayPayment ()  {  }
```

```
class feedback
        {
            private:
                    int feedbackID;
                    char GuserName[10];
                    char description[400];
                    Guest *gus1;

            public:
                    feedback();
                    feedback(int fbId, char GUName[], char desc[]);
                    void displayFeedback();
        } ;
```

```
feedback:: feedback()  {

        feedbackID = 0;

        strcpy(GuserName, “”);

        strcpy(description, “”);

}

 feedback:: feedback(int fbId, char GUName[], char desc[])  {

        feedbackID = fbId;

        strcpy(GuserName, GUName);

        strcpy(description, desc);

}

void feedback::displayFeedback()   {  }
```

```
class Inquires
        {
            private:
                    int inquire_ID;
                    char  GuserName[10];
                    char inquire_type[50];
                    char inquireDescription[400];
                    Guest *gus2;

           public:
                     Inquires();
                     Inquire(int inId, char GUName[], char inType[], char indesc);
                     void displayInquire();

        };
```

```
Inquires:: Inquires()   {

          inquire_ID = 0;

          strcpy(GuserName, “”);

          strcpy(inquire_type, “”);

          strcpy(inquireDescription, “”);

}

Inquires:: Inquires(int inId, char GUName[], char inType[], char indesc[] )   {

         inquire_ID = inId;

          strcpy(GuserName, GUName);

          strcpy(inquire_type, inType);

          strcpy(inquireDescription, indesc);

}

void Inquires:: InquiresdisplayInquire()    {   }
```

```
class Receptionist

          {

               private:

                         int Rep_ID;

                        char  Rep_Name[10];

                        char Rep_password[10];

                        Guest *gus3;

                        Reservation *Res1;

                        Inquire * inq;


              public:

                         Receptionist();

                         Receptionist(int RId, char RName[], char Rpassword[]);

                         void displayDetails();

                         void respondInquries();


         };
```

```
Receptionist:: Receptionist() {

                    Rep_ID = 0 ;

                     strcpy(Rep_Name, “”);

                     strcpy(Rep_password, “”);

}

Receptionist:: Receptionist(int RId, char RName[], char Rpassword[]) {

                     Rep_ID = RId ;

                     strcpy(Rep_Name, RName);

                     strcpy(Rep_password, Rpassword);

}

void Receptionist::displayDetails()   {   }

void Receptionist::respondInqries()  {   }
```

```
class report
        {
               private:
                              int reportId;
                               Room * Rm4;
                               Guest *gus2;
                                Reservation *Res2;
                public:
                              void displayRoomReport();
                              void displayGuestReport();
                             void displayReservationReport();
                              double calculateAnnualAmount();
        };
```

```
void report::displayRoomReport()  {  }

 void report::displayGuestReport()  {  }

 void report::displayReservationReport()  {  }

 double report::calculateAnnualAmount()  {  }
```

```
class HotelAdminister
       {
             private:
                          int admin_Id;
                           char AuserName[20];
                           char Apassword[10];
                             Room * Rm5
                             Feedback *Fb2
                             Inquire *inq1;
                            Report *Rep;
              public:
                          void addRoom();
                          void displayFeedback();
                         void modifyRoom();
                         void dispalyReport();
                         void displayInquires();
      };
```

```
void HotelAdminister::addRoom()  {  }

void HotelAdminister::displayFeedback()  {  }

void HotelAdminister::modifyRoom()   {   }

void HotelAdminister::dispalyReport()  {   }

void HotelAdminister::displayInquires()   {   }
```