BSc (Hons) in Information Technology

Object Oriented Concepts – IT1050

Assignment 2

Year 1, Semester 2 **2022-July**

Cover Page:

Topic :Event Photography Management System

Group no :MLB_02.02_02

Campus : Malabe

Submission Date :18/11/2022

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 eachone of us equally contributed to the completion of this Assignment.

Registration No	Name	Contact Number
IT21803048	Senanayaka S.M.K (leader)	076-8444343
IT21829642	Palihakkara T.U.D	070-7208000
IT21834592	Chethani A.V.S	074-1433687
IT21833366	Wijerathna G.D.K	075-6508380
IT21838248	Perera P.A.H.V	076-3381386

❖Description

Our project report is based on the topic of "Event photography management system”. Our customers can search packages and, they can make an appointment after fill the form and pay the bill amount using payment methods. After that system confirm appointment via message. In our system admin manage staff, packages, payment and feedback. Also generate reports.

PART 1

❖REQUIREMENTS

1.visitor should register to the system by using name,email,address dob and age.

2.registered member can log into system by providing username and password.

3.registered member can views packages.

4. Registered member can search packages on the website

5.Registered member can contact receptionist for more details about packages.

6.Registered member can give feedback

7.System display previous feedback.

8.Registered member choose a package and make an appointment.

9.Registered member fill the form and submit

10.system confirm appointment via message .

11.System provides various payment methods

12.Registered member make payment

13.System validates the payment details

14.Admin should register to the system by providing user name and password

15.Admin manage packages,payment, appointment,feedback and staff

16.Admin can update and create system

❖Classes

- RegisterdUser
- Payment
- Feedback
- Admin
- Appointment
- Packages

❖CRC Cards

RegisterdUser	
Responsibility	Collaboration
Register	
Add user details	
Login	
Choose a package	package
Make an appointment	appointment
Make payment	payment
Give feedback	feedback

Admin	
Responsibility	Collaboration
Login to system	
Manage packages	packages
Manage payment	payment
Manage appointment	appointment
Manage feedback	feedback
Manage staff	staff
Generate reports	reports

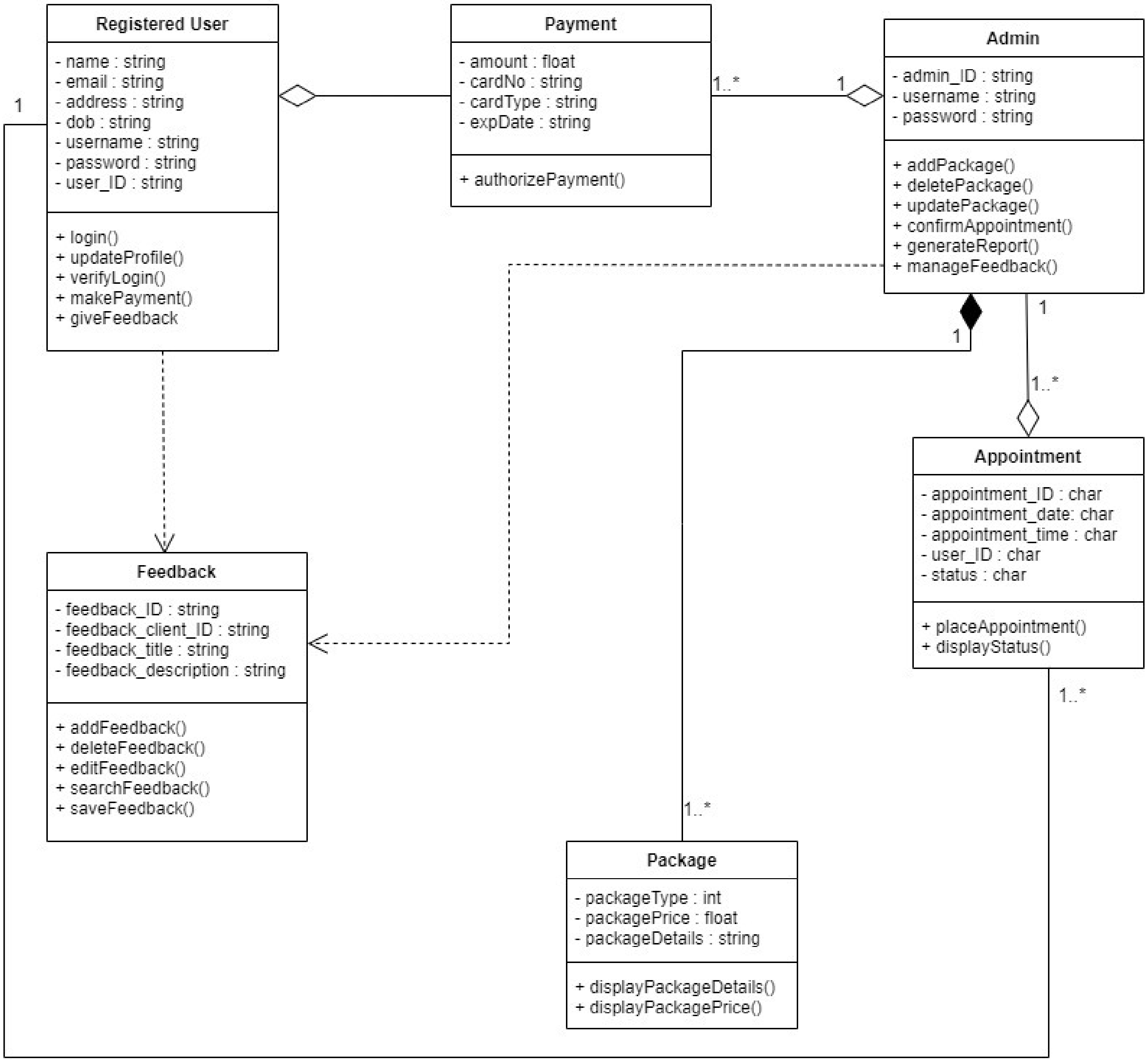
Feedback	
Responsibility	Collaboration
Give feedback	Registered user

