**Hospital Management System**

****

Session: 2022 – 2026

**Submitted by:**

Robass Atif 2022-CS-150

**Supervised by:**

Prof. Dr. Muhammad Awais Hassan

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

**Table of Contents**

**Link…….………………………………………………………………………………...……..3**

**Introduction………………………………………………………………………………...……3**

**Class Responsibility Collaboration Card……………………………...………………3**

**Object Oriented Programming………………………………………………………...…4**

**Design Pattern Implementation..………………………………………………………...5**

**Conclusion……........……………………………………………………………..……........…...6**

**Code………………………………………………………………………………………………..7**

**Image…………………………………………………………………………….………………..7**

* **Link**

https://youtu.be/yd7pYLfmX2s

* **Introduction**
* **Overview**

The Hospital Management System (HMS) is a comprehensive software solution designed to streamline and optimize the administrative, clinical, and financial operations within a healthcare facility. This system aims to enhance the overall efficiency, accuracy, and accessibility of managing patient data, medical records, appointments, and other critical aspects of hospital administration.

* **Objectives**

The objectives of this system includes

1. Enhance patient care and operational efficiency.

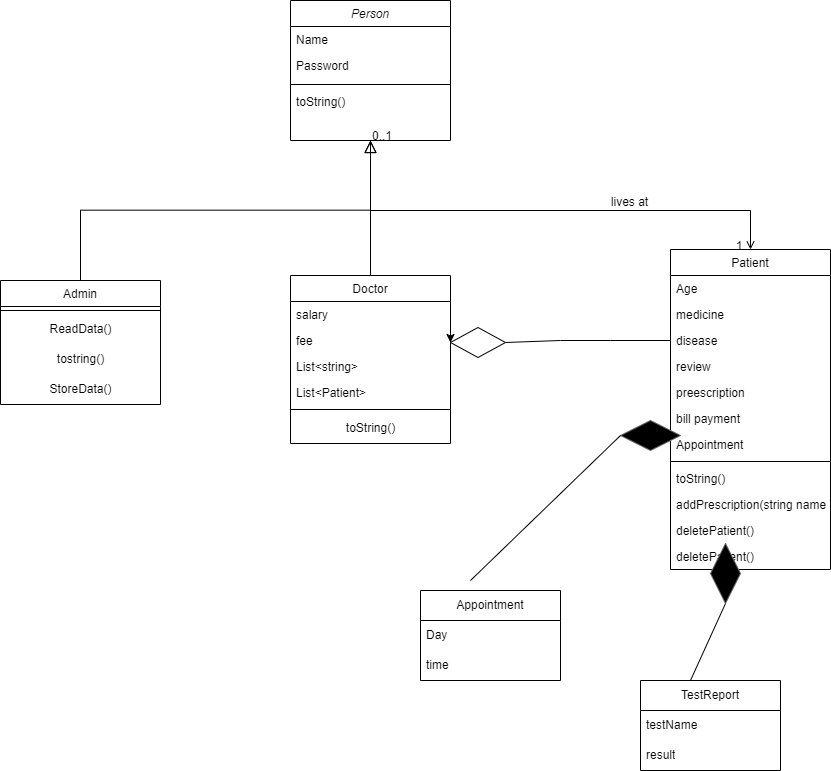
2. Ensure data security and privacy compliance.

3. Streamline administrative tasks and departmental coordination.

* **Functionality**

The intended functionality for the project includes the Sign in Sign Up system for the users.Doctor and patients added by admin. Its functionality also includes the Create, Retrieve, Update, Delete (CRUD) operations. It adds, removes, updates and prints the doctors and Patients data. It also includes the functionality of adding and showing the statistics of the Doctor and Patients.

* **Class Responsibility Collaboration Diagrams**

****

* **Object Oriented Programming**

Object-oriented programming (OOP) is a programming paradigm based on the concepts of the “objects”, which contains data and code called as attributes and behavior of the class respectively. The main concepts of OOP includes association, inheritance and polymorphism. I have used this programming paradigm in my project.

* **Association**

In Object-oriented programming, Association is a relation between two separate classes which establishes through their objects. Association can be one-to-one, one-to-many, many-to-one, many-to-many. There are two types of Association, Aggregation and Composition. I have used Composition in my project. It is used in two places..

1. Doctor class contains Patient class.

**Advantage**

If I compare this with my procedural programming concepts, I can observe that there is a clear advantage of OOP. There was disjoint data of patient and Doctor in procedural programming which is rectified in OOP. Now the objects of Patients are within the objects of Doctor

* **Inheritance**

Inheritance is one of the core concepts of Object-oriented programming approach. It is a feature that allows a new class to derive from an existing class. The new class inherits all the public or protected attributes and the member functions of the base class. I have used this OOP concept in two places in this management system.

1. Person inherit Doctors , Admin and Patient.

**Advantage**

Inheritance gives various advantages over procedural programming. It promotes code-reusability and reduces redundancy. It helps in organizing the program’s structure. It allows flexibility in the code as you will adjust in one place and the rest of the code will work smoothly.

* **Polymorphism**

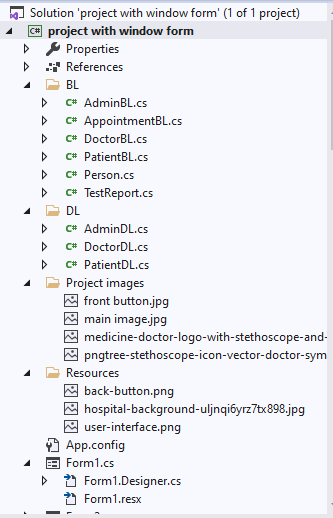
Polymorphism is also one of the core concepts of Object-oriented programming approach. This concept refers to the ability of a function to perform multiple operation under different circumstances. There are two types of Polymorphism. The type of polymorphism used to extend the functionality of common functions in parent and child classes is called Dynamic Polymorphism. I have used Dynamic Polymorphism in few places.

1. toString() functions in both inheritances.
2. The other type of Polymorphism is Static Polymorphism. I have used this polymorphism only for the constructors.

**Advantage**

Polymorphism allowed us to extend the functionalities from the base class to use them for the child classes. Due to polymorphism, our code has become shorter because it didn’t required us to copy paste the whole code and then make changes to it. Dynamic Polymorphism has enabled the programmer to use the same function in different manner. We lacked this when we were making our projects in procedural programming.

* **Design Pattern Implementation**

The directory structure for the project is given below :-

* **Conclusion**

In conclusion, I want to it was my great achievement and I learnt many thing through this project and in file handling I face great difficulty.

