Sri Lanka Institute of Information Technology

Discover Your Future

Topic : **Online voting system for award ceremony**

Group Number : **MLB_S2_WD_10.1_10**

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**
IT21185120	A.H.M.T.S.Aberathne	0714358513
IT21183690	H.M.I.K.Dhanawardhana	0716962033
IT21183836	E.M.M.V.P. Ekanayaka	0763829778

Object oriented concepts

1) User Requirement

1. An Unregistered user in an online voting system needs to first register providing details such as name, ID, e-mail address.

2.The voter should create a profile as a guest and register to the system as a member by providing valid username, password, DOB, email and NIC number.

3. System checks the validity of the user registration details and if there are some errors in the registration details user can edit user details.

4.The register user can log in to system using valid username and password.

5.If there are some errors with log in process the user should reset log in credentials.

6.The register user can view nominee profile and select only one nominee from nominee list.

7. System should allow users to cast their votes quickly.

8. Voters should not be able to vote more than once.

9.Voters should be prevented from choosing more than one nominee and may receive a warning message from the system.

10.Voters can check their votes and give feedbacks for the nominee profile.

11.Once the voter casts the vote and collection of votes are updated automatically.

12. System admin should login to the system as main admin by providing admin id and password.

13.Admin can access the details of registered members.

14.System admin also manages user and nominee accounts as well.

15.System admin should be able to add nominee id, nominee name and other relevant details.

16.Admin can view all the profiles, edit the profiles, and check the progress of website.

17.System admin can view all the feedbacks, comments, and reports.

18. System maintainer should maintain the records of voter name, voter id, and other voter details.

19. System maintainer should record and count all the votes and also always update the results during the period.

20.System admin should be possible to verify all votes have been correctly accounted for the final report.

21.Admin should calculate all the valid votes and final decision to the audience.

22.Admin should update the winner id in order and publish the final winner.

2) Noun and Verb Analysis

(Noun are in red and Verb are in blue.)

1. An Unregistered user in an online voting system needs to first register providing details such as name, ID, e-mail address.

2.The voter should create a profile as a guest and register to the system as a member by providing valid username and password, DOB, email and NIC number.

3. System checks the validity of the user registration details and if there are some errors in the registration details user can edit user details.

4. should enter valid details of username and password.

5. If there are some errors with log in process the user should reset log in credentials.

6. The registered user can view nominee profile and select only one nominee from nominee list.

7. System should allow users to cast their votes quickly.

08. Voters should not be able to vote more than once.

09.Voters should be prevented from choosing more than one nominee and may receive a warning message from the system.

10.Voters can check their votes and give feedbacks for the nominee profile.

11.Once the voter casts the vote and collection of votes are updated automatically.

12. System admin should login to the system as main admin by providing admin id and password.

13. Admin can access the details of registered members.

14.System admin also manages user and nominee accounts as well.

15.System admin should be able to add nominee id, nominee name and other relevant details.

16.Admin can view all the profiles, edit the profiles, and check the progress of website.

17. System admin can view all the feedbacks, comments, and reports.

18. System maintainer should maintain the records of voter name, voter id, and other voter details.

19. System maintainer should record and count all the votes and always update the results during the period.

20.System admin should be possible to verify all votes have been correctly accounted for the final report.

21.Admin should calculate all the valid votes and final decision to the audience.

22.Admin should update the winner id in order and publish the final winner.

3) Identified Classes using Noun Verb Analysis

Identified classes:

- Registered User
- Unregistered User
- Voter
- Nominee
- Admin
- System maintainer
- Feedback
- Winner

<u>Reasons for rejecting other nouns</u>

Redundant:

- Registered user/Unregistered user – Refers to the same person' User'
- In online voting system for award nomination guest, voter, member refers to the same person.

An event or an operation:

- Search the nominees and casting their votes.

