Topic : Event photography management system

Group no : MLB_03.01_09

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.

Group Details

	Student ID	Student Name	Email	Contact Number
1	IT21229220	Indrajith G.B.T.G	It21229220@my.sliit.lk	0769826692
2	IT21230660	Nugapitiya E.M.M.R.D.L	It21230660@my.sliit.lk	0769759661
3	IT21235238	Jinadasa H.A.K	It21235238@my.sliit.lk	0767577262
4	IT21231964	Wickramanayaka L.P.S.D	It21231964@my.sliit.lk	0775761760
5	IT21232886	Nimsara M.K.C	It21232886@my.sliit.lk	0770198427

1. Requirements

- System should be available in 24/7.
- Guest customer should register into the system by providing details.
- Registered customer can login to the system using their username and password.
- Registered customer can view all pages in the system, check schedule details and booking details.
- Customer can select the event they want, book dates, and make payments.
- Customers can choose a payment method (cash, credit card) for their bookings.
- After confirmed the payment registered customer can book dates and locations.
- Registered customer can update their booking dates and make requests.
- Admin can check booking details, customer requests and update the system.
- Booking details should confirmed by the administrator.
- Photographer should login to the system as photographer.
- Photographer can check booking details, customer requests, and send photos to the photo editor.
- Photo editor can login as the photo editor using his login credentials and then he can check new photos, edit them, and send them to the web moderator.
- Web moderator should check customer requests, give responses, and upload new photos to the system.

Classes

Registered Customer

Schedule Details

Booking Details

Event

Payment

Photo

Photographer

Feedbacks and reviews

Reasons for rejecting other nouns

1. Redundant – Customer, payments
2. An event or an operation – booking dates
3. Outside scope of system –
4. Meta-language – They, Them, his
5. An attribute – Username, Password, Locations

Methods

- Registered Customer – Login to the system

  View the system

  Select event

  Update booking

- Schedule Details – view schedule details

  Update schedule details

- Booking Details – view booking details

  Update booking details

  Confirm booking details

- Event – select event

  View event

  Update event

- Payment – add payment details

  Confirm payment

- Photo – edit photos

  Upload photos

- Photographer – view booking details

  View customer requests

  Add photos

- Feedbacks and reviews – view feedbacks and reviews

CRC Cards

Registered customer	
Responsibilities	Collaborators
Login to the page providing details	
Select necessary event	Events
Book a date with details	Booking details
Payment	Payment

Schedule details	
Responsibilities	Collaborators
Provide available dates	
Update available details	Booking details

<table>
<tr><th colspan="2">Booking details</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>Provide booking details</td><td></td></tr>
<tr><td>Update booking details for the schedules</td><td>Schedule details</td></tr>
</table>

<table>
<tr><th colspan="2">Events</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>View event</td><td></td></tr>
<tr><td>Provide event details</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Payment</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>Show how payment should be done</td><td></td></tr>
<tr><td>Confirm payment</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Photo</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>Edit photos</td><td></td></tr>
<tr><td>Upload photos</td><td></td></tr>
</table>

