

A
Project Report
On

“ PHARMACY MANAGEMENT SYSTEM”

Submitted To

**BHARATI VIDYAPEETH (DEEMED TO BE UNIVERSITY),
PUNE**

In the partial fulfillment for requirement of the degree

**BACHELOR OF COMPUTER APPLICATION
(BCA-II SEM- V 2021-22)**

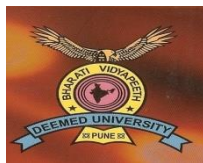
Submitted By

**Miss. KAMBLE SHUBHANGI RUPESH
Mr. NALBAND MOHAMMADKAIF SHAKIL
Miss. SAI UMAKANT GAVALI**

Under the guidance of

Dr.R.S.Pujari sir

Through The
**INSTITUTE OF MANAGEMENT AND RURAL DEVELOPMENT
ADMINISTRATION, SANGLI**



**Bharati Vidyapeeth Deemed University
Institute of Management & Rural Development
Administration, Sangli-416416**

CERTIFICATE

This is certified that a project report titled
Pharmacy Management system

BCA Sem-V in partial fulfillment for requirement of the degree of 'Bachelor of Computer Application' submitted to Bharati Vidyapeeth Deemed University, Pune has been completed under my guidance.

To the best of my knowledge and belief matter presented by her/him/them is original in nature and has not been copied down from any sources.

Place: Sangli

Date:

Dr. R. S. Pujari
(Guide)

Dr. R. S. Pujari
HoD

Prof. Dr. P. P. Jamsandekar
(Director)

Examiner :

Declaration

TO,

**THE DIRECTOR;
BVDU IMRDA, Sangli.**

Respected sir,

We the undersigned hereby declare that the Project report entitled, **PHARMACY MANAGEMENT SYSTEM** written and submitted under the guidance of **Miss. SHUBHANGI RUPESH KAMBLE ,Mr. MOHAMMADKAIF SHAKIL NALBAND** and **Miss. SAI UMAKANT GAVALI** is our Original Work. We have not copied from any report submitted to BVDU IMRDA, Sangli. We are aware that any such copying is such liable in a punishment in a way that the university authority demits this.

PLACE: Sangli

DATE:

**Miss. GAVALI SAI UMAKANT
Miss. KAMBLE SHUBHANGI RUPESH
Mr. NALBAND MOHAMMADKAIF SHAKIL**

Acknowledgement

We take this opportunity to express our deep sense of obligation to the Bharati Vidyapeeth Deemed University, Pune. The Institute of Management and Rural Development Administration Sangli. We owe our deepest gratitude towards respective guide **Dr.R.S.Pujari sir** and all of our faculty members for their valuable guidance and motivation during the completion of the Project Report.

We express our sincere thanks to the Director **Dr. P. P. Jamsandekar** for giving us an opportunity to undertaking this Project work.

Finally, we are very much thankful to everyone who helped us a lot in the completion of this Report to the great extent.

Thanking you all.

Place: Sangli.

Date:-

Miss. GAVALI SAI UMAKANT
Miss. KAMBLE SHUBHANGI RUPESH
Mr. NALBAND MOHAMMADKAIF SHAKIL

INDEX

| Sr. No. | Title | Page No. |
|----------------|--------------------------|-----------------|
| 1 | Introduction of Project | |
| 2 | Objective | |
| 3 | About The Project | |
| 4 | About the platform: JAVA | |
| 5 | Requirement Engineering | |
| 6 | System Analysis | |
| 7 | System Design | |
| 8 | Coding & Output | |
| 9 | Testing | |
| 10 | Conclusion | |
| 11 | Bibliography | |

INTRODUCTION

The main aim of the project is the management of the database of the pharmaceutical shop. This project is insight into the design and implementation of a Pharmacy Management System. This is done by creating a database of the available medicines in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store. The aim of this project is to develop software for the effective management of a pharmaceutical store. We have developed this software for ensuring effective policing by providing statistics of the drugs in stock.

This leads to various studies and researches being conducted to selected health care facilities. It is necessary to ensure a technologically appropriate, equitable, affordable, efficient and environmentally adaptable and consumer friendly system, designed to fully utilize the ICT for the maximum benefit in the health care Industry. Here Computers have great relevant on storing data's securely and ease access on them in short period of time. In order to exploit the ICT in health care system, Pharmacy Management System is being build.

Pharmacy Management System is robust, integrated technology. Pharmacy Management System is robust, integrated technology. Pharmacy Management System deals with the maintenance with drugs and consumable in the pharmacy unit. The set-up of the Pharmacy Management system will ensure availability of sufficient quantity of drugs. Pharmacy Management System will design to detect drug interaction

OBJECTIVE

> Primary objective

- To gain practical experience by modeling a software based on real world problem.
- To understand how to work on Front-end (Java) and Back-end (MySQL) by using server(wamp).

> Secondary objective

- To develop an application that deals with the day to day requirement of any pharmacy.
- To develop the easy management of the medicines (drugs).
 - To handle the inventory details like sales details, purchase details and stock expiry and quantity.
- To provide competitive advantage to the pharmacy.
- To provide details information about the stock on details necessary and help locate it in shop easily.
- To make the stock manageable and simplify the use of inventory in the pharmacy.

ABOUT PROJECT

The Pharmacy Administration System (PAS) is a clinical shop system written in NetBeans and Java. This approach gives you control over your medicine store in terms of what you can add to it and what you can remove from it. This is a straightforward task with three components:

- 1.CVS where you can manage your products and changes
- 2.Pharma, where you can add the number of pharma stores you want to.
- 3.Store admin, where you can different sort of drugs.

The Pharmacy Management System is a medical shop management system written in the Java programming language using NetBeans. Pharmacy management software gives you complete control over your pharmacy, allowing you to add and delete items from the store. You can easily manage your pharmacy with a pharmacy management system. This project offers a straightforward user interface that is simple to operate. You can customize this project by adding any department you like.

*Available Features :

CVS where you can manage your products and changes

Pharma, where you can add the number of pharma stores you want to.

Store admin, where you can different sort of drugs.

ABOUT PLATFORM

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages.

As of 2019, Java was one of the most popular programming languages in use according to GitHub,[19][20] particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems. It was released in May 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GPL-2.0-only license. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open-source software and used by most developers and is the default JVM for almost all Linux distributions.

REQUIREMENT ENGINEERING

Functional Requirements:

Generate Report: the pharmacy management system generates report weakly on information about the drugs and it exports the information as output document.

Staff : One of the core roles of a pharmacy management is to oversee the pharmacy staff and manage the people in order to maintain positive work relations and results.

Stock: Management should oversee the stock of pharmaceutical goods and other items for sale in the pharmacy. Most pharmaceutical suppliers make two deliveries daily, so replacements can be ordered and received quickly when the supply is low.

Sales and marketing: Pharmacy management should consider sales and marketing strategy to maintain regular customers and positive revenue. In pharmacies operating under major chains, central corporate bodies usually decide upon the promotional periods and guidelines to be carried out in each pharmacy.

Non Functional Requirement:

Reliability Requirement: The system should provide a reliable environment to both customers and Admin. All orders should be reaching at the admin without any errors.

Efficiency Requirement: When an Online Medical Store Management implemented customer can purchase product in an efficient manner.

Usability Requirement: The web site is designed for user friendly environment and ease of use.

Availability: This website will be available Monday to Sunday at 9:30 am to 4:30 pm.

Functional Requirements: It is the primary requirements that are fulfilled by our web site .It's allowing the users, customers to use our website at the level of ease .

SYSTEM ANALYSIS

➤ Existing System

While in the existing system, it doesn't have in built ability to look up database and work modules section and add new drugs upon confirmation of the admin. To store data to the database, one person have to look up the work and add data to populate database. Because of standalone system, no chances of using this system for multi user environment. Among all, the most important for any business process was not available such as daily logs report, not possible to make audit report and no functionality to prepare customized report as per business requirements.

➤ Proposed System

This new pharmacy management system is just like workflow software which can be treated as full version software which has generic modules to prevent underpayment that makes efficient business system. Its automatic batch processing system saves time and money for the user who will use this system through with excellent graphical user interface. Upon all, to meet real time environment working process, this system has automated durable medical equipment billing system. To make non stoppable development process, its full interaction checkup provides the flexibility of drug interaction and status, does checking, allergy checking report and medical reports, duplicate therapy and private patient messages.

➤ Modules

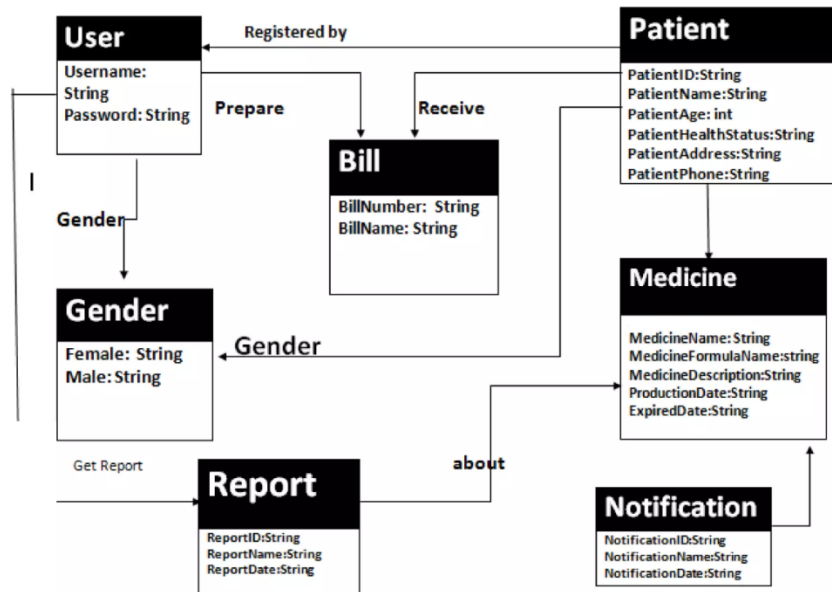
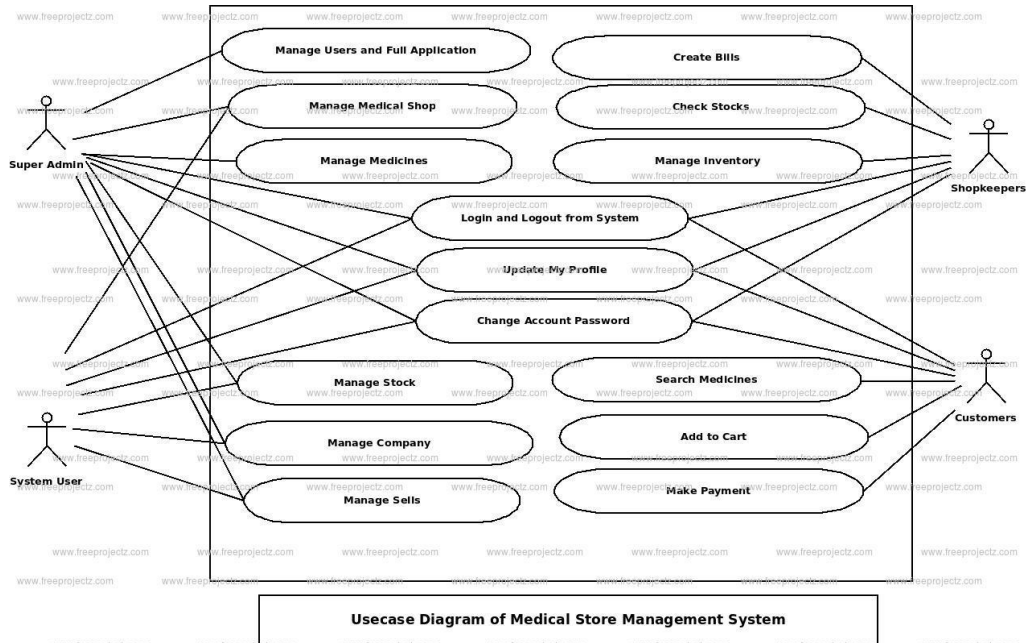
Script Assist: -Helping new assistant to work upon modules as per defined rules by helping through various resources and instructions.

