Topic : Online voting system for Award nominations

Group no :MLB_06.02_Group12

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**
IT21329456	Subawickrama N.S.P.	0702141560
IT21336218	Fernando W.S.S.	0710978668
IT21313998	Rasara S.H.K.	0778095593
IT21324888	Soyza I.N.D.K.	0774827060
IT21328084	Kattadige S.B.P.	0717520287

System Requirements

- The System should function 24/7/365.
- Unregistered users can overview the system, to use the system, they must register with the system by providing details such as Name, Address, NIC, Email, contact.
- Registered customers are of five types called Voters, Nominees, Sponsors, Organizers, and System admin; where they can log into the system by entering the correct username and password.
- Voters should be able to vote for preferred nominee, check current results, check news and events, get subscribe to exclusives. Voters should be able to filter nominees by types and genres.
- Nominees should be able to check their results levels, check news and events, get subscribe to exclusives.
- Sponsors should be able to contact Organizers for events, advertise, check current results, check news and events.
- Organizers should be able to contact Nominees, Sponsors, System admins & check current results, check news and events. Organizers should be able to give details about events.
- System admins should confirm details about events, enter & update details about users, update results page, update news and events, remove outdated data.
- Registered customers must do a payment.
- Registered customers must enter their payment details like payment type, card details.
- After the payment is confirmed by bank or other trusted resources a report of the payment details is being emailed
- After the payment is confirmed Voter ID, Organizer ID and Sponsor ID are given.

Noun & Verb Analysis

(NOUNS)

<u>System Requirements</u>

- The System should function 24/7/365.
- Unregistered users can overview the system, to use the system, they must register with the system by providing details such as Name, Address, NIC, Email, contact.
- Registered customers are of five types called Voters, Nominees, Sponsors, Organizers, and System admin; where they can log into the system by entering the correct username and password.
- Voters should be able to vote for preferred nominee, check current results, check news and events, get subscribe to exclusives. Voters should be able to filter nominees by types and genres.
- Nominees should be able to check their results levels, check news and events, get subscribe to exclusives.
- Sponsors should be able to contact Organizers for events, advertise, check current results, check news and events.
- Organizers should be able to contact Nominees, Sponsors, System admins & check current results, check news and events. Organizers should be able to give details about events.
- System admins should confirm details about events, enter & update details about users, update results page, update news and events, remove outdated data.
- Registered customers must do a payment.
- Registered customers must enter their payment details like payment type, card details.
- After the payment is confirmed by bank or other trusted resources a report of the payment details is being emailed
- After the payment is confirmed Voter ID, Organizer ID and Sponsor ID are given and subscriptions will be given after relevant payments

Identified Classes

Registered user
Unregistered user

Nominee

Sponsor

organizer

System admin

payment

Subscriptions

Events

Votes

Results

Reasons for rejecting other nouns

- **Redundant: voter**
- **An Event or an operation:** exclusives. report
- **Outside scope of system:** System, Bank, trusted resources
- **Meta-language:**
- **An attribute:** Details (Name, Address, NIC, Email, Contact), Username, password,

  Nominee details (type, genre)

  Voter ID, Organizer ID and Sponsor ID

<u>**Noun & Verb Analysis**</u>

(VERBS)

The System should function 24/7/365.

Unregistered users can overview the system, to use the system, they must register with the system by providing details such as Name, Address, NIC, Email, contact.

Registered customers are of five types called Voters, Nominees, Sponsors, Organizers, and System admin; where they can log into the system by entering the correct username and password.

Voters should be able to vote for preferred nominee, check current results, check news and events, get subscribe to exclusives. Voters should be able to filter nominees by types and genres.

Nominees should be able to check their results levels, check news and events, get subscribe to exclusives.

Sponsors should be able to contact Organizers for events, advertise, check current results, check news and events.

Organizers should be able to contact Nominees, Sponsors, System admins & check current results, check news and events. Organizers should be able to give details about events.

System admins should confirm details about events, enter & update details about users, update results page, update news and events, remove outdated data.

Registered customers must do a payment.

Registered customers must enter their payment details like payment type, card details.

After the payment is confirmed by bank or other trusted resources a report of the payment details is being emailed

After the payment is confirmed Voter ID, Organizer ID and Sponsor ID are given.

