

**BSc (Hons) in Information Technology**

**Object Oriented Concepts – IT1050**

**Assignment 2**



**2022-July**

**Year 1, Semester 2**

Topic : **Online Examination System for Employees.**

Group no : **MLB\_WE\_PRORATA\_07**\_OnlineExaminationSystemForEmployees

Campus : Malabe

Submission Date : 18/11/2022

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** |
| **IT18077698** | **T.M.B.C.K.Thennakoon** | **0717644169** |
| **IT20468460** | **M.H.M.I.D.Mapa** | **0716061065** |
| **IT18097566** | **P.M.U.N.Purandara** | **0761206451** |
| **IT20279134** | **A.K.Chaudhary** | **0763366156** |

**Exercise 1:**

**Description**

This is a system used by organizations in order to hold examinations for their employees (Trainees, Other higher level employees). Using this system the employees can view their upcoming examinations, book a time slot, make payments if necessary for the relevant examinations. Administrations can upload examination papers, update them, confirm bookings, confirm payments.

Also the employees can provide their feedback regarding the examinations held. The administration can view the feedbacks, they can track the sessions of the employees.

Even they can monitor the employees while they are taking up an examination.

**GitHub Links of each individual member**

|  |  |  |
| --- | --- | --- |
| **Member ID** | **Member Name** | **Link** |
| **IT18077698** | **T.M.B.C.K.Thennakoon** | [**https://github.com/IT1050-2022-Feb/2022-ooc-project-july-Buddhi1998**](https://github.com/IT1050-2022-Feb/2022-ooc-project-july-Buddhi1998) |
| **IT20468460** | **M.H.M.I.D.Mapa** | [**https://github.com/IT1050-2022-Feb/2022-ooc-project-july-IT20468460**](https://github.com/IT1050-2022-Feb/2022-ooc-project-july-IT20468460) |
| **IT18097566** | **P.M.U.N.Purandara** | [**https://github.com/IT1050-2022-Feb/2022-ooc-project-july-Umangi567**](https://github.com/IT1050-2022-Feb/2022-ooc-project-july-Umangi567) |
| **IT20279134** | **A.K.Chaudhary** | [**https://github.com/IT1050-2022-Feb/2022-ooc-project-july-ak-chaudhary.git**](https://github.com/IT1050-2022-Feb/2022-ooc-project-july-ak-chaudhary.git) |

**Exercise 1:**

1. **The following are the user requirements of the Online Examination System for Employees.**

* Employees go to the official website and providing details such as name, email, address and mobile number, employee id, position held
* Employees register to the website.
* Employees enter the Username & password for website login.
* Employees checks the upcoming examinations.
* Employees can view the examination details.
* Employees can download the necessary invigilation tools for online examinations.
* Employees can book the time slots for examinations according to their need.
* Employees can take up the examination
* Employees can make necessary payments for booking the time slots.
* Employees can make necessary payments for the examinations they take if needed by including Payment ID, Employee ID, Card name, Card number, CVC, Expiry date
* Employees can view the marks that they obtain for the quiz
* Employees can request for re-correction if needed.
* Employees can provide a feedback on the examination.
* Employees can delete their profiles.
* Employees can upload relevant answers if they are to be done separately.
* Administration can insert quizzes, examination papers.
* Administration can update the examination details.
* Administration can activate the examination quizzes.
* Administration can monitor the sessions of employees
* Administration can monitor the employees during the examination.
* Administration can register an employee on behalf of them
* Administration can have time count down mechanisms for the examination.
* Administration can send messages to employee’s feedback.
* Administration can delete employee’s profile
* Administration can login to the system
* Administration maintain their profile.
* Administration can register to the system entering name, id, email.

**Nouns and Verbs analysis**

|  |  |
| --- | --- |
| **Nouns** | **Verbs** |
| Employees - Class | Login |
| Administration - Class | Register |
| Website – Out of scope | Check |
| Name - Attribute | Download |
| Email- Attribute | View |
| Employee id- Attribute | Insert |
| Username- Attribute | Update |
| Password- Attribute | Book |
| Mobile number- Attribute | Make payments |
| Position held- Attribute | Obtain |
| Examinations - Class | Invigilate |
| Marks - Attribute | Monitor |
| Time slots - Attribute | Upload |
| Quizzes - Attribute | Activate |
| Examination papers - Attribute | Delete |
| Sessions - Class | Request |
| Feedback - Class |  |
| Profile – out of scope. |  |
| Tools- Attribute |  |
| Payments – Class |  |
| Payment ID- Attribute |  |
| Card name- Attribute |  |
| Card number- Attribute |  |
| CVC- Attribute |  |
| Expiry date- Attribute |  |

1. **List down the classes**

Online Examination System for Employees.