Compound Assist: -Using this feature you can add multiple prescriptions to a particular profile for testing and system with automatic decision support will help patients to take these prescriptions.

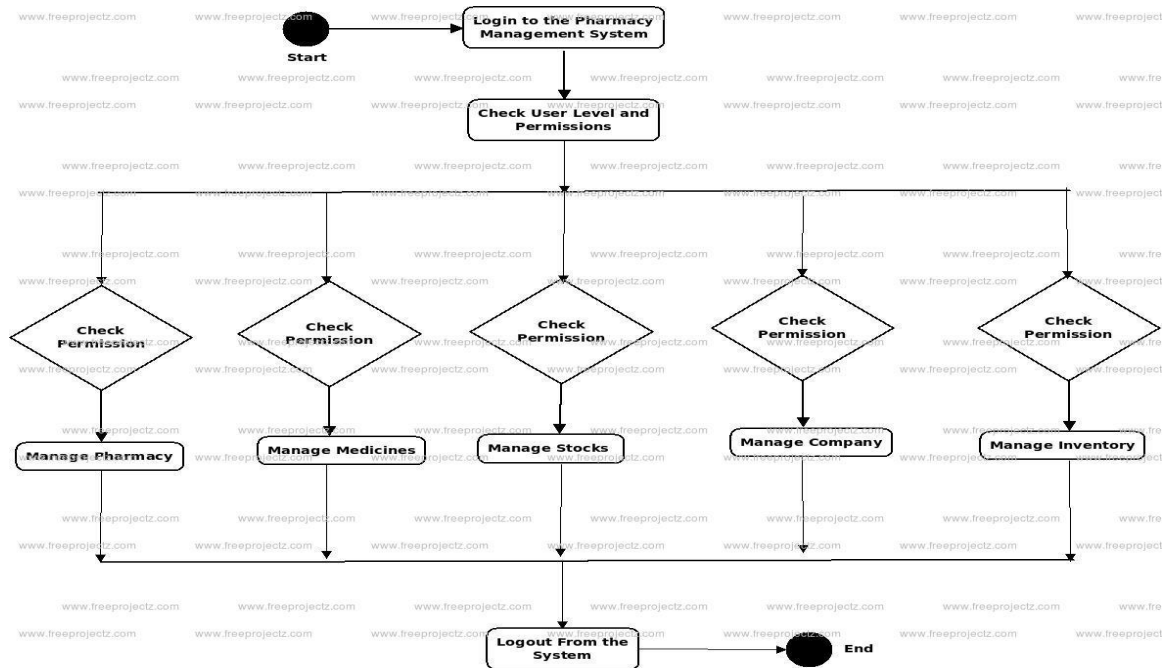
Prescription Assist: – To make its support system, more intelligent, its having feature of automatic updating its database upon completing of any particular module to make working process easier.

File Badger: -It's the part of batch processing to carry each and every tasks easily without human intervention to save time and money.

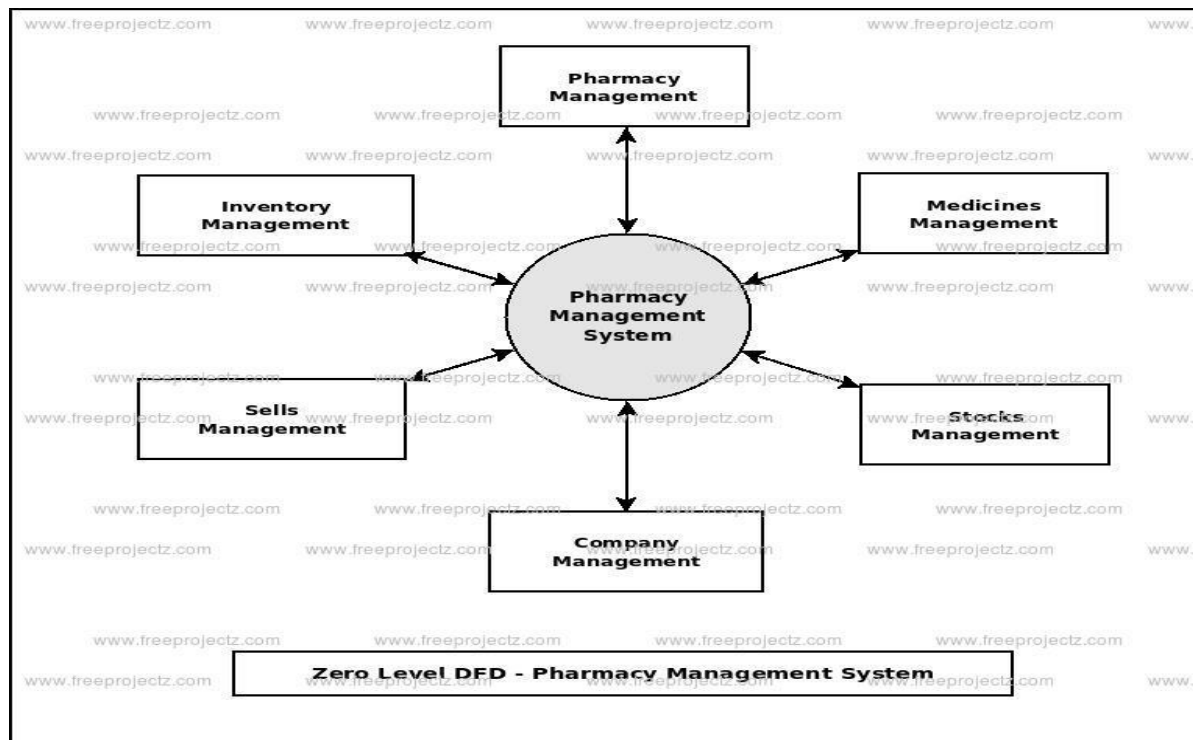
SYSTEM DESIGN



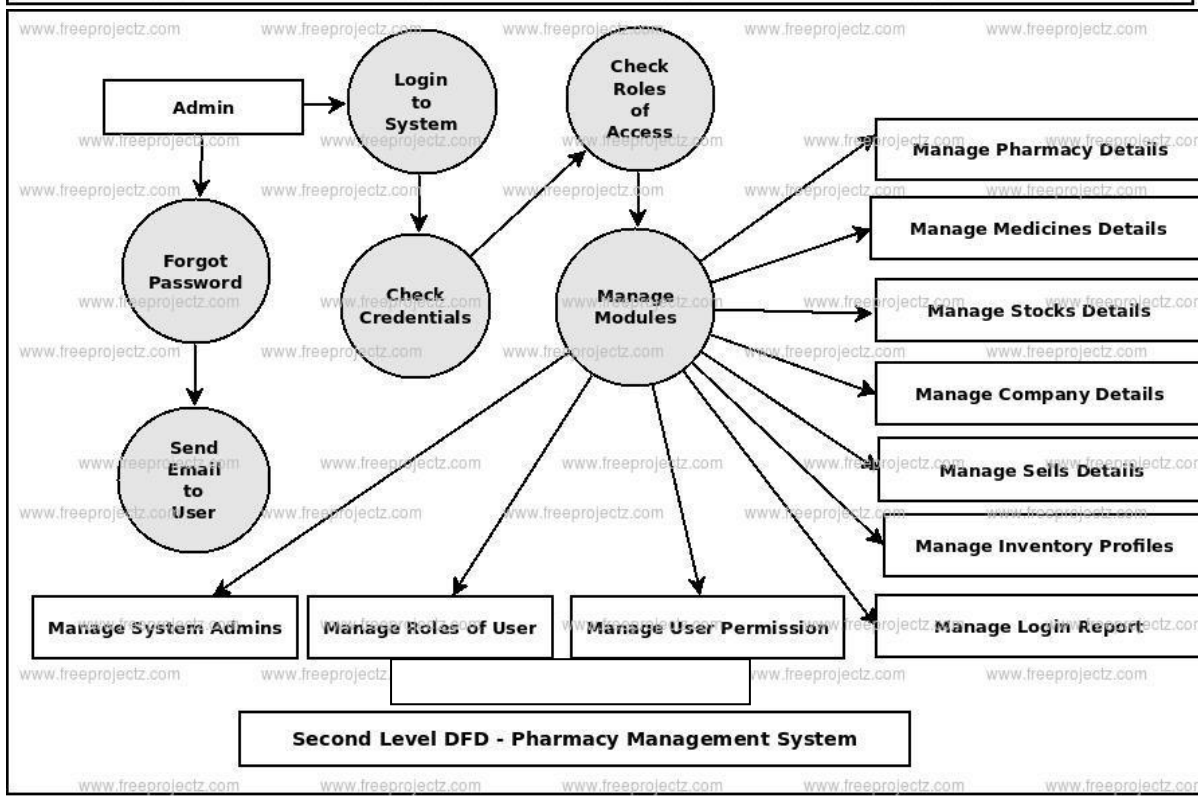
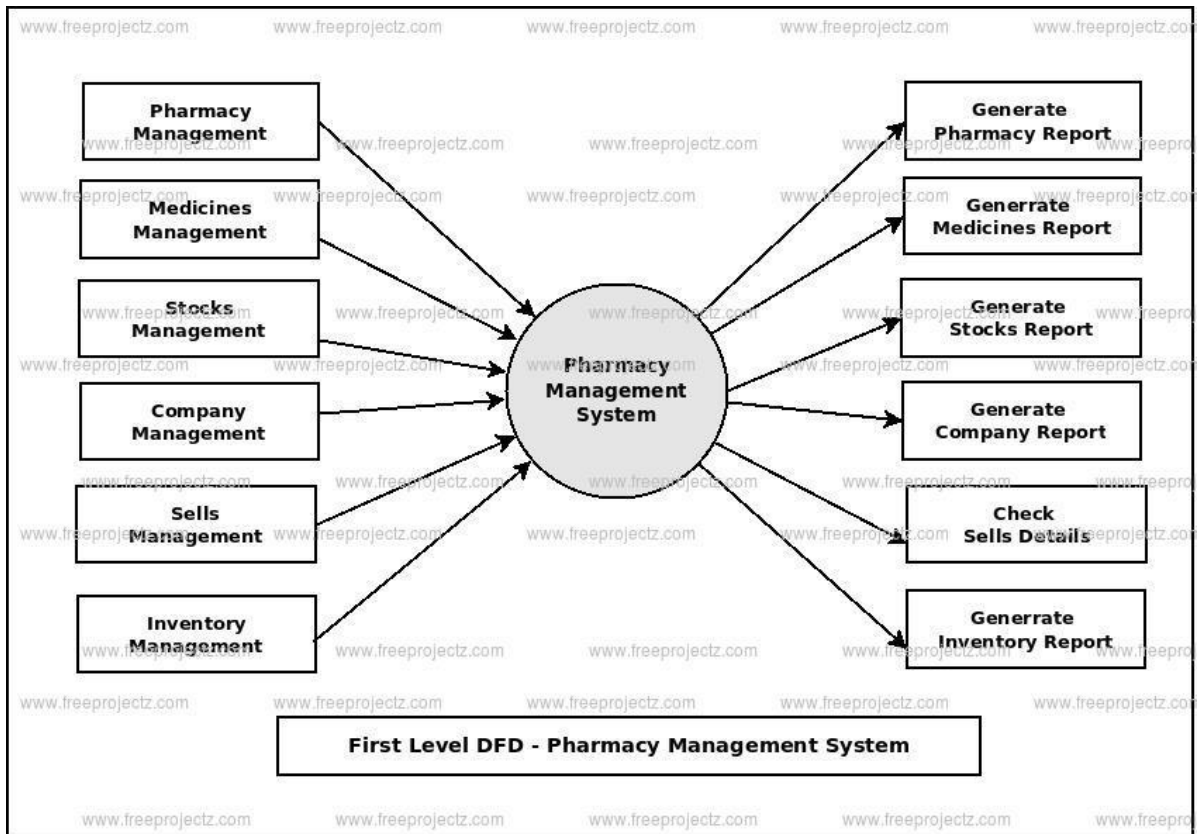
Class Diagram for Pharmacy Management System

