Topic : Online Voting System for Award Nominations

Group no : 10

Campus : Malabe

Submission Date: 2022/ 05/ 20

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
IT21307294	D.G.A.S. Gamage	070 2456332
IT21312076	E.G.S.S. Gunasinghe	075 5697095
IT21309274	K.V.A.M. Ramsith	071 3740633
IT21305078	M.G.D. Shehara	077 1482583
IT21308352	D.H.R.A. Jayathilaka	070 1804296

(1) Requirements Analysis

The "Gold Star" website provides an online voting system for award nominations. In order to vote for a nominee first users must register into the system by entering the required details. To register into the system users must provide their first name, last name, email address, phone number, role, DOB, age, gender and password, then the system will automatically generate a user ID for each registering user. Then a registered user can vote for a desired nominee. A registered user can log in to the system by giving email address and password. Users can create an account, modify or delete information within the created account

Registered users can vote for a nominee by selecting the desired nominee, nominated under each award and cast their vote by clicking the vote now button, then the system will display a successful message. Visitors can view the latest results with live updates. Users can search the content and Users can give feedbacks also Users can report an issue. If user expect to change their vote, website facilitates to cancel the vote and cast their vote for another nominee before the deadline. Anyone can provide their feedback regarding the website. User can view the home page which provides the Top 3 voted songs, latest results and voting deadline count down. Nominees' page provides all the nominated artists. Songs page provides all the nominated songs and also users can see the previous winners from the winner's page.

Administrator should provide admin ID and password to log into the system. Administrator can modify, create and delete nominees. Administrator can accept valid voters and remove invalid voters. Administrator can create new awards, edit current awards and remove unnecessary awards. Administrator can manage feedbacks and ratings. Administrator can generate reports and publish reports. Administrator can upload content.

Attributes

First name

Last name

Email address

Phone number

Role

DOB

Age

Gender

Password

User ID

Results

Ratings

Issues

Deadline

Admin ID

Out of scope

Gold Star

Website

Online voting system

System

Content

Redundant

User

Registered user

Visitors

Identified Classes

Voter

Unregistered User

Songs

Administrator

Winner

Award

Feedback

Reports

Account

Nominee

CRC Cards

Class Name: Voter	
Responsibilities	Collaborations
Enter/ Update details	Account
Delete an account	Account
Vote for nominee	Nominee
Give feedbacks and ratings	Feedback
Report an issue	Admin
Display results	Report

Class Name: Unregistered User	
Responsibilities	Collaborations
Register	
Give feedbacks and ratings	Feedback

Class Name: Songs	
Responsibilities	Collaborations
Store/Update song details	Admin
Delete songs	Admin
Display songs details	

Class Name: Administrator	
Responsibilities	Collaborations
Manage nominees	nominee
Manage voters	voter
Manage reports	report
Manage awards	award
Manage issues	Nominee/ Voters

Class Name: Winner	
Responsibilities	Collaborations
Store/Update/Delete winner details	Admin
Display winner details	

Class Name: Award	
Responsibilities	Collaborations
Store/Update award details	Admin
Delete award details	Admin
Display award details	

