OOPs PROJECT

ON

HOSPITAL MANAGEMENT SYSTEM

SUBMITTED TO:
Dr. ASHISH PHOPHALIA
Dr. NOVARUN DEV

SUBMITTED BY:
Group: PROJECT_44
MONU (201951095)
NITIN KUMAR YADAV (201951108)
SAHIL SUNIL POTDUKHE (201951114)
SANJEEV MISHRA (201951135)

INDEX

Problem Statement:

A hospital is a health care institution providing patient treatment with specialized medical and nursing staff and medical equipment. But management in hospitals is really a hectic work. Hospitals have to maintain registers for all the data of patients, about their registrations, bills, reports, and all the data of doctors. These data are prone to damage and finding and maintaining data is another headache for hospital. Not only hospitals but patients also have to go a lot of trouble while going to hospital for treatment.

Given below are the problems faced by hospitals, patients and doctors:

- **Lack of immediate retrievals: -**

The information is very difficult to retrieve and to find particular information like- E.g. - To find out about the patient's history, the user has to go through various registers. This may results in inconvenience and wastage of time.

- **Lack of immediate information storage: -**

The information generated by various transactions takes time and efforts to be stored at right place.

- **Lack of prompt updating: -**

Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved.

- **Error prone manual calculation: -**

Manual calculations are error prone and take a lot of time this may result in incorrect information. For example calculation of patient's bill based on various treatments.

- **Preparation of accurate and prompt reports: -**

This becomes a difficult task as information is difficult to collect from various register.

And the list of problems keep on going. So how do we solve these problems?

Solution:

The solution to the above mentioned problems is a Hospital Management System. The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Management of Hospital to develop software which is user friendly, simple, fast, and cost effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details. The Hospital Management System can be entered using a username and password.

Objective:

1. Define hospital
2. Register the incoming patient and store his information
3. Generating bills
4. Record reports of patient
5. Application approval of incoming doctor
6. Store the data of doctor like fees, appointments and requests
7. Access to patient and doctor’s details for administration
8. Adding of reports and bills of patients by administration
9. Access to remove doctor from hospital by administration

Concepts Used:

All the code is written in Java language and for storing data MySQL database is used.

Packages

There are four packages namely, Main Package, Patient Package, Doctor Package, Administration Package.

Interface

Driver and Connection interfaces are used in database connection.

Inheritance

Many classes will extend other classes to make use of same tasks or for generalization.

Class

All the code for each window is written in different classes and are interlinked.

Constructors

Each class has one more constructors to initialize the objects of class.

Exception Handling

Exceptions are handled over the code to handle run time errors.

Functions of type static and non-static

To perform various operations functions are used, which are either static or non-static depending upon the need.

Overloading of functions and constructors

At certain places functions and constructors are overloaded to successfully implement operations.

Multithreading

Various classes are running at the same time, jumping from one window to another, making use of multithreading.

String handling

All the data displayed or taken as input is handle using string objects.

Swing (GUI)

The output of program is show in the form of GUI, which is implemented using Swing Class of Java.

JDBC (Java Database Connectivity)

We have used JDBC to connect our code to database to store and retrieve data as and when required.

Database (MySQL)

We have used MySQL as our database, storing our data in tables.

Pseudo Code:

```
package HospitalManagementSystem;

//starting page
public class HomePage {

	int userType;
	//1-Patient
	//2-Doctor
	//3-Administration

	void callUser(int userType)
	{
		//calls user according to the type selected
		//login page open from here
	}

}

//user login page
class LoginUser {
	String userName;
	private String password;

	void login(String userName, String password)
	{
		/*
		if(credentials matched)
			calls the user profile object
		else
			message about the wrong credentials and return to login page
		*/
	}

	void newRegister()
	{
		/*
			calls the registration object according to the user type
		*/
	}

	void cancel()
	{
		/*
			return call back to HomePage
		*/
	}
}

//registration of new user
class Registration {
	String name;
	private String password;
	char gender;
	String dob;
	String email;
	String contact;
	void register()
	{
		/*
		input all the details of the user and create username and password
		save all the details of user
		*/
	}

	void cancel()
	{
		/*
			return call back to LoginUser
		*/
	}
}

//common user details and functions
class UserDetails {
	String name;
	private String password;
	char gender;
	String dob;
	String email;
	String contact;

	void viewProfile()
	{
		/*
			user can view all his/her personal details
		*/
	}

	void editProfile()
	{
		/*
			user can edit his/her some details such as contact number,
email, dob
		*/
	}

	void logout()
	{
		/*
			return call back to LoginUser
		 */
	}
}
```

```
package HospitalManagementSystem.Patient;

import HospitalManagementSystem.*;

class RegisterPatient extends Registration {
	String[] bloodgroup = {"A+ve", "A-ve", "B+ve", "B-ve", "AB+ve", "AB-ve",
"O+ve", "O-ve"};
	String address;
```

```
}

public class Patient extends UserDetails {
	String bloodGroup;
	String address;

	void viewReports()
	{
		/*
			patient can see all his/her reports
		*/
	}

	void viewBills()
	{
		/*
			patient can see his/her bills, dues and amount paid
			access Bills with patientId
		 */
	}

	void payBills()
	{
		/*
			patient can pay his/her dues
			store data in Bills object
		 */
	}

	void appointments()
	{
		/*
			patient can see all the upcoming appointments they have
			patient can even book appointments

			if(new appointment to be booked)
				call the appointment object
		 */
	}
}

class Appointments {
	String[] doctor = {"name1", "name2" , "namez"};
	String[] meetType = {"virtual meet", "physical meet"};
	String date;
	String symptoms;

	void bookNow()
	{
		/*
		book the appointment for patient with any doctor
		 */
	}

	void cancel()
	{
		/*
			return call back to Patient class
			*/
	}

	void accept()
	{
		/*
			doctor can accept appointment of patient and it will reflect 
back to patient profile
			*/
	}

	void reject()
	{
		/*
			doctor can reject appointment of patient and it will remove 
the appointment
			*/
	}
}

class Bills {
	String id;
	String patientId;
	long amount;
	long amountPaid;
	long amountDue;
	String date;
}

class Reports {
	String id;
	String patientId;
	String reportType;

	void downloadReport()
	{
		/*
			patient can download the reports
			*/
	}
}


---------------------------------------------------------------------------------

package HospitalManagementSystem.Doctor;

import HospitalManagementSystem.*;


//when doctor registers then the application will be sent to admin
class RegisterDoctor extends Registration {
	String licenseNo;
	String specialization;
	String qualification;

	//override register function to send application
	void register()
```

```java
	{
		/*
			input all the details of the user and create username and 
password
			save all the details of user
			sends application to administration
		 */
	}
}

public class Doctor extends UserDetails {

	String licenseNo;
	String specialization;
	String qualification;
	
	void appointments()
	{
		/*
			doctor can view all the appointments that he/she has to attend
		 */
	}
	
	void requests()
	{
		/*
			doctor can view all the appointment requests
			he/she can either accept or reject the requests
		 */
	}
}
```

```java
package HospitalManagementSystem.Administration;

import HospitalManagementSystem.*;

public class Admin extends UserDetails {
	String skills;
	
	void viewPatients()
	{
		/*
			admin can see all the details of patients
		 */
	}
	
	void addReports()
	{
		/*
			admin can add reports of patients that will reflect on 
patient's profile
		 */
	}
	
	void addBills()
	{
		/*
			admin can add bills of patients that will reflect on patient's
profile
		 */
	}
	
	void viewDoctors()
	{
		/*
			admin can see all the details of doctors
		 */
	}
	
	void deleteDoctors()
	{
		/*
			admin can delete doctor from hospital
		 */
	}
	
	void application()
	{
		/*
			admin can see all the applications of doctors
			admin can either accept or reject application
			call Application object's functions
		 */
	}
}


class Application extends RegisterDoctor {
	
	void accept()
	{
		/*
			admin can accept the application by calling this function
		 */
	}
	
	void reject()
	{
		/*
			admin can reject the application by calling this function
		 */
	}
}
```

Snippets from program:

