BSc (Hons) in Information Technology
Object Oriented Concepts – IT1050
Assignment 2

Year 1, Semester 2 **2023-July-Dec**

Topic : Online Medical Portal

Group no : MLB_WD_03.02.07

Campus : Malabe

Submission Date : 30/10/2023

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
IT22130716	MUNASINGHE M.A.M.C	0772611121
IT22100320	SOMASUNDARA M.M	0702420420

Table of Contents

1. Description of the requirement

- All the users can visit the website and check specialists, check available medical institutes and available medical tests.
- Anyone can register and create an account by providing username, name, DOB, address, contact no, email, and password.
- After creating an account, the registered user can log into the website using the username and the password.
- Registered users can make appointments by selecting the preferred doctor.
- Registered users can view their test results.
- Registered users can view their medical history.
- Registered users can select a preferred healthcare center to get a preferred healthcare service.
- Registered users can give feedback.
- Registered users can view, delete, or edit their profiles.
- Customer support login into the website by providing a username and password.
- Customer support manages the scheduled doctor appointments.
- Customer support can activate or deactivate the patients account.
- Customer support can update doctor details.
- Customer support manages the feedback of the customers.

2. Classes Identified

- Guest user.
- Appointments.
- Feedback.
- Doctor.
- Customer support (Inherited from the user).
- Registered user (Inherited from the user).
- Manager (Inherited from the user).

3. CRC Card

Guest User	
Responsibilities:	**Collaboration:**
Log in to the system	
Validate user	
Check doctors & specialists	Doctors
Check medical institutes	
Check available medical tests	

Appointments	
Responsibilities:	**Collaboration:**
Store appointment details.	
Show appointment details.	

Feedback	
Responsibilities:	**Collaboration:**
Store details about Feedback	
Show approved feedback	

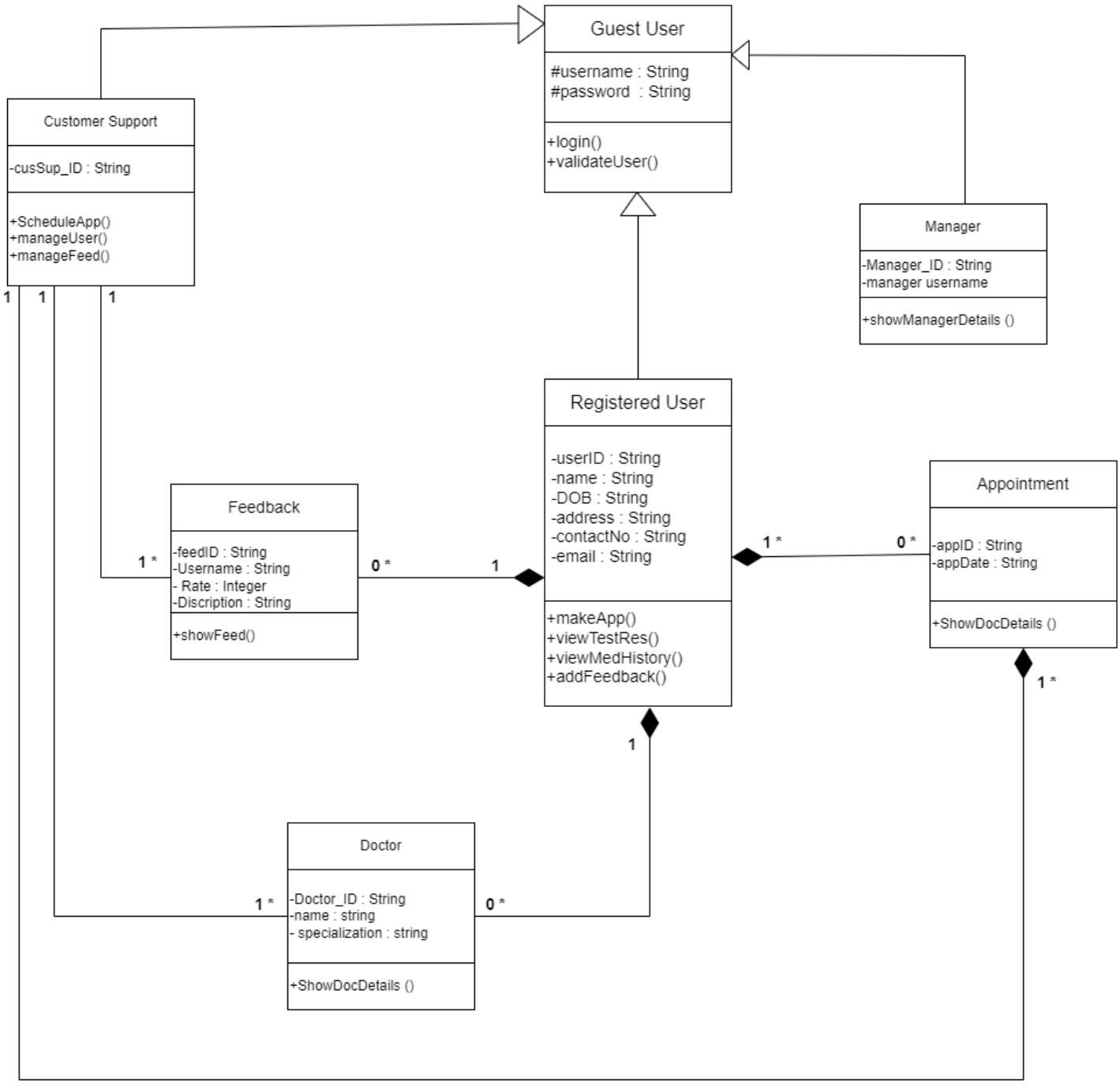
Doctor	
Responsibilities:	**Collaboration:**
Store doctor details.	
Show doctor details.	