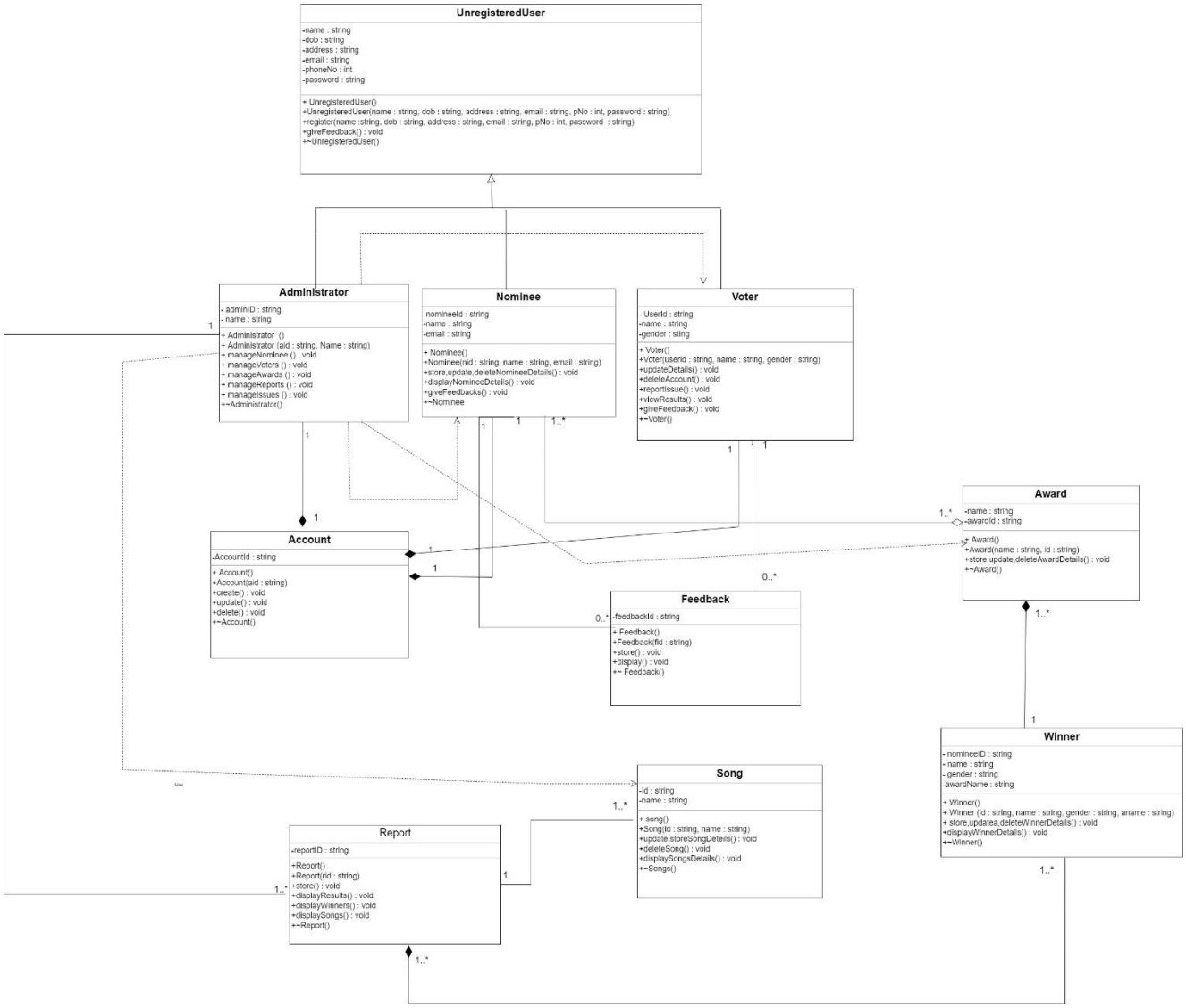
Class Name: Account	
Responsibilities	Collaborations
Create/Update/Delete account	Admin / Voter/ Registered User

Class Name: Feedback	
Responsibilities	Collaborations
Store Feedbacks	Voter/ Unregistered User/ Nominee
Display feedbacks	

Class Name: Report	
Responsibilities	Collaborations
Store details	
Display results	
Display winners	winner
Display songs	song

Class Name: Nominee	
Responsibilities	Collaborations
Store/Update/Delete nominees	Admin
Display nominee details	
Give feedbacks	Feedback

Exercise 1


UnregisteredUser
-name : string
-dob : string
-address : string
-email : string
-phoneNo : int
-password : string
+ UnregisteredUser()
+UnregisteredUser(name : string, dob : string, address : string, email : string, pNo : int, password : string)
+register(name :string, dob : string, address : string, email : string, pNo : int, password : string)
+giveFeedback() : void
+~UnregisteredUser()
Administrator
- adminID : string
- name : string
+ Administrator ()
+ Administrator (aid : string, Name : string)
+ manageNominee () : void
+ manageVoters () : void
+ manageAwards () : void
+ manageReports () : void
+ manageIssues () : void
+~Administrator()
Nominee
-nomineeId : string
-name : string
-email : string
+ Nominee()
+Nominee(nid : string, name : string, email : string)
+store,update,deleteNomineeDetails() : void
+displayNomineeDetails() : void
+giveFeedbacks() : void
+~Nominee
Voter
- UserId : string
-name : string
-gender : sting
+ Voter()
+Voter(userId : string, name : string, gender : string)
+updateDetails() : void
+deleteAccount() : void
+reportIssue() : void
+viewResults() : void
+giveFeedback() : void
+~Voter()
Award
-name : string
-awardId : string
+ Award()
+Award(name : string, id : string)
+store,update,deleteAwardDetails() : void
+~Award()
Account
-AccountId : string
+ Account()
+Account(aid : string)
+create() : void
+update() : void
+delete() : void
+~Account()
Feedback
-feedbackId : string
+ Feedback()
+Feedback(fid : string)
+store() : void
+display() : void
+~ Feedback()
Winner
- nomineeID : string
- name : string
- gender : string
-awardName : string
+ Winner()
+ Winner (id : string, name : string, gender : string, aname : string)
+ store,updatea,deleteWinnerDetails() : void
+displayWinnerDetails() : void
+~Winner()
Song
-Id : string
-name : string
+ song()
+Song(Id : string, name : string)
+update,storeSongDeteils() : void
+deleteSong() : void
+displaySongsDetails() : void
+~Songs()
Report
-reportID : string
+Report()
+Report(rid : string)
+store() : void
+displayResults() : void
+displayWinners() : void
+displaySongs() : void
+~Report()