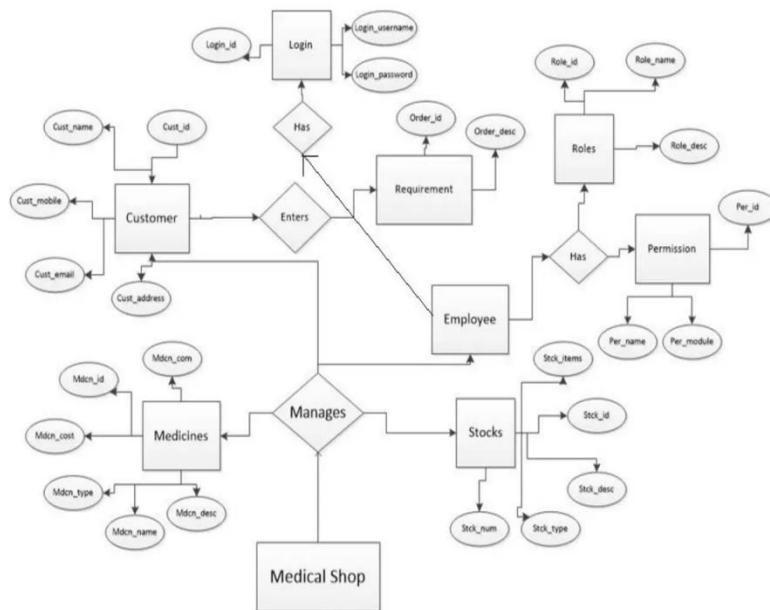


Activity Diagram for Pharmacy Management System



Zero Level DFD - Pharmacy Management System





ER Diagram of Pharmacy Management

CODING

```
package com.saeemodel;
/*
Pharmacy Management System
Admin: superuser
password: saee28
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.util.*;
public class Pharmacy
{
    static public String active="Pharmacy"; //it contains the name of last active
frame for getting back to it
    public static void main(String args[])
    {
        Updatequant uq=new Updatequant();
        uq.trunc();
        WelcomePage w=new WelcomePage();
        w.showFrame();
    }
}
class connect
{
    static Connection con; //used form storing connection path in a variable
    public Connection aconnect()
    {
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");

            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","sy
stem","12345");
        }
        catch(Exception e)
```



```

        {
            System.out.println(e);
        }
        return con;
    }
}
class Validity
{
    String myString;
    public boolean name(String s)                //NAME
    {
        boolean valid=true;
        s=s.trim();
        s=s.toUpperCase();
        if(s.length()<2 || s.length()>15){
            valid=false;
            System.out.print("Name length is out of range");
        }
        for(int i=0;i<s.length();i++)
        {
            if(s.charAt(i)>'Z' || s.charAt(i)<'A')
            {
                valid=false;
                System.out.print("Name has invalid symbol");
                break;
            }
        }
        if(valid==true)
            myString=s;

        return valid;
    }
    public boolean email(String s)                //email
    {
        boolean valid=true;
        s=s.trim();
        s=s.toLowerCase();
        int count=0;
        if(s.contains("..")==true || s.contains("@.")==true ||
s.contains(".@")==true )
            valid=false;
        if(s.length()>30)
        {
            valid=false;
            System.out.print("length is greater than 30");
        }
    }
}

```

```

    }
    if(s.endsWith(".com") && !(s.startsWith(".")))
    {
        for(int i=0;i<s.length();i++)
        {
            if(!(s.charAt(i)>='a' || s.charAt(i)<='z' || s.charAt(i)>='0'
|| s.charAt(i)<='9' || s.charAt(i)=='-' || s.charAt(i)=='@' ||s.charAt(i)=='.'||
s.charAt(i)=='_'))
                {
                    valid=false;
                    break;
                }
            else if(s.charAt(i)=='@')
            {
                count++;
                if(count>1)
                {
                    valid=false;
                    break;
                }
            }
        }
    }
    else
    {
        valid=false;
    }
    if(count<1)
        valid=false;
    if(valid==true)
        myString=s;
    return valid;
}

public boolean username(String s)                                //username
{
    boolean valid=true;
    s=s.trim();
    for(int i=0;i<s.length();i++)
    {
        if(!(s.charAt(i)<='Z' || s.charAt(i)>='A' || s.charAt(i)<='z' ||
s.charAt(i)>='a' ||s.charAt(i)<='9' ||s.charAt(i)>='0'))
        {
            valid=false;
            break;
        }
    }
}

```

```

        if(s.length()<6 || s.length()>30){
            valid=false;
            System.out.println("length of email is greater tha 30");
        }
        if(s.equals("superuser")){
            valid=false;
            System.out.println("You don't have right to creat superuser");
        }
        if(valid==true)
            myString=s;
        return valid;
    }
    public boolean authentication(String s)
    {
        s=s.trim();
        if(s.equals("ncBCA201417"))
            return true;
        else
            return false;
    }
    public boolean password(String s,String s1)
    {
        if(s.equals(s1) && s.length()>6 && s.length()<=15)
            return true;
        else
            return false;
    }
    public String setVarifiedString()
    {
        return myString;
    }
    public boolean noEmptyFields(String s1,String s2,String s3,String s4,String
s5,String s6,String s7)
    {
        boolean valid=true;
        s1=s1.trim();
        s2=s2.trim();
        s3=s3.trim();
        s4=s4.trim();
        s5=s5.trim();
        s6=s6.trim();
        s7=s7.trim();

        if(s1.equals("")||s2.equals("")||s3.equals("")||s4.equals("")||s5.equals("")||s6.e
quals("")||s7.equals(""))

```

```

        {
            valid=false;
        }
        return valid;
    }
}
//***** THE WELCOME PAGE *****
class WelcomePage extends JFrame implements ActionListener
{
    static JButton start, exit;
    public void showFrame()
    {
        this.setLayout(null);
        Container c=this.getContentPane();
        JLabel lbl=new JLabel();
        start=new JButton("Get Started");
        exit=new JButton("Exit");
        //***** Adding Image to the
Component*****
        lbl.setIcon(new ImageIcon("images\\welcome.jpg"));

        //***** Coloring ,Postioning and Resizing the Components and
*****
        start.setBackground(Color.BLUE);
        start.setForeground(Color.WHITE);
        start.setFont(new Font("Times New Roman",Font.BOLD,18));
        start.setToolTipText("Press the button to open login window");
        exit.setBackground(Color.RED);
        exit.setForeground(Color.WHITE);
        exit.setFont(new Font("Times New Roman",Font.BOLD,15));
        exit.setToolTipText("Press to close the Project");
        start.setSize(150,50);
        start.setLocation(310,350);
        exit.setSize(80,40);
        exit.setLocation(640,350);
        lbl.setSize(800,400);
        lbl.setLocation(0,0);

        //***** Adding to Container *****
        c.add(start);
        c.add(exit);
        c.add(lbl);

        //***** Frame Properties *****
        this.setTitle("Welcome");
    }
}

```

```

        this.setVisible(true);
        this.setSize(800,450);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        start.addActionListener(this);
        exit.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==start)
        {
            LoginPage h=new LoginPage();
            dispose();
            h.showFrame();
        }
        else if(e.getSource()==exit)
        {
            System.exit(0);
        }
    }
}
//***** REPORT PAGE *****
class ReportPage extends JFrame implements ActionListener, ItemListener
{
    static JButton next,back;
    static JTextField name,age,specify; // Text Fields nam,age that take input of
Name and Age respectively
    static Choice sex,type; // gen to creat choice box to select gender and spec to
select the specific illnes related with
    public void showFrame()
    {
        Container c=this.getContentPane();
        //***** Creating required componets
        //Labels
        JLabel msg=new JLabel("HOME");
        JLabel lbl1=new JLabel("Name :");
        JLabel lbl2=new JLabel("Sex :");
        JLabel lbl3=new JLabel("Age :");
        JLabel lbl4=new JLabel("Problem related to :");
        JLabel lbl5=new JLabel("Specify your problem in short if your selection
is OTHER");
        //buttons
        next=new JButton(" Next ");
        back=new JButton("Back");
    }
}

```

```

        //textbox
        name=new JTextField(20);
        age=new JTextField(5);
        specify=new JTextField(30);
        specify.setEnabled(false); //to keep the specify textbox disabled until
user select other
        //Choice box
        sex=new Choice();
        //sex.add(" ");
        sex.add("Male");
        sex.add("Female");
        // type choice box to Specify the illness is related to
        type=new Choice();
        type.add("Cough and Cold");
        type.add("Mental Illness");
        type.add("Eyes");
        type.add("Bones");
        type.add("Stomach and Appetite");
        type.add("Skin and Beauty");
        type.add("Dental");
        type.add("Sexual");
        type.add("OTHER");
        //***** Creating Button Panel
*****
        JPanel bp=new JPanel();
        bp.add(back);
        bp.add(next);
        //***** Creating data entry Panels
*****
        JPanel namePan=new JPanel();
        namePan.add(lbl1);
        namePan.add(name);
        JPanel sexPan=new JPanel();
        sexPan.add(lbl2);
        sexPan.add(sex);
        JPanel agePan=new JPanel();
        agePan.add(lbl3);
        agePan.add(age);
        JPanel typePan=new JPanel();
        typePan.add(lbl4);
        typePan.add(type);
        JPanel specPan=new JPanel(); //specify panel
        specPan.add(lbl5);
        specPan.add(specify);
        JPanel entryPan=new JPanel();