payment	
Responsibility	Collaboration
Do payment	Registered user
Add payment method	
Checkout	appointment

Packages	
Responsibility	Collaboration
Add package	admin
Update package	admin
Store package	admin
Delete package	Admin

Appointment	
Responsibility	Collaboration
Store appointment details	
Status of appointment	
Create appointment	Registered member

❖UML Diagram


Registered User
- name : string
- email : string
- address : string
- dob : string
- username : string
- password : string
- user_ID : string
+ login()
+ updateProfile()
+ verifyLogin()
+ makePayment()
+ giveFeedback
Payment
- amount : float
- cardNo : string
- cardType : string
- expDate : string
+ authorizePayment()
Admin
- admin_ID : string
- username : string
- password : string
+ addPackage()
+ deletePackage()
+ updatePackage()
+ confirmAppointment()
+ generateReport()
+ manageFeedback()
1..*
1
1
1
1
1..*
Appointment
- appointment_ID : char
- appointment_date: char
- appointment_time : char
- user_ID : char
- status : char
+ placeAppointment()
+ displayStatus()
1..*
Feedback
- feedback_ID : string
- feedback_client_ID : string
- feedback_title : string
- feedback_description : string
+ addFeedback()
+ deleteFeedback()
+ editFeedback()
+ searchFeedback()
+ saveFeedback()
1..*
Package
- packageType : int
- packagePrice : float
- packageDetails : string
+ displayPackageDetails()
+ displayPackagePrice()

- **Admin.h – IT21803048 Senanayaka S.M.K**

```
//IT21803048 SenanayakaS.M.K

//Event photography management system

//Admin.h


class Admin

{

private:

   char admin_id [10];

   char userName [20];

   char password [15];

   Payment *pay;

   Appointment *appointment;

   Package *package;

public:

   Admin();

   Admin(const char admin_id[], const char userName[],const char password[],Payment *pay,Appointment
*appointment,Package *package);

   void addPackage();

   void deletePackage();

   void updatePackage();

   void confirmAppointment();

   void generateReport();

   void manageFeedback();

   ~Admin();

};
```

- **Admin.cpp - IT21803048 Senanayaka S.M.K**

```
//IT21803048 SenanayakaS.M.K
//Event photography management system
//Admin.cpp


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


Admin::Admin()
{
   strcpy(admin_id,"");
   strcpy(userName,"");
   strcpy(password,"");
}


Admin::Admin(const char admin_id[], const char userName[],const char password[],Payment *pay,Appointment
*appointment,Package *package)
{
   strcpy (admin_id,aId);
   strcpy (userName,aUserN);
   strcpy (password,aPw);
}
void Admin::addPackage()
{}
void Admin::deletePackage()
{}
void Admin::updatePackage()
{}
void Admin::confirmAppointment()
{}
void Admin::generateReport()
```

```
{}

void Admin::manageFeedback()

{}

Admin::~Admin()

{}
```