Student ID Number	Contribution
IT21307294	Unregistered User / Song
IT21312076	Administrator / Feedback
IT21309274	Nominee / Report
IT21305078	Voter / Award
IT21308352	Account / Winner

Exercise 2

//unregisteredUser.h file

```
//Name - D.G.A.S.Gamage

//Student Id No-IT21307294


#include"feedback.h"

#include"administrator.h"

#include"voter.h"

#include"nominee.h"

#include"unregisteredUser.h"


class UnregisteredUser

{

protected:

        char name[40];

        char dob [10];

        char address [50];

        char email [35];

        int phoneNo;

        char password [30];

public:

        UnregisteredUser();

        UnregisteredUser(const char unName[], const char unDob[], const char unAddress[], const
char unEmail[],int unPhNo, char const unPassword[]);

        void register();

        void giveFeedback();

        ~UnregisteredUser();
```

```
};
```

//unregisteredUser.cpp file

```
//Name - D.G.A.S.Gamage
//Student Id No-IT21307294
#include "unregisteredUser.h"
#include <iostream>
#include <cstring>
#include"feedback.h"
#include"administrator.h"
#include"voter.h"
#include"nominee.h"
using namespace std;
UnregisteredUser::UnregisteredUser()
{
  strcpy(name, "");
  strcpy(dob, "");
        strcpy(address,"");
        strcpy(email, "");
        phoneNo = 0000000000;
        strcpy(password, "");
}
UnregisteredUser::UnregisteredUser(const char unName[], const char unDob[], const char
unAddress[],const char unEmail[], int unPhNo, const char unPassword[])
{
  strcpy(name, unName);
  strcpy(dob, unDob);
```

```
        strcpy(address, unAddress);
        strcpy(email, unEmail);
        phoneNo = unPhNo;
        strcpy(password, unPassword);
}
void UnregisteredUser::register()
{
}
void UnregisteredUser::giveFeedback()
{
}
```

//song.h file

```
// Name- D.G.A.S.Gamage
//Student Id No-IT21307294
#pragma once
#include "song.h"
#include "admin.h"
#include "report.h"
class Song
{
private:
        int Id;
        char name[20];
public:
        Song();
         Song(int id, const char Name[20]);
```

```
        void UpdateSongDetails();
        void deletesong();
        void displaysong();
        ~Song();
```

//song.cpp :

```
//Name - D.G.A.S.Gamage
//Student Id No-IT21307294
 #include <iostream>
#include <cstring>
#include "song.h"
#include "admin.h"
#include "report.h"
using namespace std;
Song::Song()
{
   Id = 0;
 strcpy(name, "");
}
Song::Song(int id, const char Name[20])
{
   Id = id;
   strcpy(name, Name);
}
void Song::UpdateSongDetails()
{
}
```

```
void Song::deletesong()
{
}
void Song::displaysong()
{
}
```

//administraor.h file

```
//Name - E.G.S.S. Gunasinghe
//Student ID No - IT21312076
#include "unregistredUser.h"
#include"account.h"
#include"report.h"
#include"voter.h"
class Administrator
{
private:
        char adminId[10];
        char name[20];
public:
        Administrator();
        Administrator(const char id[], const char adminName[]);
        void manageNominee();
        void manageVoters();
        void manageAwards();
        void manageReports();
        void manageIssues();
```

```
        void manageSongs();


        ~Administrator();
};
```

// administrator.cpp :

```
//Name - E.G.S.S. Gunasighe
//Student ID - IT21312076
#include <iostream>
#include<cstring>
#include "administrator.h"
#include"report.h"
#include"voter.h"
using namespace std;
Administrator :: Administrator()
{
   strcpy(adminId, "");
   strcpy(name, "");
}
Administrator::Administrator(const char id[], const char adminName[])
{
   strcpy(adminId, id);
   strcpy(name, adminName);
}
void Administrator::manageNominee()
{
}
```

```
void Administrator::manageVoters()
{
}
void Administrator::manageAwards()
{
}
void Administrator::manageReports()
{
}
void Administrator::manageIssues()
{
}
Void Administrator::manageSongs()
{
}
Administrator::~Administrator()
{
}
```

