Topic : Online Examination System For Employees

Group no : MLB_08.02_12

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**
IT21204852	Ezhilkumaran.G	0742124742
IT21209420	D.E.H Mallawaarachchi	0717151977
IT21211782	Wijewaradana S.P.M.G.D.	0763148962
IT21215988	Waseem M.I.M.	0774942950
IT21219498	Vithanage C.S.	0719915462

Contents

01. System Requirements

- All unregistered users can register to the system by providing details. Example: first name, last name, employee ID, username, password, NIC, date of birth.
- If members enter invalid credentials system shows, “invalid credentials”.
- Visitors can view the home page.
- Any member can log in to the website by providing login credentials.
- If registered users forgot their password, they can use forgot password option.
- Once user login to the system, the system validates their user credentials.
- All the registered examinees can make their feedback.
- Examinee can face the exam.
- Examinee register exam.
- Examinee can check exam results.
- Examinee can search for notices.
- Examinee can specify a method for each payment.
- In the system the exam supporter can check feedbacks.
- Exam supporter can update FAQ.
- Exam supporter can provide answers After analysing the problem.
- Manger can select exam type as entry-level exam, IT department exam, financial department exam, management department exam and create exam.
- Exam manager can schedule exam.
- Exam manager can update resources. Example: update create practice test.
- Exam manager can select qualify examinees.
- Admin can add and block users.
- Admin can generate reports such as list of examinees, list of qualified examinees, list of exam results.
- Admin can update notices.
- Admin can update user details.

02. Identified Classes

Unregistered users

Examinee

Exam

Exam results

Payment

Exam supporter

Exam manger

Resources

Admin

Reports

03. CRC cards

Class Name: Unregistered users	
Responsibility	**Collaborators**
Register to the system	
Search for notices	Admin

Class Name: Examinee	
Responsibility	**Collaborators**
Login to the system	
Provide feedback	Exam supporter
Register for exam	Exam, Payment
Check exam results	Exam results
Search for notices	Admin

Class Name: Exams	
Responsibility	**Collaborators**
Store created exams	Exam manager
Knows exam schedules	

Class Name: Exam results	
Responsibility	**Collaborators**
Knows exam results	

Class Name: Payment	
Responsibility	**Collaborators**
Provide payment methods	Examinee
Store payment details	
Validate payment	

Class Name: Exam supporter	
Responsibility	**Collaborators**
Provide answers for feedbacks	Examinee
Update FAQ	

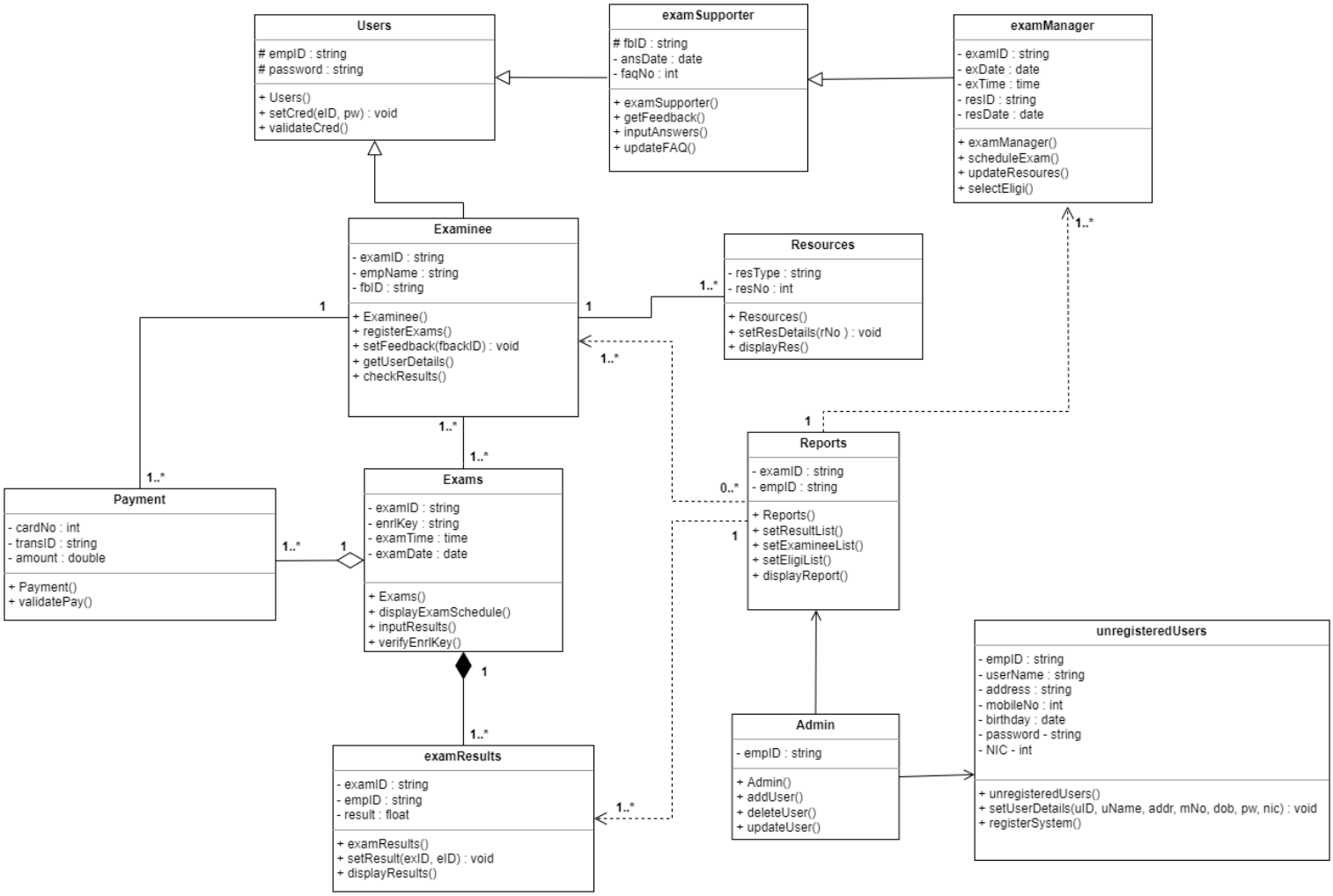
Class Name: Exam manager	
Responsibility	**Collaborators**
Create exam	
Schedule exam	Exams
Update resources	
Select qualified examinees	Exam results, Examinee

