Topic : Online Identity Issuing Service

Group no : MLB_01.02_03

Campus : Malabe

Submission Date : 20/05/2022

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**
IT21192814	Chamod A.A.A.	0715791563
IT21195198	Thathsara P.G.C.S.	0789104932
IT21193972	Pathirana D.P.D.C.	0760530071
IT21196492	Kulathunga N.A.S.	0777321113
IT21193286	Peiris W.R.L.	0778640845

Requirements of the system

1. A guest user can create an account as a registered user.
2. The system needs to store all the details of the registered user.
3. Both registered and guest users can browse the system.
4. Registered user can access to the services by login into the system.
5. System should check the validity of login account details when a user logs in to the system.
6. Registered users should be able to edit, modify or delete their personal account information in the system.
7. Registered user can select the relevant application form and enrol into it.
8. The system should provide a valid application form including the correct procedure to the user.
9. Registered users can submit the online application by filling the relevant information such as full name, date of birth, Email address, phone number, gender, and ID photo.
10. System should check the validity of submitted information of the application and store it in the system.
11. Users should select their payment method and enter payment details such as account number and verification code.
12. The system should do the verification of the payment and store the payment details of the respective user.
13. User can submit the payment slip and save and finish the application.
14. Users can give their feedbacks to the system, and these will generate a feedback ID that can be accessed by the admin.
15. Admin can edit, delete, update, or modify system features and functions according to the user feedbacks.
16. System should allow for updates and upgrades.
17. Admin staff can manage and maintain the system.

Noun Verb analysis

(Nouns)

1. A guest user can create an account as a registered user.
2. The system needs to store all the details of the registered user.
3. Both registered users and guest users can browse the system.
4. Registered user can access to the services by login into the system.
5. System should check the validity of login account details when a user logs in to the system.
6. Registered users should be able to edit, modify or delete their personal account information in the system.
7. Registered user can select the relevant application form and enrol into it.
8. The system should provide a valid application form including the correct procedure to the user.
9. Registered users can submit the online application by filling the relevant information such as full name, date of birth, Email address, phone number, gender, and ID photo.
10. System should check the validity of submitted information of the application and store it in the system.
11. Users should select their payment method and enter payment details such as account number and verification code.
12. The system should do the verification of the payment and store the payment details of the respective user.
13. User can submit the payment slip and save and finish the application.
14. Users can give their feedbacks to the system, and these will generate a feedback ID that can be accessed by the admin.
15. Admin can edit, delete, update, or modify system features and functions according to the user feedbacks.
16. System should allow for updates and upgrades.
17. Admin staff can manage and maintain the system.

(Verbs)

1. A guest user can create an account as a registered user.
2. The system needs to store all the details of the registered user.
3. Both registered and guest users can browse the system.
4. Registered user can access to the services by login into the system.
5. System should check the validity of login account details when a user logs in to the system.
6. Registered users should be able to edit, modify or delete their personal account information in the system.
7. Registered user can select the relevant application form and enrol into it.
8. The system should provide a valid application form including the correct procedure to the user.
9. Registered users can submit the online application by filling the relevant information such as full name, date of birth, Email address, phone number, gender, and ID photo.
10. System should check the validity of submitted information of the application and store it in the system.
11. Users should select their payment method and enter payment details such as account number and verification code.
12. The system should do the verification of the payment and store the payment details of the respective user.
13. User can submit the payment slip and save and finish the application.
14. Users can give their feedbacks to the system, and these will generate a feedback ID that can be accessed by the admin.
15. Admin can edit, delete, update, or modify system features and functions according to the user feedbacks.
16. System should allow for updates and upgrades.
17. Admin staff can manage and maintain the system.