Customer Support	
Responsibilities:	**Collaboration:**
Store details about customer supporter	
Schedule appointments	Appointments
Manage user accounts	Registered user
Manage feedback	Feedback

Registered User	
Responsibilities:	**Collaboration:**
Stores user details.	
Make appointments.	Appointments
View test results	
View medical history	
Give feedback	feedback

Manager	
Responsibilities:	**Collaboration:**
Store manager details.	
Generate reports	

4. Class Diagram



5. Coding for the classes

Main.cpp

```
#include <iostream>
#include <cstring>
#include "GuestUser.h"
#include "RegisteredUser.h"
#include "Appointment.h"
#include "CustomerSupport.h"
#include "Doctor.h"
#include "Feedback.h"
#include "Manager.h"

using namespace std;

int main()
{
	RegisteredUser* regUser = new RegisteredUser("JohnP", "abc@123", "RU001",
		"John Perera", "04/04/2000", "26 Main Street Kandy", "0772611121",
		"John@gmail.com");
	cout << "********* Detailes of User *********" << endl;
	regUser->display();
	cout << endl << endl;

	Manager* manage = new Manager("M001", "Tom Holand");
	cout << "********* Detailes of the manager *********" << endl;
	manage->showManagerDetails();
	cout << endl << endl;

	CustomerSupport* custsupp = new CustomerSupport("Jack Gilbert", "CS001", "abc@123");
	cout << "********* Detailes of the Customer Supporter *********" << endl;
	custsupp->display();
	cout << endl << endl;

	Doctor* doc;
	doc = new Doctor("D001", "Sam Werarathna", "Cardiologist");
	cout << "*********Detailes of the Doctor *********" << endl;
	doc->showDocDetails();
	cout << endl << endl;

	RegisteredUser* feed;
	feed = new RegisteredUser;
	feed->addFeed("F001", "Jessie", "Good service", 5);
	cout << "********* Feedback *********" << endl;
	feed->displayFeed();
	cout << endl << endl;

	RegisteredUser* appoint;
	appoint = new RegisteredUser;
	appoint->addApp("Ap001", "02/11/2023");
	cout << "********* Appointments *********" << endl;
	appoint->displayApp();

	return 0;
}
```

GuestUser.h

```
#pragma once
class GuestUser
{
protected:
	char userName[30];
	char password[15];

public:
	GuestUser();
		GuestUser(const char* username, const char* pass);
		void display();
		void login();
		void validateUser();
		~GuestUser();
};
```