//feedback.h file

```
//Name - E.G.S.S.Gunasinghe
//Student Id No- IT21312076
#include"nominee.h"
#include"voter.h"
#include"feedback.h"
class Feedback
{
```

```
private:
        int Feedbackid;
public:
        Feedback();
        Feedback(int fid);
        void store();
        void display();
        ~Feedback();
};
```

// feedback.cpp

```
//Name- E.G.S.S.Gunasinghe
//Student ID Number- IT21312076
#include <iostream>
#include<cstring>
#include "feedback.h"
#include"nominee.h"
#include"voter.h"
using namespace std;
feedback::feedback() {
        fid = 0;
}
feedback::feedback(int fid)
{
   feedbackid = fid;
}
void feedback::store()
```

```
{

}

void feedback::display()

{

}
```

//report.h

```
//Name- K.V.A.M.Ramsith

//Student ID Number-IT21309274

#pragma once

#include "report.h"

#include"song.h"

#include"administrator.h"

class Report

{

private:

        int reportId;

public:

        Report();

        Report(int rid);

        void store();

        void displayResults();

        void displayWinners();

        void displaysongs();


        ~Report();

};
```

//report.cpp

```
//Name-K.V.A.M.Ramsith
//Student ID Number-IT21309274
#include <iostream>
#include<cstring>
#include "report.h"
#include"song.h"
#include"administrator.h"
using namespace std;
Report::Report()
{
   reportId = 0;
}
Report::Report(int rid)
{
   reportId = rid;
}
void Report::displayResults()
{
}
void Report::displayWinners()
{
}
void Report::displaysongs()
{
}
```

//nominee.h

```
//Name- K.V.A.M.Ramsith
//Student ID Number-IT21309274
#pragma once
#include "nominee.h"
#include"account.h"
#include"feedback.h"
#include"award.h"
class Nominee
{
private:
        int NomineeId;
        char name[20];
        char email[20];
public:
        Nominee();
        Nominee(int Nomineeid, const char Name[20], const char email[20]);
        void StoreUpdateDeletenomineeDetails();
        void givefeedbacks();
        void displayNomineeDetails();
       ~Nominee();
```

//nominee.cpp

```
//Name-K.V.A.M.Ramsith
//Student ID Number-IT21309274
#include <iostream>
#include<cstring>
#include "nominee.h"
#include"account.h"
#include"feedback.h"
#include"award.h"
using namespace std;
Nominee::Nominee()
{
   NomineeId = 0;
   strcpy(name, "");
   strcpy(email, "");
}
Nominee::Nominee(int nid, const char Name[20], const char Email[20])
{
   NomineeId = nid;
   strcpy(name, Name);
   strcpy(email, Email);
}
void Nominee::StoreUpdateDeletenomineeDetails()
{
}
void Nominee::givefeedbacks()
{
```

```
}
void Nominee::displayNomineeDetails()
{
}
```

//voter.h file

```
//Name-M.G.D.Shehara
//Student Id No - IT21305078
#include"voter.h"
#include "feedback.h"
#include"nominee.h"
class Voter
{
private:
        int Userid;
        char name[20];
        char gender[10];
public:
        voter();
        voter(int userid, char Name[20], char Gender[10]);
        void updateDetails();
        void deleteAccount();
        void reportissue();
        void viewResults();
        void giveFeedback();
      ~voter();
};
```

//voter.cpp :

```
//Name-M.G.D.Shehara
//Student Id No - IT21305078
#include <iostream>
#include<cstring>
#include "voter.h"
#include"feedback.h"
#include"nominee.h"
using namespace std;
voter::voter(){
	Userid = 0;
	strcpy(name, "");
	strcpy(gender, "");
}
voter::voter(int userid, char Name[20], char Gender[10])
{
	Userid = userid;
	strcpy(name, Name);
	strcpy(gender, Gender);
}
void voter::updateDetails()
{
}
void voter::deleteAccount()
{
}
void voter::viewResults();
```

```
{

}

void voter::giveFeedback();

{

}
```

//award.h

```
//Name-M.G.D.Shehara

//Student Id No - IT21305078

#include "award.h"

#include"administrator.h"

#include"winner.h"

class Award

{

private:

        char awardId[11];

        char name[28];

public:

        Award();

        Award(const char id[], const char awardName[]);

        void storeawarddetails();

        void Updateawarddetails();

        void Deleteawarddetails();

        void Displayawarddetails();


        ~Award();

};
```

