Topic : Online Advertising agency

Group no : MLB_02.01_07

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

Registration No	**Name**	**Contact Number**
IT21816390	De Silva D L S S	077-3747688
IT21810664	Perera K C G S	070-1291964
IT21812262	Rasanjana J A D	075-6321827
IT21817694	Edirisinghe E A M N	076-1078501
IT21806940	Wanniarachchi P W A S V	076-3942788

Exercise 1

System Requirements

1) To obtain any service, first the customer needs to register to the system.
2) Customer should provide details such as first name, last name, NIC, date of birth, in order to register to the system.
3) Then a customer should create a password and then should verify the password by reentering the password.
4) Then the customer will get an email to verify the identity again.
5) After the password is confirmed and click on register button, the customer will get a message saying, “Hello ‘first name’, thank you for registering”.
6) After a customer is registered to the system, the one should re-direct to the user dashboard.
7) Every registered customer should log into the system, in order to obtain any service.
8) Registered customer can visit any page of the website.
9) Registered customer can contact project manager through the contact us page.
10) To request contact, the one should provide details such as name, telephone number and email and submit the form.
11) Once the customer sends a contact request, responsible team member will get into the customer through an email or phone call.
12) Registered customer can request for quotation for each service at once. As similarly, specific customer can request for different service quotations in different times.
13) Once the quotation is requested, the request will be displayed on the admin panel.
14) Then the project manager refers the quotation request and create the quotation.
15) The project manager uploads the quotation to the system and then it will be displayed in the customer’s user dashboard.
16) The customer can click on the quotation and, view it.
17) If the customer is satisfied with the service package, the one can click on the reserve service now.
18) Then the customer will be redirected to the checkout page with the reference of the quotation.
19) The system will automatically calculate the payment amount.
20) There the customer can select payment method as debit/credit card.
21) After selecting the payment method, the one should provide payment details such as the card name, bank name, card number, country and cvv to process the payment.
22) Once the customer clicks on confirm payment button, the payment will process, and both the user and the financial manager notified (record payment) that the payment has been done.

23) Meanwhile the customer receive notification regarding the status of the ongoing project.
24) Once the project is successfully completed, the project manager will notify the customer, that the project has been completed and the completed project is delivered to the client through the system.
25) Then the project manager updates the project table with reference to the customer as “completed”.
26) In addition, customer can request for services such as consultations for digital marketing, join technology-based forums.
27) As well as the customer can change or reset the password. In order to do this, the one should provide the existing password and username.

28) After the payment received, the project manager will start the project and continue.

Identified Classes

Customer

Payment

Quotation

Contact

Team member

Service

Card

Project

Reasons for rejecting other nouns

1) Registered customer - redundant
2) Project manager - redundant
3) Financial manager - redundant
4) System - outside of scope
5) Quotation request - event
6) Contact request - event
7) Project status - attribute
8) First name - attribute
9) Last name - attribute
10) NIC - attribute
11) Email - attribute
12) Telephone number - attribute
13) Password - attribute
14) Date of birth - attribute
15) Username - attribute
16) User dashboard - outside of the scope
17) Message - outside of the scope
18) Website - outside of the scope
19) Contact us page - outside of the scope
20) Checkout page - outside of the scope
21) Contact request - event
22) Phone call - event
23) Payment amount - attribute
24) Payment method - attribute
25) Card name - attribute
26) Bank name - attribute
27) CVV - attribute
28) Country - attribute
29) Card number - attribute
30) Project status - attribute
31) Send notification - event
32) Request consultations - event
33) Join forums – event

Methods

Customer - Register to the system

Create password

Verify password

Log into the system

Request consultation

Join forums

Change password

Contact - Request to contact

Contact customer

Quotation- Request quotation

Display quotation request

Upload quotation

Display quotation

Service- Reserve service

Payment- Select payment method

Calculate payment

Confirm payment

Record payments

Card- Add card details

Project- Notify project status

Update project status

Upload projects

Team member- Login to the system

CRC CARDS

Customer	
Responsibility	**Collaboration**
Register to the system	
Create password	
Verify password	
Log into the system	
Request consultation	Team member
Join forums	Project, Team member
Change password	