* **Code**

public class Person

{

protected string name;

protected string password;

public Person(string name,string password)

{

this.name = name;

this.password = password;

}

public string getName()

{

return name;

}

public string getPassword()

{

return password;

}

public void setName(string name)

{

this.name= name;

}

public void setPassword(string password)

{

this. password=password;

}

public virtual string toStirng()

{

return name+"\t"+password;

}

public class AdminBL:Person

{

public AdminBL(string name,string password):base (name,password)

{

}

}

public class DoctorBL:Person

{

private int salary;

private int fee;

private List<string> transiction=new List<string>();

public List<PatientBL> pat = new List<PatientBL>();

public void setTransiction(List<string> s1)

{

transiction = s1;

}

public List<string> getTransiction()

{

return transiction;

}

public void setList(List<PatientBL> s1)

{

pat = s1;

}

public List<PatientBL> getList()

{

return pat;

}

public void setSalary(int salary)

{

this.salary = salary;

}

public void setFee(int fee)

{

this.fee = fee;

}

public int getSalary()

{

return salary;

}

public int getFee()

{

return fee;

}

public void addPatientList(PatientBL s1)

{

pat.Add(s1);

}

public DoctorBL(string name,string password): base (name,password)

{

}

public DoctorBL(string name, string password,int salary,int fee) : base(name, password)

{

this.salary = salary;

this.fee = fee;

}

public bool deletePatient(string name)

{

foreach (PatientBL x in pat)

{

if (x.getName() == name)

{

pat.Remove(x);

return true;

}

}

return false;

}

public bool addPrescription(string name, string prescription)

{

foreach (PatientBL x in pat)

{

if (x.getName() == name && x.getPrescription()=="")

{

x.setPrescription(prescription);

return true;

}

}

return false;

}

public override string toStirng()

{

return base.toStirng()+"\t"+salary+"\t"+"fee";

}

public class PatientBL:Person

{

private string age;

private string disease;

private string medicine;

private string contact;

private string review;

private string prescription;

private int billPayment;

private AppointmentBL appointment;

private List<TestReport> test;

public PatientBL(string name,string password ): base(name,password)

{

}

public PatientBL(string name, string password,string age,string disease,string medicine,string contact) : base(name, password)

{

this.age = age;

this.disease = disease;

this.medicine = medicine;

this.contact = contact;

this.appointment=new AppointmentBL("","");

this.test=new List<TestReport>();

prescription = "";

}

public int getBill()

{

return billPayment;

}

public void setBill(int bill)

{

this.billPayment=bill;

}

public List<TestReport> getTestReport()

{

return test;

}

public void setTestReport(List<TestReport> test)

{

this.test=test;

}

public AppointmentBL getAppointment()

{

return appointment;

}

public void setAppointmnet(AppointmentBL appointment)

{

this.appointment =appointment;

}

public string getPrescription()

{

return prescription;

}

public void setPrescription(string prescription)

{

this.prescription = prescription;

}

public string getReview()

{

return review;

}

public void setReview(string review)

{

this.review = review;

}

public string getAge()

{

return age;

}

public string getDisease()

{

return disease;

}

public string getMedicine()

{

return medicine;

}

public string getContact()

{

return contact;

}

public void setAge(string age)

{

this.age = age;

}

public void setDisease(string disease)

{

this.disease = disease;

}

public void setMedicine(string medicine)

{

this.medicine = medicine;

}

public void setContact(string contact)

{

this.contact = contact;

}

public override string toStirng()

{

return base.toStirng()+"\t"+age+"\t"+disease+"\t"+medicine+"\t"+contact;

}

public class TestReport

{

private string testName;

private string result;

public TestReport(string name,string result)

{

this.testName=name;

this.result=result;

}

public string getTestName()

{

return testName;

}

public void setTestName(string testName)

public class AdminDL

{

public static List<AdminBL> admins = new List<AdminBL>();

public static void addIntoList(AdminBL p1)

{

admins.Add(p1);

}

public static bool checkAdmin(AdminBL b1)

{

foreach (AdminBL x in admins)

{

if(b1.getName()==x.getName() && b1.getPassword()==x.getPassword())

{

return true;

}

}

return false;

}

public static bool sameAdmin(AdminBL d)

{

foreach (AdminBL x in admins)

{

if (x.getName() == d.getName() )

{

return false;

}

}

return true;

}

public static void storeData(AdminBL a1)

{

string path = "admin.txt";

StreamWriter file = new StreamWriter(path, true);

file.WriteLine(a1.getName() + "," + a1.getPassword());

file.Flush();

file.Close();

}

public static void readData(string path)

{

if(File.Exists(path))

{

StreamReader file = new StreamReader(path);

string record;

while((record=file.ReadLine())!=null)

{

string[] spiltData = record.Split(',');

string name = spiltData[0];

string password = spiltData[1];

AdminBL a = new AdminBL(name, password);

addIntoList(a);

}

file.Close();

}

}

public class DoctorDL

{

public static List<DoctorBL> doctors = new List<DoctorBL>();

public static void addIntolist(DoctorBL d1)

{

doctors.Add(d1);

}

public static DoctorBL checkDoctor(DoctorBL d1)

{

foreach(DoctorBL x in doctors)

{

if(x.getName()==d1.getName() && x.getPassword()==d1.getPassword() )

{

return x;

}

}

return null;

}

public static bool sameDoctor(DoctorBL d)

{

foreach (DoctorBL x in doctors)

{

if (x.getName() == d.getName() )

{

return false;

}

}

return true;

}

public static void storeData(DoctorBL d)

{

string path = "doctor.txt";

StreamWriter file = new StreamWriter(path, true);

string patient = "";

if (d.getList().Count==0)

{

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee());

}

else if( d.getList().Count==1)

{

patient = patient + d.getList()[d.getList().Count-1].getName();

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee()+","+patient);

}

else

{

for (int i = 0; i<d.getList().Count-1; i++)

{

patient = patient + d.getList()[i].getName()+";";

}

patient = patient + d.getList()[d.getList().Count-1].getName();

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee()+","+patient);

}

file.Flush();

file.Close();

}

public static void rewriteData()

{

string path = "doctor.txt";

StreamWriter file = new StreamWriter(path, false);

for(int x=0;x<doctors.Count;x++)

{

DoctorBL d = doctors[x];

string patient = "";

if (d.getList().Count==0)

{

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee());

}

else if (d.getList().Count==1)

{

patient = patient + d.getList()[d.getList().Count-1].getName();

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee()+","+patient);

}

else

{

for (int i = 0; i<d.getList().Count-1; i++)

{

patient = patient + d.getList()[i].getName()+";";

}

patient = patient + d.getList()[d.getList().Count-1].getName();

file.WriteLine(d.getName() + "," + d.getPassword() + "," + d.getSalary() + "," + d.getFee()+","+patient);

}

}

file.Flush();

file.Close();

}

public static void readData(string path)

{

if (File.Exists(path))

{

StreamReader file = new StreamReader(path);

string record;

while ((record = file.ReadLine()) != null)

{

string[] spitlData = record.Split(',');

string name, password,patient;

int salary, fee;

if (spitlData.Length==4)

{

name = spitlData[0];

password = spitlData[1];

salary = int.Parse(spitlData[2]);

fee = int.Parse(spitlData[3]);

DoctorBL d = new DoctorBL(name, password, salary, fee);

DoctorDL.addIntolist(d);

}

else

{

name = spitlData[0];

password = spitlData[1];

salary = int.Parse(spitlData[2]);

fee = int.Parse(spitlData[3]);

patient=spitlData[4];

string[] patientSpilt = patient.Split(';');

List<PatientBL> p1 = new List<PatientBL>();

for(int i=0;i<patientSpilt.Length;i++)

{

PatientBL p = PatientDL.isPatientExist(patientSpilt[i]);

if(p!=null)

