

# Project Lab File



SUBMITTED TO:

Mr Rupesh Mishra

SUBMITTED BY:

Palak Agarwal

2K10-MRCE-CSE-090

## **ACKNOWLEDGEMENTS**

I would like to express my special thanks of gratitude to Mr. Rupesh Mishra who gave me the golden opportunity to do this wonderful project on the topic Patient Billing System which also helped me in doing a lot of Research and I came to know about so many new things I am really thankful to them.

Secondly I would also like to thank my parents and friends who helped me a lot in finishing this project within the limited time.

I am making this project not only for marks but to also increase my knowledge.

THANKS AGAIN TO ALL WHO HELPED ME.

Name: Palak Agarwal  
Roll No: 2K10-MRCE-CSE-090

## **CANDIDATE'S DECLARATION**

I hereby declare that work presented in this project report entitled "Patient Billing System", is an authentic Record of our own work carried out under the guidance of (Mrs. Mamta) at Manav Rachna College of Engineering, Faridabad.

Place: Faridabad

Date: 2<sup>nd</sup> November, 2013

Name: Palak Agarwal

Roll No: 2K10-MRCE-CSE-090

# **Project On Patient Billing System**

# Project Report

## Table of Contents

1. Introduction
  - 1.1. Title
  - 1.2. Objective and scope
  - 1.3. Introduction to project
2. Requirement Elicitation and Analysis
  - 2.1. Requirement Analysis
  - 2.2. Requirement Elicitation
    - 2.2.1. Use Case Diagrams
    - 2.2.2. Sequence Diagram for a System
    - 2.2.3. Activity Diagram
    - 2.2.4. Class Diagram
3. Design specification
  - 3.1. Entity relationship diagram
  - 3.2. E-R to Database Design
4. User interface design specification
5. Coding
  - 5.1. Printout of code (complete code)
6. System testing specification
  - 6.1. Description of test cases
7. Conclusion
  - 7.1. Summary
  - 7.2. Limitation
  - 7.3. Future work
8. References

# 1. Introduction

## 1.1. Title

Development of a feature-rich, practical Patient Billing Software (PBS) for a hospital

## 1.2. Objective and scope

This project is aimed at developing a patient billing software that is of importance to a hospital. The PBS is a local software system. This system can be used to maintain the location (bed) of each patient either in the ward or the ICU. Information about the patient and the charges to be paid is also stored.

## 1.3. Introduction to project

List of Inputs

1. Login Details
2. Details of Patient
3. Enter the Bed details
4. Tracking of Patient
5. Register New Patients
6. Enter the Patient Charges

List of Outputs

1. View Patient details
2. View Bill details of the patient
3. View test charges
4. Generate Bill Amount

## 2. Requirement Elicitation and Analysis

### 2.1. Requirement Analysis

Number of Modules:

The system after careful analysis has been identified to be presenting the following modules:

1. Administrator
2. Doctor
3. Patient

#### Administrator:

This module is having the facility of maintaining the records of patients in the hospital and can also register the new patients. He maintains all the rights to view details of patients and billing generation regarding the patients. The amount to be paid based on the treatment given towards selected test. The bill generation of the patient is also looked after by the administrator. Overall the administrator looks over the entire work of the Receptionist and the Accountant.

#### Doctor

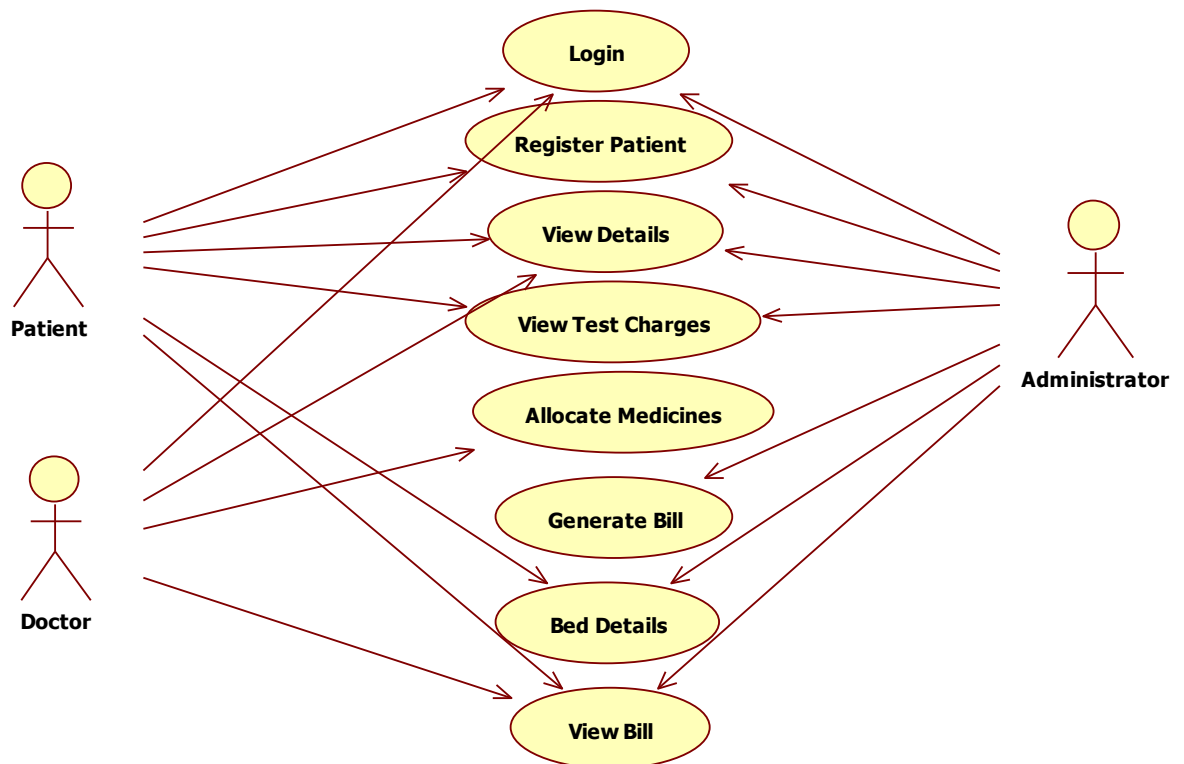
This module is dealing with the details of allocating the medicines to the patients. New Doctors can register themselves on the system and login to access all. Doctor can track the history of the patient.

#### Patient

This module is the main actor in our system. He can register itself at the system or ask the administrator to do to so. He can view its details as on the system or view different ward charges for the bed. Patient can view test charges or view the bill charges allocated to them by the hospital.