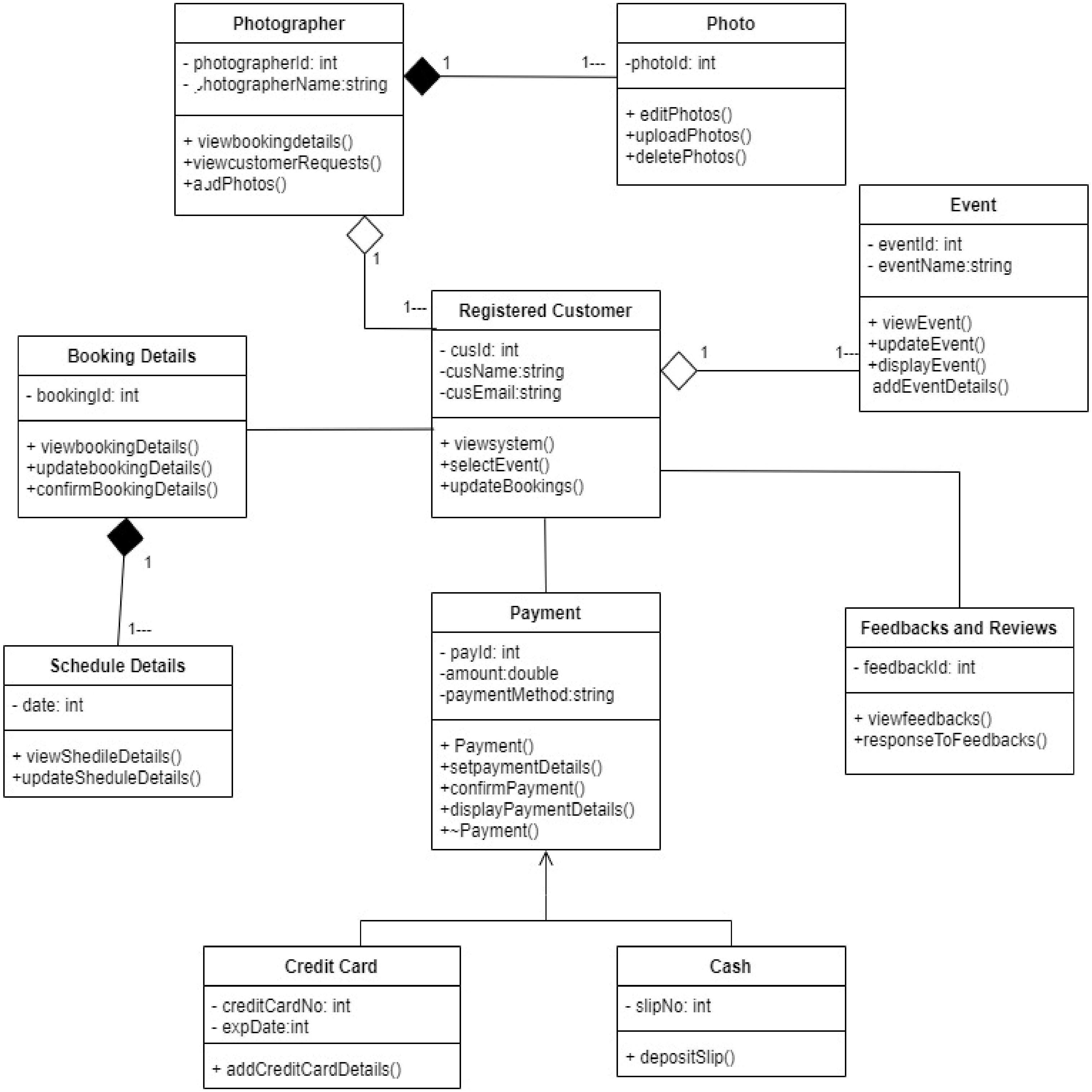
<table>
<tr><th colspan="2">Photographer</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>Login as photographer</td><td></td></tr>
<tr><td>View customer requests</td><td>Registered customer</td></tr>
<tr><td>Add photos</td><td></td></tr>
</table>

<table>
<tr><th colspan="2">Feedbacks and reviews</th></tr>
<tr><th>Responsibilities</th><th>Collaborators</th></tr>
<tr><td>View feedbacks and reviews</td><td></td></tr>
</table>

Cash	
Responsibilities	Collaborators
Deposit slip	Payment

Credit card	
Responsibilities	Collaborators
Adding credit card details	Payment

Class Diagram



```
//Booking_details.h
class Booking_details {
private :
         int bookingId;
         Booking_details*bookingDet[20];
public :
        Booking_details();
        Booking_details(int pBookingId);
        void viewBookingDetails();
        void updateBookingDetails();
        void confirmBookingDetails();
        ~Booking_details();
};

//Event.h
class Event{
private:
          int eventId;
          string eventName;
          Registered_customer*regiCustomer[20];
public :
          Event();
          Event(int pEventId, string pEventName);
          void viewEvents();
          void addEventDetails();
          void updateEvent();
          void displayEvent();
          ~Event();
};



//Registered_customer.h
```

```
class Registered_customer{
private:
        int cusId;
        string cusName;
        string cusEmail;
        Event*Events;
        Booking details*bookingDet[size];
        Photographer*Photographers[size];
        Payment*Payments[size];
        Feedbacks_and_reviews*FeedAndReview[size];
public:
       Registered_customer();
      Registered_customer(int pRegisteredId ,string pRegisteredName, string
pRegisteredEmail);
      void viewSystem();
      void selectEvent();
      void updateBooking();
      ~Registered_customer();
};

//Schedule_details.h
class Schedule_details {
private:
         int date;
         Booking_details*bookingDet[20];
public:
        Schedule_details();
       Schedule_details(int pDate);
       void viewScheduleDetails();
       void updateScheduleDetails();
       ~Schedule_details();
};


//Payment.h
```

```
class Payment {
        private:
                int payId;
                double amount;
                string paymentMethod;

        public:
                Payment();
                void setPaymentDetails(int pId, double pAmount);
                void confirmPayment();
                void displayPaymentDetails();
                ~Payment();
};

//CreditCard.h
class CreditCard{
        private:
                int creditCardNo;
                int expDate;

        public:
                CreditCard(int pcCNo,int pcExpDate);
                void addCreditCardDetails();
                ~CreditCard();
};

//Cash.h
class Cash {
        private:
                int slipNo;
        public:
                Cash(int sNo);
                void depositSlip();
                ~Cash();};
//Photo.h
```

```
class Photo{
		private:
			int photoId;
		public:
			Photo(int pId);
			void editPhotos();
			void uploadPhotos();
			void deletePhotos();
			~Photo();
	};

//Photographer.h
class Photographer {
		private:
			int photographerId;
			string photographerName;
		public:
			Photographer(int pgId,string pgName);
			void viewBookingDetails();
			void viewCustomerRequests();
			void addPhotos();
			~Photographer();
	};

//FeedbacksAndReviews.h
class FeedbacksAndReviews{
		private:
			int feedbackId;
		public:
			FeedbacksAndReviews(int fId);
			void viewFeedbacks();
			void responseToFeedbacks();
			~FeedbacksAndReviews();
	};
```