Identified Classes

- Guest user
- Registered user
- System
- Application
- Payment
- Feedback
- Admin
- Account

CRC Cards

Guest User	
Responsibilities:	**Collaborations:**
Create account in the system	
Browse the website system	
Give feedbacks	Feedback

Registered User	
Responsibilities:	**Collaborations:**
Access to the services	System
Browse the website system	
Give feedbacks	Feedback
Select the application and enrol	Application
Edit, modify, or delete their personal account information	System
Fill and submit the application	Application
Make the payment	Payment

System	
Responsibilities:	**Collaborations:**
Store all the details of the registered user	Registered User
Check the validity of login account details	Registered User
Provide a valid application form	Application
Check the validity of submitted information of the application and store it in the system	
Do verification of the payment and store the payment details	Payment
Allow for updates and upgrades	Admin

Application	
Responsibilities:	**Collaborations:**
Store user data	System, Registered user
Store payment slip	Payment

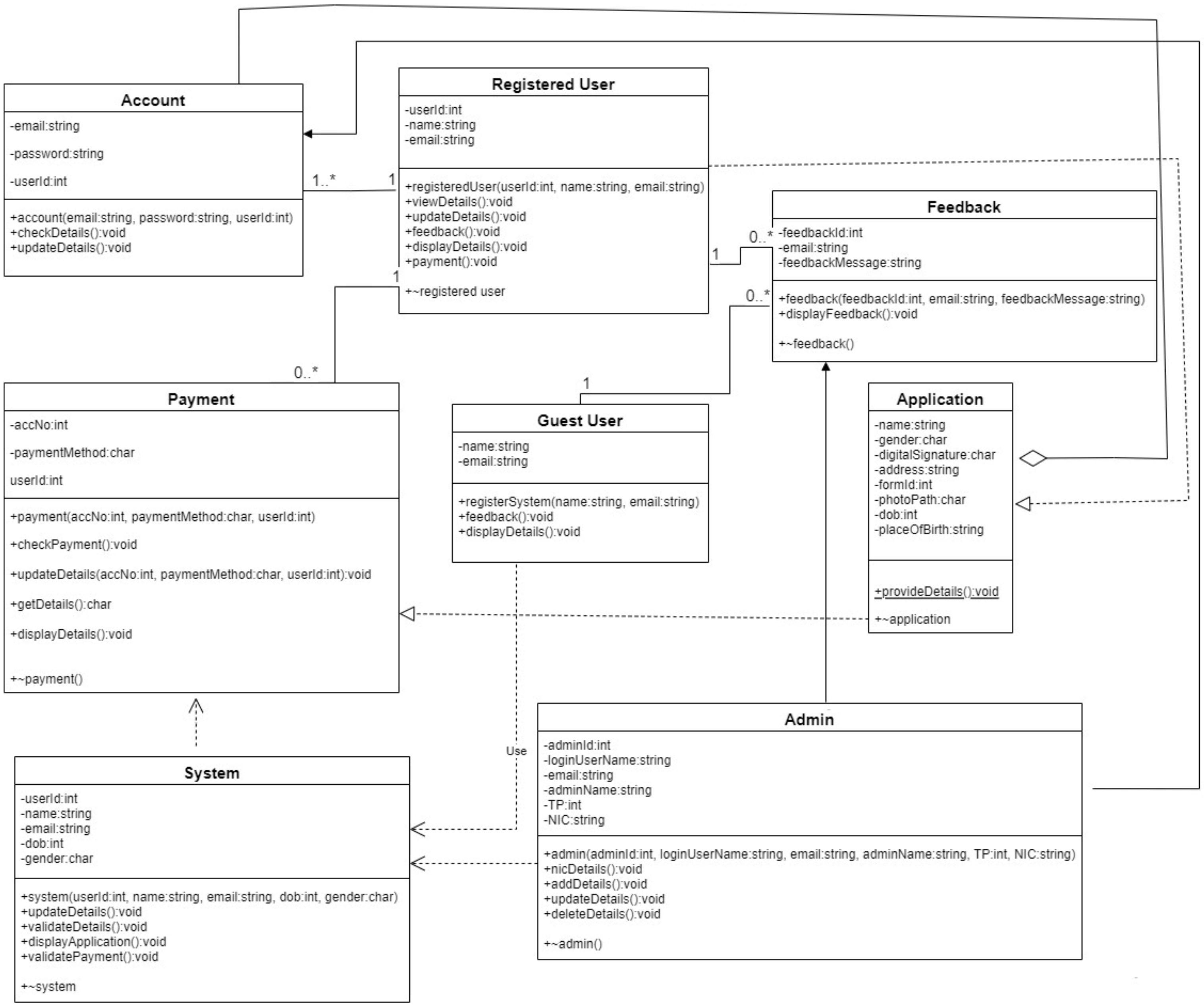
Payment	
Responsibilities:	**Collaborations:**
Allow users to select the payment method	System
Allow users to make the payment	Registered User
Issue payment slip	System