## 2.2. Requirement Elicitation

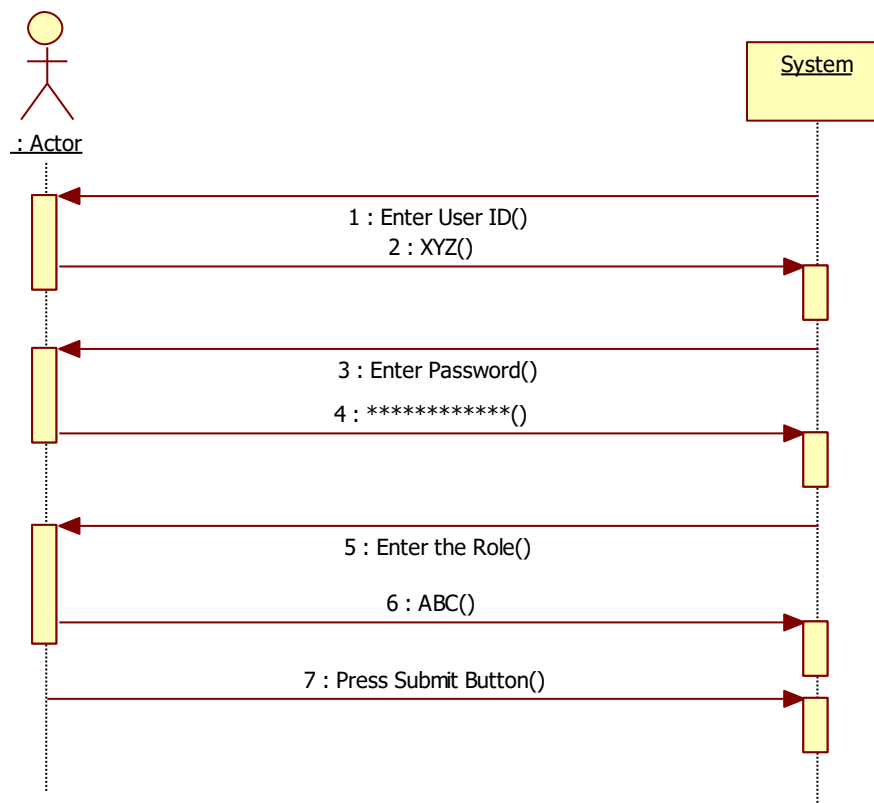
### 2.2.1. Use Case Diagrams



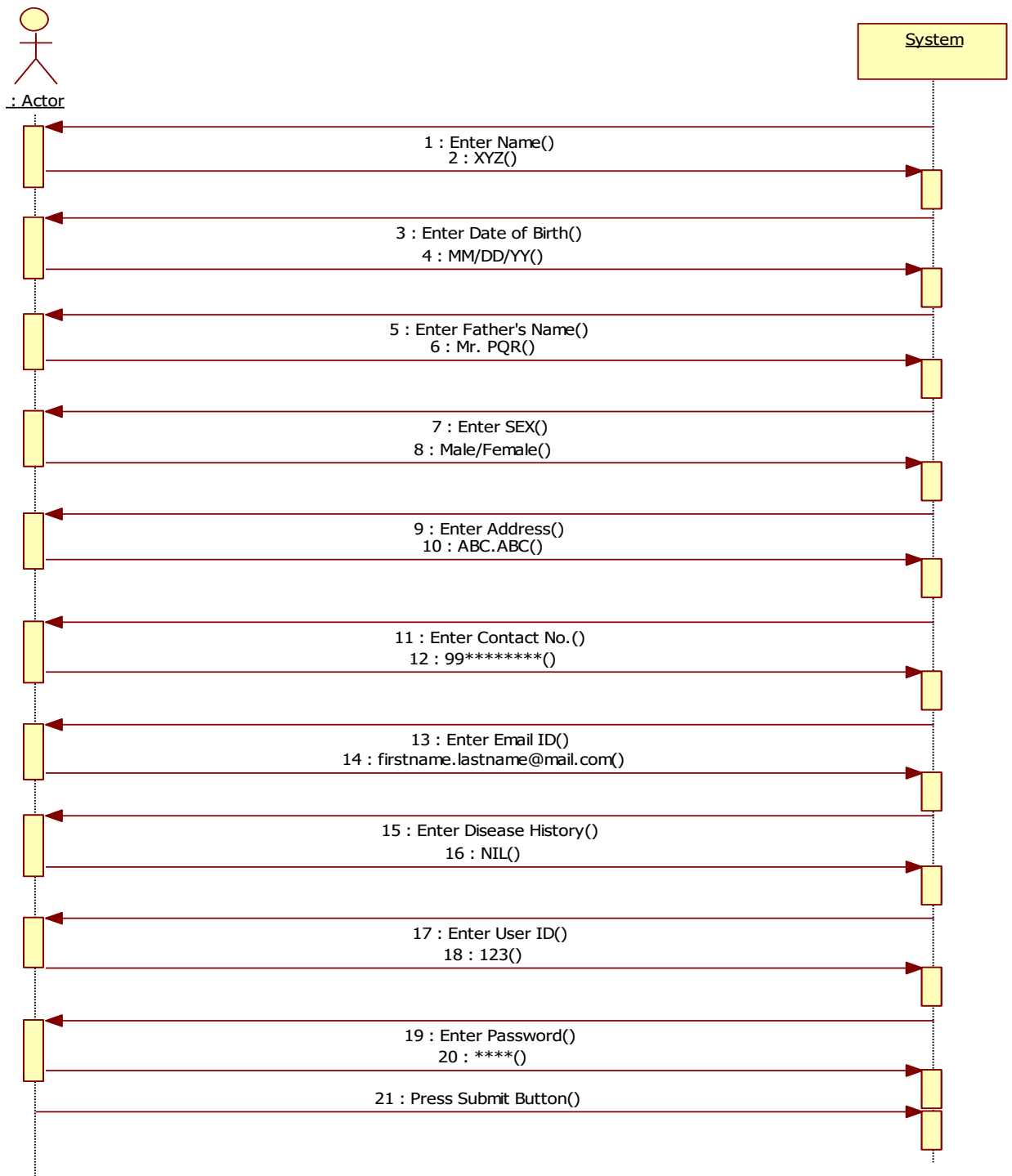


## 2.2.2 Sequence Diagram for a System

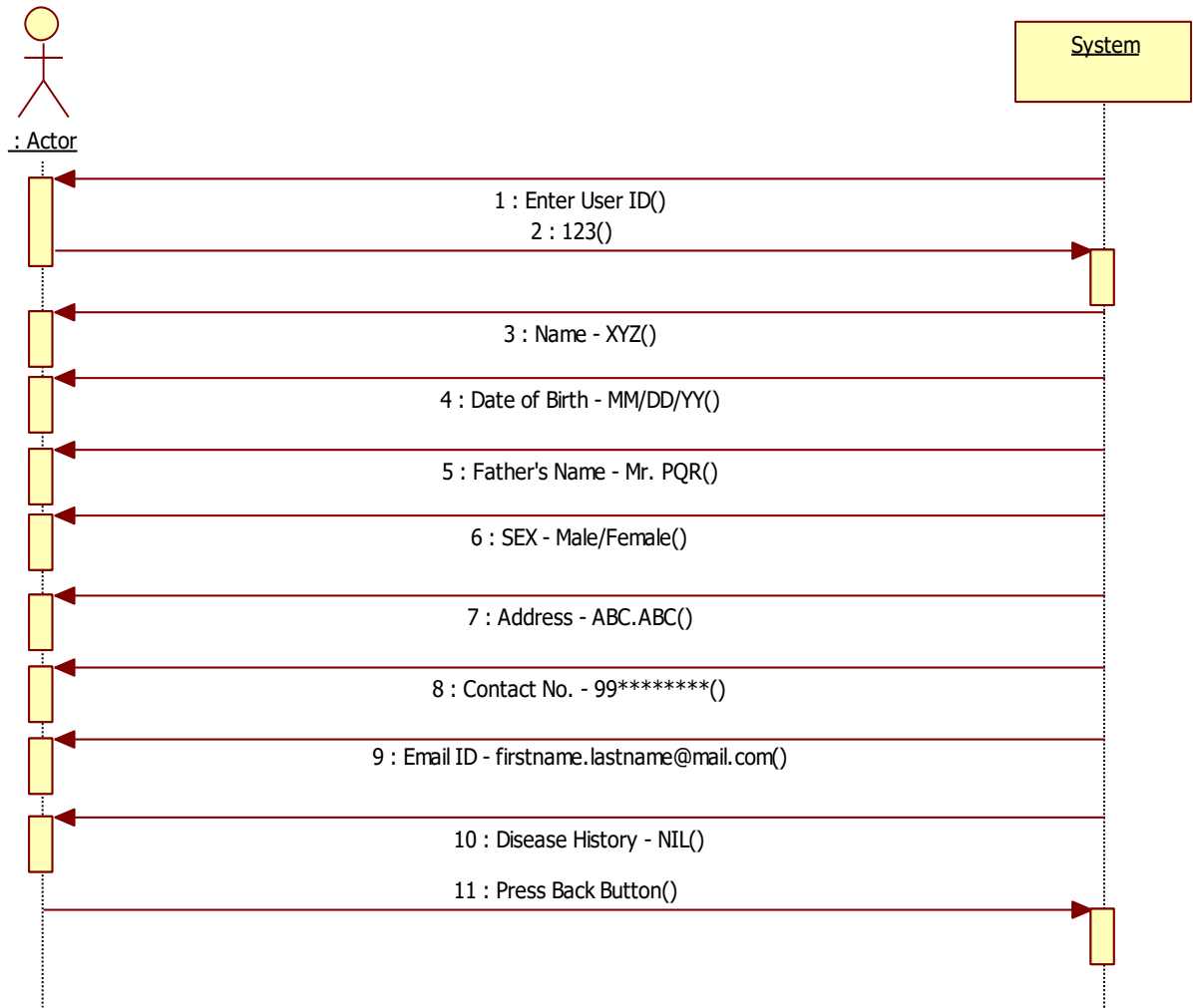
### Login



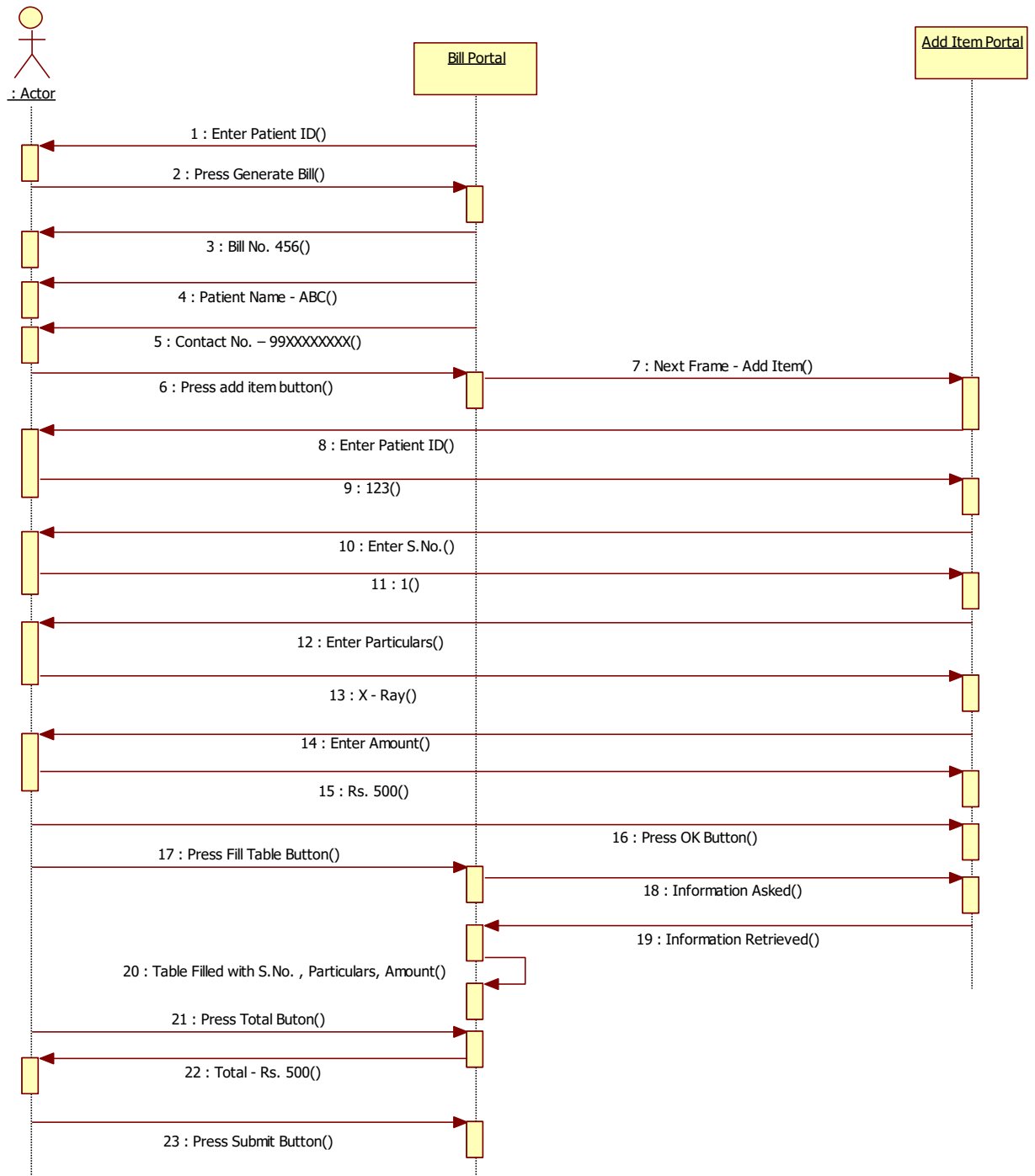
## Register Patient



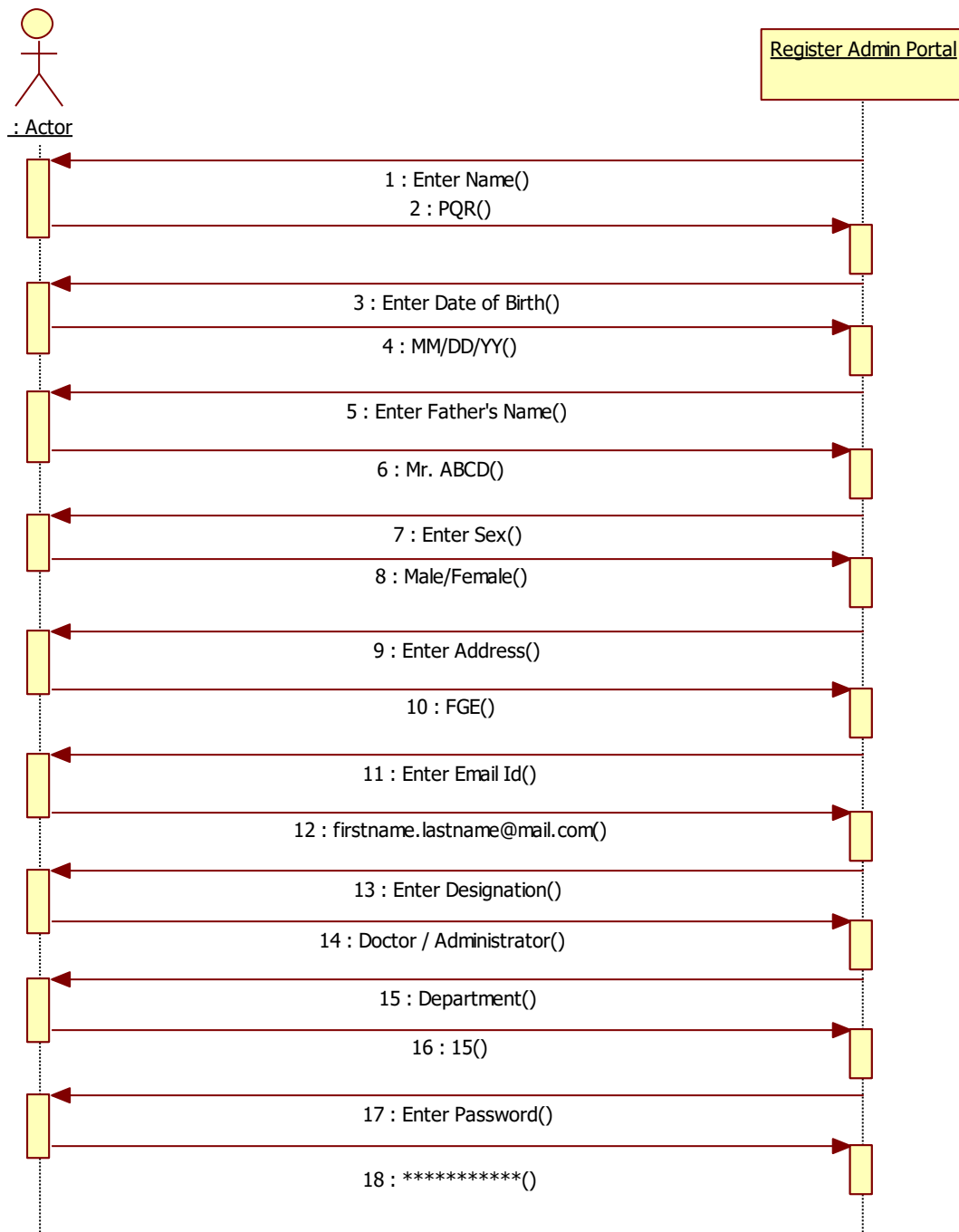
## View Details of Patient



## Generate Bill

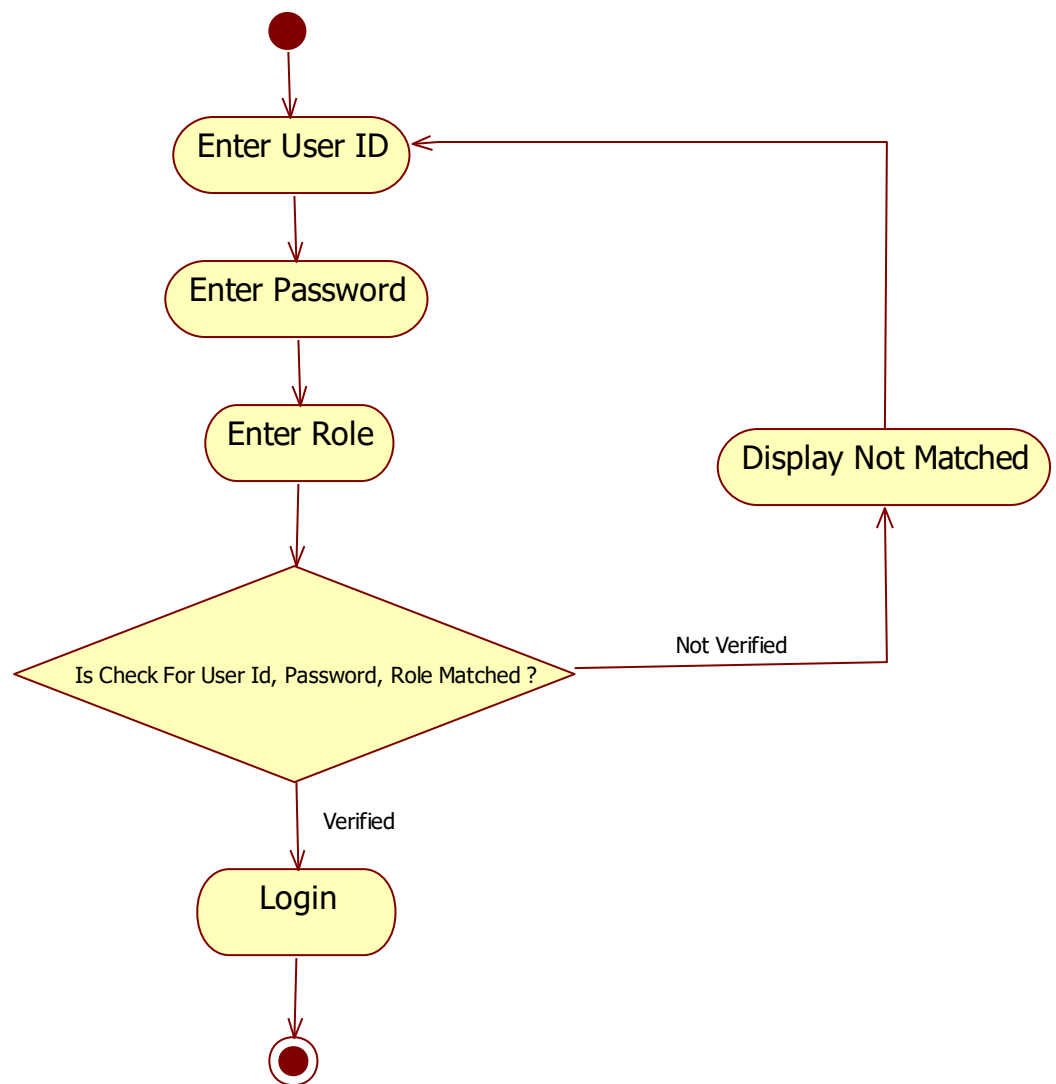


## Register Admin

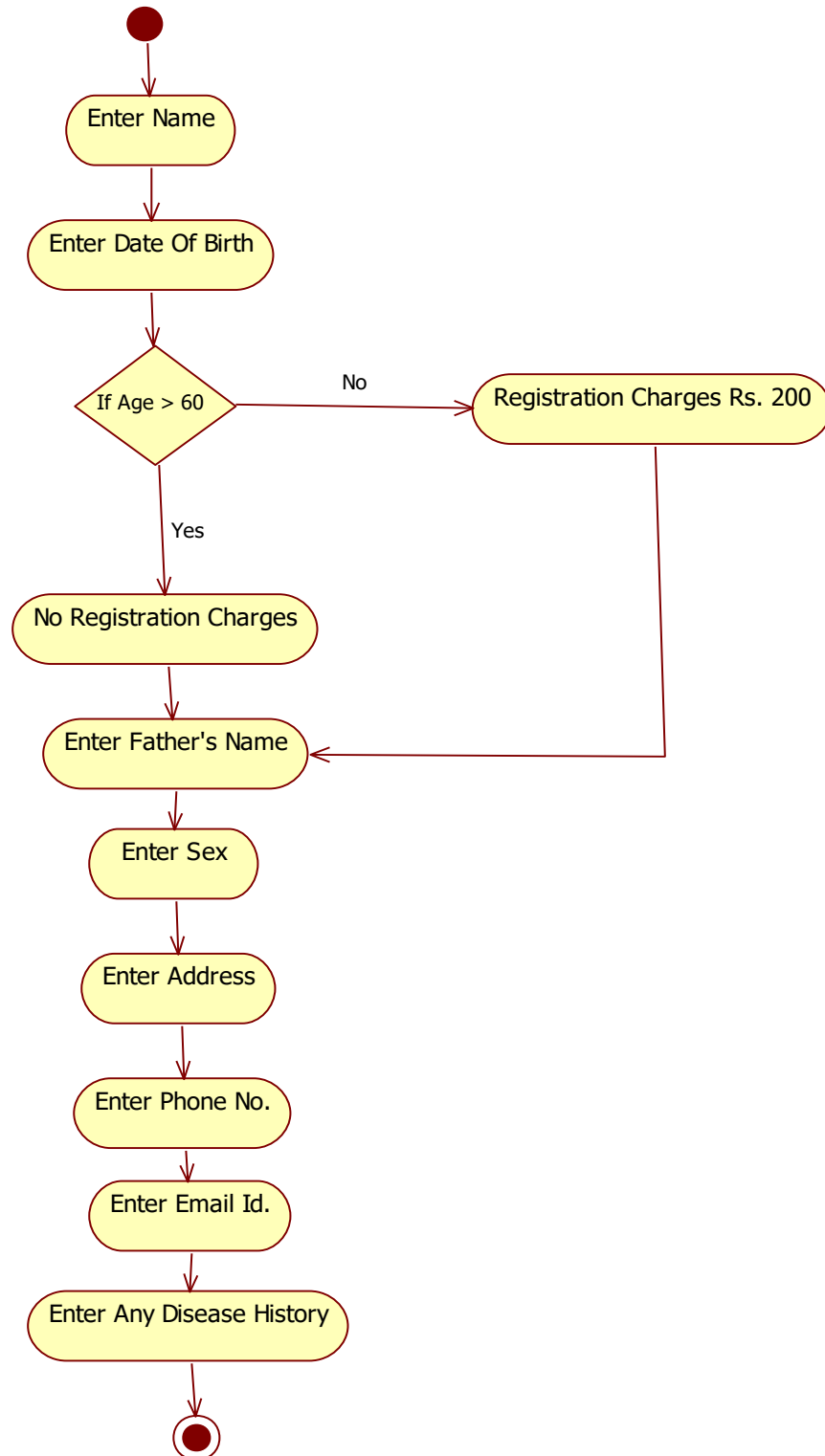


