Topic : Event Photography Management System

Group no : MLB_WD_07.02_08

Campus : Malabe

Submission Date: 17/05/2022

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 each one of us equally contributed to the completion of this Assignment.

Registration No	**Name**	**Contact Number**
IT21386022	U.W.H.N. Senevirathne	0714326276
IT21376054	A.M.A.D. Herath	0705865618
IT21385414	E.D.S. Gallage	0702316141
IT21226496	W.M.M. Gunasekara	0740459868
IT21584718	P.H.R. Peiris	0764350367

Contents

System Requirements

- The system should function 24/7/365.
- System users are two types called guest users and registered users.
- Guest user can view album, about us page and feedback page.
- When he/she views our about us page in our website, he/she can know about our team, opportunities and partners.
- In the team page, there are registered photographers with their profiles.
- If the guest user wants to hire the selected photographer, they must register with the system by providing details such as first name, last name and the email.
- In addition, they should create a strong password according to given instructions.
- In here, the system should confirm the details, and the username and password can be obtained via email.
- Then, registered user login to the system by entering correct username and password.
- System checks the validity of the login details and gives access to the system.
- Then, the registered user fills the quote form to place the order to hire the selected photographer.
- After submitting the quote form, the system redirects to the selecting packages page.
- If the registered user wants to order an album, it is included in the packages.
- When registered user select the package system shows payment details form.
- Registered user specifies a payment method (visa, Master card, American Express, Discover) for each order.
- Once the registered user confirms the order, he/she can see the pending service request status by exploring the user account.
- In here, system checks the validity of payment details and verify the registered user's service.
- In this system, also system admin should login to the system with correct username and password.
- In here, system admin can upload photos and delete photos.

Noun & Verb Analysis

(NOUNS)

- The system should function 24/7/365.
- System users are two types called guest users and registered users.
- Guest user can view album, about us page and feedback page.
- When he/she views our about us page in our website, he/she can know about our team, opportunities and partners.
- In the team page, there are registered photographers with their profiles.
- If the guest user wants to hire the selected photographer, they must register with the system by providing details such as first name, last name and the email.
- In addition, they should create a strong password according to given instructions.
- In here, the system should confirm the details, and the username and password can be obtained via email.
- Then, registered user login to the system by entering correct username and password.
- System checks the validity of the login details and gives access to the system.
- Then, the registered user fills the quote form to place the order to hire the selected photographer.
- After submitting the quote form, the system redirects to the selecting packages page.
- If the registered user wants to order an album, it is included in the packages.
- When registered user select the package and system shows payment details form.
- Registered user specifies a payment method (visa, Master card, American Express, Discover) for each order.
- Once the registered user confirms the order, he/she can see the pending service request status by exploring the user account.
- In here, system checks the validity of payment details and verify the registered user's service.
- In this system, also system admin should login to the system with correct username and password.
- In here, system admin can upload photos and delete photos.

Identified Classes

- Guest user
- Registered user
- Album
- Photographer
- Package
- Order
- Payment

Reasons for rejecting other nouns

- **Redundant:** Registered user, Album, Order, Registered photographer, packages page, payment details form, payment details
- **An Event or an operation:** pending service request status
- **Outside scope of system:** system, instructions, quote form, user account, system admin, registered user's service, photos
- **Meta-language:** he/she, they, their
- **An attribute:** details(first name, last name, email, username, password), payment method (visa, Master card, American Express, Discover),

Noun & Verb Analysis

(VERBS)