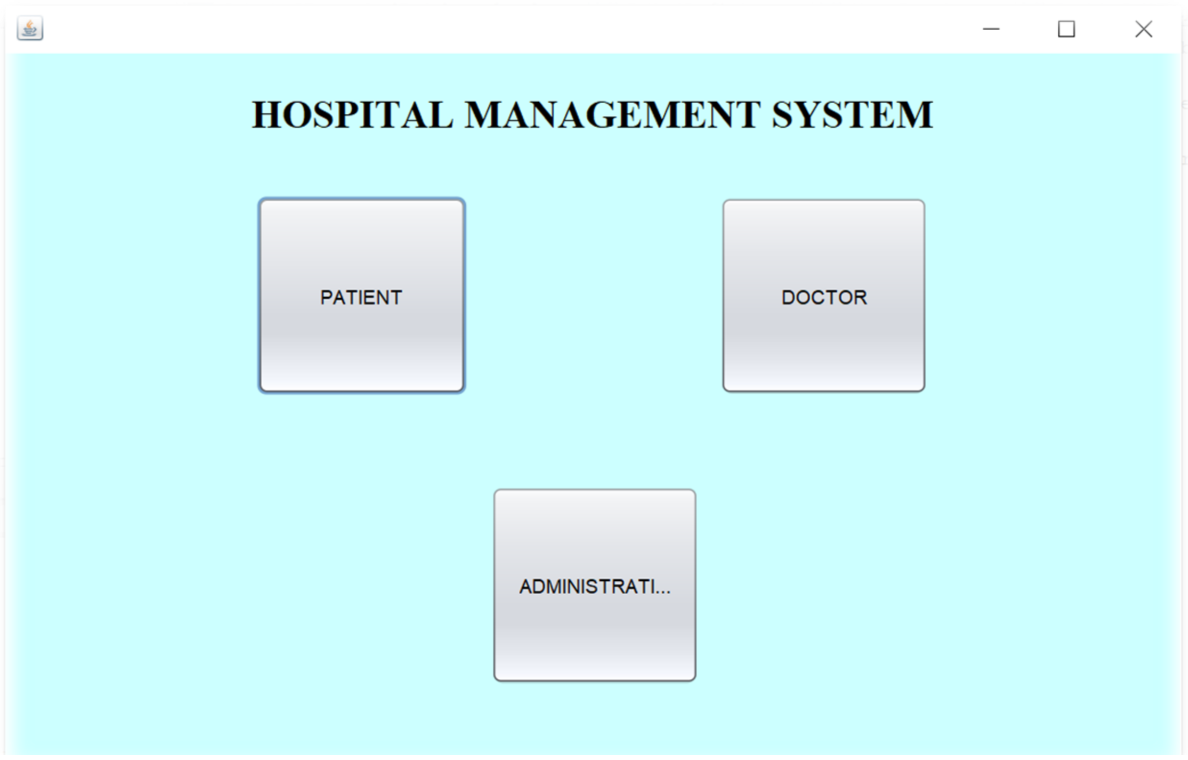


Figure 1: Home Page

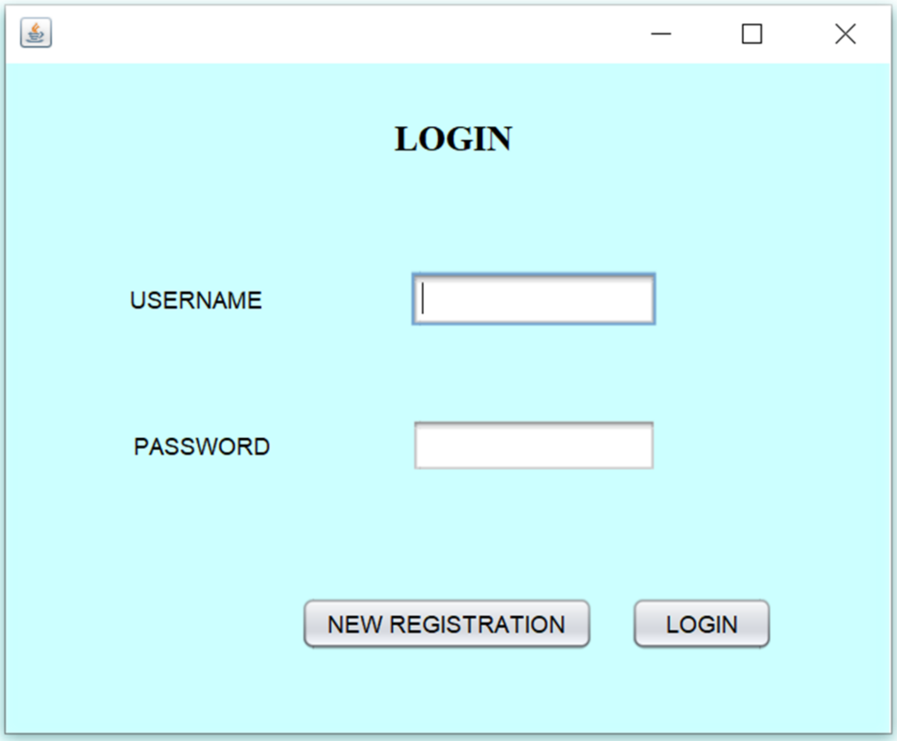
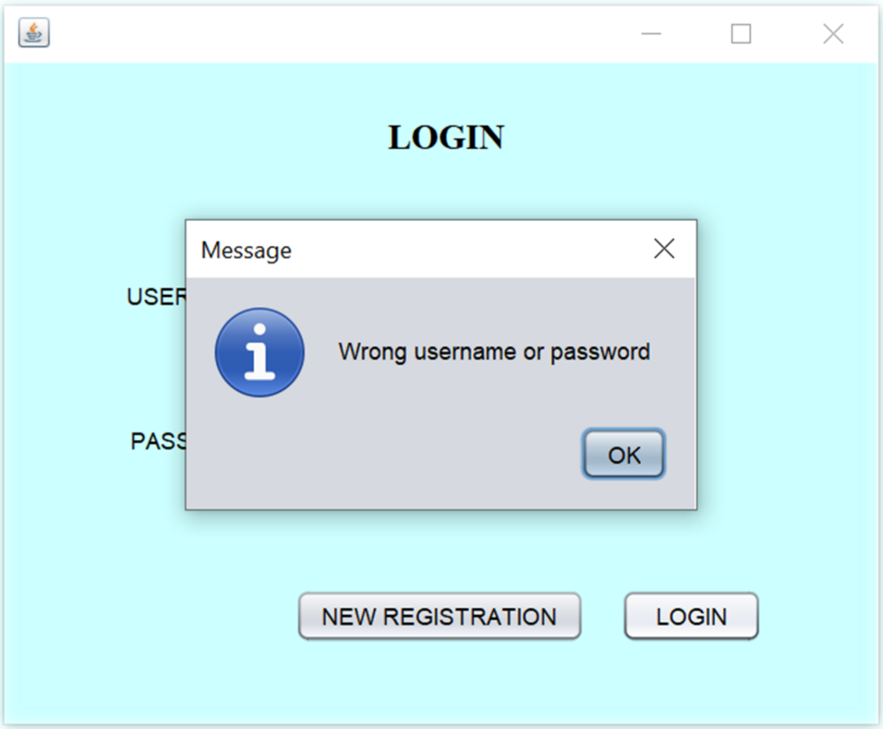


Figure 2: Login Page



Wrong username or password

Figure 3: Pop message when entered wrong credentials

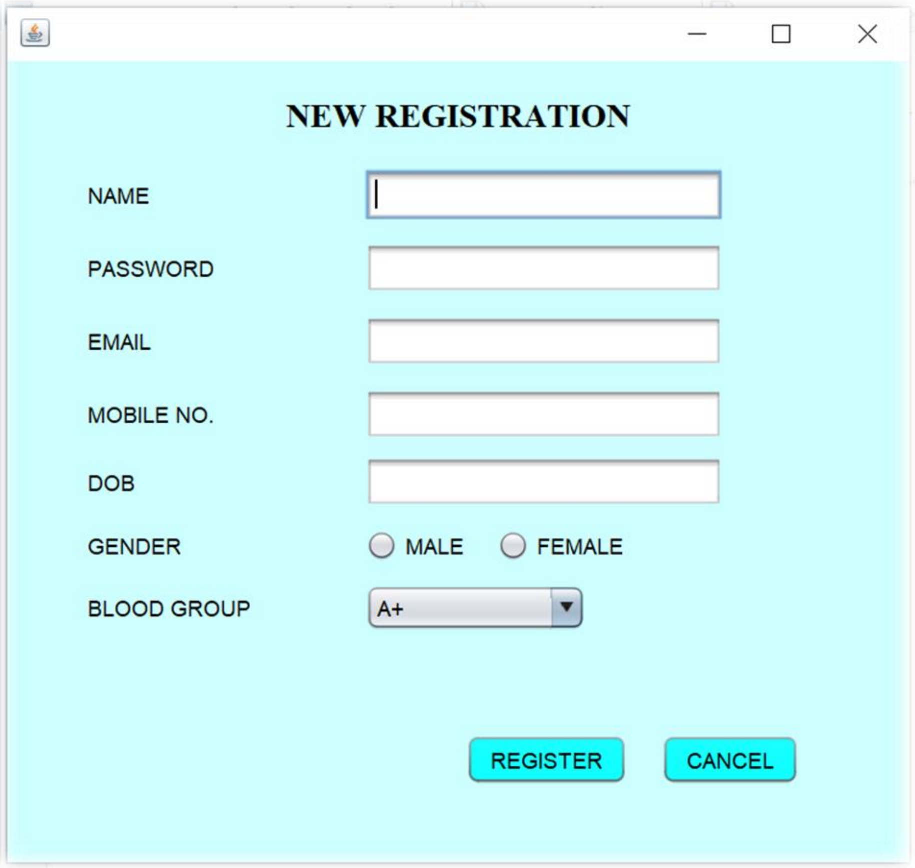
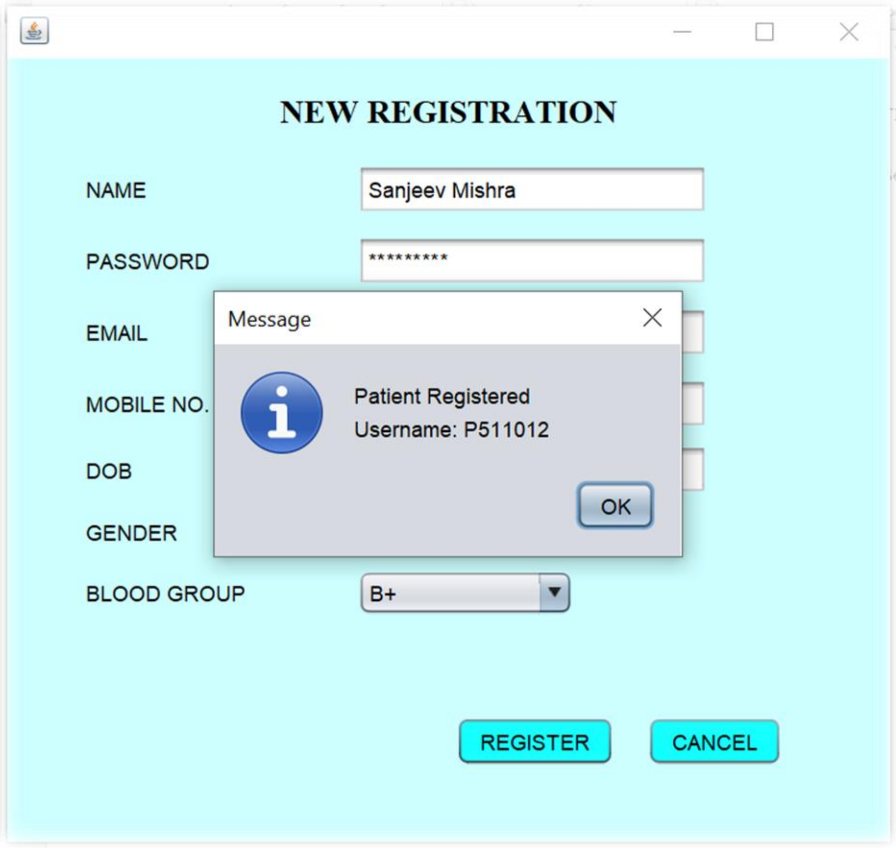


Figure 4: Registration Page (Patient)