Class Name: Resources	
Responsibility	**Collaborators**
Store resources details	

Class Name: Admin	
Responsibility	**Collaborators**
Add user	Unregistered users
Block user	Examinee
Generate report	Reports
Update user details	Examinee
Update notices	

Class Name: Reports	
Responsibility	**Collaborators**
List of exam results	Exam results
List of qualified examinees	Exam manager
List of examinees	Examinee

04. Class diagram



05. Class header files

- Admin.h

```
//admin class
#include "unregisteredUsers.h"
#include "Reports.h"
#include <string>
#include <iostream>
using namespace std;

class Admin{
private:
  string empID;
  unregisteredUsers *ur1;
  Reports *r1;

public:
  Admin(){
    empID = "";
  }
  void addUser();
  void deleteUser();
  void updateUser(string eID);
  ~Admin(){
    cout<<"Admin deleted"<<endl;
  }
};
```

- Examinee.h

```
//Examinee class
#include <string>
#include <iostream>
using namespace std;


class Examinee : public Users{
private:
  string examID;
  string empName;
  string fbID;
  Resources *recs[10];
  Payment *p1[2];
  Exams *exm1[2];



public:
  Examinee(){
  examID="";
  empName="";
  fbID="";
}

  void registerExams(string pxamID,string pempName){};
  void setFeedback(string fbackID){};
  void getUserDetails();
  void checkResults();
  ~Examinee(){
  cout<<"examinee deleted"<<endl;
  }

};
```

- examManager.h

```
//examManager class
#include "examSupporter.h"
#include<string>
#include<iostream>
using namespace std;


class examManager: public examSupporter{
private:
string examID;
string date;
string time;
string resID;
string resDate;


public:
examManager(){
examID="";
date="";
time="";
resID="";
resDate="";
}
void sheduleExam(string pexamID,string pdate,string ptime);
void update(string pxamID,string pempName);
void selectEliglist();
~examManager(){
  cout<<"examManager deleted"<<endl;
  }

};
```

- examResults.h

```
//examResults class
#include<string>
#include<iostream>
using namespace std;

class examResults{
private:
string examID;
string empID;
float results;

public:
examResults(){
  examID="";
  empID="";
}
examResults(int r){
  results = r;
}
void setResults();
float getResults();
void displayResults();
~examResults(){
  cout<<"examResults deleted"<<endl;
  }
};
```

- Exams.h

```
//Exams class
#include "Examinee.h"
#include "examResults.h"
#include "Payment.h"
#include <iostream>
#include <string>
using namespace std;
#define SIZE 5

class Exams{
private:
  examResults *results[2];
  Payment *payment[SIZE];
  string examID;
  string enrlKey;
  int eDay;
  int eMonth;
  int eYear;
  int eHour;
  int eMinutes;
  Examinee *exminee[2];

public:
  Exams(){
    examID = "";
    enrlKey = "";
    eDay = 0;
    eMonth = 0;
    eYear = 0;
    eHour = 0;
    eMinutes = 0;
    results[0] = new examResults[50];
    results[1] = new examResults[35];

  }
  void addPayment(Payment *p1, Payment *p2)
{
  payment[0] = p1;
  payment[1] = p2;
}
  void addExaminee(Examinee *exmi);
  void displayExamSchedule();
  void inputResults();
  void verifyEnrlKey();
  ~Exams(){
    cout<<"Exams deleted"<<endl;
    for(int i = 0; i < 2; i++)
       delete results[i];
    }
};
```