Employees – Class name

Employee id

Name

Email

Phone Number

Position held

Username

Password

Administration– Class name

Administration ID

Name

Email

Examination– Class name

Examination ID

Examination Name

Time

Quiz Questions

Booking– Class name

Booking ID

Employee ID

Time slot

Examination ID

Examination Name

Payment slip

Payment– Class name

Payment ID

Employee ID

Card name

Card number

CVC

Expiry date

Feedback– Class name

Employee ID

Feedback ID

Feedback message

Sessions– Class name

Session ID

Employee ID

Email

Name

Log

**CRC Cards**

**Exercise 1:**

|  |  |
| --- | --- |
| Class Name: **Employees** | |
| Responsibilities | Collaborations |
| Register to the website |  |
| Enter details |  |
| View Exams | Examination |
| Confirms the booking | Booking |
| Making Payment | Payment |

|  |  |
| --- | --- |
| Class Name: **Administration** | |
| Responsibilities | Collaborations |
| Register to the website |  |
| Enter details |  |
| Insert Examinations | Examination |
| Delete Examination | Examination |
| Update Examinations | Examination |
| Delete Employees | Employees |
| View Sessions | Sessions |
| Confirm Bookings | Bookings, Payments |

|  |  |
| --- | --- |
| Class Name: **Examinations** | |
| Responsibilities | Collaborations |
| Add examination details |  |
| Maintain time slots | Booking, Payments |

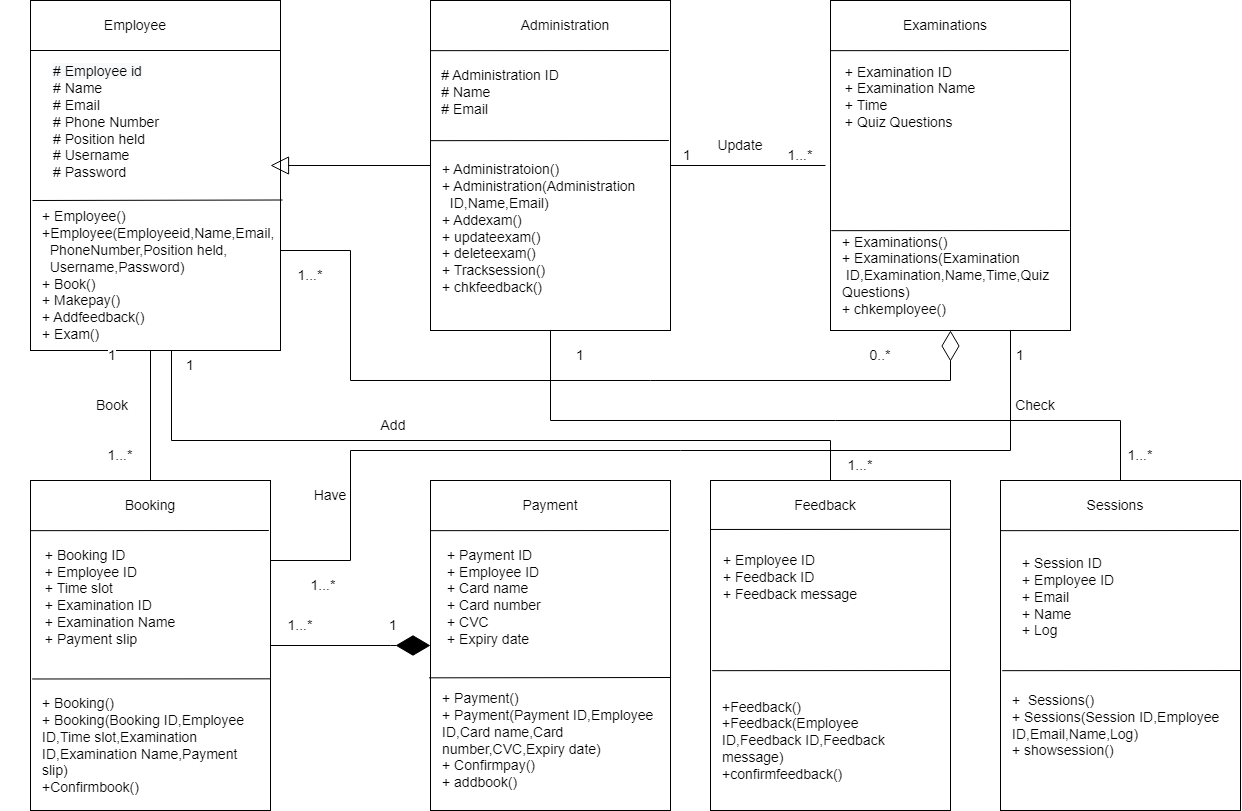
|  |  |
| --- | --- |
| Class Name: **Payment** | |
| Responsibilities | Collaborations |
| Accept the payment | Employee |
| Confirm the booking | Booking, Administration |
| System shown the details about user payment |  |