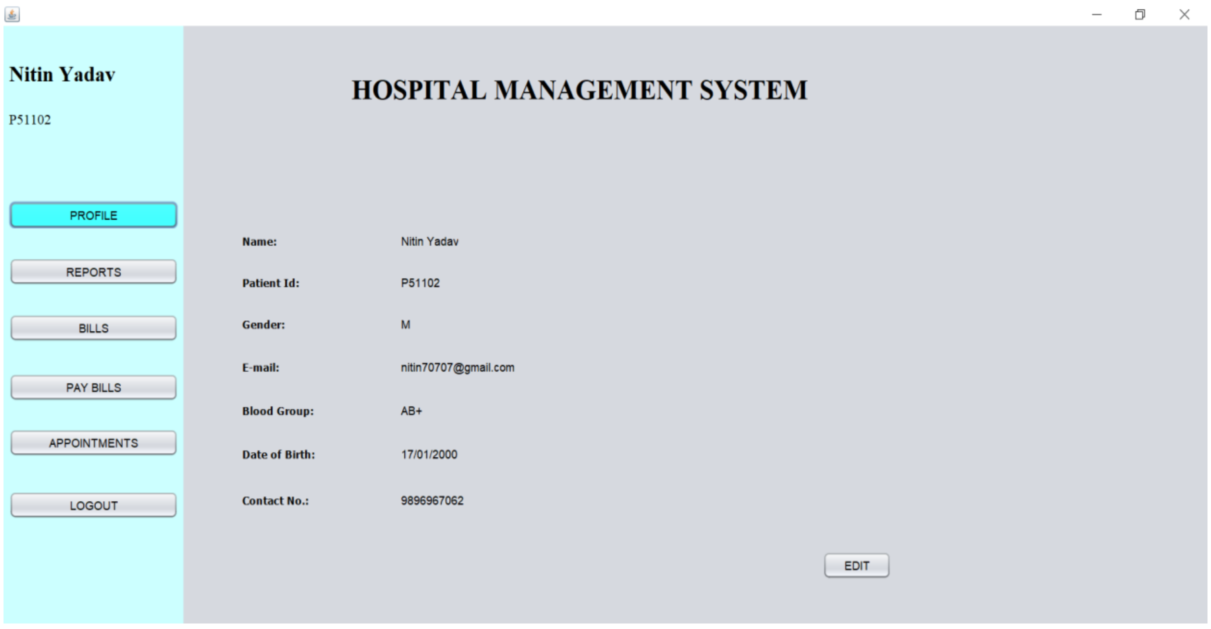
NEW REGISTRATION

NAME	Sanjeev Mishra
PASSWORD	*********
EMAIL	
MOBILE NO.	
DOB	
GENDER	
BLOOD GROUP	B+

Message

Patient Registered
Username: P511012

Figure 5: Pop message upon registration displaying generated username



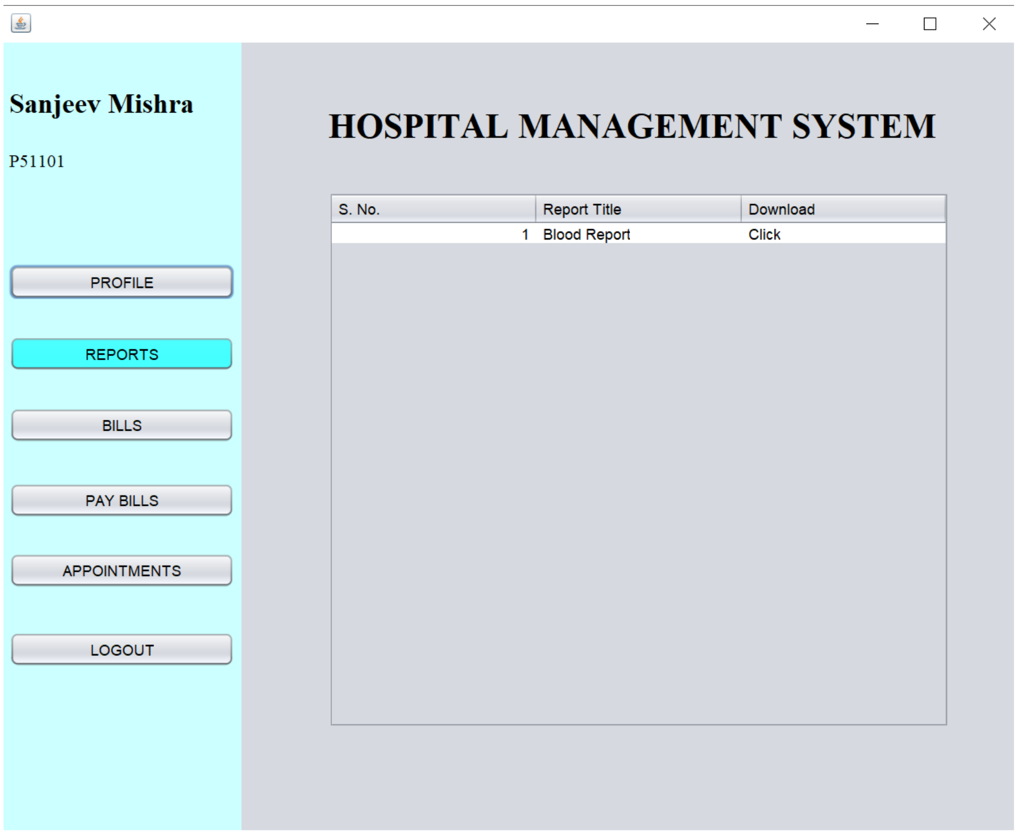
Nitin Yadav

P51102

HOSPITAL MANAGEMENT SYSTEM

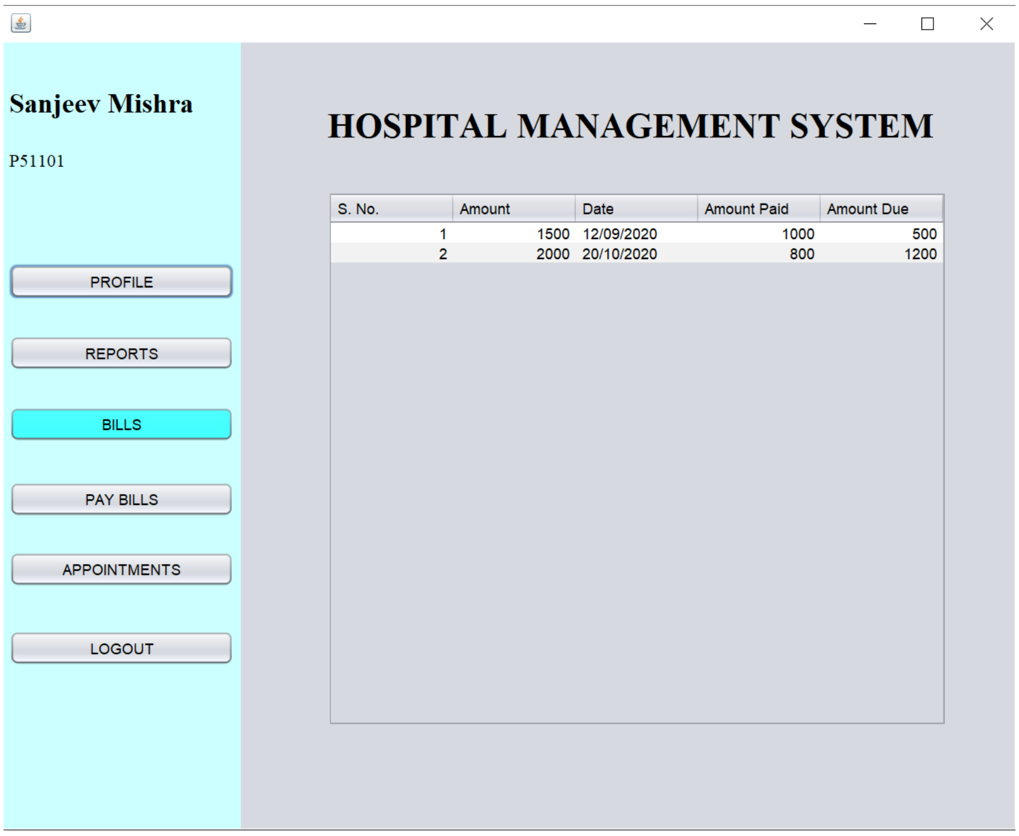
Name:	Nitin Yadav
Patient Id:	P51102
Gender:	M
E-mail:	nitin70707@gmail.com
Blood Group:	AB+
Date of Birth:	17/01/2000
Contact No.:	9896967062

Figure 6: Patient's Profile (opened upon login)



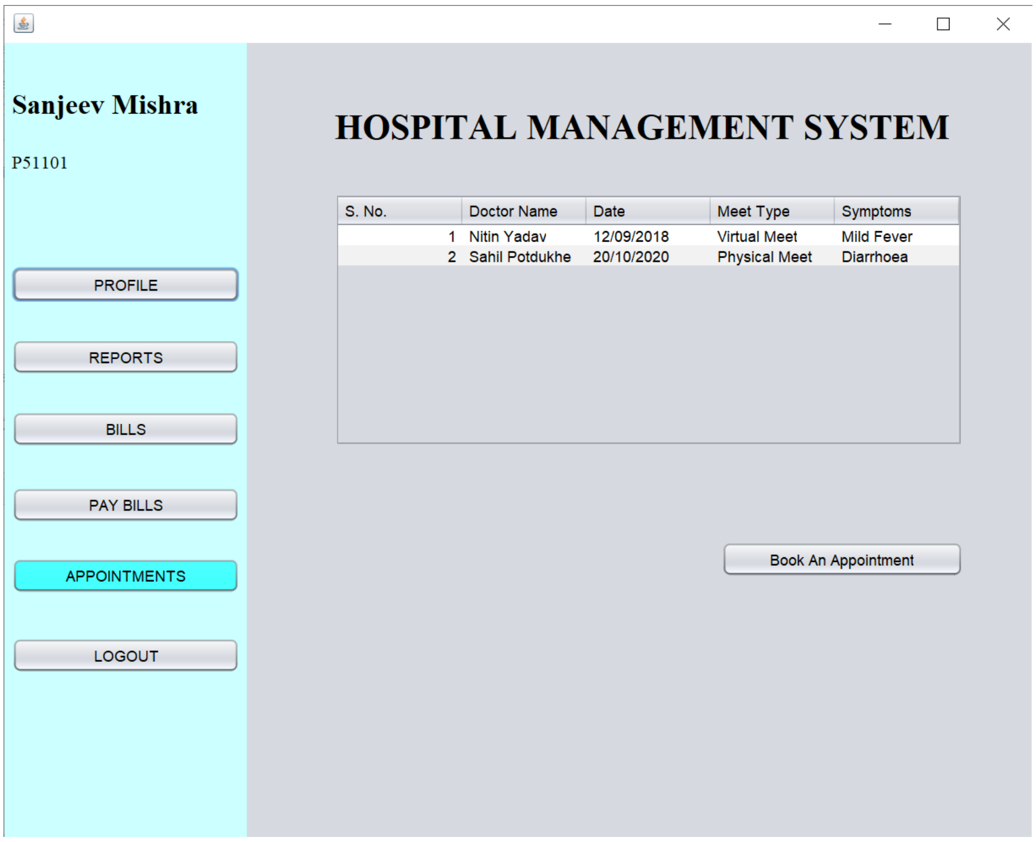
S. No.	Report Title	Download
1	Blood Report	Click