- 5

Methods

- **Registered user**: log into the system by entering the correct username and password.

- **Unregistered user**: overview the system

  register with the system by providing details

- **Voters**: vote for preferred nominee, check current results

  check news and events get subscribe filter nominees

- **Nominee:** check their results levels check news and events

  subscribe

- **Sponsor:** contact Organizers

  Advertise

check current results

check news and events

- **organizer:** contact Nominees, Sponsors, System admins check current results check news and events give details

- **System admin**

  confirm details

enter & update details

update results page

update news and events

remove outdated data

- payment

- **Subscriptions: check payment details,**

  check user ID

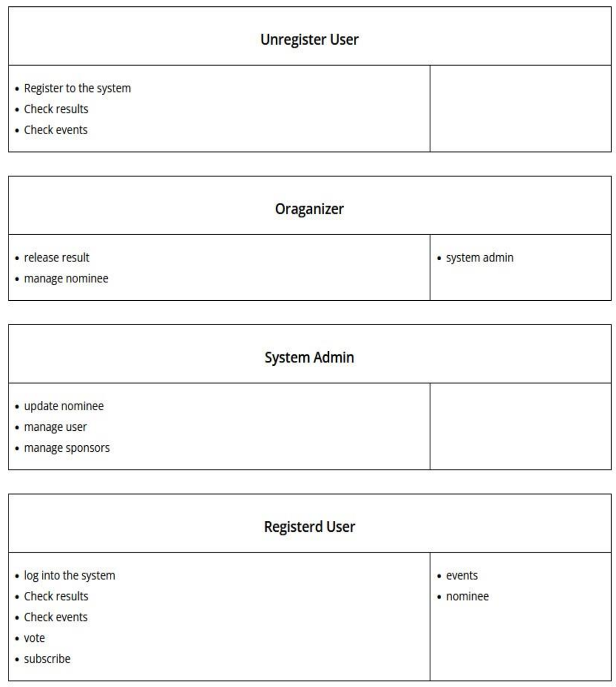
update user profiles

- **Events: enter event details**

  display event details

- **Votes**

CRC Cards

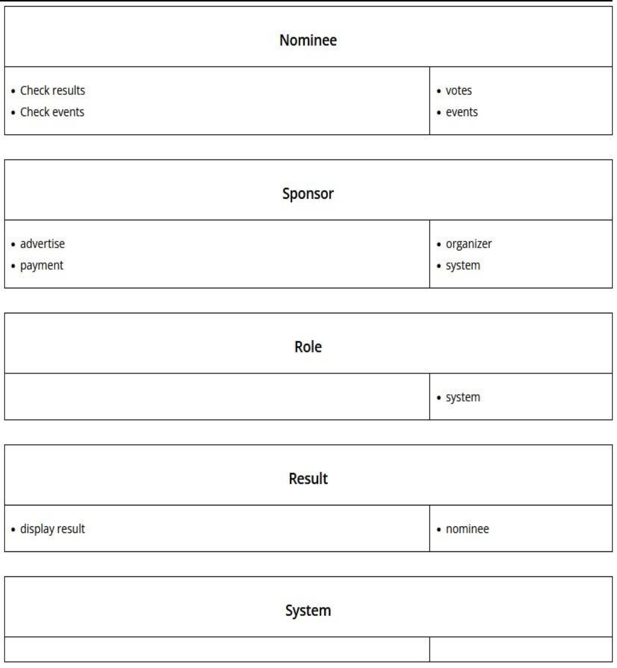


Unregister User	
• Register to the system	
• Check results	
• Check events	

Oraganizer	
• release result	• system admin
• manage nominee	

System Admin	
• update nominee	
• manage user	
• manage sponsors	

Registerd User	
• log into the system	• events
• Check results	• nominee
• Check events	
• vote	
• subscribe	



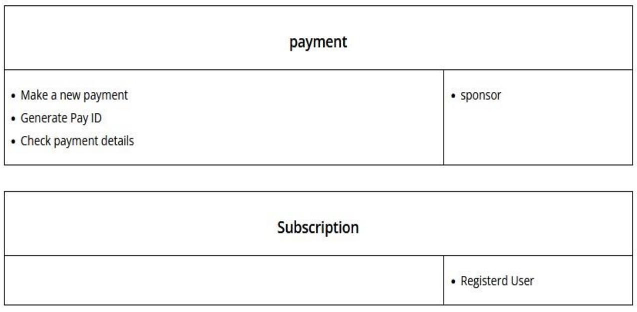
Nominee	
• Check results	• votes
• Check events	• events