- **Feedback.h - IT21803048 Senanayaka S.M.K**

```
//IT21803048 SenanayakaS.M.K

//Event photography management system

//Feedback.h


#include "RegisterdUser.h"

#include "Admin.h"


class Feedback

{

private:

		char feedback_id[20];

		char feedback_client_id[20];

		char feedback_title[50];

   char feedback_description[100];

   RegisterdUser *regMemb;

   Admin *admin;


public:

		Feedback();

		Feedback(const char fId[], const char fcId[],const char fTitle[], const char fDesc[],RegisterdUser *regMemb,Admin *admin;);

		void addFeedback();

		void deleteFeedback();

   void editFeedback();

   void searchFeedback();

   void saveFeedback();

		~Feedback();

};
```

- **Feedback.cpp - IT21803048 Senanayaka S.M.K**

```
//IT21803048 SenanayakaS.M.K
//Event photography management system
//Feedback.cpp

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

Feedback::Feedback()
{
    strcpy(feedback_id,"");
    strcpy(feedback_client_id,"");
    strcpy(feedback_title,"");
    strcpy(feedback_description,"");
}

Feedback::Feedback(const char fId[], const char fcId[],const char fTitle[], const char fDesc[],RegisterdUser *regMemb,Admin *admin)
{
    strcpy(feedback_id,fId);
    strcpy(feedback_client_id,fcId);
    strcpy(feedback_title,fTitle);
    strcpy(feedback_description,fDesc);
    regMemb = rregMemb;
    admin = aadmin;
}

void Feedback::addFeedback()
{}
void Feedback::deleteFeedback()
{}
```

```
void Feedback::editFeedback()

{}

void Feedback::searchFeedback()

{}

void Feedback::saveFeedback()

{}

Feedback::~Feedback()

{}
```

- **payment.h – IT21829642 Palihakkara T.U.D**

```
//IT21829642 PalihakkaraT.U.D

//Event photography management system

//payment.h


class Payments

{

  private:


    float amount;

    int CardNo;

    char CardType[5];

    int expDate;

    RegisteredUser *reguser;

    Admin *admin;


  public:


    Payments();

    Payments(float pAmount, int cNo, const char cType[], int eDate, RegisteredUser *ru, Admin *admin);

    void autherizePayment();

    ~Payments();

};
```

- **payment.cpp - IT21829642 Palihakkara T.U.D**

```
//IT21829642 PalihakkaraT.U.D
//Event photography management system
//payment.cpp

#include <iostream>
#include <cstring>
#include "RegisterUser.h"
#include "Admin.h"
#include "Payments.h"

using namespace std;

Payments::Payments()
{

    amount = 0.0;
    CardNo = 0;
    strcpy(CardType, "");
    expDate = 0;

    //RegisteredUser *reguser;
    //Admin *admin;
}

Payments::Payments(float pAmount, int cNo, const char cType[], int eDate, RegisteredUser *ru, Admin *admin)
{
    amount = pAmount;
    CardNo = cNo;
    strcpy(CardType, cType);
    expDate = eDate;
```

```
   reguser = ru;

   adm = a;

}


void Payments::autherizePayment()

{


}


Payments::~Payments()

{

   cout << "destructor running!" << endl;   }
```

- **Appointment.h – IT21834592 – Chethani A.V.S**

```
//IT21834592 Chethani A.V.S

//Event Photography management system

//Appointment.h


#include <iostream>

#include <cstring>

using namespace std;


class Appointment

{

        protected:

        char appointment_id[20];

        char appointment_date[20];

        char appointment_time[20];

        char user_id[20];

        char status[20];

//      Admin* ad[SIZE];

//      Registereduser* regmeb2;


        public:

                Appointment();

                Appointment(const char aId[], const char aDate[], const char aTime[], const char uId[], const char sta[]);

                void placeAppoinment(const char uId[]);

          void displayStatus();

                ~Appointment();


};
```

- **Appointment.cpp - IT21834592 – Chethani A.V.S**

```
//IT21834592 Chethani A.V.S

//Event Photography management system

//Appointment.cpp


#include "Appointment.h"

#include <iostream>

#include <cstring>

using namespace std;



Appointment::Appointment()

{

        strcpy_s(appointment_id, "");

        strcpy_s(appointment_date, "");

        strcpy_s(appointment_time, "");

        strcpy_s(user_id, "");

        strcpy(_sstatus, "");


}


Appointment::Appointment(const char aId[], const char aDate[], const char aTime[], const char uId[], const char sta[])

{

        strcpy(_sappointment_id, aId);

        strcpy_s(appointment_date, aDate);

        strcpy_s(appointment_time, aTime);

        strcpy_s(user_id, uId);

        strcpy_s(status,sta );


}


void Appointment::placeAppointment(const char uId[])
```

```
{
        cout << "User is : " << uId <<endl;
}
void Appointment::displayStatus()
{
        cout << "status " << status << endl;
}

Appointment::~Appointment()
{
                cout << "Destructor running" << endl;
}
```