{

p1.Add(p);

}

}

DoctorBL d = new DoctorBL(name, password, salary, fee);

d.setList(p1);

DoctorDL.addIntolist(d);

}

//DoctorBL d = new DoctorBL(name, password, salary, fee);

//DoctorDL.addIntolist(d);

}

file.Close();

}

}

public static bool deleteDoctor(string name)

{

foreach(DoctorBL x in doctors)

{

if(x.getName()==name)

{

doctors.Remove(x);

return true;

}

}

return false;

}

public static bool updateDoctor(string name,string fee)

{

foreach(DoctorBL x in doctors)

{

if(x.getName()==name)

{

x.setFee(int.Parse(fee));

return true;

}

}

return false;

}

public static bool updateSalary(string name, string salary)

{

foreach (DoctorBL x in doctors)

{

if (x.getName() == name)

{

x.setSalary(int.Parse(salary));

return true;

}

}

return false;

}

public class PatientDL

{

public static List<PatientBL> patient = new List<PatientBL>();

public static void addIntoList(PatientBL p1)

{

patient.Add(p1);

}

public static bool deletePatient(string name)

{

foreach (PatientBL x in patient)

{

if (x.getName() == name)

{

patient.Remove(x);

return true;

}

}

return false;

}

public static PatientBL isPatientExist(string name)

{

foreach (PatientBL x in PatientDL.patient)

{

if (x.getName()==name)

{

return x;

}

}

return null;

}

public static bool checkPatient(PatientBL b1)

{

foreach (DoctorBL y in DoctorDL.doctors)

{

foreach (PatientBL x in y.getList())

{

if (DoctorDL.doctors != null && DoctorDL.doctors.Count > 0)

{

if (b1.getName() == x.getName() && b1.getPassword() == x.getPassword())

{

return true;

}

}

}

}

return false;

}

public static bool samePatient(PatientBL d,DoctorBL d1)

{

foreach (PatientBL x in d1.getList())

{

if (x.getName() == d.getName() )

{

return false;

}

}

return true;

}

public static PatientBL isPatientExist(PatientBL p)

{

foreach(DoctorBL y in DoctorDL.doctors)

{

foreach(PatientBL x in y.getList())

{

if(p.getName()==x.getName())

{

return x;

}

}

}

return null;

}

public static bool samePatient(PatientBL p)

{

foreach(DoctorBL y in DoctorDL.doctors)

{

foreach(PatientBL x in y.getList())

{

if(x.getName()==p.getName())

{

return true;

}

}

}

return false;

}

public static DoctorBL patientDoctor(PatientBL p1)

{

foreach(DoctorBL y in DoctorDL.doctors)

{

foreach(PatientBL x in y.getList())

{

if(p1.getName()==x.getName())

{

return y;

}

}

}

return null;

}

public static bool checkTestReport(TestReport t1,string name,DoctorBL doctor)

{

foreach(PatientBL y in doctor.getList())

{

if(y.getName()==name)

{

y.getTestReport().Add(t1);

return true;

}

}

return false;

}

public static int addBillPayment(string name, DoctorBL d1)

{

foreach(PatientBL p in d1.getList())

{

if(name==p.getName())

{

return p.getBill();

}

}

return 0;

}

public static void reWriteData()

{

string path = "patient.txt";

StreamWriter file = new StreamWriter(path, false) ;

foreach(PatientBL p in patient )

{

string test = "";

if (p.getTestReport().Count==0)

{

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime());

}

else if (p.getTestReport().Count==1)

{

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult();

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime()+test);

}

else

{

for (int i = 0; i<p.getTestReport().Count; i++)

{

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult()+"\*";

}

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult();

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime()+test);

}

}

file.Flush();

file.Close();

}

public static void storeData(PatientBL p)

{

string path = "patient.txt";

string test="";

StreamWriter file = new StreamWriter(path, true);

if (p.getTestReport().Count==0)

{

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime());

}

else if (p.getTestReport().Count==1)

{

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult();

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime()+test);

}

else

{

for(int i=0;i<p.getTestReport().Count;i++)

{

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult()+"\*";

}

test=test+p.getTestReport()[p.getTestReport().Count-1].getTestName()+"\*"+p.getTestReport()[p.getTestReport().Count-1].getResult();

file.WriteLine(p.getName()+","+p.getPassword()+","+p.getAge()+","+p.getDisease()+","+p.getMedicine()+","+p.getContact()+","+p.getReview()+","+p.getPrescription()+","+p.getBill()+","+p.getAppointment().getDay()+"!"+p.getAppointment().getTime()+test);

}

file.Flush();

file.Close();

}

public partial class MainMenu : Form

{

public MainMenu()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Visible = false;

frmDoctorLogin s = new frmDoctorLogin();

s.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Visible = false;

frmPatientLogin p = new frmPatientLogin();

p.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Visible = false;

Admin a = new Admin();

a.Show();

}

public partial class Admin : Form

{

public Admin()

{

InitializeComponent();

}

private void button3\_Click(object sender, EventArgs e)

{

SignUp s = new SignUp();

s.Show();

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

MainMenu m = new MainMenu();

m.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

signIn s = new signIn();

s.Show();

this.Close();

}

private void Admin\_Load(object sender, EventArgs e)

{

this.Visible = false;

}

public partial class signIn : Form

{

public signIn()

{

InitializeComponent();

}

public AdminBL inputAdmin()

{

string name = txtname.Text;

string password = Password.Text;

AdminBL p1;

if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(password))

{

p1 = new AdminBL(name, password);

return p1;

}

return null;

}

private void signIn\_Load(object sender, EventArgs e)

{

pictureBox1.BorderStyle = BorderStyle.None;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Admin a = new Admin();

a.Show();

}

private void front\_Click(object sender, EventArgs e)

{

AdminBL b1 = inputAdmin();

if(b1!=null)

{

bool flag = AdminDL.checkAdmin(b1);

if(flag)

{

this.Close();

frmAdminInterface f = new frmAdminInterface();

f.Show();

}

else

{

result.ForeColor = Color.Red;

emptyInput();

result.Text = "You have enter wrong input";

}

}

else

{

emptyInput();

result.ForeColor = Color.Red;

result.Text = "You have enter wrong input";

}

}

public void emptyInput()

{

txtname.Text = "";

Password.Text = "";

}public partial class SignUp : Form

{

public SignUp()

{

InitializeComponent();

}

public void emptyInput()

{

txtname.Text = "";

Password.Text = "";

}

public AdminBL inputAdmin()

{

string name = txtname.Text;

string password = Password.Text;

AdminBL p1;

if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(password))

{

p1 = new AdminBL(name, password);

return p1;

}

return null;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Admin a = new Admin();

a.Show();

}

private void front\_Click(object sender, EventArgs e)

{

AdminBL b1 = inputAdmin();

if (b1 != null)

{

bool flag = AdminDL.sameAdmin(b1);

if (flag)

{

AdminDL.storeData(b1);

AdminDL.addIntoList(b1);

emptyInput();

result.Text = "Admin is added";

result.ForeColor = Color.Black;

}

else

{

emptyInput();

result.Text = "already ths admin is exist";

result.ForeColor = Color.Red;