- The system should function 24/7/365.
- System users are two types called guest users and registered users.
- Guest user can view album, about us page and feedback page.
- When he/she views our about us page in our website, he/she can know about our team, opportunities and partners.
- In the team page, there are registered photographers with their profiles.
- If the guest user wants to hire the selected photographer, they must register with the system by providing details such as first name, last name and the email.
- In addition, they should create a strong password according to given instructions.
- In here, the system should confirm the details, and the username and password can be obtained via email.
- Then, registered user login to the system by entering correct username and password.
- System checks the validity of the login details and gives access to the system.
- Then, the registered user fills the quote form to place the order to hire the selected photographer.
- After submitting the quote form, the system redirects to the selecting packages page.
- If the registered user wants to order an album, it is included in the packages.
- When registered user select the package system shows payment details form.
- Registered user specifies a payment method (visa, Master card, American Express, Discover) for each order.
- Once the registered user confirms the order, he/she can see the pending service request status by exploring the user account.
- In here, system checks the validity of payment details and verify the registered user's service.
- In this system, also system admin should login to the system with correct username and password.
- In here, system admin can upload photos and delete photos.

Methods

• **Guest User**	- Views album, about us page and feedback page
	- Registers to the system
• **Registered User**	- Hires the photographer
	- Selects packages
	- Does payments
• **Packages**	- Allows to select packages
	- Store details of packages
• **Order**	- Place orders
• **Photographer**	- Photograph as required
	- Follow details
	- Upload content
• **Album**	- Views the album
	- Store photos
• **Payment**	- Check payment details
	- Allow to do payment
	- Confirm payment details

CRC Cards

Guest user	
Responsibility	**Collaboration**
Views album, about us page and feedback page	
Registers to the system	

Registered user	
Responsibility	**Collaboration**
Hires the photographer	photographer
Selects packages	packages
Does payments	payment

Packages	
Responsibility	**Collaboration**
Allows to select packages	Registered user
Store details of packages	

Order	
Responsibility	**Collaboration**
Place orders	Photographer, Registered user

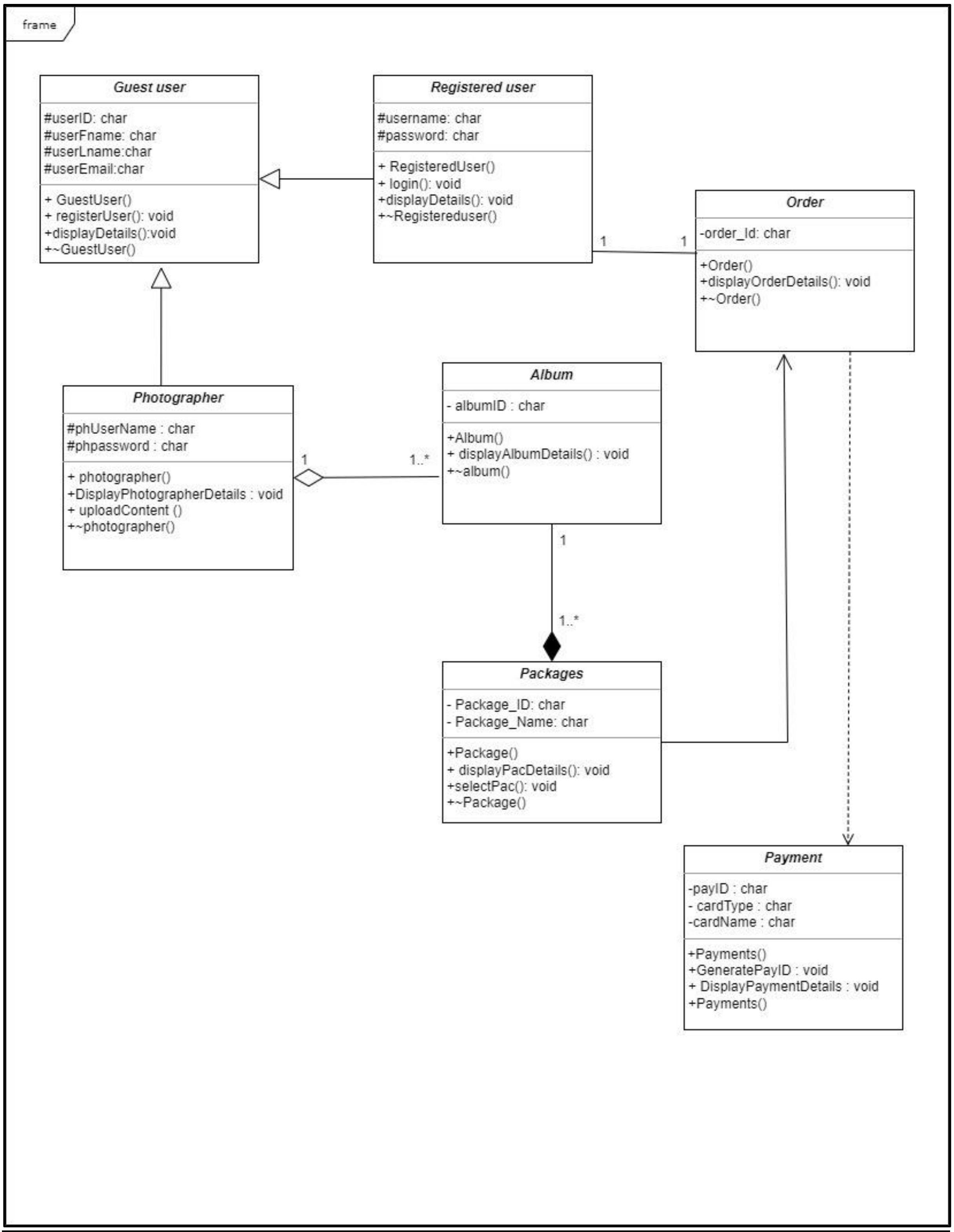
Photographer	
Responsibility	**Collaboration**
Photograph as required	Registered user
Follow details	Packages
Upload content	