Outside scope of system:

- System
- System administrator

Attributes:

- Username, password
- User account
- Registration details
- Voter id

- Nominee id
- Admin id
- System maintainer id
- Profile
- Vote id
- Records of votes
- report
- Winner id

Noun

	verbs
Unregistered user	• Create • view
Registered user	• login • view • select
Voter	• login • view • search • vote
Nominee	• login • view • edit • update • check • give feedback
System maintainer	• Login • View • Add • Update • Count • Edit • Generate report
Admin	• login • view • publish report
Winner	• login

	• view

1) CRCcards for online voting system for award nomination

Registered User	
Responsibilities	**Collaborations**
Store login details	
Edit user details	
View nominee details	Nominee profile

Unregistered User	
Responsibilities	**Collaborations**
Store Register details	

Voter	
Responsibilities	**Collaborations**
Store voter details	
Edit voter details	
View nominee profile	Nominee profile
Cast the vote	
Give feedback	

Nominee	
Responsibilities	**Collaborations**
View nominee details	
View winners in order	Admin

Nominee profile	
Responsibilities	**Collaborations**
Store nominee details	
Show nominee photos and details	
Add nominee details	Admin

Admin	
Responsibilities	**Collaborations**
View all profiles	Nominee profile
Edit all nominee profiles	
Update nominee profiles	
Give access to registered members	Registered members
View all report of nominees	
Publish the final winner	System maintainer

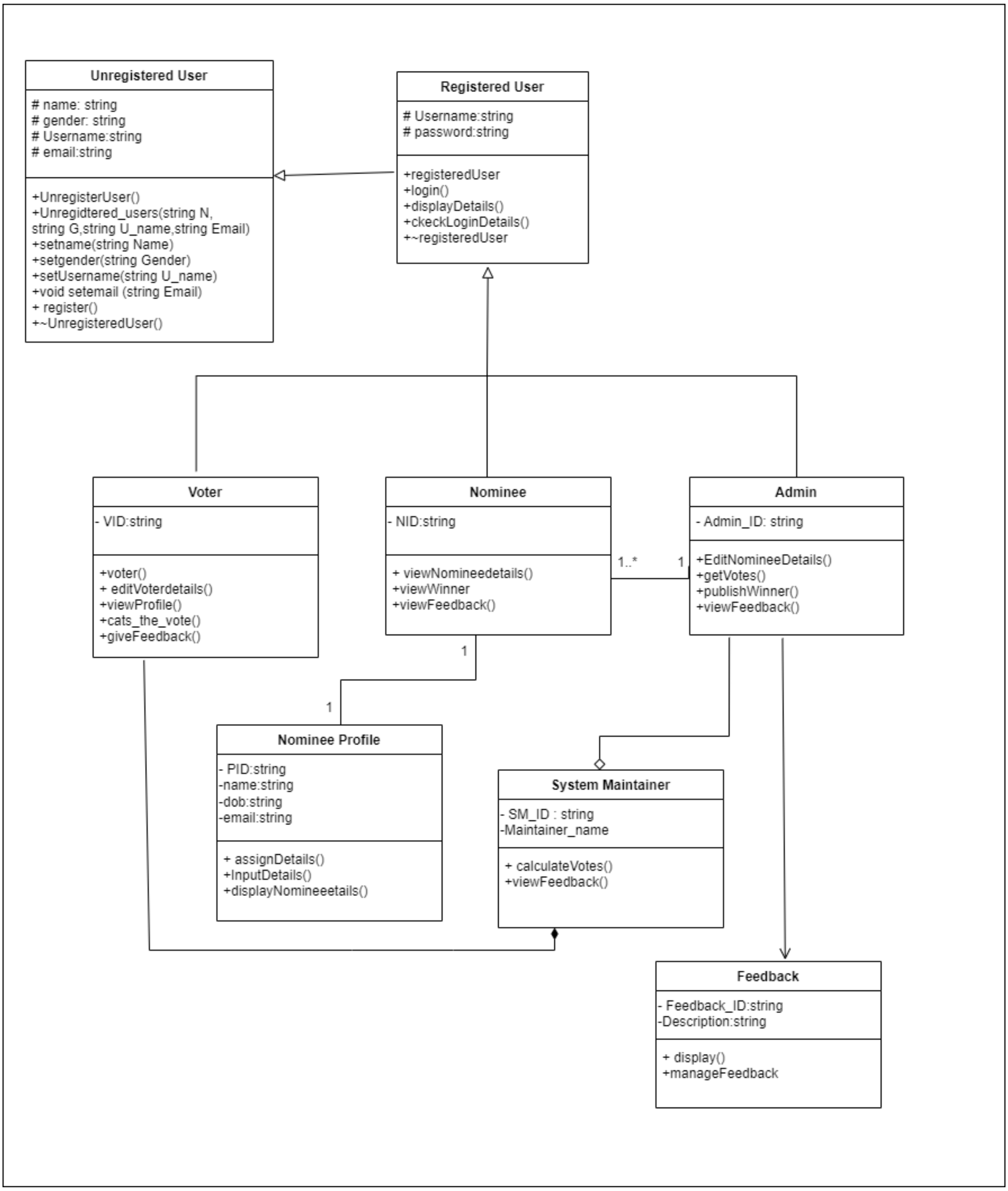
System maintainer	
Responsibilities	**Collaborations**
Record voters’ details	Voter
Calculate votes	Voter
Update result	