Sponsor	
• advertise	• organizer
• payment	• system

Role	
	• system

Result	
• display result	• nominee

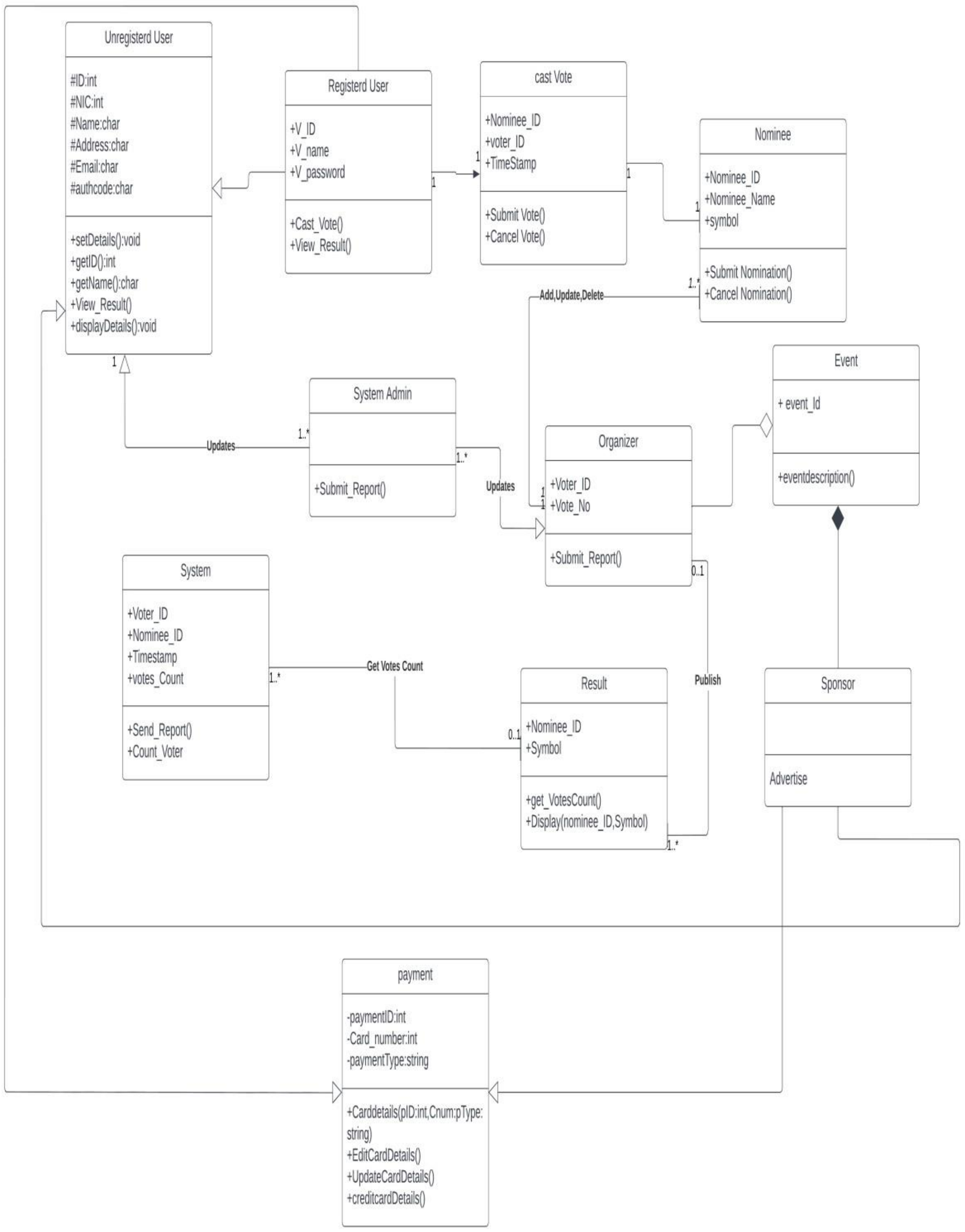
System	



payment	
• Make a new payment	• sponsor
• Generate Pay ID	
• Check payment details	