## 2.2.2 Activity Diagram

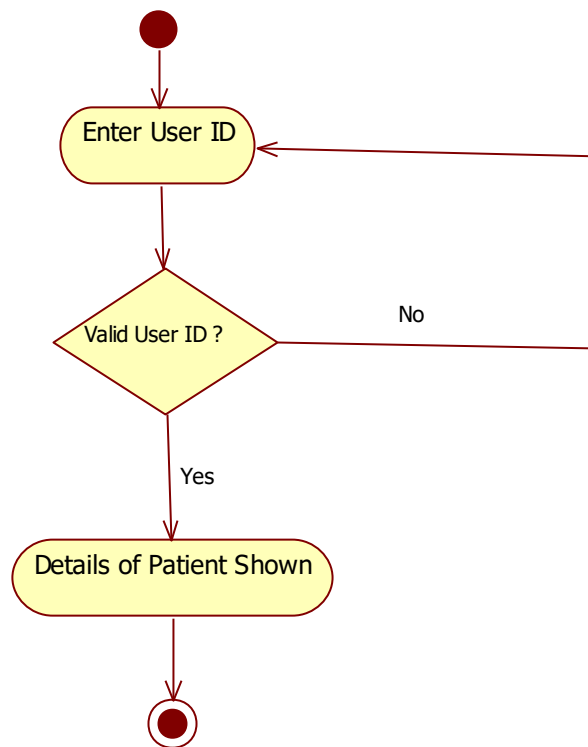
### Login



## Registration Form - Patient

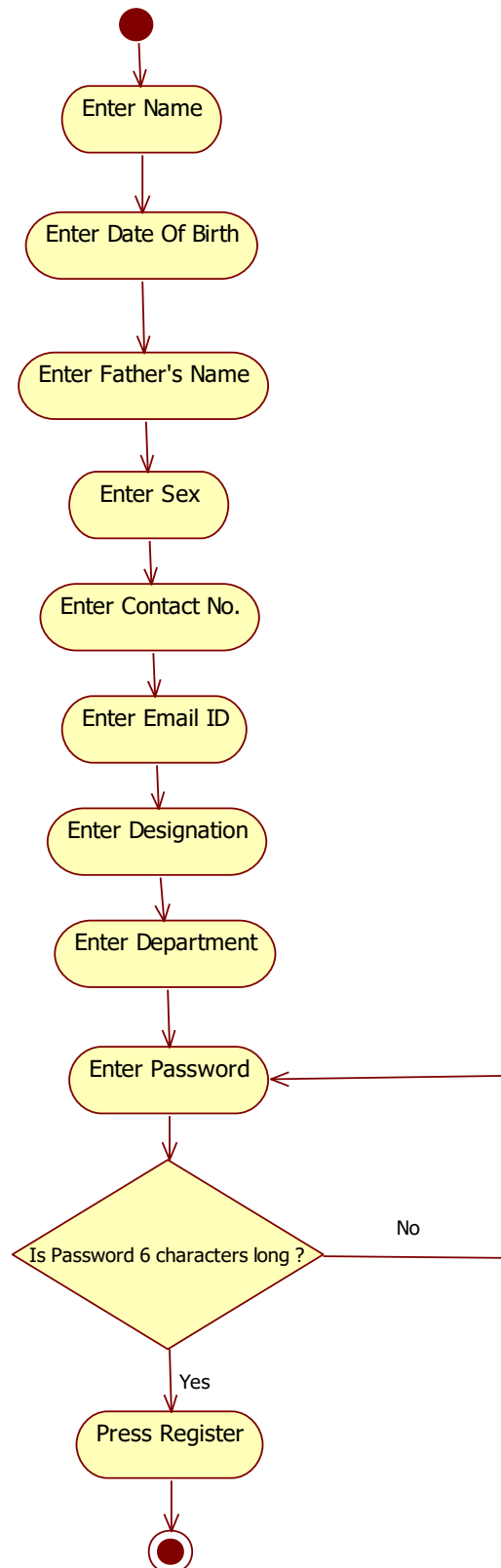


### View Patient Details

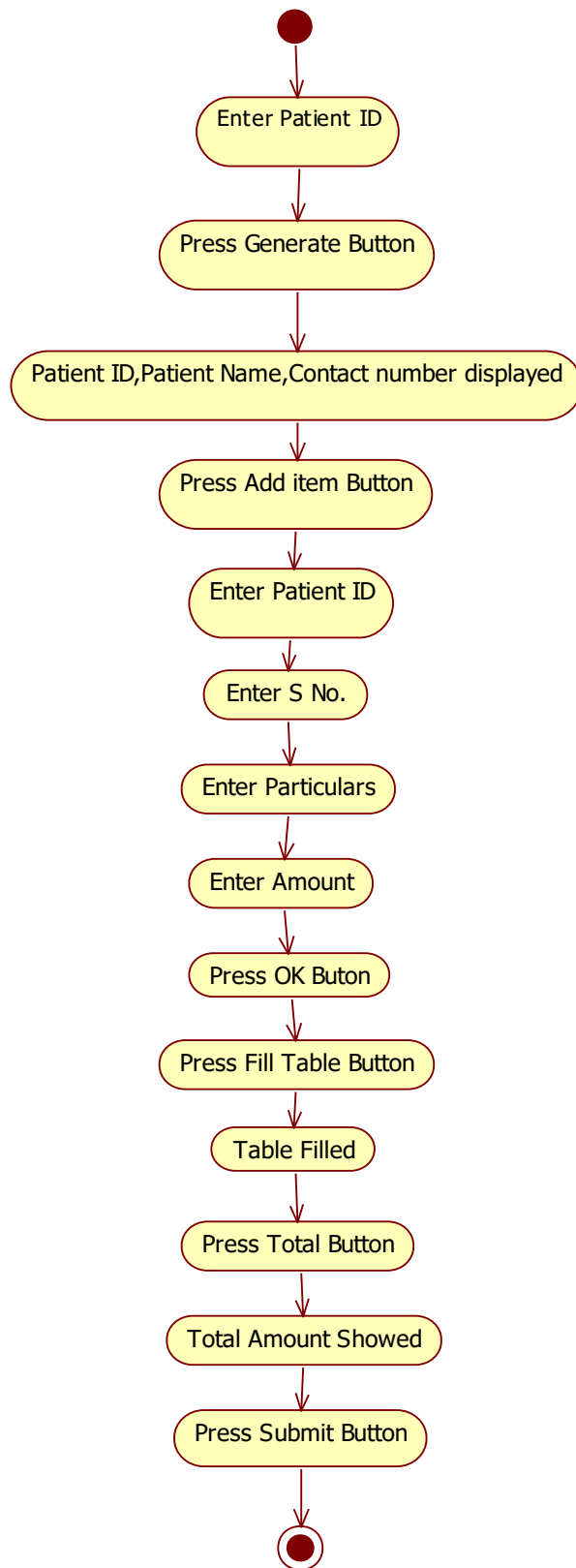


### Register Admin – Doctor / Administrator

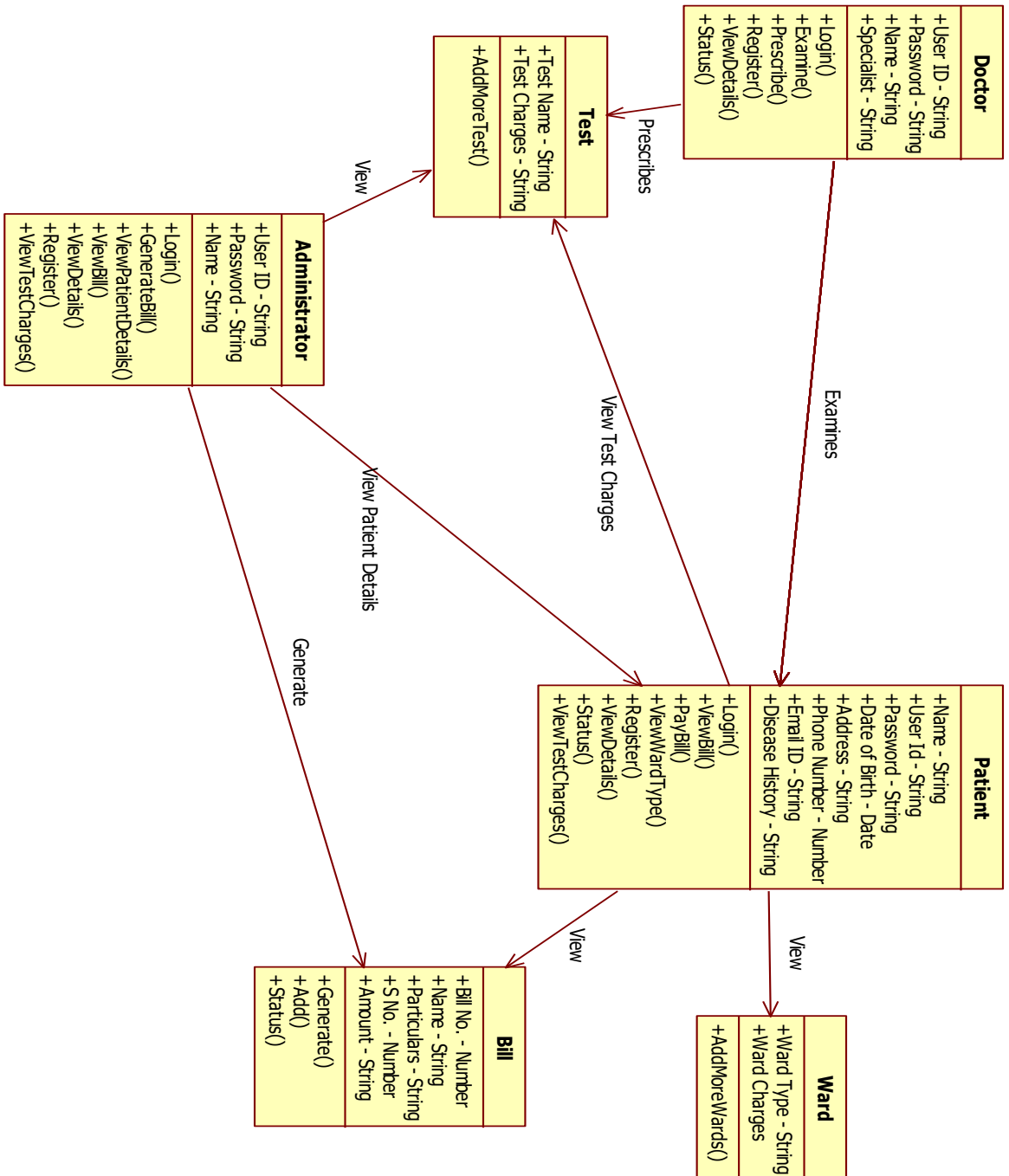




## Generate Bill

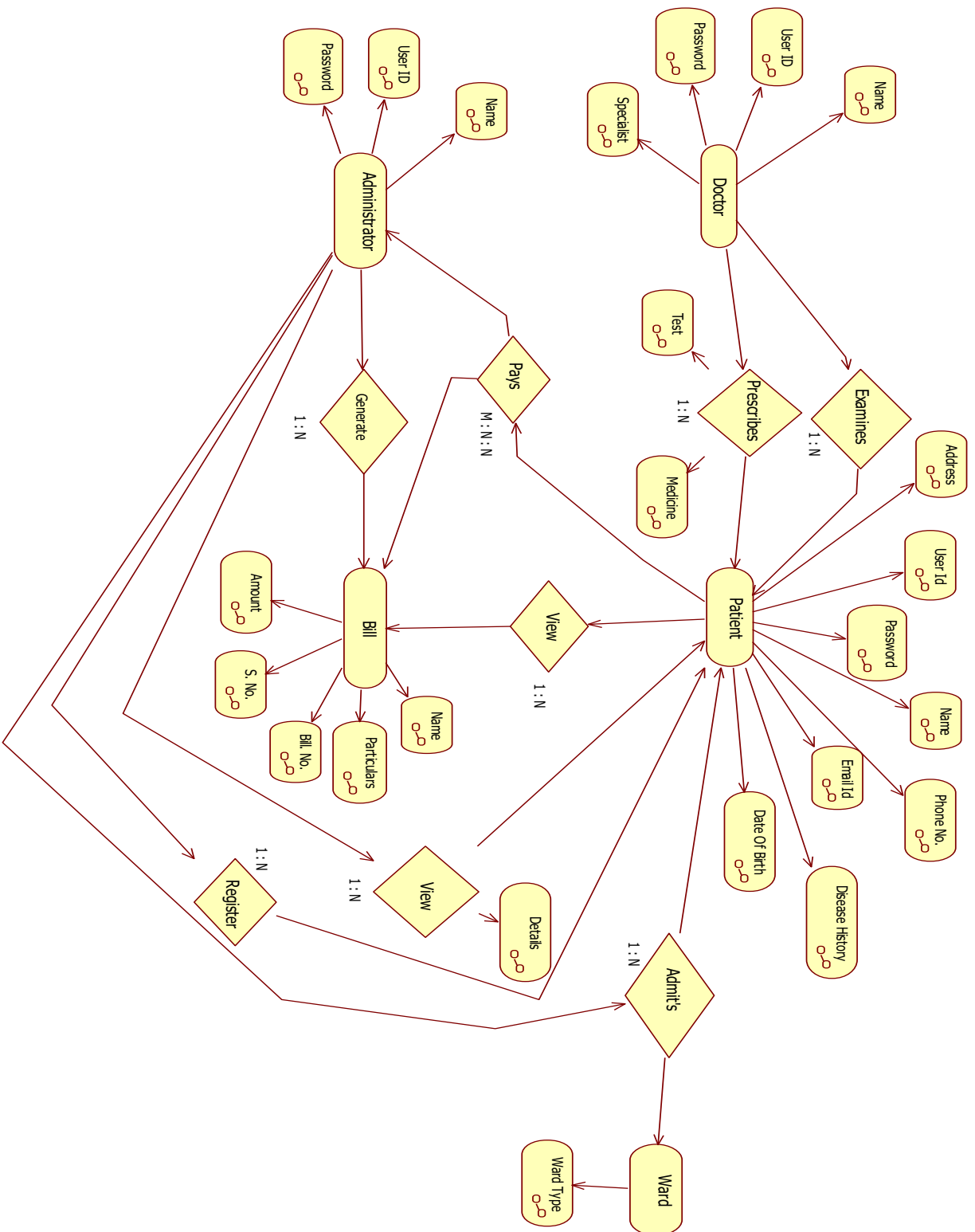


## 2.2.4 Class Diagram



## **3     Design Specification**

### **3.1     Entity Relationship Diagram**



### 3.2. E-R to Database Design

**Step1:**

Convert each strong entity into a relation.