Contact	
Responsibility	**Collaboration**
Request to contact	Customer
Contact customer	Team member, Customer

Quotation	
Responsibility	**Collaboration**
Request quotation	Customer
Display quotation request	Team member
Upload quotation	Team member, Project
Display quotation	Customer

Service	
Responsibility	**Collaboration**
Reserve service	Customer

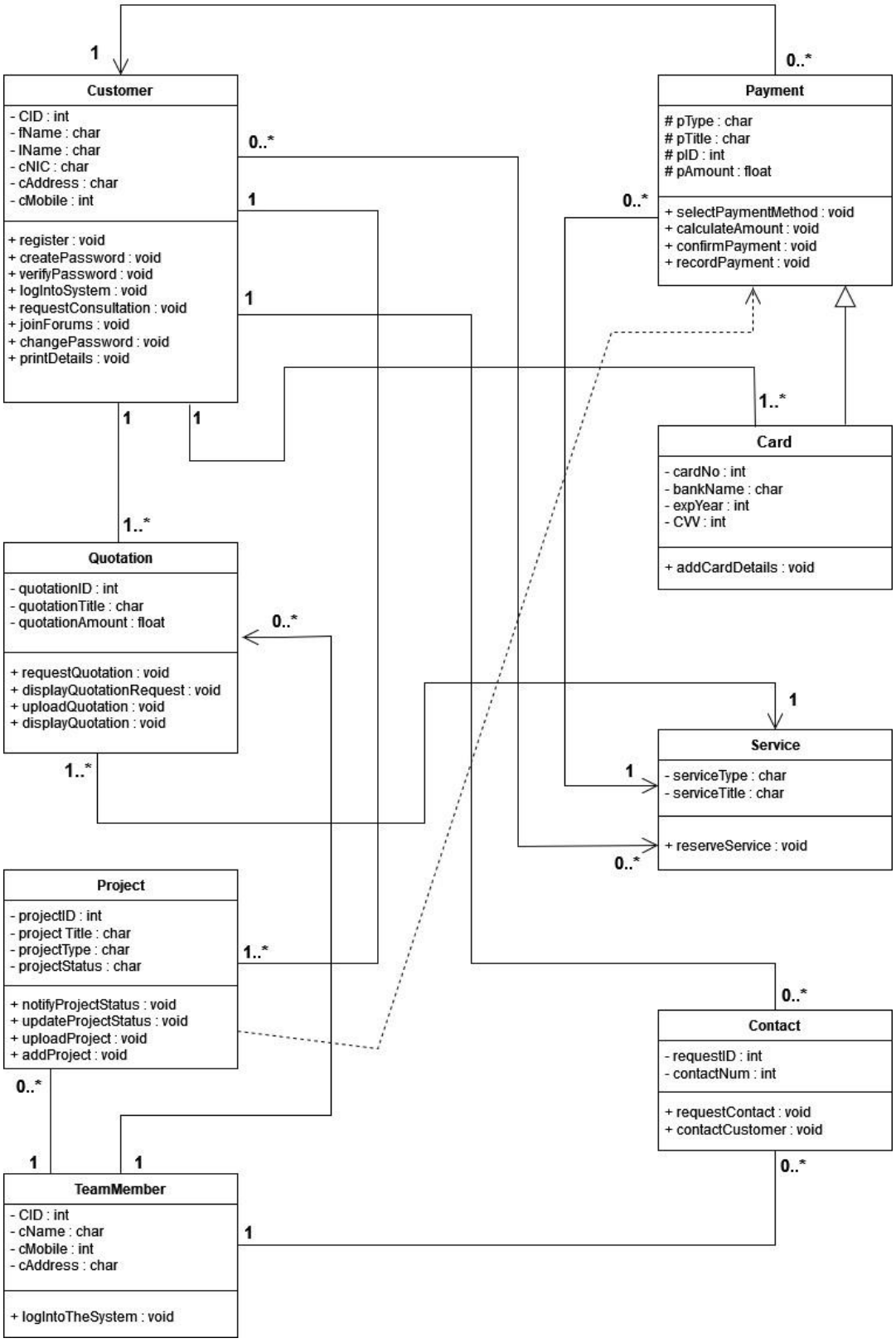
Payment	
Responsibility	**Collaboration**
Select payment method	Customer
Calculate payment	Service, Customer service, Customer
Confirm payment	Customer
Record payment	Team member, Customer, Project