Subscription	
	• Registerd User

Class diagram



Codes – Header files

01.Unregistereduser.h

#include<iostream>

```
class unregisterduser

{

protected:


int userID; char

userName[30]; char

userAddress[30];

char userEmail[30];

int userTel; char

username[10]; char

password[10];


public:


            unregisterduser();

                  unregisterduser(int uID,const char uname[],const char uaddress[],const
char umail[],int utel[],const char pword[]);

                  void regisration();

         void cancelRegistration();

         void displaydetails();

  char  getName();       int    getid();

void View_Results();
```

```
                    ~unregisterduser();





          };
```

12

02.Registereduser.h

```
#include"Event.h"

#include"Nominee.h" #include

"unregisterduser.h" class registerd:

public unregisterduser

{

protected:          int
V_IDR;          char
Name[20];       char
Password[30];
//Class Relationship
public:

              registerd();              registerd(const char uname[25],

const char cpwd[8]);

                   void login();

                   void passwordvalidation();

                   void logout();


                    void Cast_Vote();

          void View_Result();

                    ~registerd();

};
```

03.Nominee.h

```
//#include"Event.h"
class Nominee
{
protected:                    int
Nominee_ID;            char
Nominee_Name[20]; char
Symbol[30];

public:
    Nominee();
    Nominee(int n_id,char n_name,char symbol[30]);
 void Submit_Nomination();             void
Cancel_Nomination();
            ~Nominee();
};
```

04.Sponsor.h

```
#include"organizer.h"
```

```
#include"registerd.h"
#include"system.h"
class Sponsor
{
protected:


public:
Sponsor();
            void Advertise();
            ~ Sponsor();
};
```

05,Organizer.h

```
#include"SystemAdmin.h"
class Organizer
{
```

```
private: int
Voter_ID; int
Voter_no;

public:
   Organizer();    Organizer(int
v_ID,int v_no);          void
Submit_Report();



           ~Organizer();
};
```

06.Systemadmin.h

```
class
SystemAdmin
{
private:  int
Authcode[10];

public:
SystemAdmin();    void
Submit_Report();
```

```
        ~SystemAdmin();
};
```

07.Payment.h

```
class payment
{
private:
int     payment_ID;   int
Card_number;      char
PaymentType[30];

public:
   payment();
    payment(int p_id,int c_number,char  P_Type[30]);
 void Carddetails();    void EditCardDetails();
char UpdateCardDetails();   void displaydetails();
            ~payment();
};
```

08.Events.h

```
#include "Nominee.h"
//#include "registerd.h"
#include "SystemAdmin.h"
//#include "sponsor.h"
class event
{
private:
            int eventid;     char
eventname[50];          char
eventdescription[50];
```

```
            Nominee* Nominee;

            //registerd* registerd;

            SystemAdmin* systemadmin;

            //sponsor* sponsor;


public:

            event();


            void eventdetails(int eventid, const char eventname);

 //const char eventdescription, Nominee* pNominee, registeruser*
pregisteruser,

            //systemadmin* psystemadmin, sponsor*

psponsor):registerd(){          void updateEventdetails();   void

displayEventdetails();          void checkEventdetails();

            ~event();


};


/*class Event

{

private: int

Event_ID;


public:

    Event();

    Event(int e_id);

            void eventdescription();

            ~Event();
```

```
};
*/
```

09.Votes.h

```
#include "Nominee.h"

#include "registeruser.h"

#include "systemadmin.h"

#include "sponsor.h"


#define SIZE


class vote

{

        private:                int

onOfregisteruser;
```

```
		registeruser* votedregisteruser[SIZE];


		Nominee* Nominee;

		systemadmin* systemadmin;

	sponsor* sponsor;


	public:

		vote();

		vote(int usid[], const char usname[], int onOfregisteruser,		const

char uspwd[], Nominee* pNominee, systemadmin* psystemadmin,

		sponsor* psponsor);

		void addvote(registeruser* votedregisteruser);

		void login();

	void updatevote();
```

```
void displayvote();
```

```
        };                    ~vote();
```