Album	
Responsibility	**Collaboration**
Views the album	Guest user
Store photos	

Payment	
Responsibility	**Collaboration**
Check payment details	Registered user
Allow to do payment	Registered user
Confirm payment details	

Report	
Responsibility	**Collaboration**
List of orders	order
List of payments	payment

Class Diagram (UML Notation)



Class header files

GuestUser.h

```
#pragma once
class GuestUser
{
protected:
	int userID;
	char userFname[20];
	char userLname[20];
	char userEmail[30];

public:
	GuestUser();
	GuestUser(int puserid, const char puserfname[], const char puserlname[], 
const char puseremail[]);
	void registerUser();
	virtual void displayDetails();
	~GuestUser();
};
```

RegisteredUser.h

```
#include"GuestUser.h"
#pragma once
class registeredUser : public GuestUser
{
protected:
	char username[10];
	char password[10];
public:
	registeredUser();
	registeredUser(const char pusername[], const char ppassword[], int 
puserid, const char puserfname[], const char puserlname[], const char 
puseremail[]);
		void displayDetails();
		void login();
		~registeredUser();
};
```

Packages.h

```
#pragma once
class Packages
{
private:
	char Package_ID[20];
	char Package_Name[20];

public:
	Packages();
	void displayPacDetails(const char pPackage_ID[], const char 
pPackage_Name[]);
	void selectPackages(const char pPackage_ID[], const char pPackage_Name[]);
	~Packages();

};
```

Order.h

```
#pragma once
class Order
{
private:
	char order_Id[20];
public:
	Order();
	void displayOrderDetails(const char porder_id);
	~Order();

};
```

Photographer.h

```
#pragma once
#include"GuestUser.h"
class Photographer
{
private:
	char phusername[20];
	char phpassword[20];

public:
	Photographer();
	Photographer(const char pphusername[], const char pphpassword[]);
	void displayPhotographerDetails(const char pphusername[], const char
pphpassword[]);
	void uploadContent();
	~Photographer();

};
```

Album.h

```
#pragma once
class Album
{
private:
	char album_Id[10];

public:
	Album();
	void displayAlbumDetails(const char palbum_Id[]);
	~Album();
};
```