//award.cpp file

```
//Name-M.G.D.Shehara
//Student Id No - IT21305078
#include <iostream>
#include<cstring>
#include "award.h"
#include"administrator.h"
#include"winner.h"
using namespace std;
Award :: Award()
{
   strcpy(awardId, "");
   strcpy(name, "");
   }
      Award :: Award(const char id[], const char awardName[])
{
   strcpy(awardId, id);
   strcpy(name, awardName);
}
void Award:: storedetails()
{
}
void Award:: updateawarddetails()
{
}
void Award:: deleteawarddetails()
{
```

```
}
void Award::displayawarddetails()
{
}
Award :: ~Award()
{
}
```

//winner.h file

```
#include "winner.h"
#include "report.h"
#include "award.h"
class Winner
{
private:
        char nomineeId[11];
        char name[28];
        char nomineeGender[12];
        char aName[6];
public:
        nominee();
       nominee (const char nomineeid[], const char name[],const char nomineeGender[],const char aName[]);
        void storedetails();
        void Updatedetails();
        void DeleteWinnerdetails();
        void DisplayWinnerdetails();
      ~nominee();
```

```
};
```

//winner.cpp file

```
//Name-Hansali Ranradhi Jayathilaka
//IT Number-IT21308352
#include <iostream>
#include<cstring>
#include "winner.h"
#include "report.h"
#include "award.h"
using namespace std;
Winner::Winner()
{
   strcpy(nomineeId, "");
   strcpy(name, "");
   strcpy(nomineeGender, "");
   strcpy(aName, "");
}
Winner::Winner(const char nomineeid[], const char nomineeName[], const char nomineegender[],
const char awardName[])
{
   strcpy(nomineeId, id);
   strcpy(name, nomineeName);
   strcpy(nomineeGender, gender);
   strcpy(aName, awardName);
}
void Winner::storeDetails()
{
```

```
}

void Winner::updateDetails()
{
}
void Winner::deleteWinnerdetails()
{
}
void Winner::displayWinnerdetails()
{
}
Winner :: ~Winner()
{
}
```

//account.h file

```
//Name-Hansali Ranradhi Jayathilaka
//IT Number-IT21308352

#include "unregistredUser.h"
#include "account.h"
#include "nominee.h"
#include "voter.h"
#include "administrator.h"

class Account
{
```

```
private:
        char accountId[8];

public:
        Account ();
        Account (const char id[]);
        void Createaccount();
        void updateaccount();
        void Deleteaccount();

        ~Account();
};
```

// account.cpp file

```
//Name-Hansali Ranradhi Jayathilaka
//IT Number-IT21308352
#include <iostream>
#include <cstring>
#include "unregistredUser.h"
#include "account.h"
#include "nominee.h"
#include "voter.h"
#include "administrator.h"
using namespace std;
Account :: Account()
{
```

```
   strcpy(accountId, "");


}
Account::Account(const  char id[])
{
   strcpy(accountId, id);
   }
void Account :: Create account ()
{
}
void Account :: update account ()
{
}
void Account :: Delete account ()
{
}



Account:: ~Account()
{
}
```

// main .cpp file

```
#include <iostream>

using namespace std;

#include "administrator.h"

#include "unregisteredUser.h"

#include "voter.h"

#include "nominee.h"

#include "account.h"

#include "feedback.h"

#include "award.h"

#include "winner.h"

#include "song.h"

#include "report.h"


int main(void)

{

	Administrator* admin;

	UnregisteredUser* unregUse;

	Voter* voter;

	Nominee* nomi;

	Account* acc;

	Feedback* feed;

	Award* award;

	Winner* winner;

	Song* song;

	Report* report;

	admin = new Administrator("001", "Amal");
```

```
        unregUse = new UnregisteredUser("Malaka","1999/03/12", "No122/Kandy road/
Kalaniya","malaka123@gamil.com", 0711232234  , "mala234@", );

        voter =new  Voter("V001","Malaka","Male");

        nomi = new Nominee("N001", "Wenura","wenura12@gmail.com");

        acc = new Account("ACC001");

        feed = new Feedback("F001");

        award = new Award("A001", "Best Male Artist");

        winner = new Winner("W001","Saman", "Male", "Best Male Artist");

        song = new Song("S001", "Ciao Malli");

        report = new Report("R001");


        delete admin;

        delete unregUse;

        delete voter;

        delete nomi;

        delete acc;

        delete feed;

        delete award;

        delete winner;

        delete song;

        delete report;


        return 0;


}
```