Figure 7: Reports of patient (will reflect when admin adds them)



S. No.	Amount	Date	Amount Paid	Amount Due
1	1500	12/09/2020	1000	500
2	2000	20/10/2020	800	1200

Figure 8: Bills of patient (will reflect when admin adds bills)


Sanjeev Mishra

P51101

PROFILE

REPORTS

BILLS

PAY BILLS

APPOINTMENTS

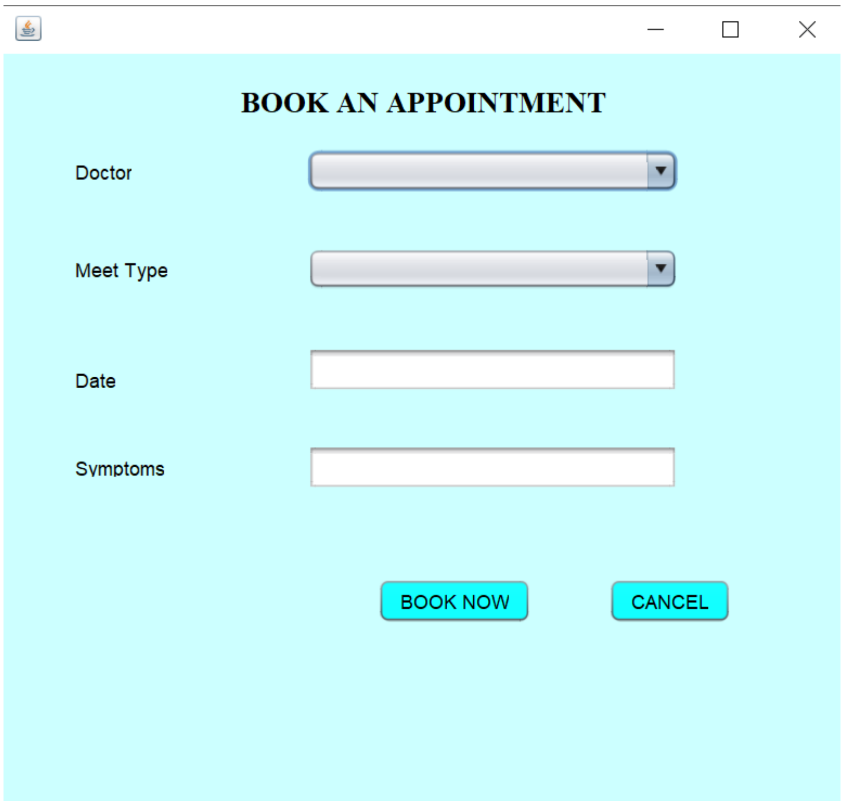
LOGOUT

HOSPITAL MANAGEMENT SYSTEM

S. No.	Doctor Name	Date	Meet Type	Symptoms
1	Nitin Yadav	12/09/2018	Virtual Meet	Mild Fever
2	Sahil Potdukhe	20/10/2020	Physical Meet	Diarrhoea

Book An Appointment

Figure 9: Appointments of patient (will reflect when booked an appointment)


BOOK AN APPOINTMENT

Doctor

Meet Type

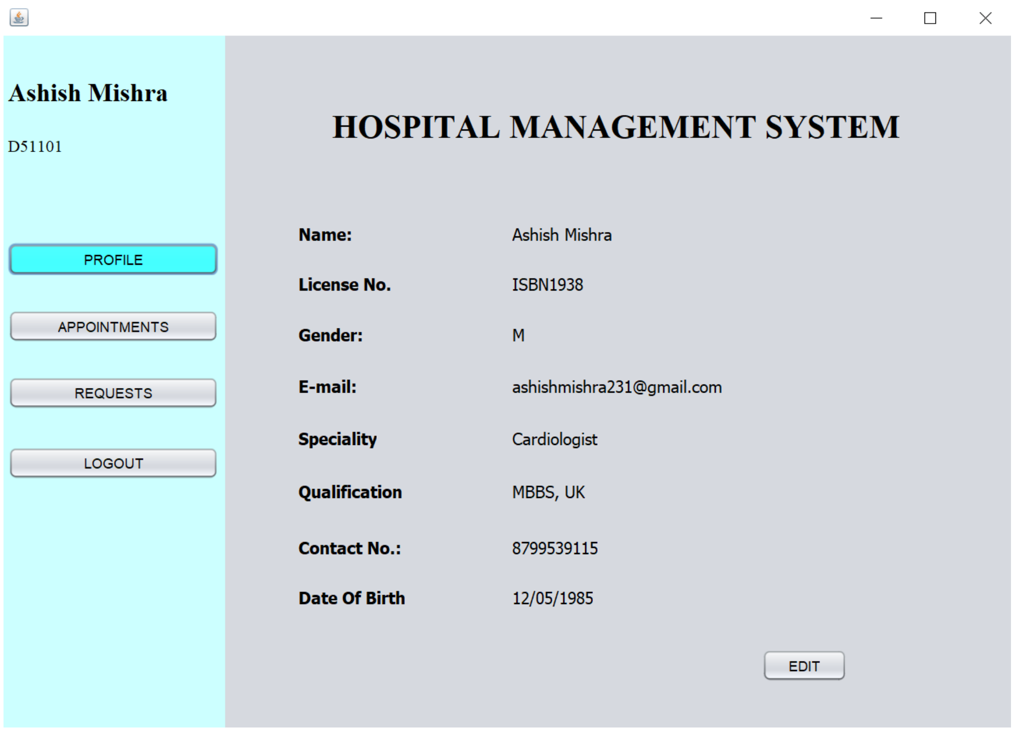
Date

Symptoms

BOOK NOW

CANCEL

Figure 10: Book an appointment

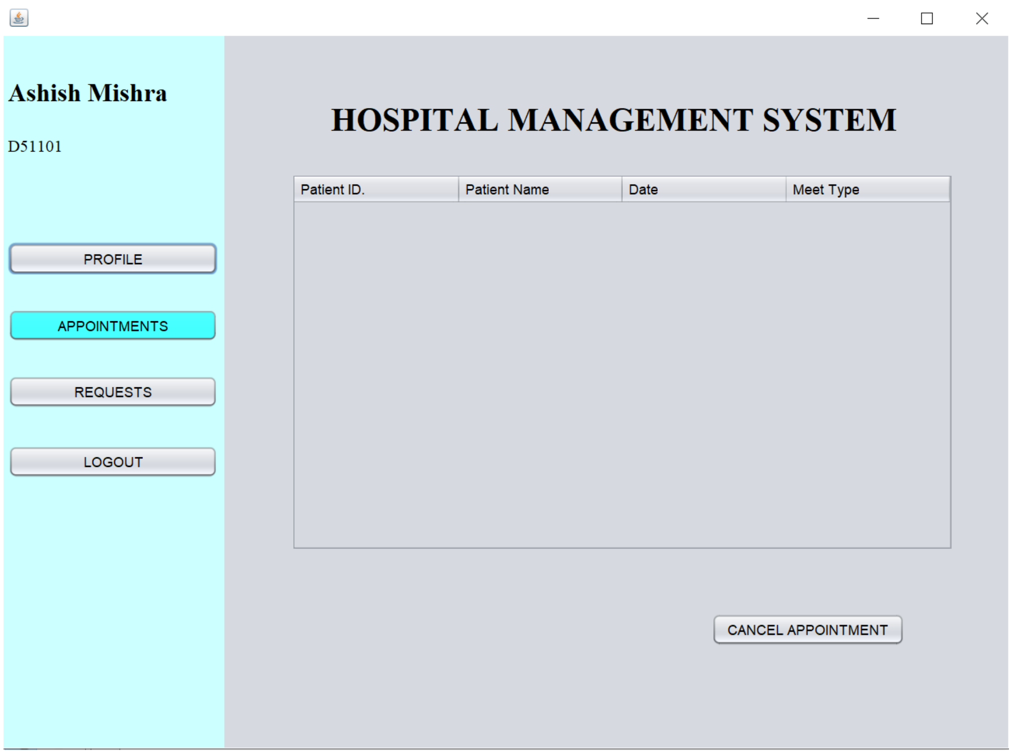

Ashish Mishra

D51101

HOSPITAL MANAGEMENT SYSTEM

Name:	Ashish Mishra
License No.	ISBN1938
Gender:	M
E-mail:	ashishmishra231@gmail.com
Speciality	Cardiologist
Qualification	MBBS, UK
Contact No.:	8799539115
Date Of Birth	12/05/1985

Figure 11: Doctor's Profile

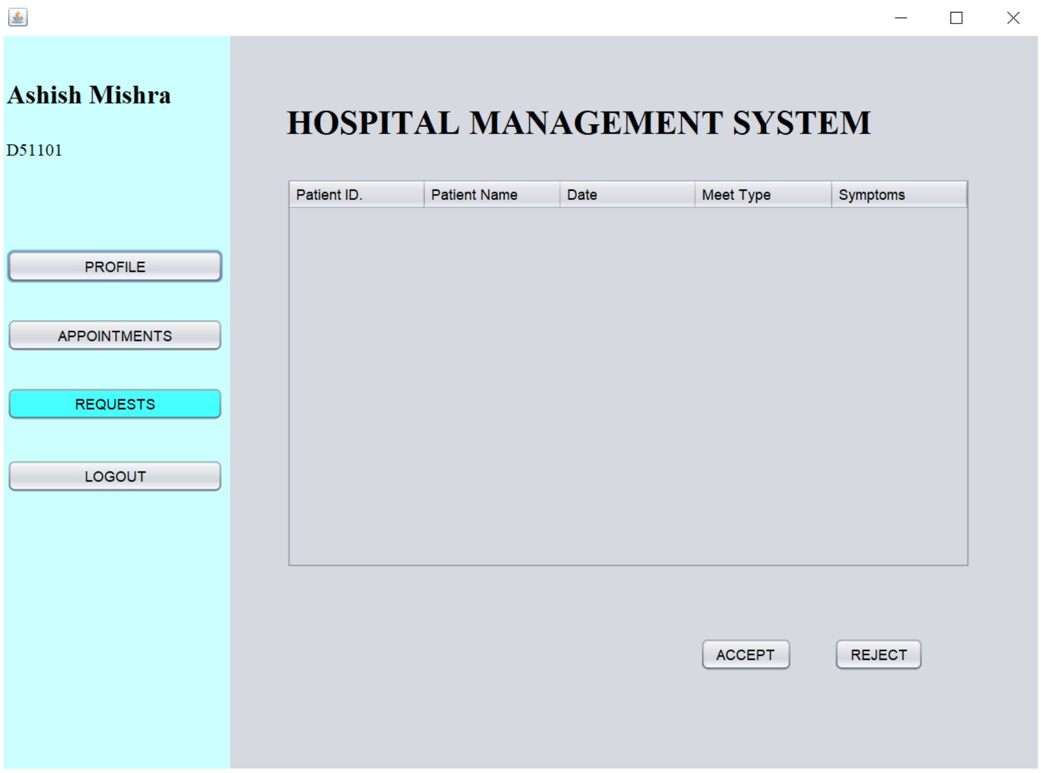

Ashish Mishra

D51101

HOSPITAL MANAGEMENT SYSTEM

Patient ID.	Patient Name	Date	Meet Type

Figure 12: Appointments of doctor booked by patients (can cancel)