}

}

else

{

emptyInput();

result.Text = "you have enter wrong input";

result.ForeColor = Color.Red;

}

}

public partial class SignUp : Form

{

public SignUp()

{

InitializeComponent();

}

public void emptyInput()

{

txtname.Text = "";

Password.Text = "";

}

public AdminBL inputAdmin()

{

string name = txtname.Text;

string password = Password.Text;

AdminBL p1;

if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(password))

{

p1 = new AdminBL(name, password);

return p1;

}

return null;

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Admin a = new Admin();

a.Show();

}

private void front\_Click(object sender, EventArgs e)

{

AdminBL b1 = inputAdmin();

if (b1 != null)

{

bool flag = AdminDL.sameAdmin(b1);

if (flag)

{

AdminDL.storeData(b1);

AdminDL.addIntoList(b1);

emptyInput();

result.Text = "Admin is added";

result.ForeColor = Color.Black;

}

else

{

emptyInput();

result.Text = "already ths admin is exist";

result.ForeColor = Color.Red;

}

}

else

{

emptyInput();

result.Text = "you have enter wrong input";

result.ForeColor = Color.Red;

}

}

public partial class frmAdminInterface : Form

{

public frmAdminInterface()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Admin s = new Admin();

s.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminViewRecord r = new frmAdminViewRecord();

r.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminViewPatient p = new frmAdminViewPatient();

p.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminViewBill a = new frmAdminViewBill();

a.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminUpdateSalary s = new frmAdminUpdateSalary();

s.Show();

}

private void button9\_Click(object sender, EventArgs e)

{

this.Close();

frmDeleteDoctor a = new frmDeleteDoctor();

a.Show();

}

private void AddDoctor\_Click(object sender, EventArgs e)

{

frmAddDoctor dot = new frmAddDoctor();

dot.Show();

this.Close();

}

private void AdminUpdateFee\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminUpdateFee f = new frmAdminUpdateFee();

f.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminViewReviews a = new frmAdminViewReviews();

a.Show();

}

public partial class frmAdminUpdateFee : Form

{

public frmAdminUpdateFee()

{

InitializeComponent();

}

public void emptyInput()

{

textBox1.Text = "";

textBox2.Text = "";

}

DataTable table = new DataTable();

private void frmAdminUpdateFee\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("salary");

table.Columns.Add("Fee");

printDoctor();

dataGridView1.DataSource = table;

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printDoctor();

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

public void printDoctor()

{

foreach (DoctorBL d in DoctorDL.doctors)

{

table.Rows.Add(d.getName(), d.getPassword(), d.getSalary(), d.getFee());

}

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

bool flag1 = AdminDL.validationOfAlphabet(name);

string fee = textBox2.Text;

bool flag2 = AdminDL.validationOfNumber(fee);

if (flag1 && flag2)

{

bool flag = DoctorDL.updateDoctor(name, fee);

if (flag == true)

{

DoctorDL.rewriteData();

updataGrid();

label3.Text="Fee update sucessfully!";

label3.ForeColor = Color.Green;

}

else

{

label3.Text="You enter wrong data";

label3.ForeColor = Color.Red;

}

}

else

{

label3.Text = "You enter wrong data";

label3.ForeColor = Color.Red;

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

public partial class frmAdminUpdateSalary : Form

{

public frmAdminUpdateSalary()

{

InitializeComponent();

}

DataTable table = new DataTable();

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("salary");

table.Columns.Add("Fee");

printDoctor();

dataGridView1.DataSource = table;

}

public void printDoctor()

{

foreach (DoctorBL d in DoctorDL.doctors)

{

table.Rows.Add(d.getName(), d.getPassword(), d.getSalary(), d.getFee());

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

public void emptyInput()

{

textBox1.Text = "";

textBox2.Text = "";

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

bool flag1 = AdminDL.validationOfAlphabet(name);

string salary = textBox2.Text;

bool flag2 = AdminDL.validationOfNumber(salary);

if (flag1 && flag2)

{

bool flag = DoctorDL.updateSalary(name, salary);

if (flag)

{

DoctorDL.rewriteData();

emptyInput();

label3.Text="SucessFully updated !";

label3.ForeColor = Color.Green;

updataGrid();

}

else

{

emptyInput();

label3.Text= "You enter wrong Data";

label3.ForeColor = Color.Red;

}

}

else

{

emptyInput();

label3.Text = "You enter wrong Data";

label3.ForeColor = Color.Red;

}

}

private void frmAdminUpdateSalary\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("salary");

table.Columns.Add("Fee");

printDoctor();

dataGridView1.DataSource = table;

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printDoctor();

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

public partial class frmAdminViewBill : Form

{

public frmAdminViewBill()

{

InitializeComponent();

}

DataTable table = new DataTable();

private void frmAdminViewBill\_Load(object sender, EventArgs e)

{

table.Columns.Add("Patient");

table.Columns.Add("Bill Payment");

printBillPayment();

dataGridView1.DataSource = table;

}

public void printBillPayment()

{

foreach (DoctorBL y in DoctorDL.doctors)

{

foreach (string x in y.getTransiction())

{

table.Rows.Add(y.getName() , x);

}

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

public partial class frmAdminViewPatient : Form

{

public frmAdminViewPatient()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

DataTable table = new DataTable();

private void frmAdminViewPatient\_Load(object sender, EventArgs e)

{

table.Columns.Add("Doctor");

table.Columns.Add("Patient");

printPatientData();

dataGridView1.DataSource = table;

}

public void printPatientData()

{

foreach (DoctorBL y in DoctorDL.doctors)

{

string doctor = y.getName();

foreach (PatientBL x in y.getList())

{

table.Rows.Add(doctor,x.getName());

}

}

}

public partial class frmAdminViewRecord : Form

{

public frmAdminViewRecord()

{

InitializeComponent();

}

DataTable table = new DataTable();

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printDoctor();

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

private void frmAdminViewRecord\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("salary");

table.Columns.Add("Fee");

updataGrid();

}

public void printDoctor()

{

foreach(DoctorBL d in DoctorDL.doctors)

{

table.Rows.Add(d.getName(), d.getPassword(), d.getSalary(), d.getFee());

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

. public partial class frmAdminViewReviews : Form

{

public frmAdminViewReviews()

{

InitializeComponent();

}

private DataTable table1 = new DataTable("table");

private void frmAdminViewReviews\_Load(object sender, EventArgs e)

{

table1.Columns.Add("Name");

table1.Columns.Add("Reviews");

printReview();

dataGridView1.DataSource = table1;

dataGridView1.Columns["Reviews"].Width = 1000;

}

public void printReview()

{

string line;

foreach (PatientBL x in PatientDL.patient)

{

line = x.getReview() ;

if (line != "")

{

table1.Rows.Add(x.getName(), line);

}

}

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table1.Rows.Clear();

dataGridView1.DataSource = table1;

dataGridView1.Refresh();

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

public partial class frmDeleteDoctor : Form

{

public frmDeleteDoctor()

{

InitializeComponent();

}

DataTable table = new DataTable();

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmAdminInterface a = new frmAdminInterface();

a.Show();

}

public void printDoctor()

{

foreach (DoctorBL d in DoctorDL.doctors)