**Step2:**

Convert each weak entity into a relation with foreign keys to its identifying relation.

**Step3:**

Convert binary 1-1 relationships into a unique foreign referenced from one relation to the other.

**Step4:**

Convert binary 1-n relationships into a one side relation.

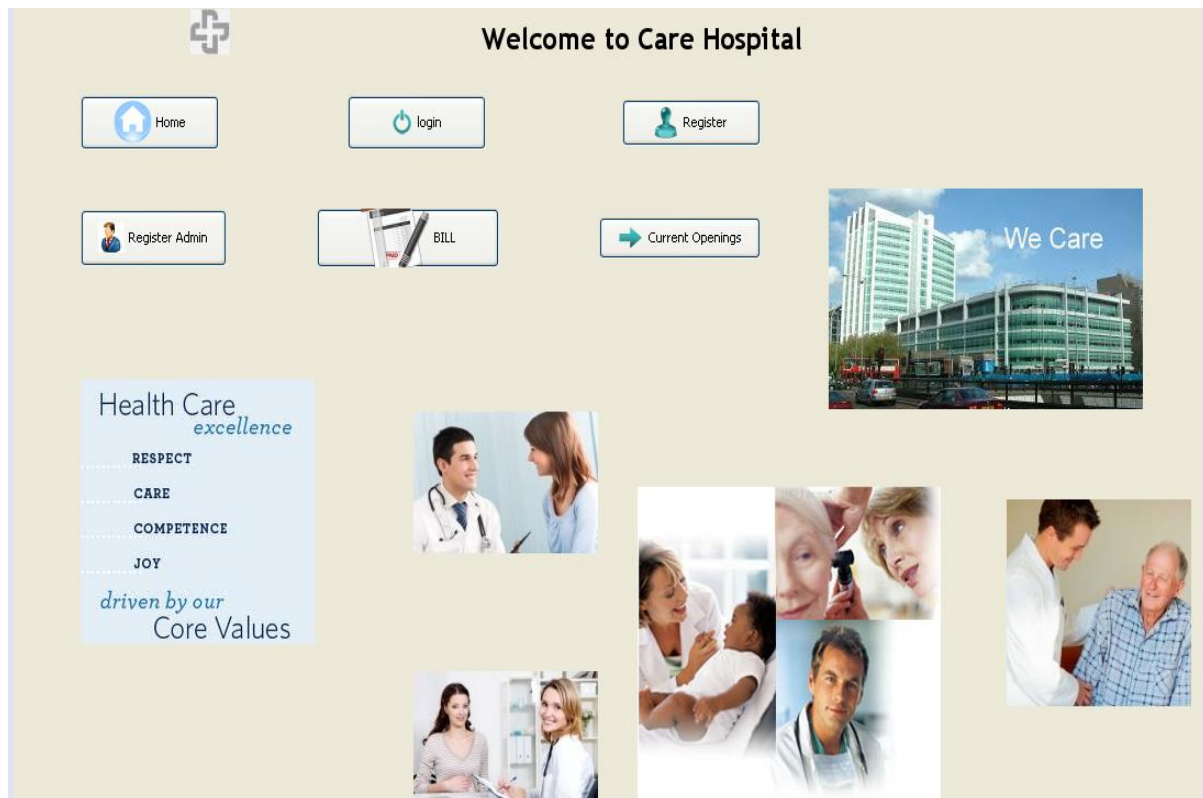
**Step5:**

Convert binary n-n relation into a new relation with foreign keys to the two participating entities.

**Step6:**

Convert the multivalued attribute into a relation with composite primary key consisting of the attribute value, plus the primary key of attribute entity.

## 4 User interface design specification



## Welcome to Care Hospital

[Home](#)[About US](#)[Facilities](#)[Test Details](#)[Bed Details](#)[Management](#)[Contact US](#)

### 24 x 7 Multifacilities / Pharmacy

The hospital has a Emergency/Pharmacy on premises which operates on 24 x 7 basis. We Serve the patients to avail multifacilities/medicines on the hospital premise it... avoiding the need to hunt for these in the city at odd hours.

### Blood Bank

The Blood Bank at Care Hospital is a modern up-to-date facility offering advanced Blood Component Therapy.

### Medical Library

The Library is equipped with the latest books and journals in medicines. It also has internet facility to enable a doctor an access to international publications in the field.

### Cafeteria

The hospital has a professionally managed cafeteria for the convenience of the patient's relatives and other visitors to the hospital.