```

```

        entryPan.setLayout(new GridLayout(5,1));
        entryPan.add(namePan);
        entryPan.add(sexPan);
        entryPan.add(agePan);
        entryPan.add(typePan);
        entryPan.add(specPan);
        //***** Formatting the components
*****
                msg.setIcon(new ImageIcon("images\\report.jpg"));
                //login.setBackground(Color.BLUE);
                //login.setForeground(Color.WHITE);
        //***** Positioning and Adding the components to Container
*****
        c.add(msg,BorderLayout.NORTH);
        c.add(entryPan,BorderLayout.CENTER);
        c.add(bp,BorderLayout.SOUTH);
        //***** Setting up the Frame Properties
        this.setTitle("Home");
        this.setVisible(true);
        this.setSize(800,400);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        //***** Adding listeners to all the three buttons
*****
        next.addActionListener(this);
        back.addActionListener(this);
        type.addItemListener(this);
    }
    public void itemStateChanged(ItemEvent e)
    {
        String s=type.getSelectedItem();
        if(s.equals("OTHER"))
            specify.setEnabled(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==next)
        {
            InsertTableHome th=new InsertTableHome();//class to insert
report information in table
            String n=name.getText();//stores value of name
            String s=sex.getSelectedItem();//stores value of sex
            String t=type.getSelectedItem();//stores value of problem
            //passing value of name and age for validation purpose
            ValidateHome vh=new ValidateHome();

```

```

        boolean va=vh.valid(n,age.getText());
        if(va==false)
        {
            JOptionPane.showMessageDialog(this,"Invalid column
field");

            name.setText("");
            age.setText("");
            sex.select("Male");
            type.select("Cough and Cold");
        }
        else
        {
            int a=Integer.parseInt(age.getText()); //stores value of
age

            boolean b=th.Home(n,a,s,t);
            if(b==false)
                JOptionPane.showMessageDialog(this,"Error
while inserting");
            else

                JOptionPane.showMessageDialog(this,"Successfully inserted");
                ExaminPage h=new ExaminPage();
                dispose();
                h.showFrame();
                h.get(n,age.getText(),s,t);
            }
        }
        else if(e.getSource()==back)
        {
            if(Pharmacy.active.equals("LoginPage"))
            {
                LoginPage h=new LoginPage();
                h.showFrame();
            }
            else if(Pharmacy.active.equals("MainMenu"))
                new MainMenu();

            dispose();
        }
    }
}

//ValidateHome class is used to validate
class ValidateHome
{
    public boolean valid(String name,String age)

```



```

    {
        boolean va=true;
        name=name.trim();
        name=name.toUpperCase();
        if(name.length()<3 || name.length()>30)           //set va=false if
length of name is less than 3
            va=false;
        for(int i=0;i<name.length();i++)
        {
            if(name.charAt(i)>'Z' || name.charAt(i)<'A' )
            {
                //na variable is used to get ascii value
                //of characted by character to check space
                int na=(int)name.charAt(i);
                if(na==32)                               //32 is ascii value of space
                    va=true;
                else
                {
                    va=false;
                    break;
                }
            }
        }
        //this for loop is used to check age value
        //should only contains numbers
        if(age.length()>4)
            va=false;
        for(int i=0;i<age.length();i++)
        {
            if(age.charAt(i)<'0' || age.charAt(i)>'9')
            {
                va =false;
                break;
            }
        }
        return va;
    }

}
//class to insert report record in table
class InsertTableHome
{
    /*Home funtion is used to accept value from text box
    of generate report window

```

```

        here Sring s1 is used to accept value of name
        int n is used to accept value of age
        String s2 is used to accept value of sex choice
        String s3 is used to accept value of type that is problem
        */
        public boolean Home(String s1,int n,String s2,String s3)//default constructor
to insert record in table
    {
        try
        {
            //connect class is used to
            //connect java with oracle
            connect cn=new connect();
            //aconect method is defined in connect class
            //that contains driver path and other necessary details
            Connection con =cn.aconnect();
            PreparedStatement stmt=con.prepareStatement("Insert into
ReportTable values(?,?,?,?)");
            //prepared statement used to execute query to insert record in
reporttable

            stmt.setString(1,s1);
            //used to set value of name in table
            stmt.setInt(2,n);
            //used to set value of age in table
            stmt.setString(3,s2);
            //used to set value of sex in table
            stmt.setString(4,s3);
            //used to set value of problem in table
            int rs=stmt.executeUpdate();
            //rs variable is used to check whether record is successfully
inserted or not

            if(rs==1)
                return true;
            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return false;
    }
}
//***** LOGIN PAGE *****
class LoginPage extends JFrame implements ActionListener
{

```

```

static JButton signup, login;
static JTextField user;
static JPasswordField pass;
static JRadioButton rb1,rb2; // to select the rights you hava with the account
static ButtonGroup g;
public void showFrame()
{
    Pharmacy.active="LoginPage";
    Container c=this.getContentPane();
    JLabel msg=new JLabel("LogIn to get access to the Stocks and Short List
the Medicines...");
    //***** Creating the button and adding to pannel *****
    login=new JButton("Log In");
    signup=new JButton("Sign Up");
    //***** Creating other Components *****
    JLabel lbl1=new JLabel("Log in as:");
    JLabel lbl2=new JLabel("Username");
    JLabel lbl3=new JLabel("Password");
    g=new ButtonGroup();
    rb1=new JRadioButton("Billing Only",true);
    rb2=new JRadioButton("Stock Handler",false);
    g.add(rb1);
    g.add(rb2);
    user=new JTextField(20);
    pass=new JPasswordField(20);
    //***** Adding these components Panel by Panel*****
    JPanel selectPan=new JPanel(); //Panel to hold the selection
radio box
    selectPan.add(lbl1);
    selectPan.add(rb1);
    selectPan.add(rb2);
    JPanel userPan=new JPanel(); //Panel to hold user name
contents
    userPan.add(lbl2);
    userPan.add(user);
    JPanel passPan=new JPanel(); //Panel to hold password
contents
    passPan.add(lbl3);
    passPan.add(pass);
    JPanel p2=new JPanel(); //Panel p2 holds all
the above panels
    p2.setLayout(new GridLayout(3,1));
    p2.add(selectPan);
    p2.add(userPan);
    p2.add(passPan);

```

```

//**** Panel p holds the three buttons ****
JPanel p1=new JPanel();
p1.add(login);
p1.add(signup);
//***** Formating with the Components
*****
msg.setIcon(new ImageIcon("images\\login.jpg"));
//***** Adding the panel p1 and p2 to the LOGIN FRAME
*****

c.add(p2, BorderLayout.CENTER);
c.add(p1, BorderLayout.SOUTH);           //Button panel added
to South
c.add(msg, BorderLayout.NORTH);         //Message added to
Centre

//***** Setting Frame properties ****
this.setTitle("Log in");
this.setVisible(true);
this.setSize(800,400);
this.setLocationRelativeTo(null);
this.setResizable(false);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
login.addActionListener(this);
signup.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==login)
    {
        String usr=user.getText();
        String psw=pass.getText();
        /*-----
        goes here. */
        code for getting username and password from the database
        user
        userchek u=new userchek();//creating object of class to check
        int b=0;//variable to check whether user is available in table or
        not
        b=u.check(usr,psw);//storing boolean type value by passing
        //user name and password
        to the funtion check of class usercheck
        boolean superRights=false;
        /*    if b=2 is returned from check method
        of userchek class then
        user hava permission to access
        all pages that is superrights is true

```

```

        else if b=1 then user have
        only permission to access billing page
        not stock handler page    */
    if(b==2)
        superRights=true;
    //-----Grants for super user -----
    if(usr.equals("superuser"))
        superRights=true;
    //if boolean type variable b returns true then it is registered user

    if(usr.equals("superuser")&&psw.equals("saee28")||b==1||b==2)
    {
        if(rb1.isSelected())
        {
            BillingPage p=new BillingPage();
            p.showFrame();
            dispose();
        }else if(rb2.isSelected())
        {
            if(superRights)
            {
                new MainMenu();
                dispose();
            }
            else
            {
                JOptionPane.showMessageDialog(this,"You
don't have the rights to login as Stock Handler!");
            }
        }
    }
    else
    {
        user.setText("");
        pass.setText("");
        JOptionPane.showMessageDialog(this,"Invalid username
or password!");
    }
}
else if(e.getSource()==signup)
{
    SignupPage h=new SignupPage();
    dispose();
    h.showFrame();
}

```

```

    }
}
//class to check existing user name
// to check user availability that whether
// user is authorized that is login id
// and password exists in table or not
class userchek
{
    /*check method is used to
    check whether user is authorized or not
    it accepts two values userid and password
    and returns integer variable
    it returns 0 if user is unauthorized
    it returns 1 if user have only permission
    to access billing page
    it returns 2 if user have full permission */
    public int check(String s1,String s2)
    {
        int b=0;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            String query="select * from SignUpTable";
            ResultSet rs=stmt.executeQuery(query);
            while(rs.next())
            {
                String uch=rs.getString(4);//variables to access user
name from current record
                String pch=rs.getString(7);//variables to access
password from current record
                String rg=rs.getString(6);//variables to store right
                if(uch.equals(s1)&&pch.equals(s2)&&rg.length()<15)
                {
                    b=1;
                    break;
                }
                else
                if(uch.equals(s1)&&pch.equals(s2)&&rg.length()>16)
                {
                    b=2;
                    break;
                }
            }
        }
    }
}

```

```

        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    return b;
}
}
//***** SIGN UP PAGE *****
class SignupPage extends JFrame implements
ActionListener,ItemListener,FocusListener
{
    static JButton signup, back, report;
    static JTextField fname,lname,email,uname,auth;
    static JPasswordField pass, pass2;
    static Checkbox accept;
    static Choice rights;
    public void showFrame()
    {
        Container c=this.getContentPane();
        JLabel msg=new JLabel("Welcome! Get Signed Up");
        //***** Creating all the components
        JLabel lb1=new JLabel("First Name :");
        JLabel lb2=new JLabel("Last Name :");
        JLabel lb3=new JLabel("Email Id :");
        JLabel lb4=new JLabel("Prefered username :");
        JLabel lb5=new JLabel("Authentication number :");
        JLabel lb6=new JLabel("User Rights :");
        JLabel lb7=new JLabel("Password :");
        JLabel lb8=new JLabel("Confirm Password :");
        fname=new JTextField(20);
        lname=new JTextField(20);
        email=new JTextField(20);
        uname=new JTextField(20);
        auth=new JTextField(20);
        //choice for user rights
        rights=new Choice();
        rights.add("Billing Only");
        rights.add("Billing and Managing Stocks");
        //passwordFields
        pass=new JPasswordField(20);
        pass2=new JPasswordField(20);
        accept=new Checkbox("I hearby declare that the information
provided by me above is correct.",false);
    }
}