Verify final decision	

Feedback	
Responsibilities	**Collaborations**
Show feedbacks	Voter
Store feedback description	Voter

Exercise 01

Class diagram for the Online Voting System for Award Nomination



Exercise 02.

Unregistered_user.h

```
//IT21183690_H.M.I.K.Dhanawardhana_Registered_user.h
#include <string>
using namespace std;

class Unregistered_users{
                    protected :
                        string name;
                        string gender;
                        string username;
                        string email;

                    public :
                      Unregidtered_users();
                        Unregidtered_users(string N,string G,string U_name,string Email);
                        Void setname(string N);
                        void setgender(string G);
                        void setusername(strin U_name);
                        void setemail (string Email);
                        register();
                        ~Unregistered_users();
};
```

Unregistered_user.cpp

```
//IT21183690_H.M.I.K.Dhanawardhana_Unregistered_user.cpp


#include "Unregistered_user.h"

#include<iostream>


using namespace std;


Unregistered_users::Unregistered_users(){


}


Unregistered_users::Unregistered_users(string N,string G,string U_name,string Email){

                                        name=N;

                                        gender=G;

                                        username=U_name;

                                        email=Email;

}

void Unregistered_users::setname(string N){

                                        name=N;

}

void Unregistered_users::setgender(string G){

                                        gender=G;

}

void Unregistered_users::setusername(string U_name){

                                        username=U_name;

}

void Unregistered_users::setemail(string Email){

                                        email=Email;
}
void Unregistered_users::regiter(){
                                        cout<<"Registration compelet"<<endl;
}


Unregistered_users::~Unregistered_users(){
                                        cout<<"object are destroyed!"<<endl;
}
```

Registered_user.h

```
//IT21183690_H.M.I.K.Dhanawardhana_Registered_user.cpp
#include <string>


using namespace std;


class Registered_users:public Unregistered_users
{
                                        protected:
                                               string Username;
                                               string Password;


                                        public :
                                           Registered_users();
                                               Registered_users(string U_name,string P_word,string
N,string G,string Email);
                                               void displayDetails();
                                               void login();
                                               string checkLoginDetails();
                                               ~Registered_user();
};
```

Registered_user.cpp

```
//IT21183690_H.M.I.K.Dhanawardhana_Registered_user.cpp
#include "Registered_user.h"
#include "Unregistered_user.h"
#include <string>

using namespace std;

Registered_users::Registered_users(){

}

Registered_users::Registered_users(string U_name,string P_word,string N,string G,string U_name,string Email){

                                        Username=U_name;
                                        Password=P_word;
}
void Registered_users::displayDetails(){



}

void Registered_users::login(){

}
```

```
void Registered_users::logout(){


}
string Registered_users::checkloginDetails{
   return 0;
}


Registered_users::Registered_users()
{
            cout<<"destroyed"<<endl;
}
```

Voter.h

```
//IT21183690_H.M.I.K.Dhanawardhana_voter.h
#include <string>


using namespace std;


class voter:public Registered_users
{
            private:
                string Voter_ID;
                feedbacks *feed[2];


            public:
              voter(string U_name,string P_word,string N,string G,string U_name,string Email);
                void editVoterDetails();
```

```
                                        void ViewProfile()
                                        void CastVote();
                                        void Give_feedback();
                                        void Addfeedback (feedbacks *f1,feedbacks *f2);
                                        ~voter();
};
```