## Welcome to Care Hospital

[Home](#)[About US](#)[Facilities](#)[Test Details](#)[Bed Details](#)[Management](#)[Contact US](#)

### Clinical Test Details

NAME OF TEST

CHARGES



## Welcome to Care Hospital

[Home](#)[About US](#)[Facilites](#)[Test Details](#)[Bed Details](#)[Management](#)[Contact US](#)

WARDS	CHARGES
GENERAL	R5. 500 PER DAY
DELUX	R5. 1000 PER DAY
I.C.U	R5.1500 PER DAY

## Welcome to Care Hospital

[Home](#)[About US](#)[Facilites](#)[Test Details](#)[Bed Details](#)[Management](#)[Contact US](#)

### Care Hospital - Directors

Dr.Tarun Garg	- Executive Chairman
Dr. Siddharth Bedi	- Managing Director
Dr. Sakshi Virmani	- Joint Managing Director
Dr. Palak Agarwal	- Executive Director, Operations

## Welcome to Care Hospital

[Home](#)[About US](#)[Facilities](#)[Test Details](#)[Bed Details](#)[Management](#)[Contact US](#)

### Care Lifeline - Emergency

**National: 1860 - 500 - 1066**

**International: 0091 - 404 - 344 - 1066**

### Request Information

**For any information about our locations, doctors or services**

**Please feel free to write. Our team will contact you as soon as possible.**

**Email ID - [info@carehospital.org](mailto:info@carehospital.org)**



## Welcome to Care Hospital

[Home](#)

### *Opening for Sr Registrars, Casualty Medical Officer , Orthopedic*

Care Hospital, Faridabad, a 350 bed tertiary care super speciality hospital, offers this golden opportunity to be a part of the Asia's largest integrated healthcare provider. We are looking forward to applications from professionals for the following requirements:

S.NO	OPENING	Qualification
1	Sr. Registrars	MS,MD/MCh
2	Medical Officer	MBBS with experience
3	Orthopedics	MBBS


#### **Applications can be sent to :**

Care Hospital

Plot No: 251, Old Sainik School Road, Faridabad

Tel : +91 674 666 0416/0415/0499

REGISTER PATIENT

 Home

NAME

DATE OF BIRTH

FATHER'S NAME

SEX

☐ MALE☐ FEMALE

ADDRESS


CONTACT NO


EMAIL ID

ANY DISEASE HISTORY

USERID

PASSWORD

 REGISTER

 RESET

one or more field requires attention

## REGISTRATION ADMIN

[HOME](#)

NAME

D.O.B

FATHER'S NAME

SEX

☐ male☐ female

ADDRESS

CONTACT NO

EMAILID

DESIGNATION

DEPARTMENT

USERID

PASSWORD




REGISTER



RESET


one or more field requires attention


 **LOGIN**


ENTER USER ID


ENTER PASSWORD

ROLE


 HOME


 SUBMIT


 RESET




**WELCOME PATEINT**

 View Details

 VIEW TEST CHARGES

 Bed Details

 View Bill



**welcome admin**



View Pateint Details



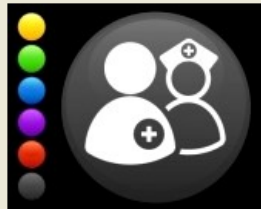
View Test Charges



View Bill



Generate Bill



**welcome doctor**



View Patient Details



View Information



Tracking Of Patient

GENERATE BILL

Home

PATIENT ID

GENERATE

BILL NO

jLabel6

PATIENT ID

PATIENT NAME:



CONTACT NO.


S.NO	PARTICULARS	AMOUNT


fill table

+ Total

submit

PATIENT ID	<input type="text"/>
S.NO	<input type="text"/>
PARTICULARS	<input type="text"/>
AMOUNT	<input type="text"/>
 add	 OK


 Home


BILL NO	<input type="text"/>
 VIEW	



VIEW DETAILS OF PATIENT

ENTER USER ID

OK

Back

NAME

D.O.B

FATHER'S NAME


SEX


ADDRESS

CONTACT NO

EMAIL ID

DISEASE HISTORY

BACK




back

TEST DETAILS

NAME OF THE TEST

CHARGES



BACK

CHOOSE BED

ENTER USER ID

SELECT BED

☐ GENERAL

☐ DELUX

☐ I.C.U

+ submit

## 5 Coding

### 5.1. Printout of code (complete code)

Welcome.java

```
public class welcome extends javax.swing.JFrame {
    public welcome() { initComponents();
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new REGISTER().setVisible(true); }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        new login().setVisible(true);}
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        new home().setVisible(true);}
    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        new curropenings().setVisible(true);}
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        new registeradmin().setVisible(true);}
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new welcome().setVisible(true);
        }
        });
    }
}
```

Home.java

```
import java.sql.*;
import javax.swing.JOptionPane;
public class home extends javax.swing.JFrame {
    public Connection con;
    public home() {
        initComponents();
        ttest.addItem("Endoscopy");
        ttest.addItem("X-RAY");
        ttest.addItem("Sonography");
        ttest.addItem("Gastroscopy");
        ttest.addItem("Blood Test");
        ttest.addItem("Stool Test");
        ttest.addItem("Urine Test");
        ttest.addItem("MRI Test");
        ttest.addItem("DNA Diagnostic Test");
        ttest.addItem("Ultrasound");

        String g=(String)ttest.getSelectedItem();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new welcome().setVisible(true);
    }
}
```

```

    }
    private void ttestItemStateChanged(java.awt.event.ItemEvent evt) {
String g = (String) ttest.getSelectedItem(); {
    if (g.equals("Endoscopy")) {
        chrg.setText("Rs. 1500");
    } else if (g.equals("X-RAY")) {
        chrg.setText("Rs. 500");
    } else if (g.equals("Sonography")) {
        chrg.setText("Rs. 1000");
    } else if (g.equals("Gastroscopy")) {
        chrg.setText("Rs. 780");
    } else if (g.equals("Blood Test")) {
        chrg.setText("Rs 200");
    } else if (g.equals("Stool Test")) {
        chrg.setText("Rs. 150");
    } else if (g.equals("Urine Test")) {
        chrg.setText("Rs. 50");
    } else if (g.equals("MRI Test")) {
        chrg.setText("Rs. 2500");
    } else if (g.equals("DNA Diagnostic Test")) {
        chrg.setText("Rs. 3000");
    } else if (g.equals("Ultrasound")) {
        chrg.setText("Rs. 1500");
    }
}
    public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new home().setVisible(true);
    }
});
}

```

## Bill1.java

```

public class BILL1 extends javax.swing.JFrame {
    public BILL1() {
        initComponents();
    }
    private void addActionPerformed(java.awt.event.ActionEvent evt) {
int total=0;
for(int i=1;i<=10;i++)
{
    total=total + Integer.parseInt(jTable1.getModel().getValueAt(i, 3).toString());
}
}
    public static void main(String args[]) {

```

```

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {
        new BILL1().setVisible(true);
    }
});
}

```

## Register.java

```

import java.sql.*;
import javax.swing.JOptionPane;
public class REGISTER extends javax.swing.JFrame {
    public Connection con;

    Statement stmt;
    ResultSet rs;
    public REGISTER() {
        initComponents();
        cmbMonth.addItem("month");
        cmbMonth.addItem("JAN");
        cmbMonth.addItem("feb");
        cmbMonth.addItem("mar");
        cmbMonth.addItem("april");
        cmbMonth.addItem("may");
        cmbMonth.addItem("june");
        cmbMonth.addItem("July");
        cmbMonth.addItem("aug");
        cmbMonth.addItem("sept");
        cmbMonth.addItem("oct");
        cmbMonth.addItem("nov");
        cmbMonth.addItem("dec");
        cmbdate.addItem("date");
        cmbyear.addItem("year");
        String [] date = new String[31];
        for(int i=1;i<=31;i++)
        {
            cmbdate.addItem(i);// TODO add your handling code here:
        }

        String [] year = new String[100];
        for(int j=1950;j<=2013;j++)
        {
            cmbyear.addItem(j);
        }
    }
}

```

```

private void txtregisterActionPerformed(java.awt.event.ActionEvent evt) {
String name,dob,fname,sex,address,contactno,emailid, history,month,date,year,mnth,pass;
String txt,userid;

name= txtname.getText();
month=cmbMonth.getSelectedItem().toString();
year=cmbyear.getSelectedItem().toString();
date=cmbdate.getSelectedItem().toString();
if (mradio.isSelected())
{
    sex = mradio.getText();
}
else
{
    sex = fradio.getText();
}

fname= txtfname.getText();
address= txtaddress.getText();
contactno= txtcontact.getText();
emailid= txtemail.getText();
history= txthistory.getText();
pass= txtpass.getText();
userid=txtuserid.getText();

try
{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    con=DriverManager.getConnection("Jdbc:Odbc:mydsn");
    System.out.println("Statement Created!!!!");
    System.out.println(name+","+month+","+date+","+year+","+fname+","+sex+","+address+","+contactno+","+emailid+","+history+","+userid+","+pass);
    String query1="insert into patient values (" +name+ "," + month + "," + date+","+year+
    "," +fname+ "," +sex+ "," +address+ "," +contactno+ "," +emailid+ "," +history+
    "," +userid+ "," +pass+ ")";
    stmt= con.createStatement();
    rs=stmt.executeQuery("select userid from patient where userid='"+userid+"'");
    System.out.println("Query Created!!!!");
    if(rs.next())
    {
        JOptionPane.showConfirmDialog(rootPane, "choose another id");
    }

    PreparedStatement ps=con.prepareStatement(query1);
    System.out.println("Prepared Statement Created!!!!");
    ps.executeUpdate();
    System.out.println("Hello");
    JOptionPane.showMessageDialog(rootPane, "RECORD INSERTED!!!!");
}
}

