Topic : Online Job Portal

Group no : MLB_01.1_04

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**
IT21058950	J.A.P.M. Jayamanne	0716606575
IT21060830	R.A.N.M. Rajapaksha	0767476847
IT21023682	P.A.D.L. Anjalee	0764126806
IT21073182	U.M.W. Bandara	0710144320
IT20631406	R. Dinushan	0763846426

Part 1

User Requirements

- New job seekers and new employers who enter the website for the first time are identified as guest users and they can search and view the posted jobs.
- To apply for a posted job, a new job seeker must register to the website by entering their details such as Name, email address, phone number and date of birth
- New employers should also register to the website by entering their details such as name, email, company name, address if they want to post jobs
- Users will get their own unique user id after registering
- Registered users also can log into the system and view or update user details
- Registered users are of two types called registered job seekers and registered employers
- Registered Job seekers can apply for a posted job by selecting a job and submitting an application containing their qualifications as well as the date
- Application should also get a unique application id from the system
- Registered job seekers can give feedbacks on posted jobs
- Registered employers can view feedbacks given by job seekers
- Registered employers should be able to post new jobs to the system with job type and the job title
- All the job details should be verified by the staff when posting a new job
- System should generate a unique id for the job after confirming
- Staff can login to the system and manage registered users

Identified Classes

- Guest user
- Registered user
- Staff
- Registered job seeker
- Registered employer
- Job
- Application
- Feedback

CRC Cards

Guest User	
Responsibility	**Collaborators**
View posted jobs	Job
Register to the system	

Registered User	
Responsibility	**Collaborators**
Login to the system	
View posted jobs	Jobs
View and update user details	

Staff	
Responsibility	**Collaborators**
Login to the system	
Add or remove users	
Verify job details	Job

Registered job seeker	
Responsibility	**Collaborators**
Submit applications for jobs	Application
Login to the system	Registered user
Give feedbacks	Feedback

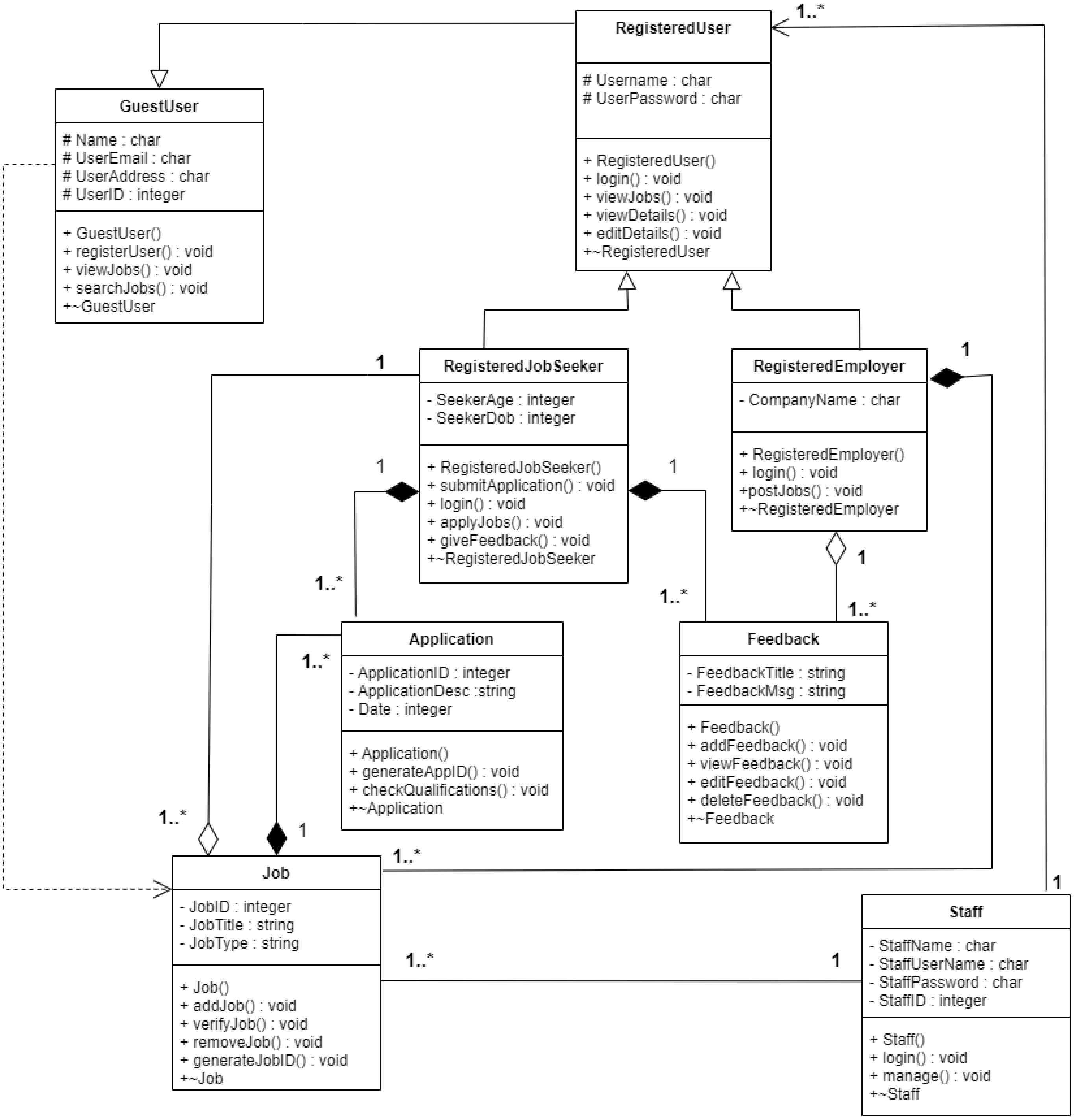
Registered employer	
Responsibility	**Collaborators**
Post new jobs	Job
Login to the system	Registered user
View feedback	Feedback