{

table.Rows.Add(d.getName(), d.getPassword(), d.getSalary(), d.getFee());

}

}

public void emptyInput()

{

textBox1.Text = "";

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printDoctor();

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

bool flag = DoctorDL.deleteDoctor(name);

if (flag)

{

DoctorDL.rewriteData();

emptyInput();

updataGrid();

label3.Text="Doctor delete sucesfully";

label3.ForeColor = Color.Green;

}

else

{

emptyInput();

label3.Text="sorry you enter wrong data!";

label3.ForeColor = Color.Red;

}

}

private void frmDeleteDoctor\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("salary");

table.Columns.Add("Fee");

printDoctor();

dataGridView1.DataSource = table;

}

public partial class frmDoctorAddAppointment : Form

{

public frmDoctorAddAppointment(DoctorBL d)

{

doctor = d;

InitializeComponent();

}

DataTable table = new DataTable();

DoctorBL doctor;

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface a = new frmDoctorInterface(doctor);

a.Show();

}

public void printPatientOfDoctor(DoctorBL d)

{

foreach (PatientBL x in d.getList())

{

table.Rows.Add(x.getName(), x.getPassword(), x.getAge(), x.getDisease(), x.getMedicine(), x.getContact());

}

}

private void frmDoctorAddAppointment\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("Age");

table.Columns.Add("Disease");

table.Columns.Add("Medicine");

table.Columns.Add("Contact");

printPatientOfDoctor(doctor);

dataGridView1.DataSource = table;

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

PatientBL p = PatientDL.Patient(name, doctor);

if (p != null)

{

if (p.getAppointment().getDay() == "" && p.getAppointment().getTime() == "")

{

string date = dateTimePicker1.Value.DayOfWeek.ToString();

string time= dateTimePicker1.Value.TimeOfDay.ToString();

AppointmentBL a1 = new AppointmentBL(date,time);

bool flag = DoctorDL.addappointmemt(name, a1, doctor);

if (flag)

{

PatientDL.reWriteData();

label3.Text=("Appointment added successfully!");

label3.ForeColor = Color.Green;

}

else

{

label3.Text=("You enter wrong input!");

label3.ForeColor = Color.Red;

}

}

else

{

label3.Text=("this patient has already has assignment!");

label3.ForeColor = Color.Red;

}

}

else

{

label3.Text=("You enter wrong input !");

label3.ForeColor = Color.Red;

}

}

public partial class frmDoctorAddBillPayment : Form

{

public frmDoctorAddBillPayment(DoctorBL d)

{

doctor = d;

InitializeComponent();

}

DoctorBL doctor;

DataTable table = new DataTable();

private void frmDoctorAddBillPayment\_Load(object sender, EventArgs e)

{

label4.Visible = false;

label5.Visible = false;

label6.Visible = false;

textBox2.Visible = false;

button3.Visible = false;

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("Age");

table.Columns.Add("Disease");

table.Columns.Add("Medicine");

table.Columns.Add("Contact");

printPatientOfDoctor(doctor);

dataGridView1.DataSource = table;

}

public void printPatientOfDoctor(DoctorBL d)

{

foreach (PatientBL x in d.getList())

{

table.Rows.Add(x.getName(), x.getPassword(), x.getAge(), x.getDisease(), x.getMedicine(), x.getContact());

}

}

private void button2\_Click(object sender, EventArgs e)

{

name = textBox1.Text;

flag = PatientDL.addBillPayment(name, doctor);

if (flag != 0)

{

Start();

}

else

{

label3.Text = ("this client has given hhis money");

label3.ForeColor = Color.Red;

}

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface a = new frmDoctorInterface(doctor);

a.Show();

}

string name;

int flag;

string amount;

public void Start()

{

label4.Visible = true;

label5.Visible = true;

label5.Text = flag.ToString();

label6.Visible = true;

textBox2.Visible = true;

button2.Visible = false;

button3.Visible = true;

}

private void button3\_Click(object sender, EventArgs e)

{

amount = textBox2.Text;

bool check = PatientDL.validationOfBill(amount);

if (check)

{

if (flag == int.Parse(amount))

{

doctor.getTransiction().Add(amount.ToString());

foreach (PatientBL p in doctor.getList())

{

if (name == p.getName())

{

p.setBill(0);

PatientDL.reWriteData();

break;

}

}

label3.Text = ("payment transfererd");

label3.ForeColor = Color.Green;

}

else

{

label3.Text = "Please enter correct input";

label3.ForeColor = Color.Red;

}

}

else

{

label3.Text = ("wrong input!");

label3.ForeColor = Color.Red;

}

}

public partial class frmDoctorAddPatient : Form

{

public frmDoctorAddPatient(DoctorBL d)

{

InitializeComponent();

doctor = d;

}

DoctorBL doctor;

private void emptyData()

{

txtName.Text = "";

Password.Text = "";

salary.Text = "";

fee.Text = "";

textBox2.Text = "";

textBox1.Text = "";

}

private PatientBL takeInput()

{

string name = txtName.Text;

bool checkName = AdminDL.validationOfAlphabet(name);

string password = Password.Text;

bool checkPassword = AdminDL.validationOfPassword(password);

string age = salary.Text;

bool checkAge = AdminDL.validationOfNumber(age);

string disease = fee.Text;

string medicine = textBox2.Text;

string contact = textBox1.Text;

bool checkContact = AdminDL.validationOfNumber(contact);

bool contactLength = PatientDL.validationOfContactLength(contact);

if (checkName && checkAge && checkContact && contactLength && checkPassword)

{

if (!string.IsNullOrEmpty(name) && !string.IsNullOrEmpty(password) && !string.IsNullOrEmpty(age) && !string.IsNullOrEmpty(disease) && !string.IsNullOrEmpty(medicine) && !string.IsNullOrEmpty(contact))

{

PatientBL p1;

p1 = new PatientBL(name, password, age, disease, medicine, contact);

return p1;

}

}

return null;

}

private void button1\_Click(object sender, EventArgs e)

{

PatientBL p1 =takeInput();

if (p1 != null)

{

bool flag = PatientDL.samePatient(p1);

if (!flag)

{

doctor.addPatientList(p1);

PatientDL.addIntoList(p1);

p1.setBill(doctor.getFee() + 100);

PatientDL.storeData(p1);

DoctorDL.rewriteData();

emptyData();

label4.Text=("Patient sucessfully login");

label4.ForeColor = Color.Green;

}

else

{

emptyData();

label4.Text=("this patient under another doctor custidiy");

label4.ForeColor = Color.Red;

}

}

else

{

emptyData();

label4.Text=("you enter wrong input");

label4.ForeColor = Color.Red;

}

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface a = new frmDoctorInterface(doctor);

a.Show();

}

public partial class frmDoctorAddPrescription : Form

{

public frmDoctorAddPrescription(DoctorBL d)

{

InitializeComponent();

doctor = d;

}

DoctorBL doctor;

DataTable table = new DataTable();

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface a = new frmDoctorInterface(doctor);

a.Show();

}

private void frmDoctorAddPrescription\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("Age");

table.Columns.Add("Disease");

table.Columns.Add("Medicine");

table.Columns.Add("Contact");

printPatientOfDoctor(doctor);

dataGridView1.DataSource = table;