GuestUser.cpp

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

using namespace std;

GuestUser::GuestUser() {
	strcpy_s(userName, "");
	strcpy_s(password, "");
}

GuestUser::GuestUser(const char* username, const char* pass) {
	strcpy_s(userName, username);
	strcpy_s(password, pass);
}

void GuestUser::display() {
	cout << userName << endl << password << endl;
}

void GuestUser::login() {

}

void GuestUser::validateUser() {

}

GuestUser::~GuestUser() {

}
```

RegisteredUser.h

```
#pragma once
#include "GuestUser.h"
#include "Appointment.h"
#include "Feedback.h"
#define SIZE 2


class RegisteredUser : public GuestUser
{
private:
	char userID[50];
	char name[50];
	char dob[50];
	char address[100];
	char contactNo[50];
	char email[50];

	Feedback *feed[SIZE];
	Appointment* appointment[SIZE];

public:
	RegisteredUser();
		RegisteredUser(const char* u_username, const char* u_pass, const char* u_id,
			const char* u_name, const char* u_dob, const char* u_address, const char*
u_contactNo,
			const char* u_email);

	void display();
	void makeapp();
	void viewTestRes();
	void viewMedHistory();
	void addFeedback();

	void addFeed(const char* feed_id, const char* feed_username, const char*
feed_description,
		int feed_rate);
	void displayFeed();

	void addApp(const char* app_id, const char* app_date);
	void displayApp();

	~RegisteredUser();
};
```

RegisteredUser.cpp

```
#include <iostream>
#include "Doctor.h"
#include "RegisteredUser.h"
#include "Appointment.h"
#include "Feedback.h"
#include <cstring>

using namespace std;

RegisteredUser::RegisteredUser() {
	strcpy_s(userID, "");
	strcpy_s(name, "");
	strcpy_s(dob, "");
	strcpy_s(address, "");
	strcpy_s(contactNo, "");
	strcpy_s(email, "");
}

RegisteredUser::RegisteredUser(const char* u_username, const char* u_pass, const char* u_id,
	const char* u_name, const char* u_dob, const char* u_address, const char* u_contactNo,
	const char* u_email)
	:GuestUser(u_username, u_pass) {
	strcpy_s(userID, u_id);
	strcpy_s(name, u_name);
	strcpy_s(dob, u_dob);
	strcpy_s(address, u_address);
	strcpy_s(contactNo, u_id);
	strcpy_s(email, u_email);
}

void RegisteredUser::display() {
	GuestUser::display();
	cout << userID << endl << name << endl << dob << endl << address << endl <<
		contactNo << endl << email << endl;
}

void RegisteredUser::makeapp() {

}
void RegisteredUser::viewTestRes() {

}
void RegisteredUser::viewMedHistory() {

}
void RegisteredUser::addFeedback() {

}

void RegisteredUser::addFeed(const char* feed_id, const char* feed_username, const char*
feed_description,
	int feed_rate) {

	feed[0] = new Feedback(feed_id, feed_username, feed_description, feed_rate);
}

void RegisteredUser::displayFeed() {
```

```
	feed[0]->showFeed();
}

void RegisteredUser::addApp(const char* app_id, const char* app_date)
{
	appointment[0] = new Appointment(app_id, app_date);
}

void RegisteredUser::displayApp() {
	appointment[0]->showDetailes();
}

RegisteredUser::~RegisteredUser() {

}
```

Manager.h

```
#pragma once
#include "GuestUser.h"
class Manager : public GuestUser {
private:
	char Manager_ID[10];
	char managerUserName[20];

public:
	Manager();
	Manager(const char* m_ID, const char* mUserNam);
	void showManagerDetails();
	~Manager();
};
```

Manager.cpp

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

using namespace std;

Manager::Manager() {

	strcpy_s(Manager_ID, "");
	strcpy_s(managerUserName, "");

}

Manager::Manager(const char* m_ID, const char* mUserNam) {

	strcpy_s(Manager_ID, m_ID);
	strcpy_s(managerUserName, mUserNam);

}
void Manager::showManagerDetails() {
	cout << Manager_ID << endl << managerUserName << endl;
}
Manager::~Manager() {
}
```