Job	
Responsibility	**Collaborators**
Add job details	Registered employer
Generate job ID	

Application	
Responsibility	**Collaborators**
Generate application ID	
Check qualifications	

Feedback	
Responsibility	**Collaborators**
Add feedback	Registered job seeker
View feedback	Registered employer

Class Diagram



Class header files

GuestUser.h

```
#include "GuestUser.h"

class GuestUser

{

protected:

	int UserID[];

	char Name[20];

	char UserAddress[30];

	char UserEmail[30];


public:

	GuestUser();

	GuestUser(int puserid, const char pname[], const char
	puserAddress[], const char puserEmail[]);

	void searchJobs(Job* pjb);

	void registerUser();

	void viewJobs();

	~GuestUser();

};
```

RegisteredUser.h

```
#include"GuestUser.h"

class RegisteredUser :public GuestUser

{

protected:

     char Username[10];

     char UserPassword[10];


public:

     RegisteredUser();

     RegisteredUser(const char pusername[], const char
     puserPassword[], int puserid, const char puserName[], const
     char puserAddress[], const char puserEmail[]);

     void viewDetails();

     void login();

     void editDetails();

     ~RegisteredCustomer();

};
```

Application.h

```
#include "RegisteredjobSeeker.h"

class Application

{

protected:

    int ApplicationID;
    int ApplicationDesc;
    int Date;
public:
    Application();
    Application(int pAppID, const int pAppDesc, int pDate);
    void generateAppID();
    void checkQualifications();
    ~Application();
};
```

Feedback.h

```
#include "RegisteredjobSeeker.h"
#include "RegisteredEmployer.h"
class Feedback
{
protected:
    int FeedbackTitle;
    int FeedbackMsg;

public:
    Feedback();
    Feedback(int pFeedbackTitle, int pFeedbackMsg);
    void addfeedback();
    void viewFeedback();
    void editFeedback();
    void deleteFeedback();
    ~Feedback();
};
```

RegisteredJobSeeker.h

```
#include"RegisteredUser.h"
class RegisteredJobSeeker : public RegisteredUser
{
protected :
      int SeekerAge [10];
      int SeekerDob [20];

public :
      RegisteredJobSeeker();
      RegisteredJobSeeker(int pseekerAge[],int
      pseekerDob[], const char puserName[], const
      char puserPasswor[]),
      void submitAppilication();
      void login();
      void applyJobs();
      void giveFeedback();
      ~RegisteredJobSeeker();
};
```

RegisteredEmployer.h

```
#include "RehisteredUser"
class RegisteredEmployer : public RegisteredUser
{
protected :
	char CompanyName [30];

public :
	RegisteredEmployer();
	RegisteredEmployer(const char pCompanyName[], const
	char pusername[], const char puserPassword[]);
	void login();
	void postJobs();
	~RegisteredEmployer();
};
```

staff.h

```
#include "job.h"
class Staff
{
private:
int staffID;
char staffName[20];
char staffUsername[20];
char staffPassword[20];
public:
Staff();
Staff(int pstaffID, const char pstaffName[], const char
pstaffUsername[], const char pstaffPassword[]);
void login(const char stfUsername, const char stfPassword );
void manage();
~Staff();
};
```

job.h

```
#include "Registeredemployer.h"
class job
{
protected:
    int JobID;
    char JobTitle;
    char JobType;
public:
    Job();
    Job(int JobID, const char JobTItle, char JobType);
    void addJob();
    void verifyJob();
    void removeJob();
    void generateJobID();
    ~Job();
};
```

Class cpp files

GuestUser.cpp

```
 #include <GuestUser.h>
 #include "cstring"
 GuestUser :: GuestUser()
 {
     user_ID = 0;
     strcpy(GuestUser_Name,"");
     strcpy(GuestUser_Address,"");
     strcpy(GuestUser_Email,"");
 }
 void GuestUser::searchJobs(int userID, const char UserName, const
char userAddress, const char userEmail)
 {


 }
 void GuestUser::registerUser()
 {


 }
 void GuestUser::viewJobs()
 {


 }
```