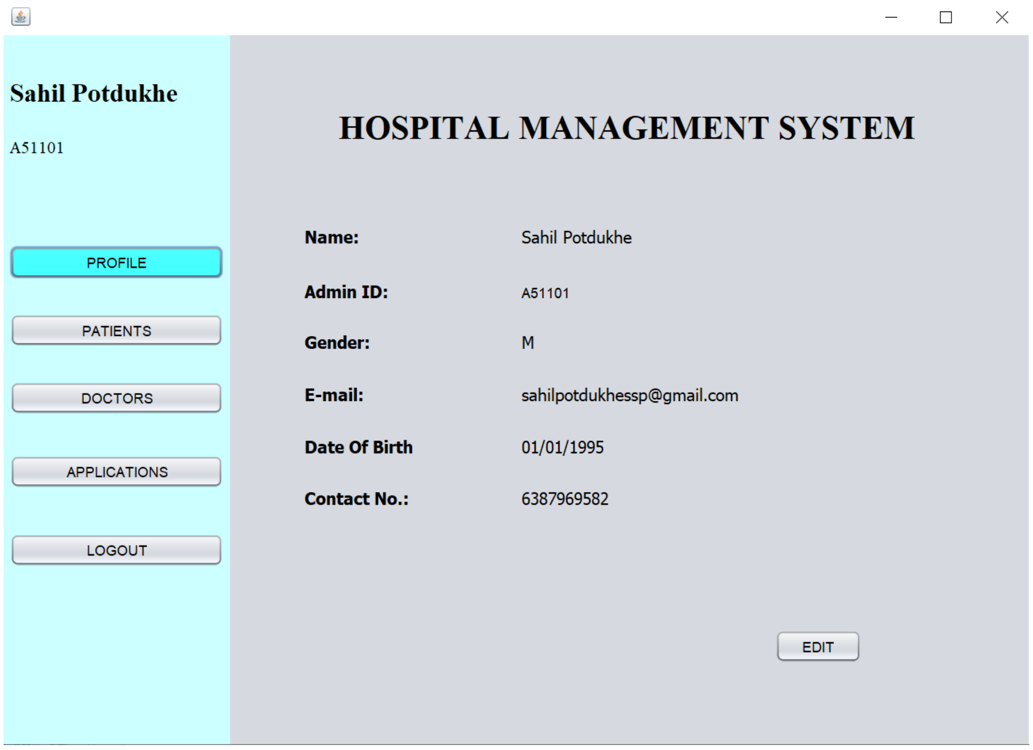

Ashish Mishra

D51101

HOSPITAL MANAGEMENT SYSTEM

Patient ID.	Patient Name	Date	Meet Type	Symptoms

Figure 13: Appointment Requests (with option to accept and reject)

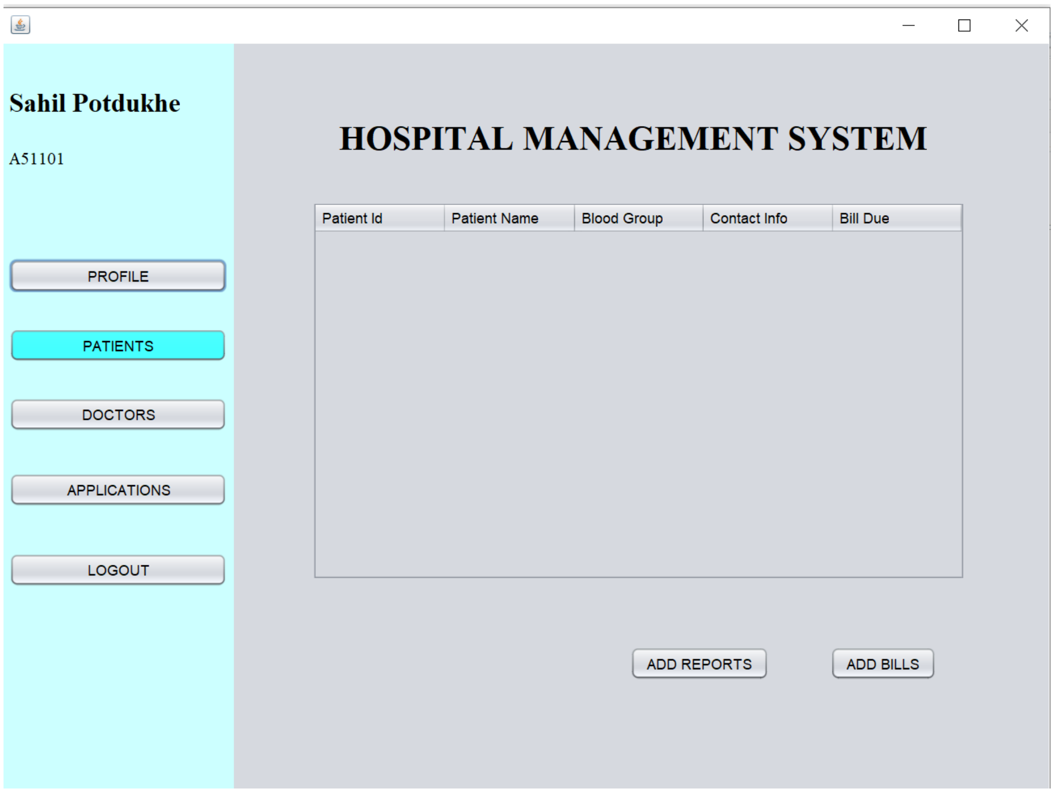

Sahil Potdukhe

A51101

HOSPITAL MANAGEMENT SYSTEM

Name:	Sahil Potdukhe
Admin ID:	A51101
Gender:	M
E-mail:	sahilpotdukhessp@gmail.com
Date Of Birth	01/01/1995
Contact No.:	6387969582

Figure 14: Admin Profile

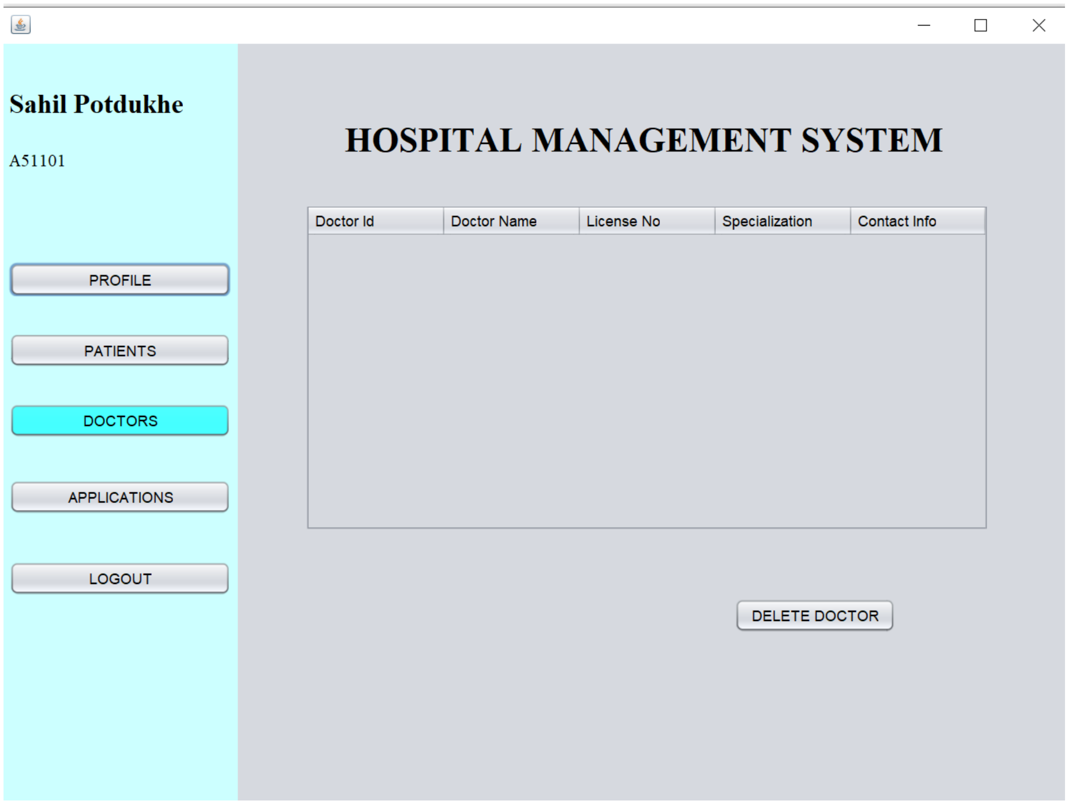

Sahil Potdukhe

A51101

HOSPITAL MANAGEMENT SYSTEM

Patient Id	Patient Name	Blood Group	Contact Info	Bill Due

Figure 15: Patients Details (will reflect when patients are added)