Payment.h

```
#pragma once
class Payment
{
private:
	char payID[20];
	char cardType[40];
	char cardName[50];

public:
	Payment();
	void generatePayId(const char ppayId);
	void displayPaydetails(const char ppayId[], const char pcardType[], const
char pcardName[]);
	~Payment();
};
```

Cpp Files

GuestUser.cpp

```
#include "GuestUser.h"
#include <cstring>
GuestUser::GuestUser()
{
	userID = 0;
	strcpy(userFname, "");
	strcpy(userLname, "");
	strcpy(userEmail, "");
}
GuestUser::GuestUser(int puserid, const char puserfname[], const char
puserlname[], const char puseremail[]){
	userID = puserid;
	strcpy(userFname, puserfname);
	strcpy(userLname, puserlname);
	strcpy(userEmail, puseremail);
}

void GuestUser::registerUser()
{
}
void GuestUser::displayDetails()
{
}
GuestUser::~GuestUser()
{
	//Destructor
}
```

RegisteredUser.cpp

```
#include "registeredUser.h"
#include <cstring>
registeredUser::registeredUser()
{
	strcpy(username, "");
	strcpy(password, "");
}
registeredUser::registeredUser(const char pusername[], const char ppassword[],
int puserid, const char puserfname[], const char puserlname[], const char
puseremail[]) : GuestUser(puserid, puserfname, puserlname, puseremail)
{
	strcpy(username, pusername);
	strcpy(password, ppassword);
}
void registeredUser::displayDetails()
{
}
void registeredUser::login()
{
}

registeredUser::~registeredUser()
{
	//Destructor
}
```

Packages.cpp

```
#include "Packages.h"
#include <cstring>
Packages::Packages()
{
	strcpy(Package_ID, "");
	strcpy(Package_Name, "");
}

void Packages::displayPacDetails(const char pPackage_ID[], const char
pPackage_Name[])
{
	strcpy(Package_ID, "pPackage_ID");
	strcpy(Package_Name, "pPackage_Name");
}

void Packages::selectPackages(const char pPackage_ID[], const char
pPackage_Name[])
{
	strcpy(Package_ID, "pPackage_ID");
	strcpy(Package_Name, "pPackage_Name");
}

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

Order.cpp

```
#include "Order.h"
#include<cstring>

Order::Order()
{
	strcpy(order_Id, "");
}

void Order::displayOrderDetails(const char porder_id)
{
	strcpy(order_Id, "porder_id");
}

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

Photographer.cpp

```cpp
#include "Photographer.h"
#include <cstring>

Photographer::Photographer()
{
	strcpy(phusername, "");
	strcpy(phpassword, "");
}

Photographer::Photographer(const char pphusername[], const char pphpassword[])
{
	strcpy(phusername, pphusername);
	strcpy(phpassword, pphpassword);
}

void Photographer::displayPhotographerDetails(const char pphusername[], const
char pphpassword[])
{
	strcpy(phusername, pphusername);
	strcpy(phpassword, pphpassword);
}

void Photographer::uploadContent()
{

}

Photographer::~Photographer()
{
	//Destructor
}
```

Album.cpp

```cpp
#include "Album.h"
#include<cstring>

Album::Album()
{
	strcpy(album_Id, "");
}

void Album::displayAlbumDetails(const char palbum_Id[])
{
	strcpy(album_Id, "palbum_Id");
}

Album::~Album()
```

```
{
	//Destructor
}
```

Payment.cpp

```
#include "Payment.h"
#include<cstring>

Payment::Payment()
{
	strcpy(payID, "");
	strcpy(cardType, "");
	strcpy(cardName, "");
}

void Payment::generatePayId(const char ppayId)
{
	strcpy(payID, "ppayId");
}

void Payment::displayPaydetails(const char ppayId[], const char pcardType[],
const char pcardName[])
{
	strcpy(payID, "");
	strcpy(cardType, "");
	strcpy(cardName, "");
}

Payment::~Payment()
{
	//Destructor
}
```