```

```
        JLabel lbl=new JLabel("And only I will be responsible for any  
kind inconvenience caused by incorrect information");
```

```
//***** Creating the button *****
```

```
signup=new JButton("Sign UP");          signup.setEnabled(false);
```

```
back=new JButton("Back");
```

```
report=new JButton("Generate Report");
```

```
//***** Adding all the components panel p2*****
```

```
        JPanel p2=new JPanel();
```

```
        p2.setLayout(null);
```

```
//setting size of components
```

```
        //labels
```

```
        lb1.setSize(150,23);
```

```
        lb2.setSize(150,23);
```

```
        lb3.setSize(150,23);
```

```
        lb4.setSize(150,23);
```

```
        lb5.setSize(150,23);
```

```
        lb6.setSize(150,23);
```

```
        lb7.setSize(150,23);
```

```
        lb8.setSize(150,23);
```

```
        //textFields etc.
```

```
        fname.setSize(280,23);
```

```
        lname.setSize(280,23);
```

```
        email.setSize(280,23);
```

```
        uname.setSize(280,23);
```

```
        auth.setSize(280,23);
```

```
        rights.setSize(280,23);
```

```
        pass.setSize(280,23);
```

```
        pass2.setSize(280,23);
```

```
//setting location of components
```

```
        //labels
```

```
        lb1.setLocation(100,30);
```

```
        lb2.setLocation(100,60);
```

```
        lb3.setLocation(100,90);
```

```
        lb4.setLocation(100,120);
```

```
        lb5.setLocation(100,150);
```

```
        lb6.setLocation(100,180);
```

```
        lb7.setLocation(100,210);
```

```
        lb8.setLocation(100,240);
```

```
        //textFiels etc.
```

```
        fname.setLocation(300,30);
```

```
        lname.setLocation(300,60);
```

```
        email.setLocation(300,90);
```

```
        uname.setLocation(300,120);
```

```
        auth.setLocation(300,150);
```

```
        rights.setLocation(300,180);
```



```

        pass.setLocation(300,210);
        pass2.setLocation(300,240);
        //agreement
        accept.setSize(500,22);
        accept.setLocation(80,300);
        lbl.setSize(510,22);
        lbl.setLocation(75,324);
    //adding all these components to JPanel p2
    p2.add(lb1);
    p2.add(lb2);
    p2.add(lb3);
    p2.add(lb4);
    p2.add(lb5);
    p2.add(lb6);
    p2.add(lb7);
    p2.add(lb8);
    p2.add(fname);
    p2.add(lname);
    p2.add(email);
    p2.add(uname);
    p2.add(auth);
    p2.add(rights);
    p2.add(pass);
    p2.add(pass2);
    p2.add(accept);
    p2.add(lbl);
    //***** Panel p1 holds the three buttons *****
    JPanel p1=new JPanel();
    p1.add(signup);
    p1.add(back);
    p1.add(report);
    //***** Formatting with the components
    *****
    msg.setIcon(new ImageIcon("images\\signup.jpg"));
    //***** Adding the panel p1,p2 to the Sign up FRAME
    *****

        c.add(p1,BorderLayout.SOUTH);                //panel added to
South
        c.add(msg,BorderLayout.NORTH);                //Message Label
added to NORTH
        c.add(p2,BorderLayout.CENTER);                //information panel
added to center
        this.setTitle("Sign Up");
        this.setVisible(true);
        this.setSize(700,600);

```

```

        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //*****Adding Listeners
        signup.addActionListener(this);
        back.addActionListener(this);
        report.addActionListener(this);
        accept.addItemListener(this);
        fname.addFocusListener(this);
        lname.addFocusListener(this);
        email.addFocusListener(this);
        uname.addFocusListener(this);
        auth.addFocusListener(this);
        pass.addFocusListener(this);
        pass2.addFocusListener(this);
    }
    public void focusGained(FocusEvent e){}
    public void focusLost(FocusEvent e)
    {
        /*
            Displays the error message if the authentication is false and the
            textfield is not empty.
            Sets the textfield blank if there is invalid entry
            Else, sets the textfield with varified entry
        */
        Validity v=new Validity();
        Component c=e.getComponent();
        if(c==fname)
        {
            if(v.name(fname.getText())==false &&
            !(fname.getText().equals(""))))
            {
                JOptionPane.showMessageDialog(this,"Invalid First
                Name!");
                fname.setText(" ");
            }
            else
            {
                fname.setText(v.setVarifiedString());
            }
        }
        else if(c==lname)
        {
            if(v.name(lname.getText())==false &&
            !(lname.getText().equals(""))))

```

```

        {
            JOptionPane.showMessageDialog(this,"Invalid Last
Name!");
            lname.setText("");
        }
        else
        {
            lname.setText(v.setVarifiedString());
        }
    }
    else if(c==email)
    {
        if(v.email(email.getText())==false &&
!(email.getText().equals("")))
        {
            JOptionPane.showMessageDialog(this,"Invalid Email
Address!");
            email.setText("");
        }
        else
        {
            email.setText(v.setVarifiedString());
        }
    }
    else if(c==uname)
    {
        if(v.username(uname.getText())==false &&
!(uname.getText().equals("")))
        {
            JOptionPane.showMessageDialog(this,"Username must
be Alphanumeric and minimum 6 char long.");
            uname.setText("");
        }
        else
        {
            userexist u=new userexist();//creating object of class to check
user
            boolean b=true;//variable to check whether user is available in
table or not

            b=u.user(uname.getText());//storing integer type value by
passing
            if(b==true)

```

```

        {
            JOptionPane.showMessageDialog(this,"username
already exist");
            uname.setText("");
        }
        else
            uname.setText(v.setVarifiedString());
        }
        else if(c==auth)
        {
            if(v.authentication(auth.getText())==false &&
!(auth.getText().equals("")))
            {
                JOptionPane.showMessageDialog(this,"Incorrect
authentication number!");
                auth.setText("");
            }
        }
        else if(c==pass)
        {
            if(pass.getText().length()<6 && !(pass.getText().equals("")))
            {
                JOptionPane.showMessageDialog(this,"Password must
be minimum 6 characters long");
                pass.setText("");
            }
        }
        else if(c==pass2)
        {
            if(!(pass.getText().equals(pass2.getText())) &&
!(pass2.getText().equals("")))
            {
                JOptionPane.showMessageDialog(this,"Confirmation
Password does not match!");
            }
        }
    }
    public void itemStateChanged(ItemEvent e)
    {
        if(accept.getState())
        {
            signup.setEnabled(true);
        }
        else
    
```

```

        {
            signup.setEnabled(false);
        }
    }
    public void actionPerformed(ActionEvent e)
    {
        Validity v=new Validity();
        if(e.getSource()==signup)
        {
            String s1,s2,s3,s4,s5,s6,s7;
            s1=fname.getText();
            s2=lname.getText();
            s3=email.getText();
            s4=uname.getText();
            s5=auth.getText();
            s6=pass.getText();
            s7=pass2.getText();
            if(v.noEmptyFields(s1,s2,s3,s4,s5,s6,s7))
            {

                if(v.name(s1)&&v.name(s2)&&v.email(s3)&&v.username(s4)&&v.authentication(s5)&&v.password(s6,s7))
                {
                    /*-----Code and methods for storing this signup
data goes here-----
                    after validating all fields of sign up frame
                    InsertsignUp class object i is created
                    and after that all values
                    that is first name for s1
                    last name for s2
                    email for s3,username for s4,
                    authentication number for s5,
                    users right for r
                    and passord for s6 */
                    InsertSignUp i=new InsertSignUp();
                    String r=rights.getSelectedItemAt();
                    i.InsertUser(s1,s2,s3,s4,s5,r,s7);
                    SignedUpPage h=new SignedUpPage();
                    dispose();
                    h.str1=uname.getText();
                    h.str2=fname.getText();
                    h.str3=lname.getText();
                    h.showFrame();
                }
            }
            else

```

```

        JOptionPane.showMessageDialog(this,"Incorrect
Password!");
    }
    else
        JOptionPane.showMessageDialog(this,"All the fields are
Mandatory!");
    }
    else if(e.getSource()==back)
    {
        LoginPage h=new LoginPage();
        dispose();
        h.showFrame();
    }
    else if(e.getSource()==report)
    {
        ReportPage h=new ReportPage();
        dispose();
        h.showFrame();
    }
}
}
// class to insert record in signup page
class InsertSignUp
{
    /*InsertUser function is used to
    insert value of SignUpTable in sql
    this function accepts 7 value and doesnot
    return any value*/
    public void InsertUser(String s1,String s2,String s3,String s4,String s5,String
ra,String s7)
    {
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Scanner in=new Scanner(System.in);
            PreparedStatement stmt=con.prepareStatement("Insert into
SignUpTable values(?,?,?,?,?,?,?)");
            stmt.setString(1,s1);
            stmt.setString(2,s2);
            stmt.setString(3,s3);
            stmt.setString(4,s4);
            stmt.setString(5,s5);
            stmt.setString(6,ra);
            stmt.setString(7,s7);

```

```

        int r=stmt.executeUpdate();
        if(r==1)
            System.out.println("Success");
        con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}

//***** EXAMIN PAGE *****
class ExaminPage extends JFrame implements ActionListener
{
    static JButton back,print,login;
    static JTextField    name,age,sex,problem;
    static JTextArea detail;
    public void showFrame()
    {
        Container c=this.getContentPane();
        //***** Creating required componets
        //Labels
        JLabel lbl1=new JLabel("Name  :");
        JLabel lbl3=new JLabel("Sex  :");
        JLabel lbl2=new JLabel("Age  :");
        JLabel lbl4=new JLabel("Problem:");
        //***** Discription Box
        detail=new JTextArea(20,4);
        //buttons
        back=new JButton(" Back ");
        print=new JButton(" Print ");
        login=new JButton("Log In");
        //textbox
        name=new JTextField(30);
        age=new JTextField(5);
        sex=new JTextField(5);
        problem=new JTextField(30);
        name.setEditable(false);
        age.setEditable(false);
        sex.setEditable(false);
        problem.setEditable(false);
        //***** Adding components panel by panel
        *****
        JPanel p1=new JPanel();
        p1.add(lbl1);

```

```

        p1.add(name);
        p1.add(lbl2);
        p1.add(age);
JPanel p2=new JPanel();
        p2.add(lbl3);
        p2.add(sex);
        p2.add(lbl4);
        p2.add(problem);
//adding p1 and p2 to panel p3
JPanel p3=new JPanel();
p3.setLayout(new BorderLayout());
p3.add(p1,BorderLayout.NORTH);
p3.add(p2,BorderLayout.SOUTH);
//Title of askPan
JPanel tp=new JPanel();
tp.add(new JLabel("Please answer these simple questions. It will take
few seconds."));
//components of askpane
JPanel ex=new JPanel();
ex.setLayout(new GridLayout(14,1));
JTextField tf0=new JTextField();
JTextField tf1=new JTextField();
JTextField tf2=new JTextField();
JTextField tf3=new JTextField();
JTextField tf4=new JTextField();
JTextField tf5=new JTextField();
JTextField tf6=new JTextField();
ex.add(new JLabel("        Enter Your height:"));
ex.add(tf0);
ex.add(new JLabel("        Enter your weight:"));
ex.add(tf1);
ex.add(new JLabel("        Enter your Blood Group:"));
ex.add(tf2);
ex.add(new JLabel("        Describe Your Problem"));
ex.add(tf3);
ex.add(new JLabel("        For how long you are facing this proble"));
ex.add(tf4);
ex.add(new JLabel("        Have you had any medication for this this
earlier. If yes, Mention."));
ex.add(tf5);
ex.add(new JLabel("        Are going through any medication right now.
If yes, Mention."));
ex.add(tf6);
//An askPan aka askPanel for interaction with the patient
JPanel askPan=new JPanel();