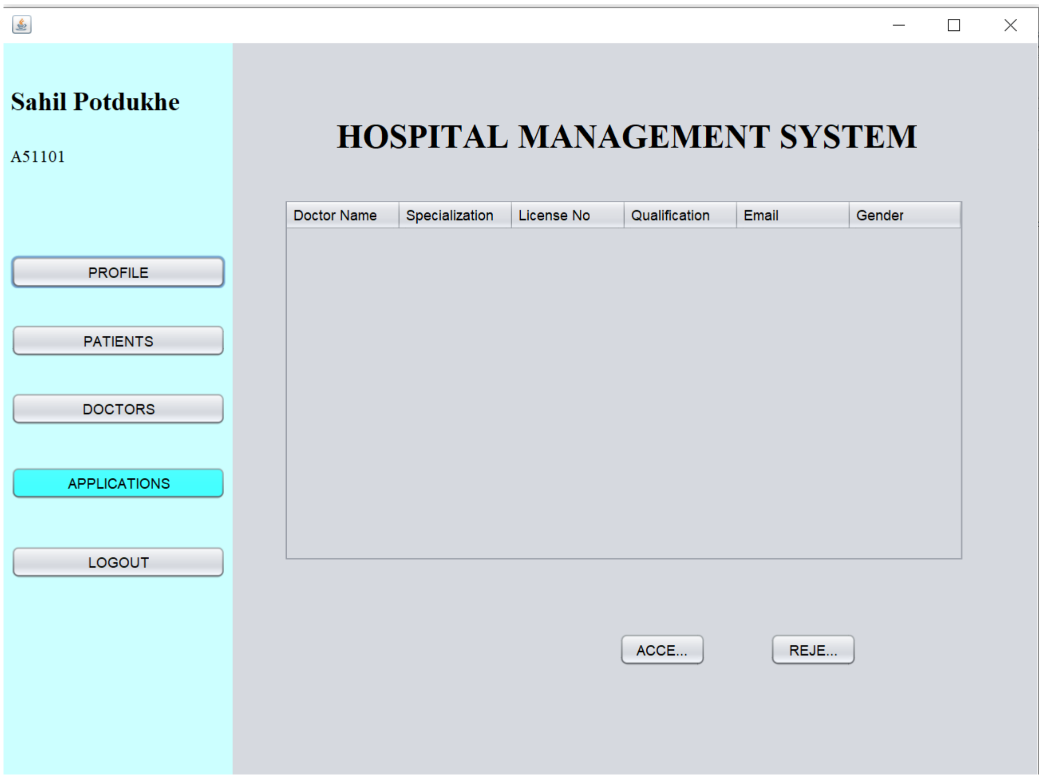

Sahil Potdukhe

A51101

HOSPITAL MANAGEMENT SYSTEM

Doctor Id	Doctor Name	License No	Specialization	Contact Info

Figure 16: Doctors Details (will reflect when doctors are added)


Sahil Potdukhe

A51101

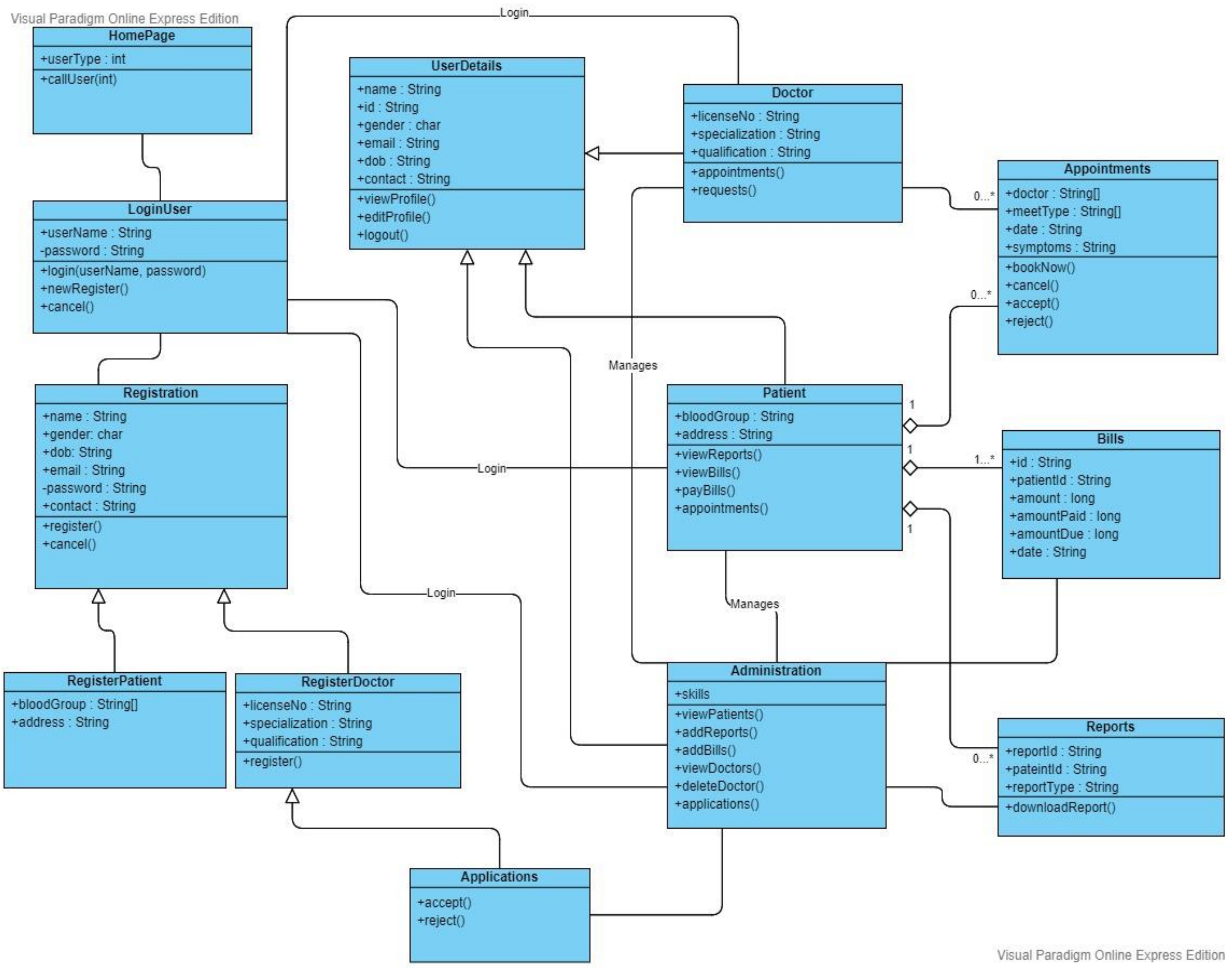
HOSPITAL MANAGEMENT SYSTEM

Doctor Name	Specialization	License No	Qualification	Email	Gender

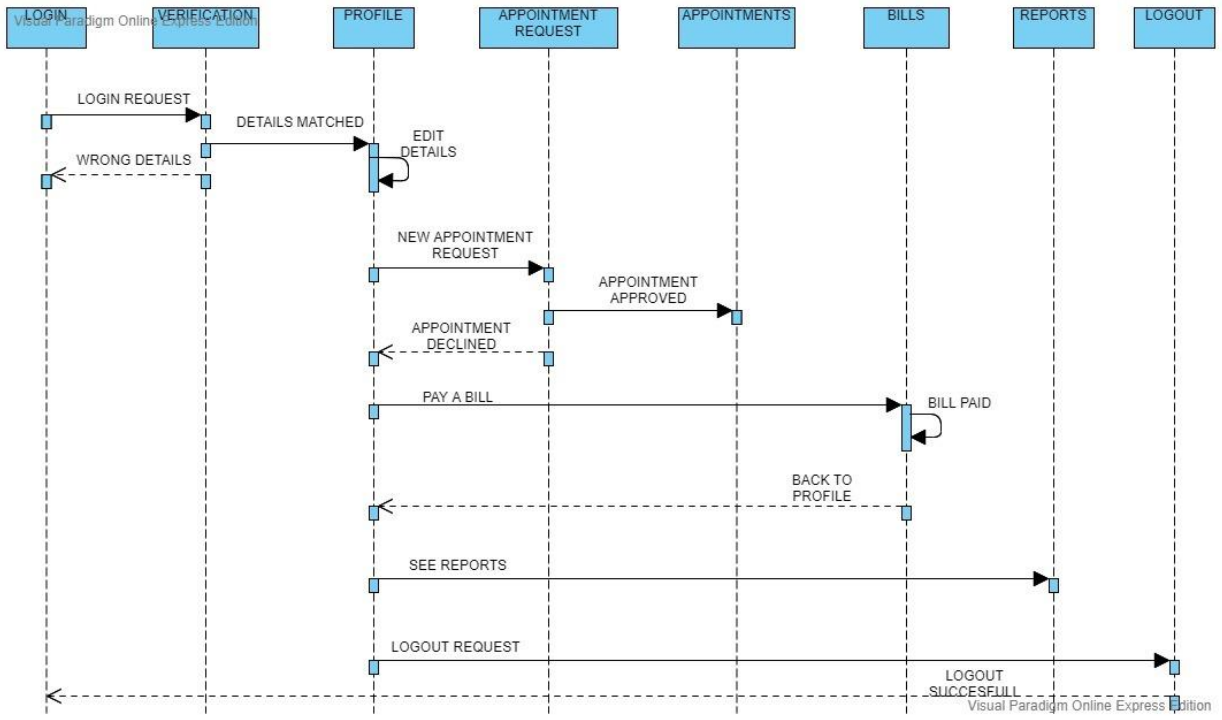
Figure 17: Applications of Doctors(will reflect when doctors will register)

UML:

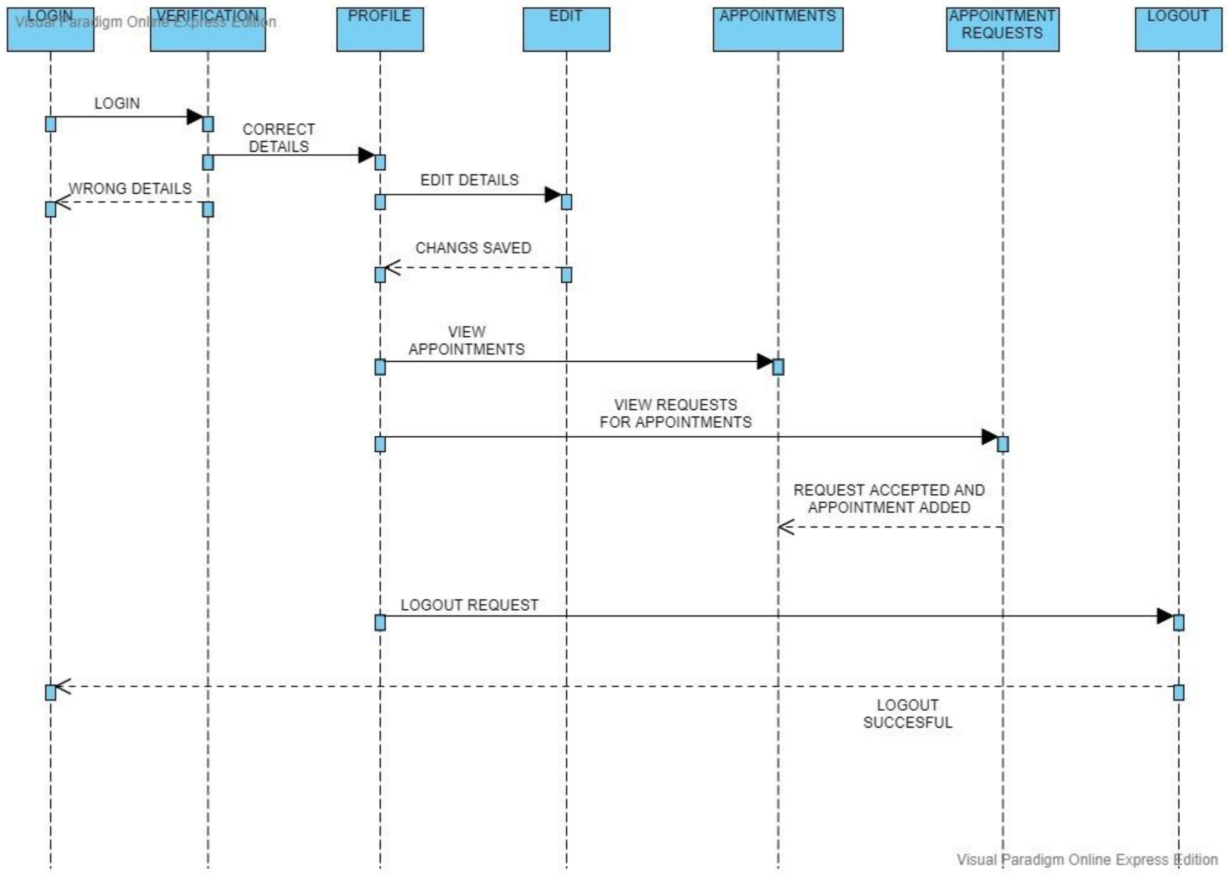
1. Class diagram



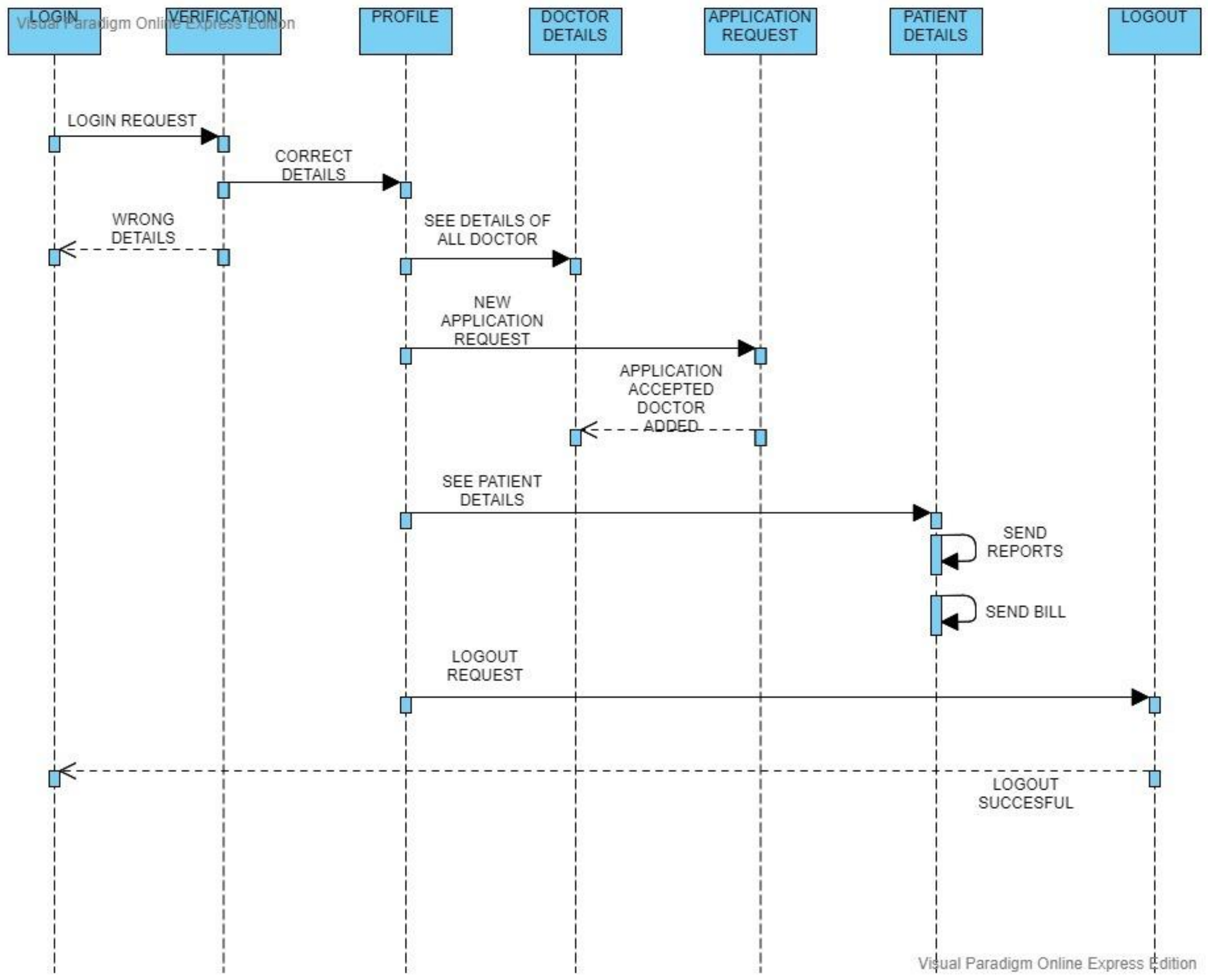
2. Sequence Diagram (Patient)



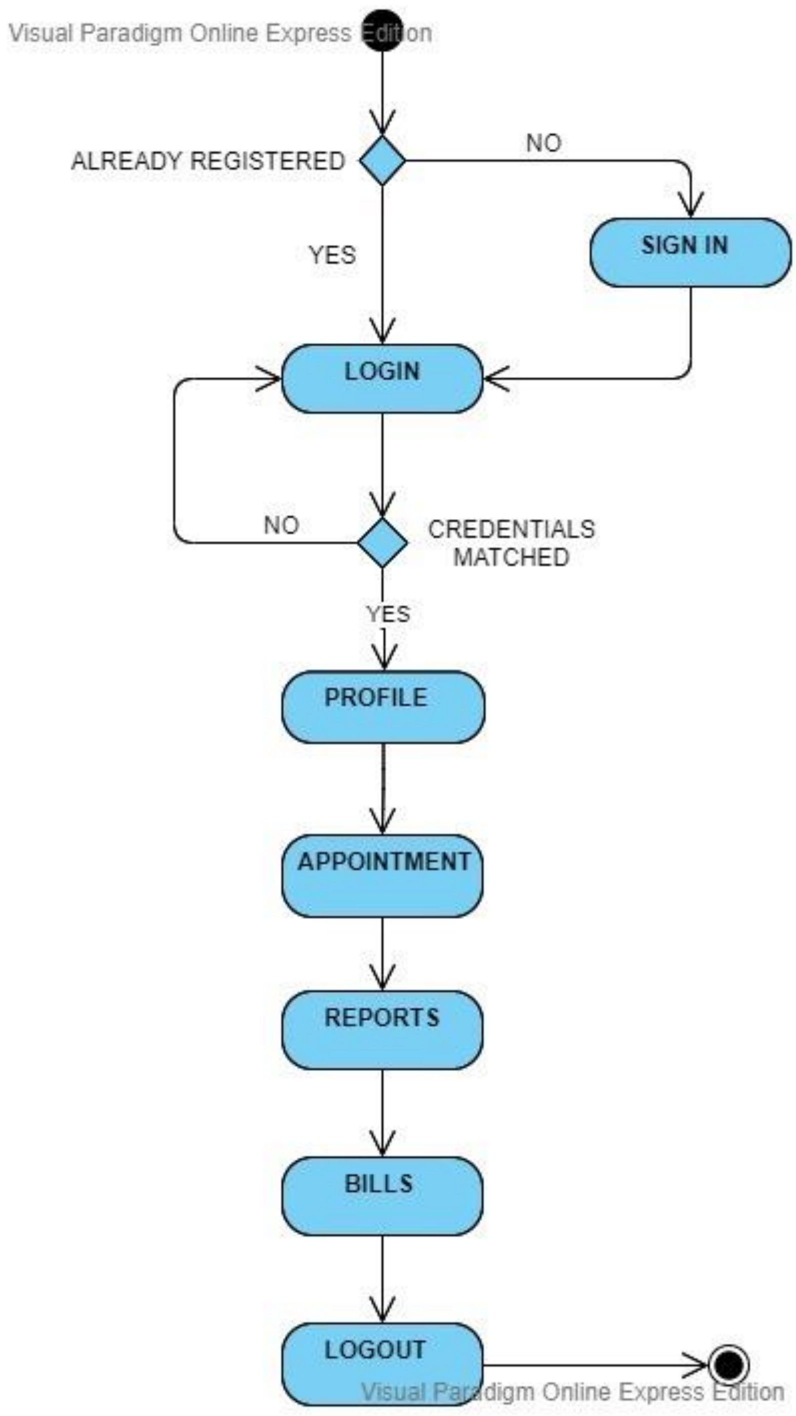
3. Sequence Diagram (Doctor)