Feedback	
Responsibilities:	**Collaborations:**
Allow users to give their feedbacks to the system	Guest user, Registered User
Generate a feedback ID	System
Store feedbacks and direct them to the admin	Admin

Admin	
Responsibilities:	**Collaborations:**
Access the feedback IDs and check user feedbacks	Feedback
Edit, delete, update, or modify system features and functions according to the user feedbacks	Registered user, Guest User, Feedback
Manage and maintain the system	System

Account	
Responsibilities:	**Collaborations:**
Allow user to create account in the system	System
Allow system to validate user account details	System
Allow users to edit, modify or delete their personal account information in the system	Registered user

Class Diagrams



Codes

payment.h

```
class payment

{

private:

        int accNo;

        char paymentMethod;

        int userId;


public:

        void setpayment(int accNo,char paymentMethod,int userId);

        void checkPayment();

        void updateDetails(int accNo,char paymentMethod, int userId);

        char getDetails();

        void displayDetails();


        ~payment();

};
```

payment.cpp

```
#include<iostream>

#include "payment.h"


using namespace std;


int main() {

        payment::payment(int accNo,char paymentMethod,int userId) {
        }

                Account_No = accNo;

                Payment_Method = paymentMethod;

                User_ID = userId;

        }


        void payment::checkPayment() {


        }


        void payment::updateDetails(int accNo,char paymentMethod, int userId)
{


        }


        char payment::getDetails() {


        }


        void payment::displayDetails(){


        }


        payment::~payment(){


        }


}
```

feedback.h

```
class feedback

{

private:

        int feedbackID;

        string email;

        string feedbackMessage;


public:

        void setfeedback(int feedbackID, string email, string feedbackMessage);

        void displayFeedback();


        ~feedback();

};
```

feedback.cpp

```
#include<iostream>

#include "feedback.h"


using namespace std;


int main() {

        feedback::feedback(int feedbackID, string email, string
feedbackMessage) {

                Feedback_ID = feedbackID;

                Email = email;

                Feedback_Message = feedbackMessage;
        }

        void Feedback::displayFeedback() {

        }

        feedback::~feedback() {

        }

}
```

system.h

```
class System{
  private:
    int DOB;
    int userID;
    char gender;
    int update,vdetail,vpayment,display;

  public:
    void setSystem( int u);
    void setupdateDetails(int up);
    void setvalidateDetails(int vd);
    void setvalidatePayment(int vp);
    void setdisplayApplication(int da);
    int getSystem();

    int getupdateDetails();

    int getvalidateDetails();

    int getvalidatePayment();

    int getdisplayApplication();


};
```

system.cpp

```
#include "system.h"


void System::setSystem(int u){

  userID=u;


}

void System::setupdateDetails(int up){

  update=up;

}

void System::setvalidateDetails(int vd){

  vdetail=vd;

}

void System::setvalidatePayment(int vp){

  vpayment=vp;

}

void System::setdisplayApplication(int da){

  display=da;

}
```

```
int System::getSystem(){
  return userID;


}
int System::getupdateDetails(){
   return update;
}
int System::getvalidateDetails(){
   return vdetail;
}
int System::getvalidatePayment(){
   return vpayment;
}
int System::getdisplayApplication(){
   return display;
}
```

guest.h

```
class guest{


        private:
                string name;
                string email;


        public:
                void setusermail(string e);
                void setusername(string na);

                string getusermail();
                string getusername();

};
```

guest.cpp

```
void guest::setusermail(string e){

        email=e;



}

void  guest::setusername(string na){

        name=na;




}
```

```
string guest::getusermail(){

          return email ;

          }
string guest::getusername(){

          return name;

}
```

registerUser.h

```
class registerUser
{
 private
    string userid;
    int name;
    int mail;
public
    void setuserid(int ui);
    void setname(int n);
    void setmail(int m);

    int getuserid();
    int getsetname();
    int getmail();
}
```

registerUser.cpp

```
#include "registerUser.h"
void registerUser::setuserid(int ui)
{
  userid = ui;
}
void registerUser::setname(int n)
{
  name = n;
}
void registerUser::setmail(int m)
{
  mail = m;
}
int registerUser::getuserid()
{
  return userid;
}
int registerUser::getname()
{
  return name;
}
int registerUser::getmail()
{
  return mail;
}
```