Voter.cpp

```
//IT21183690_H.M.I.K.Dhanawardhana_voter.cpp
#include "voter.h">
#include "Unregistered_user.h"
#include "Registered_user.h"
#include "Feedbacks.h"
#include <iostream>


 voter::voter(string string U_name,string P_word,string N,string G,string U_name,string
Email)
 {
                                   Voter_ID=V_ID;
 }
 void voter::editVoterDetails()
 {


 }


 void voter::ViewProfile()
 {


 }
```

```
void voter::CastVote(){


}


 void voter::(Givefeedback)
{


}
void voter::Addfeedback (feedbacks *f1,feedbacks *f2)
{
                                        feed[0]=f1;
                                        feed[2]=f2;
}


voter::voter()
{
                                        cout<<"destroyed"<<endl;
}
```

Admin.h

```
//it21183836_E.M.M.V.P.Ekanayake_Admin.h
#include<Feedback.h>
#include”SystemMaintainer.h”


#define SIZE1 3
#define SIZE2 3
```

```
class Admin{

                                    private:
                                          string adminID;
                                          string adminName;
                                          string adminEmail;
                                          string adminPassword;

                                          Nominee* adm[SIZE1];
                                          Feedback* fed[SIZE2];

                                    public:
                                          Admin();
                                          Admin(string aID, string aName, string aEmail, string
aPassword);
                                          void editNomineeDetails();
                                          void getVotes();
                                          void publishWinner();

};
```

Admin.cpp

```
//it21183836_E.M.M.V.P.Ekanayake_Admin.cpp
#include<Feedback.h>
#include<SystemMaintainer.h>
#include<Admin.h>

#include<string>
```

```
Admin::Admin()
{
                                        adminID = 0;
                                        strcpy(adminName,"");
                                        strcpy(adminEmail,"");
                                        strcpy(adminPassword,"");
}

Admin::Admin(string aID, string aName, string aEmail, string aPassword)
{
                                        strcpy(adminID,aID);
                                        strcpy(adminName,aNmae);
                                        strcpy(adminEmail,aEmail);
                                        strcpy(adminPasword,aPassword);
}

void Admin::editNomineeDetails()
{

}

void Admin::getVotes()
{

}

void Admin::publishWinner()
{
}
```

SystemMaintainer.h

```
//it21183836_E.M.M.V.P.Ekanayake_SystemMaintainer.h
#include"Admin.h"
#includde"voter.h"
#define SIZE1 3
#define SIZE2 3

class SystemMaintainer{

                                        private:
                                                string maintainerID;
                                                string maintainerName;

                                                Voter* vote[SIZE1];
                                                Admin* ad[SIZE2];



                                        public:
                                                SystemMaintainer(string mID, string mName);
                                                void addVoters(){
                                                          Voter[0] = new Voter();
                                              Voter[1] = new Voter();
                                              Voter[2] = new Voter();
                                                }
                                                void calculateVotes();
                                                void viewFeedback();
};
```

SystemMaintainer.cpp

```
//it21183836_E.M.M.V.P.Ekanayake_SystemMaintainer.h
#include"Admin.h"
#includde"voter.h"
#include"SystemMaintainer.h"

#include<cstring>

SystemMaintainer::SystemMaintainer(string mID, string mName)
{
                                        strcpy(maintainerID,mID);
                                        strcpy(maintainerName,mName);
}

void SystemMaintainer::calculateVotes()
{

}

void SystemMaintainer::viewFeedback()
{

}
```

Feedback.h

```
//it21183836_E.M.M.V.P.Ekanayake_Feedback.h
class Feedback{

                    private:
                        string feedbackID;
                        string description;

                    public:
                        void manageFeedback(string fID, string fdescript);
                        void display();



};
```

Feedback.cpp

```
//it21183836_E.M.M.V.P.Ekanayake_Feedback.cpp
#include"Feedback.h"
#include"cstring"



void Feedback::manageFeedback(string fID, string fdescript)
{
                    feedbackID = fID;
                    description = fdescript;
}

void Feedback::display()
{
```

```
                                            cout << "" << feedbackID << "" << description << endl;
}
```

Nominee.h

```
#include "Nominee.h"

#include "Admin.h"

#include "NProfile.h"


#include <iostream>

#include <string>

#define SIZE 1


class Nominee

{

private:

   string nomineeID;

   string nomineeName;

   string nomineeEmail;

   string nomineeNumber;

public:

  void  NomineeDetails(string N_ID,string N_Name,string N_Email,string N_Email,string
N_Number);


   void Nominee::view Details();

   void Nominee::view winner();

   void Nominee::view feedback();

   ~Nominee();

};
```