- examSupporter.h

```
//examSupporter class
#include "Users.h"
#include <string>
using namespace std;

class examSupporter : public Users{
protected:
string fbID;

private:
string ansDate;
int faqNo;

public:
examSupporter(){
  fbID = "";
  ansDate = "";
  faqNo = 0;
}
void getFeedback();
void inputAnswers();
void updateFAQ();
~examSupporter(){
  cout<<"examSupporter deleted"<<endl;
}
};
```

- Payment.h

```
//Payment class
#include <string>
#include <iostream>
using namespace std;

class Payment{
private:
  int cardNo;
  string transID;
  double amount;
  Examinee *ex1;

public:
  Payment(){};
  Payment(int cdNo, double amt){
    cardNo = cdNo;
    amount = amt;
  }
  void validatePay();
  ~Payment(){
    cout<<"Payment deleted"<<endl;
  }
};
```

- Reports.h

```
//Reports class
#include<string>
#include "examResults.h"
#include "Examinee.h"
#include "examManager.h"
#include<iostream>
using namespace std;

class Reports{
private:
string examID;
string empID;
public:
Reports(){
examID="";
empID="";
}
Reports(string pexamID,string pempID){
  examID = pexamID;
  empID = pempID;
}
void setResultList(examResults *er);
void setExamineeList(Examinee *exami);
void setEligList(examManager *em);
void displayReport();
~Reports(){
  cout<<"reports deleted"<<endl;
  }
};
```

- Resources.h

```
//Resources class
#include <string>
using namespace std;

class Resources {
private :
  string resType;
  int resNo;
  Examinee *exmi2;

public :
  Resources(){
    resType = "";
    resNo = 0;
  };
  void setResDetails(int rNo);
  void displayRes();
 ~Resources(){
   cout<<"Resources deleted"<<endl;
 }

};
```

- unregisteredUsers.h

```
//unregisteredUsers class
#include <string>
#include <iostream>
using namespace std;

class unregisteredUsers{
private:
  string empID;
  string userName;
  string address;
  string password;
  int mobileNo;
  int NIC;
  int bDay;
  int bMonth;
  int bYear;

public:
  unregisteredUsers(){
    empID = "";
    userName = "";
    address = "";
    password = "";
    mobileNo = 0;
    NIC = 0;
    bDay = 0;
    bMonth = 0;
    bYear = 0;
  }
  void setUserDetails(string eID, string uName, string addr, string pw, string mNo, int NIC, int bD, int
bM, int bY);
  void registerSystem();
  ~unregisteredUsers(){
    cout<<"unregisteredUsers deleted"<<endl;
  }
};
```

- Users.h

```
//Users class
#include <string>
#include <iostream>
using namespace std;


class Users {
protected:
string empID;
string password;

public:
Users(){
  empID = "";
  password = "";
};
  void setCred(string eID,string pw);
  void validateCred();
  ~Users(){
  cout<<"Users deleted"<<endl;
};
};
```

06. Main program

```
//main program
#include "Exams.h"
#include "Examinee.h"
#include "Resources.h"
#include "examManager.h"
#include "Admin.h"
#include <iostream>

int main() {
  Admin *ad1 = new Admin();
  unregisteredUsers *urUser = new unregisteredUsers();
  Reports *RP = new Reports();
  Exams *ex1 = new Exams();
  Payment *pay1 = new Payment(454585, 8000.00);
  Payment *pay2 = new Payment(425178, 30300.00);
  Examinee *exmi1 = new Examinee();
  Resources *res = new Resources();
  examSupporter *exsup1 = new examSupporter();
  Users *usr1 = new Users();
  examManager *exma1 = new examManager();
  examSupporter *exsup2 = new examSupporter();

  RP -> setExamineeList(examinee1);
  RP -> setResultList(examResults1);
  RP -> setEligList(examManager1);
  ex1 -> addPayment(pay1, pay2);


  delete ad1;
  delete urUser;
  delete RP;
  delete ex1;
  delete pay1;
  delete pay2;
  delete exmi1;
  delete res;
  delete exsup1;
  delete usr1;
  delete exma1;
  delete exsup2;

}
```

07. Contribution

Registration No	Name	Contribution relationship
IT21204852	Ezhilkumaran.G	Admin class - unregisteredUsers class Admin class - Report class
IT21209420	D.E.H Mallawaarachchi	Reports class - examResults class Reports class – examManager class Report class – Examinee class
IT21211782	Wijewaradana S.P.M.G.D.	Users class - Examinee class Examinee class – Resources class
IT21215988	Waseem M.I.M.	Users class - examSupporter class examSupporter class - examManager class
IT21219498	Vithanage C.S.	Exams class – examResults class Exams class - Payment class Exams class – Examinee class Examinee class – Payment class