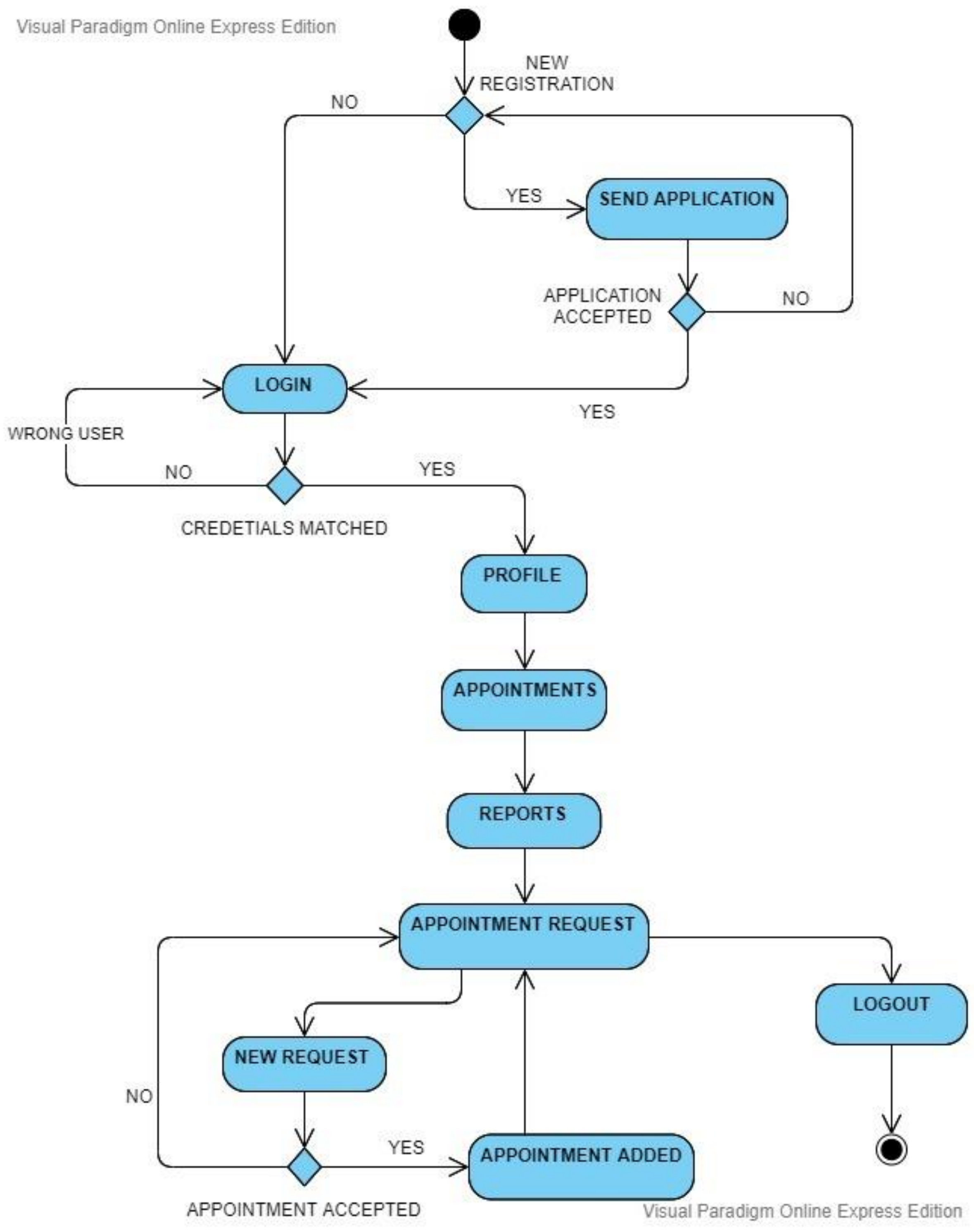
4. Sequence Diagram (Administration)



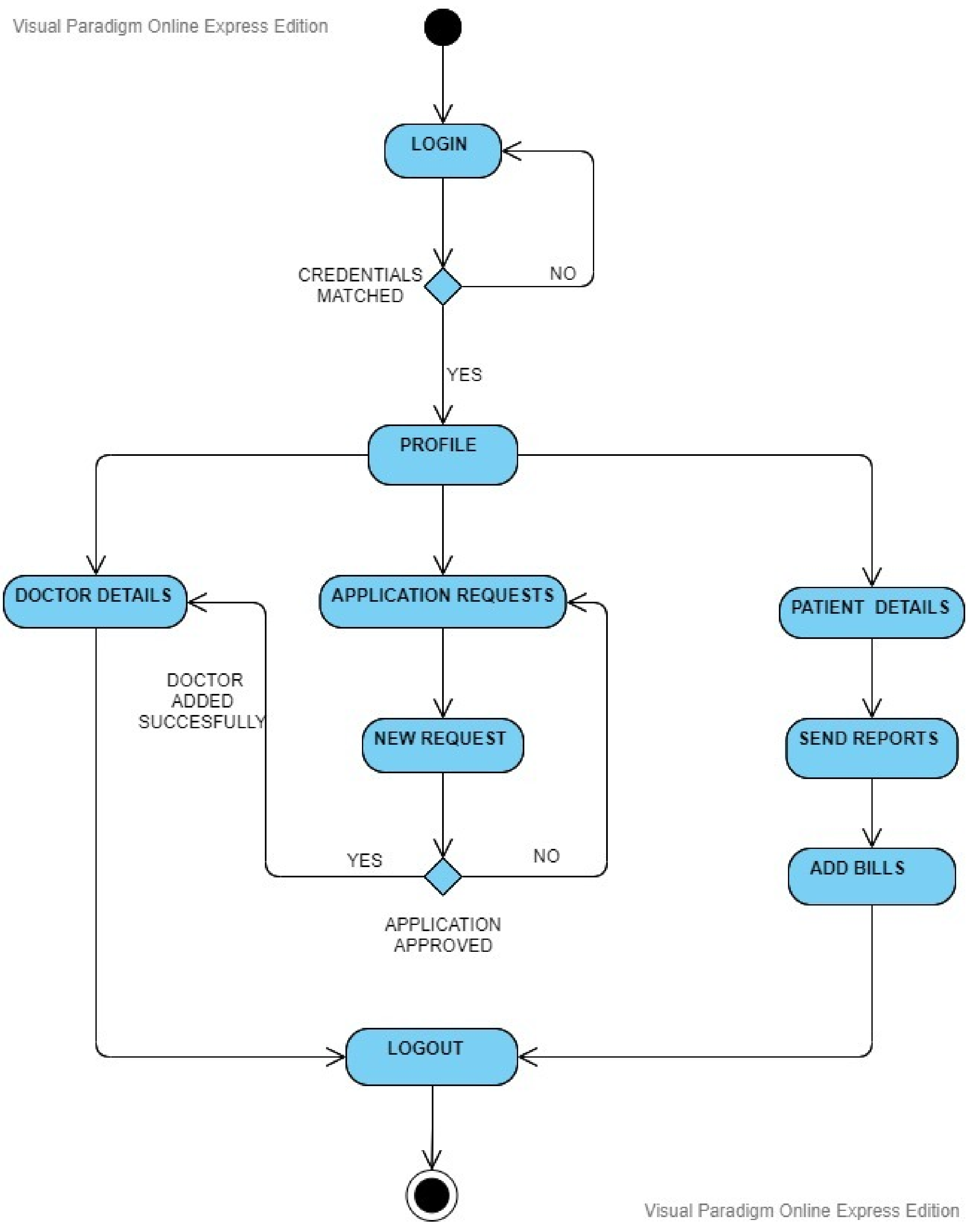
5. Activity Diagram (Patient)



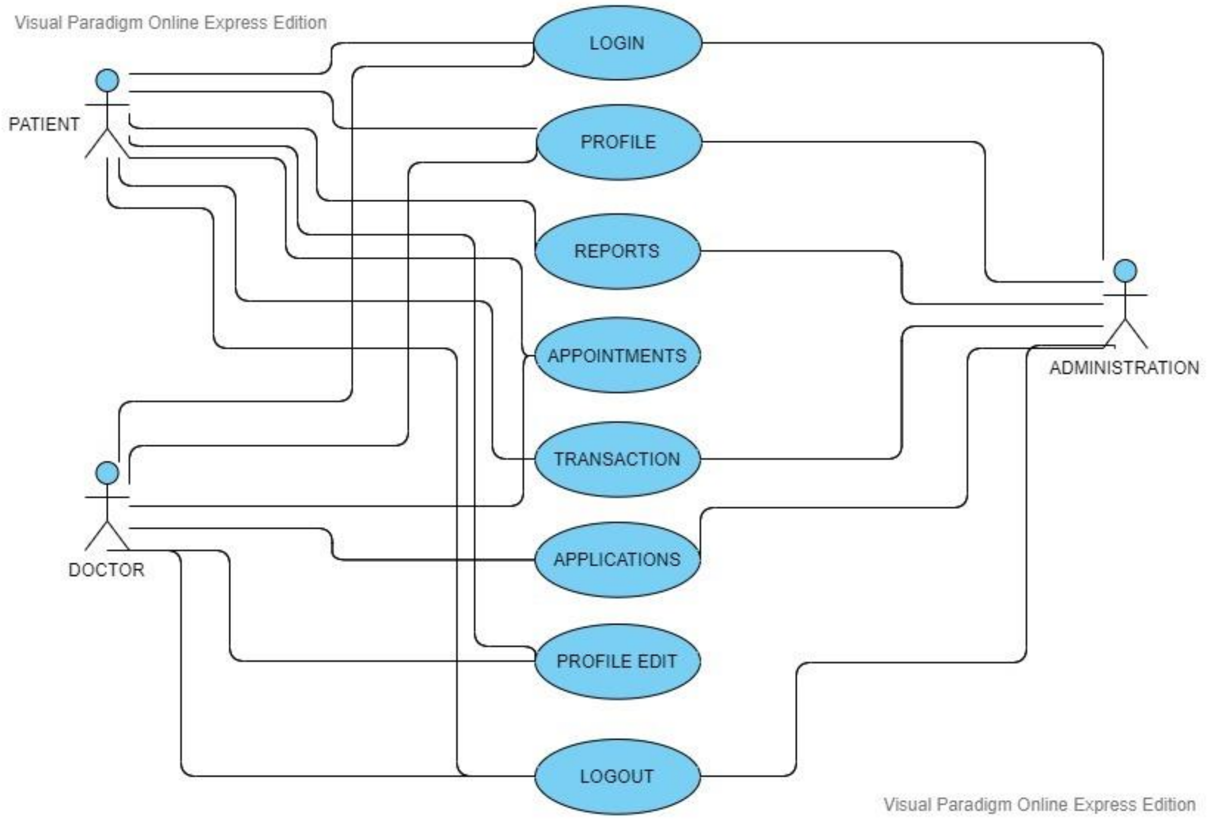
6. Activity Diagram (Doctor)



7. Activity Diagram (Administration)



8. Use case diagram



Programming Code:

All the files are contained in the zip file.

Conclusion:

Taking into account all the mentioned details, we can make the conclusion that the hospital management system is the inevitable part of the lifecycle of the modern medical institution. It automates numerous daily operations and enables smooth interactions of the users. Developing the hospital system software is a great opportunity to create the distinct, efficient and fast delivering healthcare model. Implementation of hospital management system project helps to store all the kinds of records, provide coordination and user communication, implement policies, improve day-to-day operations, arrange the supply chain, manage financial and human resources, and market hospital services. This beneficial decision covers the needs of the patients, staff and hospital authorities and simplifies their interactions. It has become the usual approach to manage the hospital. Many clinics have already experienced its advantages and continue developing new hospital management system project modules.