|  |  |
| --- | --- |
| Class Name: **Booking** | |
| Responsibilities | Collaborations |
| Check the new bookings | Employees |
| Accept bookings | Payments |
| Confirms the Bookings | Payments |
| Add employee to relevant booking | Employee |
| Add the exam to the booking | Examinations, Administration |

|  |  |
| --- | --- |
| Class Name: **Feedback** | |
| Responsibilities | Collaborations |
| View feedback |  |
| Upload feedbacks | Employees |

|  |  |
| --- | --- |
| Class Name: **Sessions** | |
| Responsibilities | Collaborations |
| Check the logs |  |
| Check the employees activities | Employees |

**Exercise 3:**



**Exercise 4:**

**Main.cpp**

#include <iostream>

#include <cstring>

#include "Employee.h"

#include "Administration.h"

#include "Examinations.h"

#include "Booking.h"

#include "Sessions.h"

#include "Feedback.h"

#include "Payment.h"

using namespace std;

int main() {

Employee\* emp;

emp = new Employee(1, "Pasindu", "pasindu98@gmail.com", "123456", "manager", "pasindu123",”123pasindu”);

Administration\* admin;

admin = new Feedback(1, "Kevin", "kevin98@gmail.com");

Examination\* ex;

ex = new Examination(1, "OOC", "2pm", "Simple Questions");

Booking\* book1;

book1 = new Booking(BK102,1,"2pm",1,”OOC”,”img.jpg”);

Payment\* pay1;

pay1 = new Payment (2,1, "visa",1234567,202,20201012);

Feedback\* ord1;

ord1 = new Feedback(1, 1, "Need improvement");

Sessions\* sesss;

sesss = new Sessions(1, 1, "pasindu98@gmail.com", "Pasindu", "taken ooc");}

**Employee.h**

#pragma once

class Employee

{

protected:

int employeeid;

char Name[20];

char Email[20];

int phonenumber;

char Username[20];

char Password[20];

Payment\* payment;

Booking\* booking;

Feedback \* feedback;

Payment \* pay;

public:

Employee();

Employee(int eemployeeid,char Nameemp[20],char Emailemp[20],int pphonenumber, char Usernameemp[20],char Passwordemp[20]);

void Book();

void makepay();

void addfeedback();

void Exam();

~Employee();

};

**Employee.cpp**

//Implementation part

#include <iostream>

#include"Employee.h"

#include<cstring>

using namspace std;

Employee :: Employee(int eemployeeid,char Nameemp[20],char Emailemp[20],int pphonenumber, char Usernameemp[20],char Passwordemp[20]);

{

employeeid = eemployeeid;

strcpy(Name, Nameemp);

strcpy(Email, Emailemp);

strcpy(phonenumber, pphonenumber);

strcpy(Username, Usernameemp);

strcpy(Password, Passwordemp);

}

void Employee :: Book()

{

}

void Employee :: Makepay()

{

}

void Employee :: Addfeedback()

{

}

void Employee :: Exam()

{

}

Employee::~Employee(){}

**Administration.h**

#program once

class Administration: Public Employee

{

  protected:

int administrationid;

char Name[10];

char Email[20];

Session \*session;

public:

Administration();

Administration(int adminid, char aname[10], char aemail[20]);

void Addexam();

void updateexam();

void deleteexam();

void tracksession()

void chkfeedback();

};

**Administration.cpp**

//Implementation part

#include "Administration.h"

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

Administration:: Administration () {

administrationid = 0;

strcpy\_s(Name, "");

strcpy\_s(Email, "");