}

public void printPatientOfDoctor(DoctorBL d)

{

foreach (PatientBL x in d.getList())

{

table.Rows.Add(x.getName(), x.getPassword(), x.getAge(), x.getDisease(), x.getMedicine(), x.getContact());

}

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

string prescription = richTextBox1.Text;

bool flag = doctor.addPrescription(name, prescription);

if (flag)

{

PatientDL.reWriteData();

label3.Text=("SUcessfully added!"); ;

label3.ForeColor = Color.Green;

}

else

{

label3.Text=("Sorry you enter wrong patient!");

label3.ForeColor = Color.Red;

}

}

public partial class frmDoctorCancelAppointment : Form

{

public frmDoctorCancelAppointment(DoctorBL d)

{

doctor = d;

InitializeComponent();

}

DoctorBL doctor;

DataTable table = new DataTable();

private void frmDoctorCancelAppointment\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Appointments");

printAppointment(doctor);

dataGridView1.DataSource = table;

dataGridView1.Columns[1].Width = 500;

}

public void printAppointment(DoctorBL d1)

{

foreach (PatientBL y in d1.getList())

{

if (y.getAppointment().getDay() != "" && y.getAppointment().getDay() != "")

{

string line = " has appointment with you at that " + y.getAppointment().getDay() + " and " + y.getAppointment().getTime();

table.Rows.Add(y.getName(), line);

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

PatientBL p1 = DoctorDL.deleteAppointment(doctor, name);

if (p1 != null)

{

bool flag = DoctorDL.deletePatientAppointment(p1);

if (flag)

{

PatientDL.reWriteData();

emptyInput();

updataGrid();

label3.Text="Appointment delete succesfully!";

label3.ForeColor = Color.Green;

}

else

{

emptyInput();

label3.Text="this patient has no appointment";

label3.ForeColor = Color.Red;

}

}

else

{

emptyInput();

label3.ForeColor = Color.Red;

label3.Text=("sorry there is no such type of patient!");

}

}

public void emptyInput()

{

textBox1.Text = "";

}

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface d = new frmDoctorInterface(doctor);

d.Show();

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printAppointment(doctor);

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

public partial class frmDoctorDeletePatient : Form

{

public frmDoctorDeletePatient(DoctorBL d)

{

InitializeComponent();

doctor = d;

}

DoctorBL doctor;

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmDoctorInterface a = new frmDoctorInterface(doctor);

a.Show();

}

DataTable table = new DataTable();

private void emptyInput()

{

textBox1.Text = "";

}

private void frmDoctorDeletePatient\_Load(object sender, EventArgs e)

{

table.Columns.Add("Name");

table.Columns.Add("Password");

table.Columns.Add("Age");

table.Columns.Add("Disease");

table.Columns.Add("Medicine");

table.Columns.Add("Contact");

printPatientOfDoctor(doctor);

dataGridView1.DataSource = table;

}

public void printPatientOfDoctor(DoctorBL d)

{

foreach (PatientBL x in d.getList())

{

table.Rows.Add(x.getName(), x.getPassword(), x.getAge(), x.getDisease(), x.getMedicine(), x.getContact());

}

}

private void button2\_Click(object sender, EventArgs e)

{

string name = textBox1.Text;

bool flag= doctor.deletePatient(name);

bool flag1 = PatientDL.deletePatient(name);

if (flag && flag1)

{

emptyInput();

PatientDL.reWriteData();

updataGrid();

label3.Text = "Patient successfully deleted";

label3.ForeColor = Color.Green;

}

else

{

emptyInput();

label3.Text = "you enter wrong Input";

label3.ForeColor = Color.Red;

}

}

private void updataGrid()

{

dataGridView1.DataSource = null;

table.Rows.Clear();

printPatientOfDoctor(doctor);

dataGridView1.DataSource = table;

dataGridView1.Refresh();

}

public partial class frmPatientViewPrescription : Form

{

public frmPatientViewPrescription(PatientBL p)

{

InitializeComponent();

patient = p;

}

PatientBL patient;

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

frmPatientInterface a = new frmPatientInterface(patient);

a.Show();

}

private void frmPatientViewPrescription\_Load(object sender, EventArgs e)

{

richTextBox1.Text = patient.getPrescription();