Card	
Responsibility	**Collaboration**
Add card details	Customer, Service

Project	
Responsibility	**Collaboration**
Notify project status	Customer, Team member
Update project status	Team member
Upload project	Customer, Team member

Team Member	
Responsibility	**Collaboration**
Log into the system	

Class diagram (UML notations)



Exercise 2

Code Implementation

Customer.h

```
class Customer
{
   private:

int cID;
char fName[20];
char lName[20];
char cNIC[12];
char cAddress[100];
int cMobile;
Contact *con[100];
Quotation *quo[100];
Service *ser[100];
Project *pro[100];
Card *ca[5];

   public:

      Customer();
      Customer( int cusID, const char cFName[], const char cLName[], const char cusNIC[], const char cusAddress[], int cusMobile, Contact *c[], Quotation *q[], Payment *p[], Service *s[], Project *p[], Card *car[] );
      void register();
```

```
void createPassword();
void verifyPassword();
void logIntoSystem();
void        requestConsultation();
void joinForums();
 void changePassword();
int getCID();
 int          getCMobile();
void printDetails();
~Customer();


};
```

Customer.cpp

```
#include <iostream>
#include <cstring>
#include "Customer.h"
#include "Contact.h"
#include "Quotation.h"
#include "Service.h" #include
"Project.h"


using namespace std;


Customer::Customer()
{       cID=0;
      strcpy(fName, " ");
```

```
strcpy(lName, " ");

strcpy(cNIC, " ");

strcpy(cAddress, " ");

cMobile = 0;



}

Customer::Customer( int cusID, const char cFName[], const char cLName[], const char cusNIC[],
const char cusAddress[], int cusMobile, Contact *c[], Quotation *q[], Payment *p[], Service *s[],
Project *p[], Card car[] )

{

cID = cusID;

strcpy(char fName, cFName);

strcpy(lName, cLName);

strcpy(cNIC, cusNIC);

strcpy(cAddress, cusAddress);

cMobile = cusMobile;


con = c;

quo = q;

ser = s;

pro = p;

ca = car;


}
```

```
void Customer::register()
{

}

void Customer::createPassword()
{

}

void Customer::verifyPassword()
{

}

void Customer::logIntoSystem()
{

}
void Customer::requestConsultation()
{

}

void Customer::joinForums()
{
```

```
}

void Customer::changePassword()
{

}
int Customer::getCID()
{
   return cID;
}

int Customer::getCMobile()
{
   return cMobile;
}

void Customer::printDetails()
{

}

Customer::~Customer()
{

}
```

Payment.h

```
class Payment
{
   protected:

char pType[10];
char pTitle[50];
int pID;
float pAmount;
Service *ser[30];

   public:

Payment();
Payment( const char payType[], const char payTitle[], int payID, float payAmount, Service
*s );
void selectPaymentMethod();
void calculateAmount();
void confirmPayment();
void recordPayment();
~Payment();

};
```

Payment.cpp

```
#include <iostream>
#include <cstring>
#include "Service.h"
#include "Card.h"

using namespace std;

Payment::Payment()
{
strcpy(pType, " ");
strcpy(pTitle, " ");
pID = 0;
pAmount = 0;
}

Payment::Payment( const char payType[], const char payTitle[], int payID, float payAmount,
Service *s )
{
strcpy(pType, payType );
strcpy(pTitle, payTitle );
pID = payID;
pAmount = payAmount;

   ser = s;
}
void Payment::selectPaymentMethod()
```

```
{

}

void Payment::calculateAmount()
{

}

void Payment::confirmPayment()
{

}

void Payment::recordPayment()
{

}

Payment::~Payment()
{

}
```