}

Administration:: Administration (int adminid, char aname[10], char aemail[20]){

 administrationid  = adminid;

strcpy\_s(Name, aname);

strcpy\_s(Email, aemail);

}

void Administration:: Addexam(){}

void Administration:: updateexam(){}

void Administration:: deleteexam(){}

void Administration:: tracksession(){}

void Administration:: chkfeedback(){}

Administration::~ Administration (){}

**Examinations.h**

#program once

#define SIZE 2

class Examinations {

private:

int examinationid;

char ename[15];

char time[10];

char questions[225];

Administration \* ad;

Employee \* ex;

Booking \* be;

public:

Examinations();

Examinations(int examinationid, char exname[], char etime, char equestions);

void chkemployee();

};

**Examinations.cpp**

#include "Examination.h"

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std;

Examinations:: Examinations ()

{

examinationid = 0;

strcpy\_s(ename, "");

strcpy\_s(time, "");

strcpy\_s(questions, "");}

Examinations::Examinations(int examinationid, char exname[], char etime, char equestions)

{

examinationid = examinationid;

strcpy\_s(name, exname);

strcpy\_s(time, etime);

strcpy\_s(questions, equestions);

}

void chkemployee(){}

**Booking.h**

#pragma once

class Booking

{

  private:

int bookingId;

int employeeid;

char timeslot;

int examinationid;

char examinationname;

char paymentslip;

Employee\* Employee;

Examination \* exam;

public:

Booking();

Booking(int bid, int eid, char time, char exid,char ename,char pslip,);

void confirmbooking();};

**Booking.cpp**

//Implementation part

#include<iostream>

#include"Booking.h"

#include<cstring>

using namespace std;

Booking::Booking(){

}

Booking::Booking(int bid, int eid, char time, char exid,char ename,char pslip,)

{

bookingId=bid

employeeid=eid

timeslot=time

examinationid=exid

examinationname=ename

paymentslip=pslip

}

void Booking:: confirmbooking ()

{

}

Booking::~Booking()

{

}

**Payment.h**

#pragma once

class Payment

{

protected:

int paymentid;

int employeeid;

char cardename;

float date;

int cvc

public:

Payment()

Payment(int pid,int eid,char cname,float pdate,int ccvc);

void confirmpay();

void addbook();

~Payment();

};

**Payment.cpp**

//Implementation part

#include<iostream>

#include<cstring>

#include"Payment.h"

using namespace std;

Payment::Payment(){

paymentid=0;

employeeid=0;

strcpy\_s(cardename, "");

date=0;

cvc=0;

}

Payment::Payment(int pid,int eid,char cname,float pdate,int ccvc){

Paymentid=pid;

Employeeid=eid;

strcpy\_s(cardename, cname);

date=pdate;

cvc=ccvc;}

void Payment:: confirmpay ()

{

}

void Payment:: addbook ()

{

}

Payment::~Payment()

{}

**Feedback.h**

#pragma once

class Feedback

{

private:

int emploid;

int feedbackid;

char message[20];

Employee\* Employee[10];

 public:

Feedback();

Feedback(int emploid, int feedbackid, char message [20]);

void confirmfeed();

~Feedback();

};

**Feedback.cpp**

//Implementation part

#include<iostream>

#include<cstring>

#include"Feedback.h"

using namespace std;

Feedback::Feedback(){

emploid=0;

Feedbackid=0;

strcpy(message, “”);

}

Feedback::Feedback(int emploid, int feedbackid, char message [20])

{

emploid= emploid;

feedbackid= feedbackid;

strcpy(message, message);

}

void Feedback::confirmfeed()

{

}

Feedback::~Feedback()

{

}

**Sessions.h**

#pragma once

class Sessions

{

private:

int sessionid;

int employeeid;

char email[20];

char name[20];

char log[20];

Administrations\* admin[10];

 public:

Features();

Features(int sessionid,int employeeid,char email[20],char name[20],char log[20]);

void showsession();

~Session();

};

**Sessions.cpp**

//Implementation part

#include<iostream>

#include<cstring>

#include"Sessions.h"

using namespace std;

Sessions::Sessions(int sessionid,int employeeid,char email[20],char name[20],char log[20])

{

Sessionid= sessionid;

Employeeid= employeeid;

strcpy(email, email);

strcpy(name, name);

strcpy(log, log);

}

void Sesions::showsession()

{

}

Sessions::~Sessions()

{

}