10.Results.h

```
class Result
{
private:
int  Nominee_ID; char
Symbol[30];

public:
   Result();
   Result(int N_ID,char symbol);     void
get_VoidCount();       void display(int
Nominee_ID,char Sysmbol);

           ~Result();
};
```

CPP files

1. Unregistereduser.cpp

```
#include <cstring> #include
"unregisterduser.h"
unregisterduser::unregisterduser()
{ userID = 0; strcpy(userName, "
"); strcpy(userAddress, " ");
strcpy(userEmail, " "); userTel =
0; strcpy(userName, "");
strcpy(password, "");
}

void unregisterduser::regisration()
{
}

void unregisterduser::cancelRegistration()
{
}
```

2. Registereduser.cpp

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

```
unregisterduser::unregisterduser()
{ userID = 0; strcpy(userName, "
"); strcpy(userAddress, " ");
strcpy(userEmail, " "); userTel =
0; strcpy(userName, "");
strcpy(password, "");
}

void unregisterduser::regisration()
{
}

void unregisterduser::cancelRegistration()
{
}
```

3. Nominee.cpp

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

```
unregisterduser::unregisterduser()
{ userID = 0; strcpy(userName, "
"); strcpy(userAddress, " ");
strcpy(userEmail, " "); userTel =
0; strcpy(userName, "");
strcpy(password, "");
}

void unregisterduser::regisration()
{
}

void unregisterduser::cancelRegistration()
{
}
```

4. Sponsor.cpp

```
#include <iostream>
#include <cstring>
#include "system.h"

System :: System(){

} void
System::Send_Report(){

} void
System::votes_Count(){
}
```

5. organizer.cpp

```
#include <iostream>
#include <cstring>
#include "organizer.h"

Organizer :: Organizer(){

}
void Organizer:: Submit_Report(){


}
```

}

6. System admin.cpp

```
#include <iostream>
#include <cstring>
#include "SystemAdmin.h"

SystemAdmin::SystemAdmin(){

}
void SystemAdmin::Submit_Report(){

}
```

7. payment.cpp

```
#include "payment.h" #include
<cstring>
payment::payment() {
payment_ID = 0;
strcpy(PaymentType, "");
Card_number = 0;
}
 payment:: payment(int id, const char type[], int CNumber)
{
payment_ID = id;
strcpy(PaymentType, type);
Card_number = CNumber;
}
void  payment::Carddetails()
{ }
void  payment::EditCardDetails()
{ }
void  payment::displaydetails()
```

```
{
}
```

8. Events.cpp

```
#include <iostream>
#include <cstring>
#include "Event.h"

event::event(){

}
void event::displayEventdetails(){

}
void event::updateEventdetails(){

}
void event::checkEventdetails(){

}
```

9. Results.cpp

```
#include <iostream>
#include <cstring>
#include "result.h"

Result :: Result(){

}
void Result:: get_VoidCount(){

} void
Result::display(){

}
```

<u>Main program</u>

```
#include <iostream>

#include "unregisterduser.h"

#include "Nominee.h"

#include "organizer.h"

#include "payment.h"

#include "registerd.h"

#include "result.h"

#include "sponsor.h"

#include "system.h"

#include "SystemAdmin.h"

#include "Event.h"


using std::cout;

using std::cin;

using std::endl;
```

```
int main()

{

   cout<<"Welcome to the Voting System"<<endl;

   //Registereduser Class Object

registerd* registerd;


   //Unregistereduser Class Object

unregisterduser* UnregisteredUser;    //Payment

Class Object    payment* payment;


   //Event class Object

event* e1 = new event();


  //Result class Object

  Result* r1= new Result();


  //Sponsor class Object

  Sponsor* s1 = new Sponsor();


  //Organizer class Object

  Organizer* o1 = new Organizer();




//-----------------Method Calling-----------------
registerd->login(); registerd-
>logout();

UnregisteredUser->regisration();
UnregisteredUser->cancelRegistration();

payment->Carddetails(); payment-
>displaydetails(); e1->updateEventdetails(); e1-
>displayEventdetails(); e1->checkEventdetails();

r1->get_VoidCount();

s1->Advertise();

o1->Submit_Report();


return 0; }
```