Card.h

```
class Card::public Payment
{
   private:

int cardNo;
char bankName[30];
int expYear;
int CVV;
Customer *cust;

   public:

Card();
Card( int no, const char bName[], int eYear, int cvv, Customer *cu );        void
addCardDetails();
~Card();
};
```

Card.cpp

```
#include <iostream>
#include <cstring>
#include "Card.h"
#include "Payment.h"
#include "Customer.h"
using namespace std;

Card::Card()
{
CardNo=0;
strcpy(bankName, "");
expYear = 0;
CVV = 0;

}

Card::Card( int no, const char bName[], int eYear, int cvv, Customer *cu )
{
CardNo = no;     strcpy(bankName,
bName);
expYear = eYear;
CVV = cvv;

   cust = cu;
}
```

```
void Card::addCard Details()
{

}

Card::~Card()
{

}
```

Quotation.h

```
class Quotation
{
   private:
int quotationID;
char quotationTitle[100];
float quotationAmount;
Customer *cust;

   public:
Quotation();
Quotation(const char quoTitle[], int qID, int qAmount, Customer * c);
void requestQuatation();
void displayQuotationRequest();
void uploadQuotation();
 void displayQutation();
~Quotation;
```

```
};
```

Quotation.cpp

```
#include<iostream>
#include<cstring>
#include "Customer.h"
#include "TeamMember.h"
#include "Service.h"

using namespace std;

void Quotation::displayQuotationRequest()
{

}

void Quotation::uploadQuotation()
{

}

void Quotation::displayQutation()
{

}

Quotation::Quotation()
{
```

```
    strcpy(quotationTitle,"");
}

Quotation::Quotation(const char quoTitle[], int qID, int qAmount, Customer * c)
{
    strcpy(quotationTitle, quoTitle);
    quatationID = qID;
    qutationAmount = qAmount;
    cust = c;
}

Quotation::~Quotation
{

}
```

Project.h

```
class Project
{
  private:
int projectID;     char
proTitle[100];     char
proType[30];     char
proStatus[30];
Customer * cust;
TeamMember * team;
```

```
 public:
Project();
Project(const char proTitle[],const char proType[],const char proStatus[], int pID,  Customer * c, TeamMember * t );
void notifyProjectStatus();
void addProject( Payment *p );
void updateProjectStatus();
void uploadProject();
~Project;
};
```

Project.cpp

```
#include<iostream>
#include<cstring>
#include "Customer.h"
#include "TeamMember.h"
#include "Payment.h"

using namespace std;

 void Project::notifyProjectStatus()
 {

 }

 void Project::addProject( Payment *p )
 {
```

```
 }

void Project::updateProjectStatus()
{

}

void Project::uploadProject()
{

}

Project::Project()
{
strcpy(projectTitle, "");
strcpy(projectType, "");
strcpy(projectStatus, "");
}

Project::Project(const char proTitle[],const char proType[],const char proStatus[],int pID,
Customer * c, TeamMember * t, Payment * p)
{
strcpy(projectTitle, proTitle);
strcpy(projectType, proType);
strcpy(projectStatus,proStatus);
projectID = pID;
cust = c;
```

```
team = t;
paym = p;
}

Project::~Project
{

}
```

TeamMember.h

```
class TeamMember
{
   private:
int eID;
int eMobile;
char eName[20];
char eAddress[30];
Contact * con[20];
Qutation * quo[20];
Project  * proj[20];

   public:
TeamMember();
TeamMember(const char empName[], const char empAddress[], int eId, int eMobile,
Contact * c[], Qutation * q[], Project * p[]);
void loginToTheSystem();
~TeamMember();
};
```

TeamMember.cpp

```
#include<iostream>
#include<cstring>
#include "Contact.h"
#include "Quatation.h"
#include "Project.h"

using namespace std;

void TeamMember::loginToTheSystem()
{

}

TeamMember::TeamMember()
{
strcpy(eName,"");
strcpy(eAddress, "");
}

TeamMember::TeamMember(empName[], empAddress[], int eId, int eMobile, Contact * c[],
Qutation * q[], Project * p[] )
{
strcpy(eName, empName);
strcpy(eAddress, empAddress);
eID = 0;     eMobile
= 0;
```

```
con = c;
quo = q;
pro = p;
}

TeamMember::~TeamMember()
{
}
```