RegisteredUser.cpp

```
 #include <RegisteredUser.h>

 #include "cstring"


 RegisteredUser :: RegisteredUser()

 {

      strcpy(regUser_Name,"");

      strcpy(regUser_Password,"");

      regUser_ID=0;

      strcpy(regUser_Address,"");

      strcpy(regUser_Email,"");

 }

 void RegisteredUser::viewDetails(const char userName, const char
userPassword)

 {


 }

 void RegisteredUser::login()

 {


 }

 void RegisteredUser::editDetails()

 {


 }
```

Application.cpp

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

Application :: Application()
 {
     AppID=0;
     strcpy(App_Description,"");
     strcpy(App_Date,"");
 }
 void Application::genarateAppID(int AppID, int Date)
 {

 }
 void Application::checkQualifications()
 {

 }
```

```
}
```

Feedback.cpp

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

Feedback :: Feedback(int fb_title, int fb_name)
{
      FeedbackTitle = fb_title;
      feedbackMessage = fb_msg;
}

void Feedback :: addFeedback()
{


}
void Feedback :: viewFeedback()
{


}
void Feedback :: editFeedback()
{


}
void Feedback :: deleteFeedback()
{


}
```

RegisteredJobSeeker.cpp

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

RegisteredJobSeeker :: RegisteredJobSeeker()
{
      strcpy(pseeker_Age,"");
      strcpy(pseeker_Dob,"");
      strcpy(pseeker_Name,"");
      strcpy(pseeker_Password,"");
}

void RegisteredJobSeeker :: login(const char puserName, const char
puserPassword)
{


}
void RegisteredJobSeeker :: submitApplication()
{


}
void RegisteredJobSeeker :: applyJobs()
{


}
void RegisteredJobSeeker :: giveFeedback()
{


}
```

RegisteredEmployer.cpp

```
 #include <RegisteredEmployer.h>

 #include "cstring"


RegisteredEmployer :: RegisteredEmployer()

{

      strcpy(RegisteredEmployer_Company Name,"");

      strcpy(RegisteredEmployer_Name,"");

      strcpy(RegisteredEmployer_Password"");

}


void RegisteredEmployer :: login(const char puserName, const char
puser password)

{


}

void RegisteredEmployer :: postJobs()

{


}
```

Staff.cpp

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

Staff :: Staff()
{
      staff_ID = 0;
      strcpy(staff_Name,"");
      strcpy(staff_Username,"");
      strcpy(staff_Password,"");
}

void Staff :: login(const char puserName, const char puser password)
{

}

void Staff :: manage()
{

}
```

Job.cpp

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

Job :: Job()
{
     job_ID = 0;
     strcpy(job_Title,"");
     strcpy(job_Type,"");
}
void Job :: addJobs(const char jobTitle, const jobType)
{

}
void Job :: verifyJob()
{

}
void Job :: removeJob()
{

}
void Job :: grnarateJobID()
{

}
```

Main Programme

```
int main()

{

     GuestUser*guser;

     guser = new GuestUser(401, "Bhanuka Silva","221/B, Galle
road, Katunayake", "bhanuska123@gmail.com");

     guser -> searchJobs("Job type", "Job title");

     guser -> viewJobs("Part time jobs", "Full time jobs");


     RegisteredUser*ruser;

     ruser = new RegisteredUser("Dilshan Bandara", "jp452", 403,
"128/2, Kandy road, Malabe", "dilshan34@gmail.com");

     ruser -> viewDetails("Jobtypes", "Description");

     ruser -> login("user name", "password");


     Staff*stf;

     stf = new Staff(ST201, "Manager", "M01", "123jps");

     stf -> login("user name", "password");


     RegistredJobSeeker*rjs;

     rjs = new RegisteredJobSeeker(21, 5/14/2001, "Kevin
Silva");

     rjs -> login("user name", "password");

     rjs -> submitApplication();

     rjs -> applyJobs("Full time jobs", "Part time jobs");

     rjs -> giveFeedback("Great job to be in");
     RegisteredEmployer*rEmp;
     rEmp = new RegisteredEmployer("JOBSTAR pvt.ltd", "Kamal
Perera", "RE362");
     rEmp -> login("user name", "password");
     rEmp -> postJobs("Project manager", "Full time");


     Job*jb
     jb = new Job(JB12, "Your dream job", "Full time job");
     jb -> addJob("Full time", "Developer");
     jb -> genarateJobID("JB12");


     Appication*app;
     app = new Application("AP856", "description", 3/14/2022);
     app -> genarateAppID("AP856");
     app -> checkQualifications("working ewxperience",
"communication skiils");


     FeedBack*fbc;
     fbc = new FeedBack("Good service", "Thank you for being
honest with us");
     fbc -> addFeedback();
     fbc -> viewFeedback();


     delete guser, ruser, stf, rjs, rEmp, jb, app;


}
```