```



```

        askPan.setLayout(new BorderLayout());
        askPan.add(tp,BorderLayout.NORTH);
        askPan.add(ex,BorderLayout.CENTER);
        //buttons
        JPanel bp=new JPanel();
        bp.add(back);
        bp.add(print);
        bp.add(login);
        //***** Formatting the components
        *****
        //***** Positioning and Adding the components to Container
        *****
        c.add(p3,BorderLayout.NORTH);
        c.add(askPan,BorderLayout.CENTER);
        c.add(bp,BorderLayout.SOUTH);
        //***** Setting up the Frame Properties
        this.setTitle("Create Report");
        this.setVisible(true);
        this.setSize(800,450);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //***** Adding listeners to all the three buttons
        *****
        back.addActionListener(this);
        print.addActionListener(this);
        login.addActionListener(this);
    }
    /* get function is used to accept value
    from Report page so that value
    inserted by user or patient
    can be feeded in report page
    it accepts four string type values
    n for name,a for age,s for sex and p for problem*/
    public void get(String n,String a,String s,String p)
    {
        name.setText(n);
        age.setText(a);
        sex.setText(s);
        problem.setText(p);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==back)
        {

```

```

        if(Pharmacy.active.equals("LoginPage"))
        {
            LoginPage p=new LoginPage();
            p.showFrame();
        }
        else if(Pharmacy.active.equals("MainMenu"))
            new MainMenu();
        dispose();
    }
    else if(e.getSource()==print)
    {
        JOptionPane.showMessageDialog(this,"No printer available at
the moment");
    }
    else if(e.getSource()==login)
    {
        LoginPage h=new LoginPage();
        dispose();
        h.showFrame();
    }
}
}
class userexist
{
    public boolean user(String s)
    {
        boolean b=false;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            PreparedStatement
stmt=con.prepareStatement("select * from SignUpTable where userid=?");
            stmt.setString(1,s);
            ResultSet n=stmt.executeQuery();
            if(n.next())
            {
                b=true;
            }
            else
                b=false;
        }
        catch(Exception ec)
        {
            System.out.println(ec);

```

```

        }
        return b;
    }
}
//***** SIGNED UP PAGE *****
class SignUpPage extends JFrame implements ActionListener
{
    static String str1="superuser";
    static String str2,str3;
    static JButton login,report;
    public void showFrame()
    {
        Container c=this.getContentPane();
        login=new JButton("Login");
        report=new JButton("Generate Report");
        JPanel buttons=new JPanel();
        buttons.add(login);
        buttons.add(report);
        //----- username label-----
        JLabel user=new JLabel(str1);
        JPanel detail=new JPanel();
        detail.add(new JLabel("User username is :  "));
        detail.add(user);
        JPanel msgPan=new JPanel();
        msgPan.add(new JLabel(" Congratulations, "+str2+" "+str3+" "));
        msgPan.add(new JLabel("Your Account created successfully!"));
        c.add(msgPan,BorderLayout.NORTH);
        c.add(buttons,BorderLayout.SOUTH);
        c.add(detail,BorderLayout.CENTER);
        //setting up Frame Properties
        this.setVisible(true);
        this.setSize(500,130);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setTitle("SignUp successful!!");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //adding listeners
        report.addActionListener(this);
        login.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==login)
        {
            LoginPage h=new LoginPage();

```

```

        dispose();
        h.showFrame();
    }
    else if(e.getSource()==report)
    {

        ReportPage h=new ReportPage();
        dispose();
        h.showFrame();
    }
}
//Billing Page
class BillingPage extends JFrame implements ActionListener,FocusListener
{
    static JButton viewC,addToC,back;
    static JTextField pCode,pName,inStock,price,netPrice;
    static Choice quan;
    static Double net;
    int row=0;
    public void showFrame()
    {
        Container c=this.getContentPane();
        //creating header Components
        JLabel title=new JLabel("Product Billing");
        title.setFont(new Font("Times New Roman",Font.BOLD,20));
        //creating footer Components
        viewC=new JButton("View Cart");
        addToC=new JButton("Add To Cart");
        back=new JButton("BACK");
        //Creating Body Components
        //--Labels
        JLabel lbl1=new JLabel("Product Code :");
        JLabel lbl2=new JLabel("Product Name :");
        JLabel lbl3=new JLabel("Quantity  :");
        JLabel lbl4=new JLabel("Left in Stock:");
        JLabel lbl5=new JLabel("Price(Rs)  :");
        JLabel lbl6=new JLabel("Net Price(Rs):");
        //--TextFields
        pCode=new JTextField(20);
        pName=new JTextField(20);
        inStock=new JTextField(6);
        price=new JTextField(10);
        netPrice=new JTextField(10);
        //quantity choice box
    }
}

```

```
quan=new Choice();
quan.add("0");
quan.add("1");
quan.add("2");
quan.add("3");
quan.add("4");
quan.add("5");
pName.setEditable(false);
inStock.setEditable(false);
price.setEditable(false);
netPrice.setEditable(false);
// Positioning the Components
//-----Title-----
title.setLocation(10,10);
title.setSize(150,35);
//-----Buttons-----
viewC.setLocation(50,470);
viewC.setSize(100,33);
//-----
addToC.setLocation(50+125,470);
addToC.setSize(100,33);
//-----
back.setLocation(550,470);
back.setSize(100,33);
//-----Labels and TextBox-----
//size label
lbl1.setSize(100,30);
lbl2.setSize(100,30);
lbl3.setSize(100,30);
lbl4.setSize(100,30);
lbl5.setSize(100,30);
lbl6.setSize(100,30);
//location label
lbl1.setLocation(50,100);
lbl2.setLocation(50,150);
lbl3.setLocation(50,200);
lbl4.setLocation(50,250);
lbl5.setLocation(50,300);
lbl6.setLocation(350,350);
//size textbox
pCode.setSize(400,28);
pName.setSize(400,28);
quan.setSize(100,28);
inStock.setSize(100,28);
price.setSize(100,28);
```

```

        netPrice.setSize(100,28);
        //location textbox
        pCode.setLocation(150,100);
        pName.setLocation(150,150);
        quan.setLocation(150,200);
        inStock.setLocation(150,250);
        price.setLocation(150,300);
        netPrice.setLocation(440,350);
        netPrice.setForeground(Color.RED);
//Adding Components to the Container c
this.setLayout(null);
c.add(back);
c.add(viewC);
c.add(addToC);
c.add(title);
c.add(lbl1);
c.add(pCode);
c.add(lbl2);
c.add(pName);
c.add(lbl3);
c.add(quan);
c.add(lbl4);
c.add(inStock);
c.add(lbl5);
c.add(price);
c.add(lbl6);
c.add(netPrice);
//Setting Frame Properties
this.setTitle("Billing Page");
this.setSize(700,550);
this.setVisible(true);
this.setLocationRelativeTo(null);
this.setResizable(false);
this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//Adding Listeners to the buttons and other components
viewC.addActionListener(this);
addToC.addActionListener(this);
back.addActionListener(this);
pCode.addFocusListener(this);
quan.addFocusListener(this);
}
/*function to set value of medicine name and other details from table
it accepts 3 values
and does not return any value*/
public void get(String name,int quantity,Double rate)

```

```

    {
        pName.setText(name);
        if(quantity<1)
            quantity=0;
        inStock.setText(String.valueOf(quantity));
        price.setText(String.valueOf(rate));
        net=rate;    //value of rate is stored in net double type variable
    }
    public void focusGained(FocusEvent e){}
    public void focusLost(FocusEvent e)
    {
        //if focus is lost from quantity choice
        // then control comes here
        if(e.getComponent()==quan)
        {
            //if value of price is null then
            // netprice is automaticall set to 0
            if(price.getText().equals(""))
                netPrice.setText("0");
            else
            {
                int q=Integer.parseInt(quan.getSelectedItem());
                if(!(inStock.getText().equals(""))))
                {
                    int is=Integer.parseInt(inStock.getText());
                    is=is-q;
                    if(is<1)
                        is=0;
                    inStock.setText(String.valueOf(is));
                }
                /* total double type variable is
                used to store number of
                items to be purchased by
                the customer */
                Double
total=Double.parseDouble(quan.getSelectedItem());
                /* pric double type variable is
                used to calculate total value
                from rate and number of items
                that is quantity
                double type net variable already contains
                price value */
                Double pric=total*net;
                //ceil method of math class is used
                //to cut extra numbers after decimal place
            }
        }
    }

```

```

        pric=Math.ceil(pric);
        //pr variable is used to convert double
        //type variable pric into string type
        //as textfield accept string
        String pr=String.valueOf(pric);
        netPrice.setText(pr);
    }
}
else if(e.getComponent()==pCode)
{
    String s=pCode.getText();
    s=s.trim();
    s=s.toUpperCase();
    if(s.equals(""))
        JOptionPane.showMessageDialog(this,"Invalid product
code");
    else
    {
        pCode.setText(s);
        SearchStock ss=new SearchStock();
        /* searchstock class object ss is
        created then a string type variable
        bill is used that contains "bill" as value
        this particular bill variable
        is used to distinguish easily
        for searchstock class that it's
        object is created for billing page or
        stock handler page code is also
        concated with the variable bill then
        passes to search method*/
        String bill="bill";    //to distinguish between billing
page search and stockhandler page search
        bill=bill+s;
        boolean b=ss.Search(bill);
        if(b==false)
        {
            JOptionPane.showMessageDialog(this,"Product
code not found");

            String nam="NULL";
            String stk="NULL";
            String mrp="NULL";
            String tot="NULL";
            pName.setText(nam);
            inStock.setText(stk);
            price.setText(mrp);

```



```

        netPrice.setText(tot);
    }
}

}

}

public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==back)
    {
        /* this class
        is used to truncate temporary
        table of insert
        cart so that another bill is easily
        created without any error */
        Updatequant uq=new Updatequant();
        uq.trunc();
        //LoginPage p=new LoginPage();
        if(Pharmacy.active.equals("LoginPage"))
        {
            LoginPage p=new LoginPage();
            p.showFrame();
        }
        else if(Pharmacy.active.equals("MainMenu"))
        {
            new MainMenu();
        }

        dispose();
        //p.showFrame();
    }
    else if(e.getSource()==viewC)
    {
        CartPage p=new CartPage();
        /* get method of cartpage class
        accepts int type variable row
        that contains how many records
        has been inserted to the insertcart table*/
        p.get(row);
        p.showFrame();
        dispose();
    }
    else if(e.getSource()==addToC)
    {
        String nam=pName.getText();
        if(nam.equals("")||nam.equals("NULL"))

```

```

        {
            JOptionPane.showMessageDialog(this,"No
Product to Add to Cart");
            quan.select(0);
        }
        else
        {
            //----- Code for adding to cart table goes
            here -----
            String pc=pCode.getText();
            String n=pName.getText();
            String q=quan.getSelectedItem();
            //pr variable is used to store actual rate
            String pr=price.getText();
            //netprice variable is used to store
            // total price after multiplication with quantity
            String ne=netPrice.getText();
            int qu=Integer.parseInt(q);
            int left=Integer.parseInt(inStock.getText());
            /* if quantity contains 0 items then it is
            value
            mandatory for customer to select atleast one

            and if left in stock is less than required amount
            then also adding to cart is not possible*/
            if(q.equals("0"))

                JOptionPane.showMessageDialog(this,"Please select atleast 1 quantity");
            else if(left<qu||left<1)
                JOptionPane.showMessageDialog(this,"
Stock is insufficient");
            else
            {
                Insertcart ic=new Insertcart();
                //passing values to insert method of
                //insert cart table that accepts 5 string
                type
                //variables
                ic.insert(pc,n,q,pr,ne);
                //after inserting row integer type
                //variable is incremented so that
                //row should be counted
                row++;
                //-----
            }
        }
    }
}