Contact.h

```
class Contact
{
   private:
int requestID;
int contactNum;
Customer *cust;
TeamMember *team;

   public:
Contact();
Contact(rID, cNum);
void requestContact();
void contactCustomer();
~Contact();
};
```

Contact.cpp

```
#include<iostream>
#include"Customer.h"
#include"TeamMember.h"

using namespace std;

void Contact::requestContact()
 {

 }

void Contact::contactCustomer()
{

}

Contact::Contact(int rID, int cNum, Customer * c, TeamMember * t)
{
requestID = rID;
contactNum = cNum;
cust = c;
team = t;
}

Contact::Contact()
{
```

```
requestID = 0;
contactNum = 0;
}

Contact::~Contact
{

}
```

Service.h

```
class Service
{
   private:
char serviceType[30];
char serviceTitle[50];
Sevice * serv;

   public:
Service();
Sevice( const char serType[], const char serTitle[], Service * s);
void reserveService();
~Service;
};
```

Service.cpp

```
#include <iostream>

#include <cstring>

#include "Quotation.h"

#include "Customer.h"


using namespace std;


void Service::reserveService()

{


}


Service::Service()

{

strcpy(serviceType, " ");     strcpy(serviceTitle,

" ");

}


Service::Service(const char serType[], const char serTitle[], Service * s)

{

strcpy(serviceType, serType );

strcpy(serviceTitle, serTitle );

Service * s;

}


Service::~Service()
```

```
{


}
```

Main.cpp

```
#include <iostream>
#include <cstring>

#include "Customer.h"
#include "Service.h"
#include "Payment.h"
#include "Card.h"
#include "TeamMember.h"
#include "Project.h"
#include "Contact.h"
#include "Quotation.h"

using namespace std;

int main()
{
   // craeting dynamic objects
   Customer *cus1 = new Customer(); // object for customer
   Service *ser1 = new Service(); // object for service
   Payment *pay1 = new Payment(); // object for payment
   Card *car1 = new Card(); // object for card
   TeamMember *mem1 new TeamMember(); // object for team member
```

```
    Project *pro1 = new Project(); // object for project
    Contact *con1 = new Contact(); // object for contact
    Quotation *quo1 = new quotation(); // object for quotation



    // calling methods

    // methods of customer
    cus1->register();     cus1-
>createPassword();     cus1-
>verifyPassword();     cus1-
>logIntoSystem();     cus1-
>requestConsultation();     cus1-
>joinForums();     cus1->changePassword();
    cus1->printDetails();

    // methods of payment     pay1-
>selectPaymentMethod();     pay1-
>calculateAmount();     pay1->confirmPayment();
pay1->recordPayment();

    //methods of card     car1-
>addCard Details();     // methoda of
team member     mem1-
>loginToTheSystem();
```

```
   //methods of service     ser1-
>reserveService();

   //methods of project     pro1-
>notifyProjectStatus();     pro1-
>updateProjectStatus();     pro1-
>uploadProject();     pro1->addProject();



   //methods of quotation     quo1-
>requestQuatation();     quo1-
>displayQuotationRequest();     quo1-
>uploadQuotation();     quo1->displayQuotation();

   //methods of contact     con1-
>requestContact();     con1->contactCustomer();

  return 0;

}
```

Individual contributions

IT21816390 - De Silva D L S S

- Payment class
- Noun verb analysis

IT21810664 - Perera K C G S

- Customer class
- Main.cpp program
- Requirement analysis
- identifying responsibilities

IT21812262 - Rasanjana J A D

- Service class
- Quotation class
- UML diagram

IT21817694 - Edirisinghe E A M N

- Card class
- Contact class
- UML diagram

IT21806940 - Wanniarachchi P W A S V

- Project class
- TeamMember class
- CRC cards