- **RegisterdUser.h – IT21833366 Wijerathna G.D.K**

```
//It21833366 Wijerathna G.D.K

//Event Photography Management System

//RegisterdUser.h


#include "Feedback.h"

#include "Payments.h"


class RegisterdUser

{

        private:

                char name[20];

                char email[20];

                char address[20];

                char dob[20];

                char username[20];

                char password[20];

                int user_id;


                Feedback* fd[0];

                Payments* pay[1];


        public:

                RegisterdUser();

                RegisterdUser(const char pname[],const char pemail[],const char paddress[],const char pdob[],const char pusername[],const char ppassword[]);


                void setName(const char pname[]);

                char getName();


                void setEmail(const char pemail[]);

                char getEmail();
```

```
            void setAddress(const char paddress[]);
            char getAddress();

            void setDob(const char pdob[]);
            char getDob();

            void setUsername(const char pusername[]);
            char getUsername();

            void setPassword(const char ppassword[]);
            char getPassword();

            int login(int puId);
            void updateProfile(const char pname[],const char pemail[],const char paddress[],const char pdob[],const
char pusername[],const char ppassword[]);
            void verifyLogin();

            float makePayements(Payments* pay[]);
            void givefeedbacks(Feedback* fd[]);

            ~RegisterdUser();

};
```

- **RegisterdUser.cpp - IT21833366 Wijerathna G.D.K**

```
//It21833366 Wijerathna G.D.K
//Event Photography Management System
//RegisterdUser.cpp

#include "RegisterdUser.h"
#include "Feedback.h"
#include "Payments.h"

#include <iostream>
#include <cstring>
using namespace std;

RegisterdUser::RegisterdUser(){
    strcpy(name[],"");
                            strcpy(email[],"");
                            strcpy(address[],"");
                            strcpy(dob[],"");
                            strcpy(username[],"");
                            strcpy(password[],"");
                            user_id=0;

} //default constructor

RegisterdUser::RegisterdUser(pname[],pemail[],paddress[],pdob[],pusername[],ppassword[],puId){
          strcpy(name[],pname);
          strcpy(email[],pemail);
          strcpy(address[],paddress);
          strcpy(dob[],pdob);
          strcpy(username[],pusername);
          strcpy(password[],ppassword);
```

```
        user_id=puId;

} // overload constructor

//getters
char RegisterdUser::getName(){}
char RegisterdUser::getEmail(){}
char RegisterdUser::getAddress(){}
char RegisterdUser::getDob(){}
char RegisterdUser:: getUsername(){     }
char RegisterdUser::getPassword(){}

//setters
void RegisterdUser::setName(const char pname[]){}
void RegisterdUser::setEmail(const char pemail[]){}
void RegisterdUser::setAddress(const char paddress[]){}
void RegisterdUser::setDob(const char pdob[]){}
void RegisterdUser:: setUsername(const char pusername[]){}
void RegisterdUser::setPassword(const char ppassword[]){}

int RegisterdUser::login(int puId){} //login function which use to login the system

void RegisterdUser::updateProfile(const char pname[],const char pemail[],const char paddress[],const char pdob[],const char
pusername[],const char ppassword[]){} // use to update the current details in profile page

void RegisterdUser::verifyLogin(){} // verify the login details

float RegisterdUser::makePayements(Payments* pay[]){} // use to make a payments

void RegisterdUser::givefeedbacks(Feedback* fd[]){} // use to give a feedback

RegisterdUser::~RegisterdUser(){} // destructor
```

- **Package.h - IT21838248 Perera P.A.H.V**

```
#include <iostream>

#include <cstring>

using namespace std;


//IT21838248 Perera P.A.H.V

//Event phototgraphy management system

//Package.h


class Package

{

private:

  float packagePrice;

  int packageType;

  string packageDetails;


public:

  Package();

  Package(float p_price, int p_type);

  void setPackagePrice(float p_price);

  void setPackageType(int p_type);

  int getPackageType();

  float getPackagePrice();

  void displayPackageDetails();

  void displayPackagePrice();

  ~Package();

};
```