```

```

    }
    catch(Exception e) {
        JOptionPane.showMessageDialog(rootPane, e.getMessage());
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        new welcome().setVisible(true);
    }
    private void txtresetActionPerformed(java.awt.event.ActionEvent evt) {
        txtname.setText(null);
        cmbMonth.setSelectedItem("month");
        cmbdate.setSelectedItem("date");
        cmbyear.setSelectedItem("year");
        txtname.setText(null);
        txtfname.setText(null);
        txtaddress.setText(null);
        mradio.setSelected(false);
        fradio.setSelected(false);
        txtcontact.setText(null);
        txtemail.setText(null);
        txthistory.setText(null);
        txtuserid.setText(null);
        txtpass.setText(null);
    }
    private void txtemailFocusLost(java.awt.event.FocusEvent evt) {
        jemail.setVisible(false);
        jmsg.setVisible(false);
        txtregister.setVisible(true);
        String x=txtemail.getText();
        Integer atpos=x.indexOf("@");
        Integer dotpos=x.lastIndexOf(".");
        if (atpos<1 || dotpos<atpos+2 || dotpos+2>=x.length())
        {
            jemail.setText("Not a valid e-mail address");
            jemail.setVisible(true);
            jmsg.setVisible(true);
            txtregister.setVisible(false);
        }
    }

    private void txtpassMouseExited(java.awt.event.MouseEvent evt) {
        jLabel13.setVisible(false);
        jmsg.setVisible(false);
        txtregister.setVisible(true);
        String x=txtpass.getText();
        if (x.length()<6)
        {
            jLabel13.setText("passwords must be atleast 6 characters long");
            jLabel13.setVisible(true);
        }
    }

```

```

        jmsg.setVisible(true);
        txtregister.setVisible(false);
    }}
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {
                new REGISTER().setVisible(true);
            }
        });
    }

```

## VDetails.java

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class VDETAILS extends javax.swing.JFrame {
    public Connection con;

```

```

    Statement stmt,stmt1;
    ResultSet rs,rs1;

```

```

    public VDETAILS() {
        initComponents();
    }
    public Connection getConnection()
    {
        String Name = null,dob,fname,sex,address,contactno,emailid,
        history,month,date,year,mnth,pass;
        String txt,userid = null;

        try
        {
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            con=DriverManager.getConnection("Jdbc:Odbc:mydsn");
            //Statement stmt=con.createStatement();
            System.out.println("Statement Created!!!!");

        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(rootPane, e.getMessage());
        }
    }

```



```

        return con;
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Connection con=getConnection();
        try
        {
            String txt,userid = null;
            String g= id.getText();
            stmt1= con.createStatement();
            rs1=stmt1.executeQuery("Select
            Name,Month,Date,Year,Father_Name,Sex,Address,Contact,Email_Id,Disease_History from
            patient where userid='"+g+"'");
            System.out.println("Query Created!!!!");

            while(rs1.next())
            {
                System.out.println("Hello");
                txtname.setText(rs1.getString(1));
                txtmonth.setText(rs1.getString(2));
                txtdate.setText(rs1.getString(3));
                txtyear.setText(rs1.getString(4));
                txtfname.setText(rs1.getString(5));
                txtsex.setText(rs1.getString(6));
                txtaddress.setText(rs1.getString(7));
                txtcontact.setText(rs1.getString(8));
                txtemail.setText(rs1.getString(9));
                txthistory.setText(rs1.getString(10));

            }
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(rootPane, e.getMessage());
        }
        private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
            new patient().setVisible(true);    // TODO add your handling code here:
        }
        public static void main(String args[]) {
            java.awt.EventQueue.invokeLater(new Runnable() {

                public void run() {
                    new VDETAILS().setVisible(true);
                }
            });
        }
    }

```

## Test.java

```
public class test extends javax.swing.JPanel {
    public test() {
        initComponents();
        ttest.addItem("Endoscopy");
        ttest.addItem("X-RAY");
        ttest.addItem("Sonography");
        ttest.addItem("Gastroscopy");
        ttest.addItem("Blood Test");
        ttest.addItem("Stool Test");
        ttest.addItem("Urine Test");
        ttest.addItem("MRI Test");
        ttest.addItem("DNA Diagnostic Test");
        ttest.addItem("Ultrasound");

        String g=(String)ttest.getSelectedItem();
    }
    private void ttestItemStateChanged(java.awt.event.ItemEvent evt) {

        String g = (String) ttest.getSelectedItem();
        {
            if (g.equals("Endoscopy")) {
                chrg.setText("Rs. 1500");
            } else if (g.equals("X-RAY")) {
                chrg.setText("Rs. 500");
            } else if (g.equals("Sonography")) {
                chrg.setText("Rs. 1000");
            } else if (g.equals("Gastroscopy")) {
                chrg.setText("Rs. 780");
            } else if (g.equals("Blood Test")) {
                chrg.setText("Rs 200");
            } else if (g.equals("Stool Test")) {
                chrg.setText("Rs. 150");
            } else if (g.equals("Urine Test")) {
                chrg.setText("Rs. 50");
            } else if (g.equals("MRI Test")) {
                chrg.setText("Rs. 2500");
            } else if (g.equals("DNA Diagnostic Test")) {
                chrg.setText("Rs. 3000");
            } else if (g.equals("Ultrasound")) {
                chrg.setText("Rs. 1500");
            }
        }
    }

    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
```

```

@Override
public void run() {
    new test().setVisible(true);
}
});
}

```

## 6 System testing specification

### 6.1. Description of test cases

Test Case ID	Description	Input	Expected Output	Actual Output	Result
1	The login to the system with the login assigned by the admin and the correct password	Valid user id and valid data	Login should be successful and the user should enter in to the system	A new page opens up depending on the designation of the user.	Passed
2	The login to the system with a wrong password	Valid user id and invalid password	Login should fail.	Error message "Invalid Login" is prompted on the screen.	Passed
3	The login to the system with a wrong log id	Invalid login id and valid password	Login should fail.	Error message "Invalid Login" is prompted on the screen.	Passed

4	Validation of the Name being entered in the Registration form.	Name entered in lower case.	Name should not be accepted and error should be shown.	An error message "First character should be capital" is displayed.	Passed
5.	Validation of the Father's Name being entered in the Registration form.	Name entered in lower case.	Name should not be accepted and error should be shown.	An error message "First character should be capital" is displayed.	Passed
6.	Validation of the Name being entered in the Registration form.	Text including numbers and special characters entered.	Name should not be accepted and error should be shown.	An error message "Invalid Name" is displayed.	Passed
7	Validation of the Contact number being entered in the Registration form.	A number less than ten digits.	Contact number should not be accepted.	An error message "Contact number should be ten digit numbers" is displayed.	Passed
8	Validation of the Contact number being entered in the Registration form.	Text including alphabets and special characters is entered.	Contact number should not be accepted.	An error message "Invalid Phone number" is displayed.	Passed

9	Validation of the Email ID being entered in the Registration form.	Text entered in incorrect format for an email id.	Email Id should not be accepted.	An error message "Please enter a valid Email Id" is displayed.	Passed
10	Validation of the Password being entered in the Registration form.	A password of less than six digits.	The Password should not be accepted.	An error message "Password should be of at least six digits" is displayed.	Passed
11	Unsuccessful Login attempted three times.	Invalid login id or password entered three consecutive times.	The user should be authenticated.	An error message "You have reached the maximum login attempts. Please enter the phone no also to continue" is displayed.	Passed
12	Validation while password is being confirmed.	A Password not matching the password entered earlier.	The confirmation of password should fail	An error message "Password does not match" is displayed.	Passed

13	Validation of the process of Generating a report.	Employee ID and Task ID for which report has already been generated.	The report should not be generated again.	An error message "Reports already were generated for completed tasks".	Passed
----	---------------------------------------------------	----------------------------------------------------------------------	-------------------------------------------	------------------------------------------------------------------------	--------

## 7 Conclusion

### 7.1. Summary

This project was aimed at developing a patient billing software that is of importance to a hospital. The Patient Billing System is a local software system. This system can be used to maintain the location (bed) of each patient either in the ward or the ICU. Information about the patient and the charges to be paid is also stored. New patients as well as new doctors or administrators can register and work on the system. This project was given a website look in view for those patients who can register or view their bills from home.

### 7.2. Limitation

This project was aimed at developing a patient billing software that is of importance to a hospital. The big limitation faced by this project is the online payment of bill by the patient. The patient right now can view his bill but to pay the bill he has to go to the administrator online.

### **7.3. Future Work**

The future work for this project Patient Billing System is the work to cover up the limitation, that is to work on upgrading the system to accommodate online payment of bill by the patient be it at the hospital or at home.

## **8. References**

1. [www.w3cschools.com/](http://www.w3cschools.com/)
2. [www.wikipedia.com/](http://www.wikipedia.com/)
3. [www.slideshare.net/](http://www.slideshare.net/)
4. [www.google.com/](http://www.google.com/)