CustomerSupport.h

```
#pragma once
#include "GuestUser.h"
#include "Feedback.h"
#include "Doctor.h"
#include "Appointment.h"

class CustomerSupport : public GuestUser {
private:
	char cusSup_ID[10];

public:
	CustomerSupport();
	CustomerSupport(const char* M_username, const char* M_pass, const char* cs_ID);
	void display();
	void ScheduleApp();
	void manageUser();
	void manageFeed();
	~CustomerSupport();
};
```

CustomerSupport.cpp

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

using namespace std;

CustomerSupport::CustomerSupport() {
	strcpy_s(cusSup_ID, "");
}

CustomerSupport::CustomerSupport(const char* cs_username, const char* cs_pass, const char*
cs_ID)
	:GuestUser(cs_username, cs_pass) {
	strcpy_s(cusSup_ID, cs_ID);
}

void CustomerSupport::display() {
	GuestUser::display();
	cout << cusSup_ID << endl;
}
void CustomerSupport::ScheduleApp() {
}
void CustomerSupport::manageUser() {
}
void CustomerSupport::manageFeed() {
}
CustomerSupport::~CustomerSupport() {
}
```

Doctor.h

```
#pragma once
#include "GuestUser.h"
class Doctor : public GuestUser{

private:

	char doctor_ID[10];
	char name[20];
	char specialization[20];

public:
	Doctor();
	Doctor(const char* d_ID, const char* nam, const char* spec);
	void showDocDetails();
	~Doctor();
};
```

Doctor.cpp

```
#include "Doctor.h"
#include <iostream>
#include <cstring>
#include "RegisteredUser.h"

using namespace std;

Doctor::Doctor() {

	strcpy_s(doctor_ID, "");
	strcpy_s(name, "");
	strcpy_s(specialization, "");
}

Doctor::Doctor(const char* d_ID, const char* nam, const char* spec) {

	strcpy_s(doctor_ID, d_ID);
	strcpy_s(name, nam);
	strcpy_s(specialization, spec);

}
void Doctor::showDocDetails() {

	cout << doctor_ID << endl << name << endl << specialization << endl;

}
Doctor::~Doctor() {
}
```

Appointment.h

```
#pragma once
class Appointment
{
private:
	char appID[50];
	char appDate[50];

public:
	Appointment();
		Appointment(const char* aID, const char* aDate);
		void showDetailes();
		~Appointment();
};
```

Appointment.cpp

```
#include <iostream>
#include "Appointment.h"
#include "RegisteredUser.h"
#include <cstring>

using namespace std;

Appointment::Appointment() {
	strcpy_s(appID, "");
	strcpy_s(appDate, "");
}

Appointment::Appointment(const char* aID, const char* aDate)
{
	strcpy_s(appID, aID);
	strcpy_s(appDate, aDate);
}

void Appointment::showDetailes() {
	cout << appID << endl << appDate << endl;
}

Appointment::~Appointment() {

}
```

Feedback.h

```
#pragma once
class Feedback
{
private:
	char feed_ID[10];
	char userName[20];
	char description[100];
	int rate;

public:
	Feedback();
	Feedback(const char* f_id, const char* f_username, const char* f_description, int
f_rate);
	void showFeed();
	~Feedback();
};
```

Feedback.cpp

```
#include <iostream>
#include "Feedback.h"
#include "RegisteredUser.h"
#include <cstring>

using namespace std;

Feedback::Feedback() {
	strcpy_s(feed_ID, "");
	strcpy_s(userName, "");
	strcpy_s(description, "");
	rate = 0;
}
Feedback::Feedback(const char* f_id, const char* f_username, const char* f_description, int
f_rate) {
	strcpy_s(feed_ID, f_id);
	strcpy_s(userName, f_username);
	strcpy_s(description, f_description);
	rate = f_rate;

}
void Feedback::showFeed() {
	cout << userName << endl << description << endl << rate << endl;
}
Feedback :: ~Feedback() {

}
```