admin.h

```
include<iostrem>
class admin
{
 private:
   int adminid;
   string loginusername;
   string email;
   int tp;
 public:
   void setnicdetails(int d);
   void setadddetails(int a);
   void setupdatedetails(int ud);
   void setdeletedetails(int dd)
   int getnicdetails(int d);
   int getadddetails(int a);
   int getupdatedetails(int ud);
   int getdeletedetails(int dd)
};
```

admin.cpp

```
include "admin"
void admin::setnicdetails(int d)
{
  nicdetails = d
}
void admin::setadddetails(int a)
```

```
{
  adddetails = a
}
void admin::setupdatedetails(int ud)
{
  updatdettails = ud
}
void admin::setdeletedetails(int dd)
{
  deletedetails = dd
}
int admin::getnicdetails()
{
  return nicdetails;
}
int admin::getadddetails()
{
  return adddetails;
}
int admin::getupdatedetails()
{
  return  updatedetails;
}
int admin::getdeletedetails()
{
  return deletedetails
}
```

account.h

```
class Account
{
private:
	char Email[20];
	char Password[20];
	char UserID[20];
public:
	Account(const char AEmail[], const char APassword[], const char
AUserID[]);
	void checkDetails();
	void updateDetails();
};
```

account.cpp

```
#include <iostream>
#include<cstring>
using namespace std;
Account::Account(const char AEmail[], const char APassword[], const char
AUserID[])
{
	strcpy_s(Email, AEmail);
	strcpy_s(Password, APassword);
	strcpy_s(UserID, AUserID);
}
void Account::checkDetails()
```

```
{
}
void Account::updateDetails()
{
}
int main()
{
        Account A1("Rumesh@my.sliit.lk","123456789","1");
        return 0;
}
```

application.h

```
class Application
{
{
private:
        string Name[20];
        char Gender[6];
        char DigitalSignature[20];
          string Address[40];
        int  FormID;
        int DOB;
        int PhotoPath[20];
        string PlaceOfBirth[30];
public:
        Application(const char AName[], const char AGender[], const char
ADigitalSignature[], const char AAddress[], int AFormID, int ADOB, const char
APhotoPath[], const char APlaceOfBirth[]);
```

```
        void ProvideDetails();

};
```

application.cpp

```
#include <iostream>

#include<cstring>

using namespace std;

Application::Application(const char AName[], const char AGender[], const char
ADigitalSignature[], const char AAddress[],int AFormID, int ADOB, const char
APhotoPath[], const char APlaceOfBirth[])

{

        strcpy_s(Name, AName);

        strcpy_s(Gender, AGender);

        strcpy_s(DigitalSignature, ADigitalSignature);

        strcpy_s(Address, AAddress);

        FormID=AFormID ;

        DOB=ADOB;

        strcpy_s(PhotoPath, APhotoPath);

        strcpy_s(PlaceOfBirth, APlaceOfBirth);

}

void Application::ProvideDetails()

}

Application
A2("Rumesh","Male","Wer234pgh","Wennappuwa",1,20010409,"C/img","Chil
aw");

        return 0;

}
```

main.cpp

```
#include<iostream>
#include<iomanip>
#include<string>
#include "system.h"
#include "guest.h"
#include "payment.h"
#include "feedback.h"
#include "admin.h"
#include "registeruser"
#include "account"
#include "application"

using namespace std;

int main(){
  System s1;
  int ua,di;
  cout<<"input UserID :";
  cin>>ua;
  cout<<"update paiment price :";
  cin>>di;
  s1.setSystem(ua);
  s1.setvalidatePayment(di);
  s1.setvalidateDetails(1);
  cout<<"user ID is :"<<setw(5)<<setfill('0')<<s1.getSystem()<<endl<<"your
paiment is :"<< s1.getvalidatePayment()<<endl;

  Guest g1;
  g1.setusermail('a');

  Payment p1;
  p1.setpayment(1,'a',1);

  Feedback f1;
  f1.setfeedback(1,'a','a');

  Admin a1;
  a1.setnicdetails(1);

  registerUser r1;
  r1.setuserid(1);

  account c1;
  c1.setupdateDetails();

  aplication l1;
  l2.setprovideDetails();

  return 0;
}
```