Nominee.cpp

```
#include "NProfile.h"

#include "Admin.h"

#include "Registered_User.h"

#include<cstring>

#include<iostream>

using namespace std;


N::Nominee (string N_ID,string N_Name,string N_Email, string nominee_G,string
nomiineeNumber)


void NProfile:: inputdetails()

{

 cout<<"please enter your ID,Name,Email,NomineeNumber:";

 cin>>ID,Name,Email,NomineeNumber;

}

void Nprofile:: viewDetails()

{

  cout<<"Profile ID:"<<pID<<endl;

  cout<<"Name:"<<name<<endl;

  cout<<"Email:"<<email<<endl;

  cout<<"NomineeNumber:"<<NomineeN<<endl;

}
```

Nominne_Profile.h

```
//IT21185120_A.H.M.T.S.Aberathne_Nominee_Profile.h

#include "nominee.h"

#include <string>

#define SIZE1


CLASS Nominee_Profile

{

                    private:

                              string nomineeP_ID;

                              char nomineeP_Name;

                              int nomineeP_Number;

                              char nomineeP_Email;


                    public:

                              NomineeP(string NomineeP_ID,string
NomineeP_Name,string Nomineep_Number,string nomineeEmail);


                               void inputdetails();

                               void displaydetails();

                               void assign();


};
```

Nominne_Profile.cpp

```
//IT21185120_A.H.M.T.S.Aberathne_Nominee.cpp

#include"nominee.h"

#include"cstring"
```

```
#include"iostream"

using namespace std;


NomineeP::nomineeP (string nomineeP_ID,string nomineeP_Name,string
nomineeP_Email,string nominee_DOB, string nominee_Gender )

{


  NomineeP=new
NomineeP(nominee_PID,nomineeP_Name,nomineeP_Number,nomineeP_Email,nomineeP_
DOB, nomineeP_Gender);

}


void NomineeP:: assignDetails()

{

     NomineeP->assignDetails()

}

void NomineeP:: displaydetails()

{

     NomineeP->displaydetails()

}

NomineeP::~NomineeP()

{

     COUT<<"Deleting NomineeP" <<endl;

}
```

Main.cpp

```
//MLB_10.1_10_Online voting system for award nomination
#include <iostream>
#include "Unregistered_user.h"
#include "Registered_user.h"
#include "voter.h"
#include "Admin.h"
#include "SystemMaintainer.h"
#include "Feedback.h"
#include "Nominee.h"
#include "Nominee_Profile.h"

using namespace std;


int main()
{
                              //Unregistered_user objects
                              Unregistered_users user1, user2;


                              user1.Unregistered_users("Vihanga","male","Viha@1011","
Vihanga@gmail.com");
                              user2.Unregistered_users("Vihanga","male","Viha@1011","
Vihanga@gmail.com");



                              cout<<"-----------------------------"<<endl;


  //Registered_user objects
  Registered_users R_user1, R_user2;
```

```
R_user1.Registered_users("Nuwan@3033","1234#nuwa","Nuwan","male","nuwa@gmail.com");
                                        R_user1.Registered_users("Hansi#3235","4534@Han","Hansika","femail","hansi@gmail.com");


                                        R_user1->displayDetails();
                                        R_user2->displayDetails();


                                        //voter objects
                                        voter v;
                                        v.voter("IP1234","Srima@3033","1234#Srima","Srima","female","srima@gmail.com")
                                        v.CastVote();



//object Admin class
  Admin* adm = new Admin();


 Admin->editNomineeDetails();
 Admin->getVotes();
 Admin->publishWinner();


 //object SystemMaintainer class
SystemMaintainer* main = new SystemMaintainer();


 Admin->calculateVotes();
 Admin->viewFeedback();


  //object Feedback class
Feedback* fd = new Feedback();
```

```
  //object Nominee class

  Nominee N ;
N.Nominee("223ihs","sachini","sach.hd@gmail.com","female","0712436722")
N.viewDetails();

  //objects Nominee_Profile classs

  Nominee_Profile N_p;
  N_p.displaydetails();



  //delete dynamic objects
   //delete Unregistered_users
  delete user1;
  delete user2;

   //delete Registered_users
  delete R_user1;
  delete R_user2;

  //delete nominee
  delete N;

  //delete nominee_Profile
  delete N_p;
```

```
}
```