```

```

        JOptionPane.showMessageDialog(this,"Added to Cart.");
        quan.select(0);
        pCode.setText("");
        pName.setText("");
        price.setText("");
        netPrice.setText("");
        inStock.setText("");
    }
}

}

}

}

//*****Insert cart class is used to
// store value of cart page that is billing page values
class Insertcart
{
    /* insert method is used to store five values
    accepted by this particular method to the
    table this method returns boolean type variable*/
    public boolean insert(String code,String name,String quant,String price,String
netprice) //function to insert record in table
    {
        boolean b=false;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Scanner in=new Scanner(System.in);
            PreparedStatement stmt=con.prepareStatement("Insert into
InsertCart values(?,?,?,?,?)");
            stmt.setString(1,code);
            stmt.setString(2,name);
            stmt.setString(3,quant);
            stmt.setString(4,price);
            stmt.setString(5,netprice);
            int rs=stmt.executeUpdate();
            if(rs==1)
                b=true;
            con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

        return b;
    }
}
//***** STOCK HANDLER PAGE *****
class StockHandlerPage extends JFrame implements ActionListener,ItemListener
{
    /*
    NOTE: in card2 , del(RadioButton)
           i.e. 'delete' is used for
           search, delete as well as modificaion.
           while modification is transfered to card1 add(Radio Button)
    */
    //Declarations for the Entire Frame
    static JRadioButton add,del,view;
    static CardLayout clo;
    static JPanel cards; //card1 and card2
    static JButton back;
    //Declarations for Card1
    static JButton addr;
    static JTextField code1,name1,quan1,rate1,mrp1;
    static Choice exp1,exp2;
    //Declarations for card2
    static JTextField searchbox;
    static JButton search;
    static JTextField name,quan,rate,mrp,exp;
    static JButton delete,modify;
    static String date;
    //used for containing date
    public void showFrame()
    {
        Container c=this.getContentPane();
        //-----header -----
        JLabel title=new JLabel("Add, Delete or Modify the Records here :");
        //Title
        ButtonGroup g=new ButtonGroup();
        //RadioButton
        add=new JRadioButton("Add",true);
        del=new JRadioButton("Search/Delete/Modify Records",false);
        view=new JRadioButton("View Stock",false);
        g.add(add);
        g.add(del);
        g.add(view);
        JPanel rbutton=new JPanel();
        rbutton.add(add);
        rbutton.add(del);
    }
}

```

```

rbutton.add(view);
JPanel p=new JPanel();
//Title + RadioButton Panel
    p.add(title);
    p.add(rbutton);
//----- Back BUTTON
back=new JButton("Back");
JPanel toolbar=new JPanel();
toolbar.setLayout(new BorderLayout());
toolbar.add(back,BorderLayout.EAST);
toolbar.add(p);
//-----Body-----
JPanel card1,card2;
card1=new JPanel();
card2=new JPanel();
clo=new CardLayout();
cards=new JPanel();
cards.setLayout(clo);
cards.add(card1,"ADD");
cards.add(card2,"DELETE");
card1.setLayout(null); card1.setBackground(Color.WHITE);
card2.setLayout(null); card2.setBackground(Color.WHITE);
//--- Designing THE CARDS----- CARD1 and CARD2 -----
    //##### CARD 1 #####
        //ADD button and Title
        JLabel c1title=new JLabel("Enter the Details of the
Medicine:"); //title
        c1title.setFont(new Font("Times New
Roman",Font.BOLD,18));
        c1title.setSize(320,35);
        c1title.setLocation(350,10+30);
        addr=new JButton("ADD RECORD");
        addr.setSize(150,30);
        addr.setLocation(630,310);
//-----Setting up size of other labels
and textFields
        JLabel ll0=new JLabel("Product Code :");
        JLabel ll1=new JLabel("Medicine Name :");
        JLabel ll2=new JLabel("Quantity :");
        JLabel ll3=new JLabel("Rate :");
        JLabel ll4=new JLabel("MRP :");
        JLabel ll5=new JLabel("Exp. (MM/YYYY) :");
        code1=new JTextField();
        name1=new JTextField();
        quan1=new JTextField();

```

```

rate1=new JTextField();
mrp1=new JTextField();
exp1=new Choice();
exp2=new Choice();
exp1.add("January");
exp1.add("February");
exp1.add("March");
exp1.add("April");
exp1.add("May");
exp1.add("June");
exp1.add("July");
exp1.add("August");
exp1.add("September");
exp1.add("October");
exp1.add("November");
exp1.add("December");
//this for loop is used
//to insert value in year combo box
for(int i=2017;i<=2029;i++)
{
    String st=String.valueOf(i);
    exp2.add(st);
}

//Label size
ll0.setSize(100,30);
ll1.setSize(100,30);
ll2.setSize(100,30);
ll3.setSize(100,30);
ll4.setSize(100,30);
ll5.setSize(100,30);
//TextField size
code1.setSize(280,28);
name1.setSize(280,28);
quan1.setSize(100,28);
rate1.setSize(100,28);
mrp1.setSize(100,28);
exp1.setSize(90,28);
exp2.setSize(58,28);
//Label Location
ll0.setLocation(50+100,60+50);
ll1.setLocation(50+100,100+50);
ll2.setLocation(50+100,140+50);
ll3.setLocation(50+100,180+50);
ll4.setLocation(50+100,220+50);
ll5.setLocation(50+100,260+50);

```

```

//textField location
code1.setLocation(200+100,60+50);
name1.setLocation(200+100,100+50);
quan1.setLocation(200+100,140+50);
rate1.setLocation(200+100,180+50);
mrp1.setLocation(200+100,220+50);
exp1.setLocation(200+100,260+50);
exp2.setLocation(200+190,260+50);

//---Adding components to card2
card1.add(c1title);
card1.add(ll0);
card1.add(ll1);
card1.add(ll2);
card1.add(ll3);
card1.add(ll4);
card1.add(ll5);
card1.add(code1);
card1.add(name1);
card1.add(quan1);
card1.add(rate1);
card1.add(mrp1);
card1.add(exp1);
card1.add(exp2);
card1.add(addr);

//##### CARD 2 #####
JLabel c2title=new JLabel("Enter the Product Code to
search for : "); //title
c2title.setFont(new Font("Times New
Roman",Font.BOLD,18));
c2title.setSize(320,35);
c2title.setLocation(350,10+30);
searchbox=new JTextField();
//search box
searchbox.setSize(350,30);
searchbox.setLocation(250,50+30);
search=new JButton("SEARCH RECORD");
//search button
search.setSize(150,30);
search.setLocation(630,50+30);
//DELETE and MODIFY button
delete=new JButton("DELETE RECORD");
modify=new JButton("MODIFY RECORD");
modify.setEnabled(false);
delete.setSize(150,30);
modify.setSize(150,30);

```

and textFields

```
delete.setLocation(630,240);
modify.setLocation(630,310);
//-----Setting up size of other labels
```

```
JLabel l1=new JLabel("Medicine Name :");
JLabel l2=new JLabel("Quantity :");
JLabel l3=new JLabel("Rate :");
JLabel l4=new JLabel("MRP :");
JLabel l5=new JLabel("Exp. (MM-YYYY) :");
name=new JTextField();
quan=new JTextField();
rate=new JTextField();
mrp=new JTextField();
exp=new JTextField();
```

```
//Label size
```

```
l1.setSize(100,30);
l2.setSize(100,30);
l3.setSize(100,30);
l4.setSize(100,30);
l5.setSize(100,30);
//TextField size
name.setSize(300,28);
```

```
name.setEditable(false);
```

```
quan.setSize(100,28);
```

```
quan.setEditable(false);
```

```
rate.setSize(100,28);
```

```
rate.setEditable(false);
```

```
mrp.setSize(100,28);
```

```
mrp.setEditable(false);
```

```
exp.setSize(100,28);
```

```
exp.setEditable(false);
```

```
//Label Location
```

```
l1.setLocation(50+100,100+50);
l2.setLocation(50+100,140+50);
l3.setLocation(50+100,180+50);
l4.setLocation(50+100,220+50);
l5.setLocation(50+100,260+50);
//textField location
name.setLocation(200+100,100+50);
quan.setLocation(200+100,140+50);
rate.setLocation(200+100,180+50);
mrp.setLocation(200+100,220+50);
exp.setLocation(200+100,260+50);
```

```
//---Adding components to card2
card2.add(c2title);
```



```

        card2.add(searchbox);
        card2.add(search);
        card2.add(l1);
        card2.add(l2);
        card2.add(l3);
        card2.add(l4);
        card2.add(l5);
        card2.add(name);
        card2.add(quan);
        card2.add(rate);
        card2.add(mrp);
        card2.add(exp);
        card2.add(delete);
        card2.add(modify);
c.add(toolbar, BorderLayout.NORTH);
c.add(cards, BorderLayout.CENTER);
//-----Setting up frame properties-----
setTitle("Test");
setVisible(true);
setSize(1000,600);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLocationRelativeTo(null);
setResizable(false);
//----- Adding Listeners -----
addr.addActionListener(this);
search.addActionListener(this);
modify.addActionListener(this);
delete.addActionListener(this);
back.addActionListener(this);
add.addItemListener(this);
view.addItemListener(this);
del.addItemListener(this);
exp1.addItemListener(this);
exp2.addItemListener(this);
}
/*get function is used to get value from
searchstock table so that it can be
inserted in the textbox      it accepts 5 variables
nam is for name,qua for quantity,ra for rate
mr for mrp and exp2 for containing expiry variables*/
public void get(String nam,int qua,double ra,double mr,String exp2)
//funtion to set text value after search
{
    name.setText(nam);
    quan.setText(String.valueOf(qua));

```

```

        rate.setText(String.valueOf(ra));
        mrp.setText(String.valueOf(mr));
        exp.setText(exp2);
    }
    public void itemStateChanged(ItemEvent e)
    {
        if(add.isSelected())
        {
            clo.show(cards,"ADD");
        }
        else if(del.isSelected())
        {
            clo.show(cards,"DELETE");
        }
        else if(view.isSelected())
        {
            /* when view radio button is selected
            then stockjtable class object sj
            is created and table is shown in jtable
            format */
            StockJTable sj=new StockJTable();
            sj.showFrame();
            dispose();
        }
    }
    /* clear function is used to clear
    all previos values of text box and list box
    to previous stage to perform another operation */
    public void clear()
    {
        code1.setText("");
        name1.setText("");
        quan1.setText("");
        rate1.setText("");
        mrp1.setText("");
        exp1.select("January");
        exp2.select("2017");
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==back)
        {
            new MainMenu();
            dispose();
        }
    }