Class cpp file

```
//Implementation methods in each class

Booking_details::Booking_details(){}
Booking_details::Booking_details(int pBookingId)
{
  bookingId=pBookingId;
}
void Booking_details::viewBookingDetails(){}
void Booking_details::updateBookingDetails(){}
void Booking_details::confirmBookingDetails(){}

//Event.cpp

Event::Event(){}
Event::Event(int pEventId, string pEventName)
{
      eventId= pEventId;
      eventName= pEventName;
}

void Event::viewEvents(){}
void Event::addEventDetails(){}
void Event::updateEvent(){}
void Event::displayEvent(){}

//Registered_customer.cpp

Registered_customer::Registered_customer(){}
Registered_customer::Registered_customer(int pRegisteredId ,string
pRegisteredName, string  pregisteredEmail)
{
       cusId= pRegisteredId;
       cusName= pRegisteredName;
       cusEmail= pregisteredEmail;
}
void Registered_customer::viewSystem(){}
void Registered_customer::selectEvent(){}
void Registered_customer ::updateBooking(){}

//Schedule_details.cpp

Schedule_details::Schedule_details(){}
Schedule_details::Schedule_details (int pDate)
{
   date= pDate;
}
void Schedule_details:: viewScheduleDetails(){}
void Schedule_details::updateScheduleDetails(){}

//Payment.cpp

void Payment::setPaymentDetails(int pId, double pAmount)
{
        payId=pId;
        amount=pAmount;
}

void Payment::confirmPayment(){}

void Payment::displayPaymentDetails()
{
        cout<<"payId="<<payId<<endl;
        cout<<"amount="<<amount<<endl;
}
Payment::~Payment(){}



//CreditCard.cpp
```

```
CreditCard::CreditCard(int pcCNo, int pcExpDate)
{
        creditCardNo = pcCNo;
        expDate = pcExpDate;
}
void CreditCard::addCreditCardDetails(){}
CreditCard::~CreditCard(){}


//Cash.cpp

Cash::Cash(int sNo)
        {
                slipNo=sNo;
        }
void Cash::depositSlip(){}

Cash::~Cash(){}

//Photo.cpp
Photo::Photo(int pID)
{
        photoId=pID;
}
void Photo::editPhotos(){}
void Photo::uploadPhotos(){}
void Photo::deletePhotos(){}
Photo::~Photo(){}
//Photographer.cpp
Photographer::Photographer(int pgId,string pgName)
{
        photographerId=pgId;
        photographerName=pgName;
}
```

```
void Photographer::viewBookingDetails(){}
void Photographer::viewCustomerRequests(){}
void Photographer::addPhotos(){}
Photographer::~Photographer(){}

//FeedbacksAndReviews.cpp
FeedbacksAndReviews::FeedbacksAndReviews(int fID)
{
	feedbackId=fID;
}
void FeedbacksAndReviews::viewFeedbacks(){}
void FeedbacksAndReviews::responseToFeedbacks(){}
FeedbacksAndReviews::~FeedbacksAndReviews(){}
```

Main Program

```
//main program
int main()
{
	Registered_customer regCus1;
	Schedule_details scheDet1;
	Booking_details bkDet1;
	Event eve1;
	Payment pay1;
	Photo pht1;
	Photographer phtgr1;
	FeedbacksAndReviews fbARe1;

	return 0;
};
```