}

* **Images**

****

Figure :1

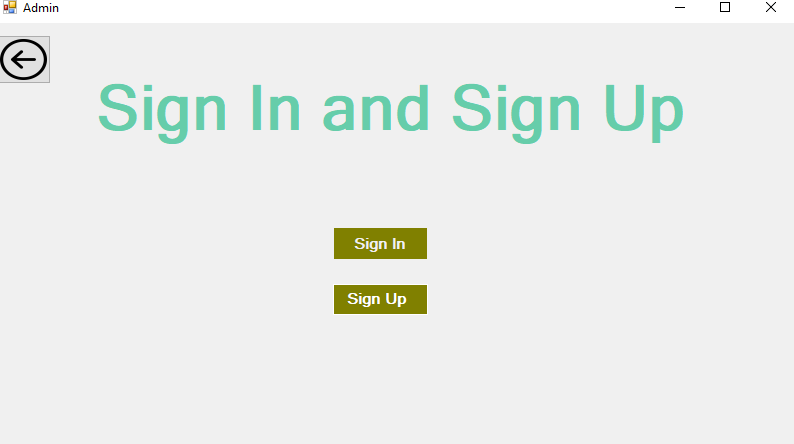


Figure :2

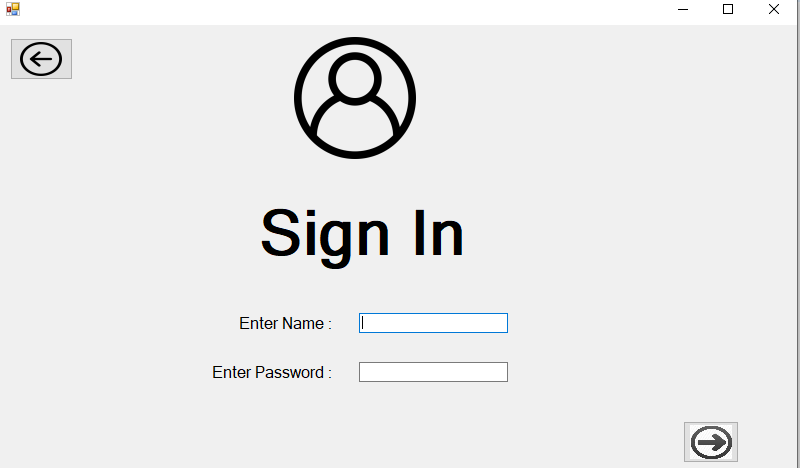


Figure :3

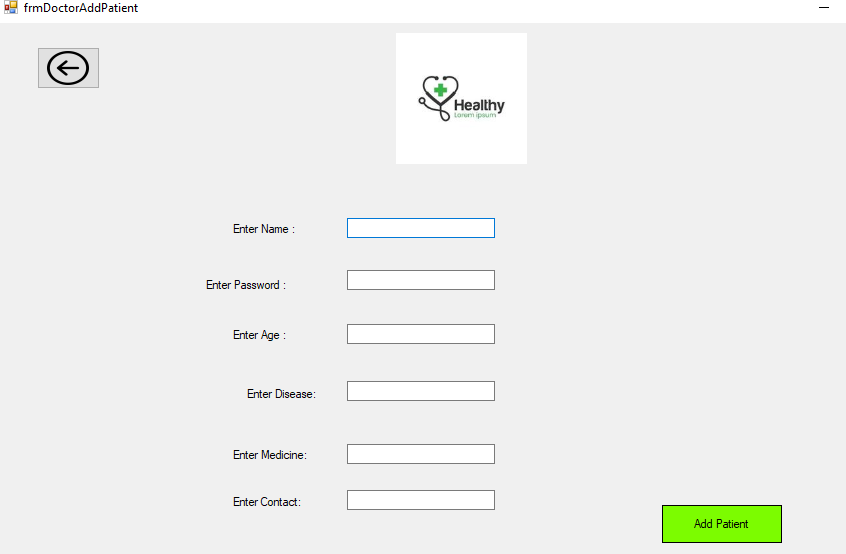


Figure :4

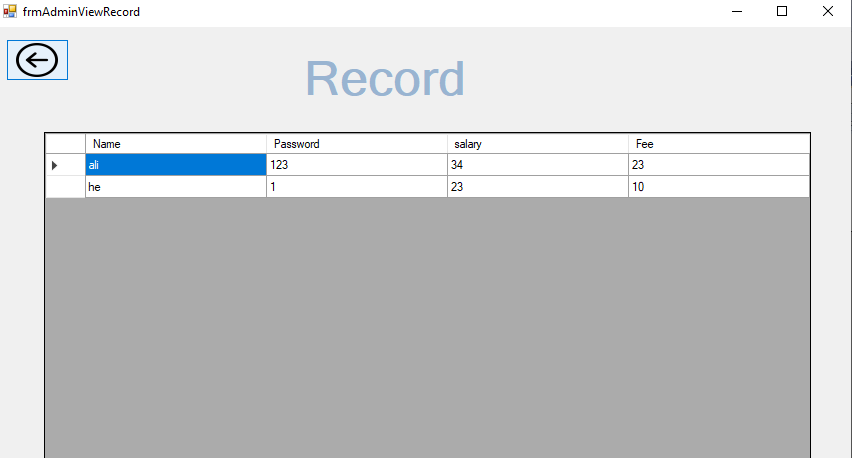


Figure :5

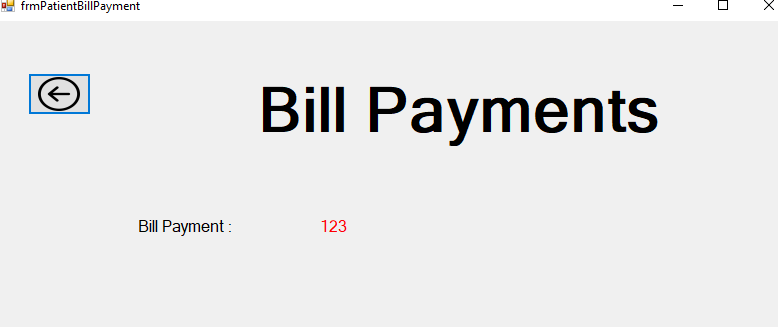


Figure :6

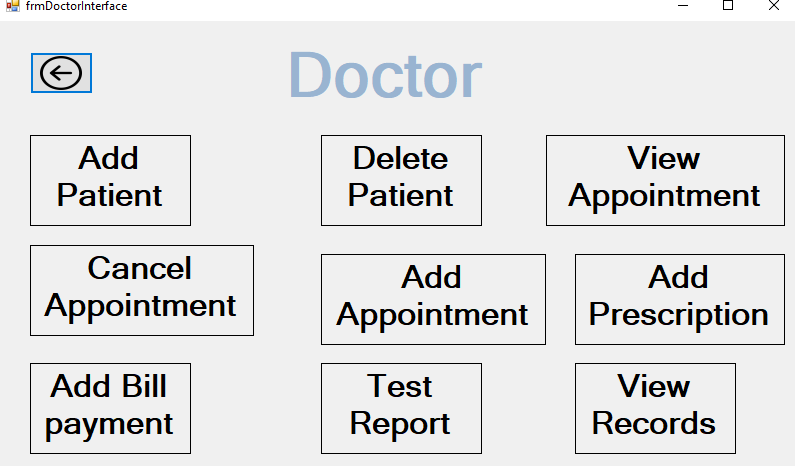


Figure :7