- **Package.cpp – IT21838248 Perera P.A.H.V**

```
//IT21838248 Perera P.A.H.V

//Event phototgraphy management system

//Package.cpp


#include "Package.h"

#include <iostream>

#include <cstring>

using namespace std;


//Default constructor

Package::Package()

{

  packagePrice = 0.0;

  packageType = 0;

}


//Overloaded constructor

Package::Package(float p_price, int p_type)

{

  packagePrice = p_price;

  packageType = p_type;

}


//setters


void Package::setPackagePrice(float p_price)

{

  packagePrice = p_price;

}
```

```
void Package::setPackageType(int p_type)
{
 packageType = p_type;
}

//getters
int Package:: getPackageType()
{
 return packageType;
}

float Package:: getPackagePrice()
{
 if (packageType==1)
  packagePrice=10000.00;
 else
 {if (packageType==2)
    packagePrice=15000.00;
   else
    packagePrice=25000.00;
 }

 return packagePrice;
}

void Package::displayPackageDetails()
{
 cout<<"Our Packages : "<<endl<<endl;

 cout<<"Package 1 : This package is specially for events like Parties"<<endl;
 cout<<"Package Price : Rs. 10,000.00"<<endl<<endl;
```

```
  cout<<"Package 2: This package is specially for concerts"<<endl;
  cout<<"Package Price : Rs. 15,000.00"<<endl<<endl;


  cout<<"Package 3: This package is specially for weddings"<<endl;
  cout<<"Package Price : Rs. 25,000.00"<<endl<<endl;


}
void Package::displayPackagePrice()
{
  cout<<endl;
  cout<<"You selected Package Type : " << packageType << endl;
  cout<<"Package Price : " << packagePrice <<endl;
  cout<<"***********************************"<<endl;
}

//Destructor
Package::~Package()
{
  cout<<"The selected package is removed"<<endl;
}
```

- **Main.cpp**

```
#include <iostream>
#include<cstring>
#include "RegisterdUser.h"
#include "Admin.h"
#include "Appointment.h"
#include "Package.h"
#include "Payments.h"
#include "Feedback.h"

#include <cstring>
#include <iostream>
using namespace std;

int main() {
        //create package object
        Package* pac[2];
        pac[0] = new Package();
        pac[1] = new Package(2300.00, 2);

        //acces method in package class
        pac[0]->getPackageType();
        pac[0]->getPackagePrice();
        pac[0]->displayPackagePrice();
        pac[0]->displayPackageDetails();

        //create appointment object
        Appointment* appointment[2];
        appointment[0] = new Appointment();
```

```
appointment[1]= new Appointment("ap001", "2022-09-17", "22:18:09", "001", "pending");

//acces method in appointment class
appointment[0]->placeAppointment(“1”);
appointment[1]->displayStatus();

//create admin object
Admin* admin[2];
admin[0] = new Admin();
admin[1]=new Admin("002","kalum","k@123",payments[0],appointment[0],package[0]);

//acces method in admin class
admin[0]->addPackage();
admin[0]->updatePackage();
admin[0]->confirmAppointment();
admin[0]->deletePackage();
admin[0]->generateReport();
admin[0]->manageFeedback();

//create payments object
Payments* payments[2];
payments[0] = new Payments();
payments[1] = new Payments(2500.00,3,"credit", 20220613,reguser[0],admin[0]);

//acces method in payemnt class
payments[0]->autherizePayment();

//create feedback object
Feedback* feedback[2];
feedback[0] = new Feedback();
feedback[1] = new Feedback("001", "003","good", "best customer service",reguser[0],admin[0]);

//acces method in feedback class
```

```
        feedback[0]->addFeedback();
        feedback[0]->editFeedback();
        feedback[1]->deleteFeedback();
        feedback[1]->searchFeedback();
        feedback[0]->saveFeedback();

        //create registerd user object
        RegisterdUser* reguser[2];
        reguser[0] = new RegisterdUser();
        reguser[1] = new
    RegisterdUser("smanali","samanali@gmail.com","no97/5,galle","1999/10/12","samanali@99","Sam@991012");

        //acces method in Registerd user class
        reguser[0]->getUsername();
        reguser[0]->getPassword();
        reguser[0]->login(12);
        reguser[0]->verifyLogin();
        reguser[0]->updateProfile("sama","sama@gmail.com","no97/5,galle","1999/10/12","sama@99","Sama@991012");
        reguser[0]->makePayements(payments[0]);
        reguser[0]->givefeedbacks(feedback[0]);



        //clear dynamic memory space after complete the process
        delete pac[0];
        delete pac[1];
        delete reguser[0]
        delete reguser[1];
        delete appointment[0];
        delete appointment[1];
        delete admin[0];
        delete admin[1];
        delete payments[0];
        delete      payments[1];
```

```
                delete feedback[0];

                delete feedback[1];



return 0;


}
```