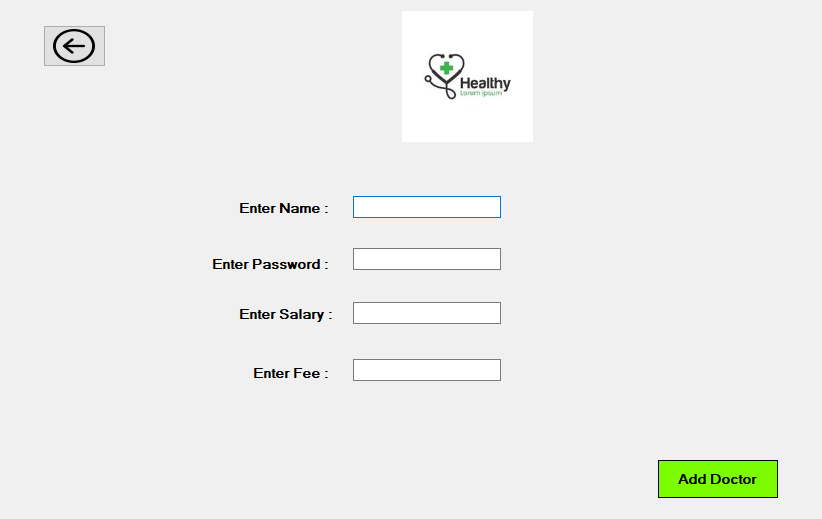


Figure :8

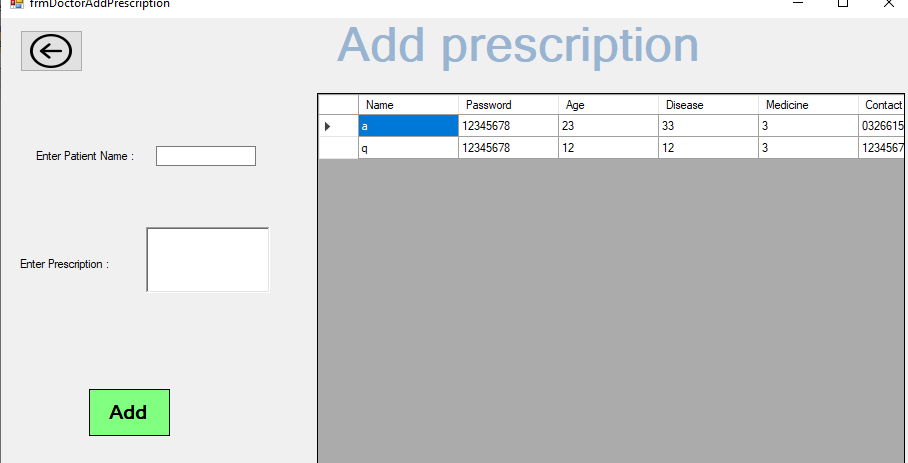


Figure :9

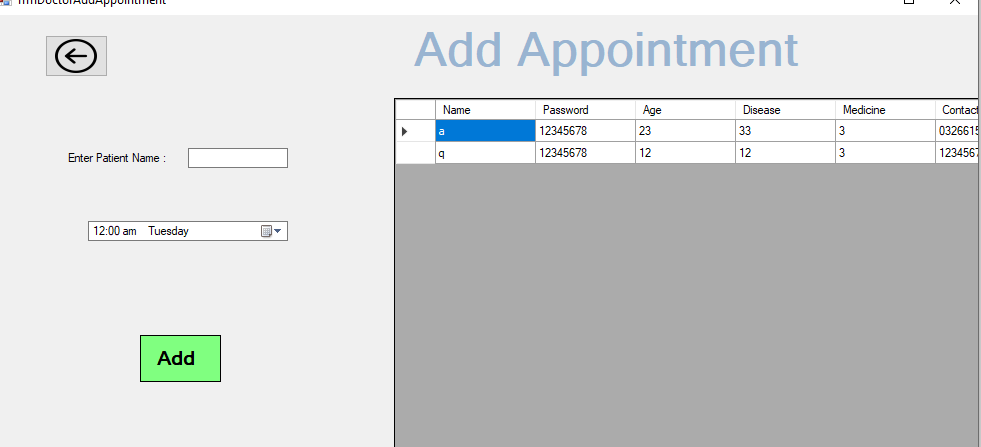


Figure :10

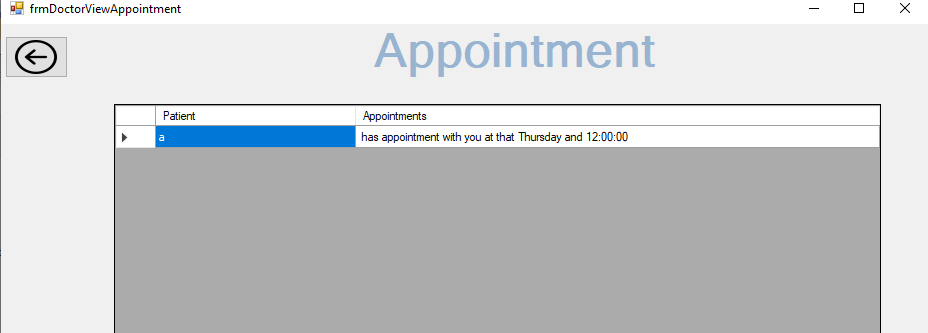


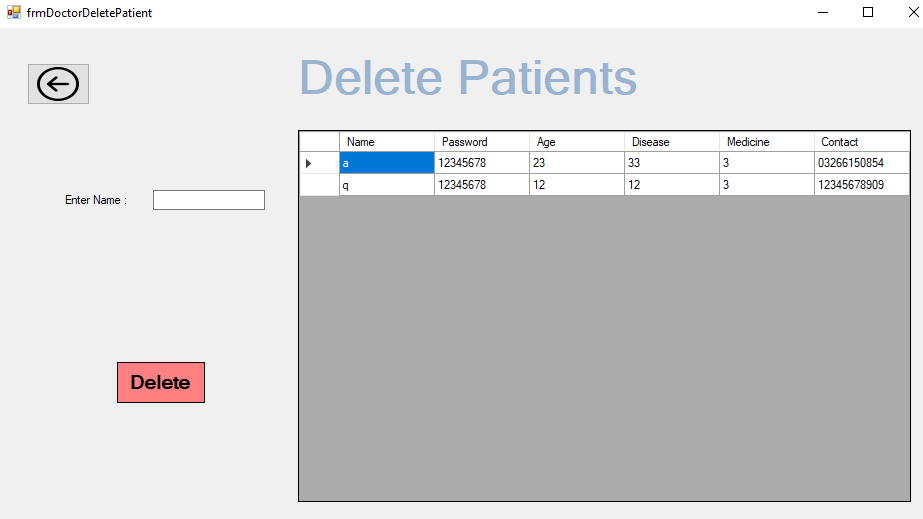
Figure :11

Figure :12

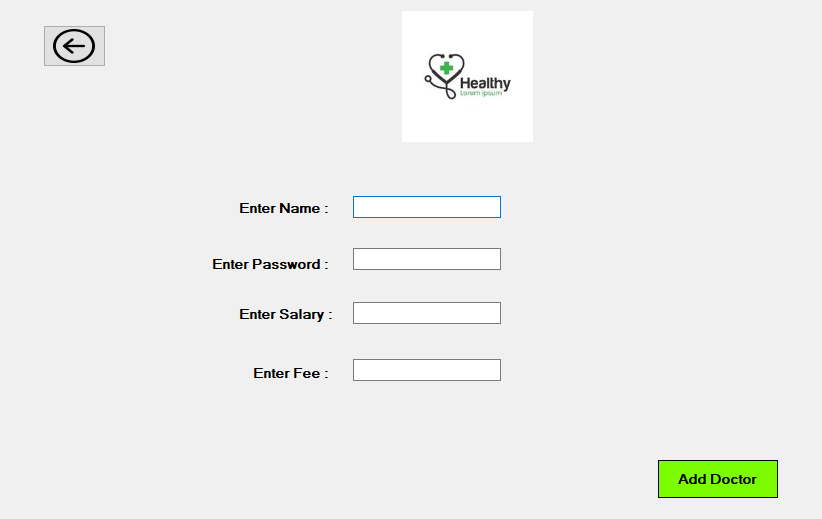


Figure :13

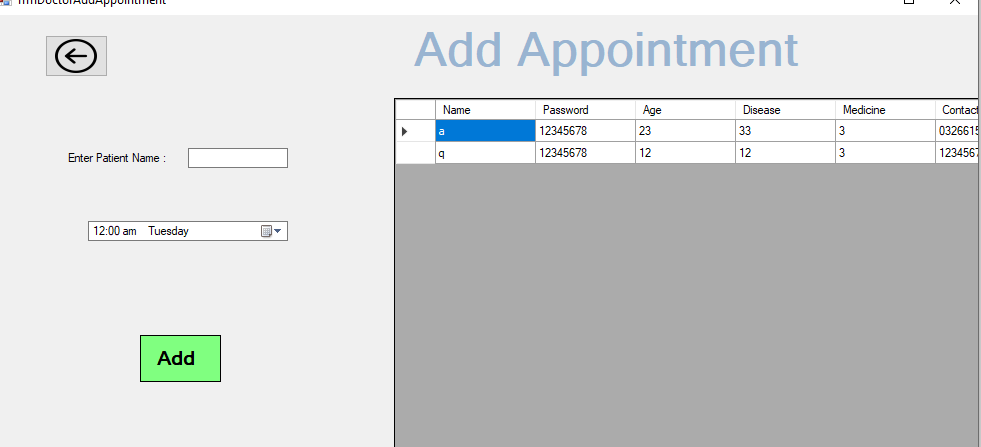


Figure :14