```

```

else if(e.getSource()==delete)
{
    /* Deletestock class object d
    is created then
    code value is passed to delete method
    if it is not equal to null */
    DeleteStock d=new DeleteStock();
    String c=searchbox.getText();
    if(c.equals(""))
    {
        JOptionPane.showMessageDialog(this,"Product
code invalid");

        searchbox.setText("");
    }
    else
    {
        boolean b=d.Delete(c);
        if(b==true)

JOptionPane.showMessageDialog(this,"Successfully deleted");
        else if(b==false)

JOptionPane.showMessageDialog(this,"Product code not available");
        name.setText("");
        quan.setText("");
        rate.setText("");
        mrp.setText("");
        exp.setText("");
        searchbox.setText("");
    }
}
else if(e.getSource()==addr)
{
    /* first code value is given
    to variable c then check that
    if length is less than 3 or does
    not containing null values */
    String c=code1.getText();
    if(c.length()<3 || c.equals("") ||c.length()>13)
    {
        JOptionPane.showMessageDialog(this,"Invalid
product code");

        //clear();
    }
    else

```

```

        {
            /* checkcode class object ch is created
            then method check is called and passed
            code that whether code already exists in table
            or not boolean type variable cod get value
            from check method if it returns false
            then code doesnot exists in table and new record
            is to be inserted otherwise update the
            values in the existing record of table*/
            CheckCode ch=new CheckCode();
            boolean cod=ch.Check(c);
            if(cod==false)
            {
                /* after that valid method of validatestock
                class is called and check that whether
                all value of inserted field is valid or not
                it passes 4 values of textbox valid
                method returns boolean type variable
                if it returns true then value is inserted
                in the table else error message in the
                form of message dialog box is generated

                */
                ValidateStock vs=new ValidateStock();
                boolean
v=vs.valid(name1.getText(),quan1.getText(),rate1.getText(),mrp1.getText());
                if(v==true)
                {
                    String n=name1.getText();
                    int
                    double
                    double
                    String ex;//used to concate date
                    String
                    String
                    ex=month+year;
                    InsertStock Is=new
                    //passing value to insert table
                    function

```

```

        c=c.trim();
        c=c.toUpperCase();
        boolean bo=Is.Stock(c,n,q,r,m,ex);
        /* after inserting in table by passing
all
insertstock
successfully inserted
message is displayed*/

        values in stock method of
        class if it returns true then
        message is shown otherwise error
        if(bo==true)

JOptionPane.showMessageDialog(this," Successfully Added/Modified");
        else

        JOptionPane.showMessageDialog(this,"Error while inserting");

        //after adding all textfields become
empty
        clear();
    }
    else
    {

JOptionPane.showMessageDialog(this,"Invalid Data Provided");
        //clear();
    }
}
else
{
    /*modify coding
        if code already exists in the table
        then automatically control comes to else
        part to update table record
        once again valid method of validatestock
        class is called and
        all values is passed to validate all
        fields it also returns boolean type variable
        it returns true if all are valid otherwise
        returns false in case of true operation of
        next step is to be performed otherwise

generates
        error message */
        ValidateStock vst=new ValidateStock();

```

```

                                boolean
v=vst.valid(name1.getText(),quan1.getText(),rate1.getText(),mrp1.getText());
                                if(v==true)
                                {
UpdateStock cs=new
UpdateStock();
                                String ca=code1.getText();
                                String n=name1.getText();
                                String q=quan1.getText();
                                String r=rate1.getText();
                                String m=mrp1.getText();
                                String
exw=exp1.getSelectedItem()+exp2.getSelectedItem();
                                /* after storing all values in string
type
                                variable it is passed to upstock
method
                                of upstock table */
                                boolean
up=cs.upstock(ca,n,q,r,m,exw);
                                if(up==true)

JOptionPane.showMessageDialog(this,"Record Successfully updated");
                                else

JOptionPane.showMessageDialog(this,"Error while updating");
                                clear();
                                }
                                else
                                {

JOptionPane.showMessageDialog(this,"Invalid data feeded while modifying");
                                clear();
                                }
                                }
                                }
}
else if(e.getSource()==search)
{
/* after clicking on search modify
button is enabled so that modify
operatio can be performed only after searching
operation*/
modify.setEnabled(true);
SearchStock se=new SearchStock();

```

```

        if(searchbox.getText().equals(""))
        {
            JOptionPane.showMessageDialog(this,"Invalid
product code!");
            searchbox.setText(" ");
        }
        else
        {
            String c=searchbox.getText();
            c=c.trim();
            c=c.toUpperCase();
            if(se.Search(c)==false)

JOptionPane.showMessageDialog(this,"code not available");
            else

JOptionPane.showMessageDialog(this,"Successfully find");
        }
    }
    else if(e.getSource()==modify)
    {
        code1.setText(searchbox.getText());
        name1.setText(name.getText());
        quan1.setText(quan.getText());
        rate1.setText(rate.getText());
        mrp1.setText(mrp.getText());
        String c=searchbox.getText();
        SearchStock ss=new SearchStock();
        boolean bl=ss.Search(c);
        /* c variable contains
        code value is to be sarched
        if it contains null value then
        searching operation is not performed
        if search method of serchstock class
        returns false then record is not
        found error message is generated
        otherwise success message is generated*/
        if(c.equals(""))
        {
            JOptionPane.showMessageDialog(this,"Product
code invalid");
            searchbox.setText("");
        }
        else if(bl==false)

```

```

        {
            JOptionPane.showMessageDialog(this,"Product
code not available");
            searchbox.setText("");
        }

        else
            add.setSelected(true);
        /* to modify control is transfered to
        add record card after clicking on add button
        modify operation is performed then modify button
        is once again disabled and enabled again only
        when searching operation is to be performed */
        modify.setEnabled(false);
    }
}

}

//***** Stock PAGE *****/
/*this class is used to show values of stock table
when ever user clicks on view stock radio button in
stockhandler page */
class StockJTable extends JFrame implements ActionListener
{
    static JButton back;
    static JTable table;
    int row=0;
    static String s[][];
    public void showFrame()
    {
        Container c=this.getContentPane();
        //creating header Components
        back=new JButton("BACK");
        JPanel p1=new JPanel();
        p1.setLayout(new BorderLayout());
        p1.add(back,BorderLayout.WEST);
        //Creating Body Components
        String head[]={"Code","Name","Quantity","Rate","Mrp","Expiry"};
        /* tablestock class object ts is created
        to use method to display record in stock
        getrow method returns number of records
        in the form of integer so that
        jtable row can be adjusted in that format */
        Tablestock ts=new Tablestock();
        row=ts.getrow();
        /* string s double dimension array object

```



```

        is created to pass and accept values from
        views method of table stock class */
        s=new String[row][6];
        if(row!=0)
            s=ts.views(s);
        else
            JOptionPane.showMessageDialog(this,"No Product in
stock");

        String body[][]=s;
        table=new JTable(body,head);
        this.add(new JScrollPane(table));
        //Adding Components to the Container c
        c.add(p1,BorderLayout.NORTH);
        //Setting Frame Properties
        this.setTitle("StockJTable");
        this.setSize(700,550);
        this.setVisible(true);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //Adding Listeners to the buttons and other components
        back.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==back)
        {
            StockHandlerPage p=new StockHandlerPage();
            p.showFrame();
            dispose();
        }
    }
}
/* tablestock class is used to extract number of
records in table and return record in the form
of 2d array */
class Tablestock
{
    /* getrow function is used to count number of
    rows in existing table so that array arr is
    created according to the row
    this function does not accept any value but
    return integer type variable that is number of
    records in the table */
    public int getrow()

```

```

    {
        int row=0;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from StockTable");
            while(rs.next())
                row++;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return row;
    }
    /* this function views accepts
    one string type double dimension array
    and also returns a double dimension
    string type variable */
    public String[][] views(String arr[][])
    {
        int row=getrow();
        arr=new String [row][6];
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from
StockTable");

            int c=0;
            while(rs.next())
            {
                arr[c][0]=rs.getString(1);
                arr[c][1]=rs.getString(2);
                arr[c][2]=rs.getString(3);
                arr[c][3]=rs.getString(4);
                arr[c][4]=rs.getString(5);
                arr[c][5]=rs.getString(6);
                c++;
            }
        }
        catch(Exception e)
    }

```

```

        {
            System.out.print(e);
        }

        return arr;
    }
}
/* updatetock class is used to update
the record of stocktable
after searching operation and user press modify button
*/
class UpdateStock
{
    /* upstock method is used to update record
of stock table
it accepts 6 values that is code,name
quantity,rate,mrp,exp */
    public boolean upstock(String c,String n,String q,String r,String m,String e)
    {
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            PreparedStatement stmt=con.prepareStatement("update
StockTable set code=?,name=?,quant=?,rate=?,mrp=?,exp=?where code=?");
            stmt.setString(1,c);
            stmt.setString(2,n);
            stmt.setString(3,q);
            stmt.setString(4,r);
            stmt.setString(5,m);
            stmt.setString(6,e);
            stmt.setString(7,c);
            int rn=stmt.executeUpdate();
            if(rn==1)
                return true;
        }
        catch(Exception exx)
        {
            System.out.println(exx);
        }
        return false;
    }
}
/* validate stock class is used to validate
all fields in stock class while inputing in

```

textfield after pressing add record button
a valid method is defined in this class that
accepts 4 values and return 1 boolean type
variable */

class ValidateStock

```
{
    public boolean valid(String name,String quant,String rate,String mrp)
    {
        boolean b=true;
        name=name.trim();
        //name=name.toUpperCase();
        /* validity of name is based on
        name must contain alphabets
        its length should be between 3 and 19 */
        if(name.length()>40 ||name.length()<3 || name.equals("") )
        {
            b=false;
            System.out.println("Symbols allowed in Product name and
length range is 3-40");
        }
        /* quantity length must be between 1 and 4 and must contain
        only numeric value and don't contain null values */
        if(quant.length()<1 || quant.equals("") || quant.length()>4)
            b=false;
        else
        {
            for(int i=0;i<quant.length();i++)
            {
                if(quant.charAt(i)<'0' || quant.charAt(i)>'9')
                {
                    b=false;
                    break;
                }
            }
        }
        boolean ch=check(rate);
        if(ch==false)
            b=false;
        boolean cha=check(mrp);
        if (cha==false)
            b=false;
        if(b!=false)
        {
            Double r=Double.parseDouble(rate);
            Double m=Double.parseDouble(mrp);
```

```

        if(r>m)
            b=false;
    }
    return b;
}
public boolean check(String r)
{
    boolean b=true;
    if(r.length()<1 || r.equals("") || r.length()>7)
        b=false;
    else
    {
        for(int i=0;i<r.length();i++)
        {
            if(r.charAt(i)<'0' || r.charAt(i)>'9')
            {
                int na=(int)r.charAt(i);
                if(na==46)                //46 is ascii value of
                    dot
                    b=true;
                else
                {
                    b=false;
                    break;
                }
            }
        }
        return b;
    }
}
// this class is used to check whether code is
// already existing or not
class CheckCode
{
    public boolean Check(String scode)
    {
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            PreparedStatement stmt=con.prepareStatement("select * from
StockTable where code=?");
            stmt.setString(1,scode);
            ResultSet rs=stmt.executeQuery();

```

```

        if(rs.next())
            return true;
    }
    catch(Exception et)
    {
        System.out.println(et);
    }
    return false;
}

}

//class to delete stock records in the table
class DeleteStock
{
    public boolean Delete(String code)                //function to insert
    record in table
    {
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Scanner in=new Scanner(System.in);
            PreparedStatement stmt=con.prepareStatement("delete from
StockTable where code=?");
            String r=code;
            stmt.setString(1,r);
            int rs=stmt.executeUpdate();
            if(rs==1)
                return true;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return false;
    }
}

//class to search stock
class SearchStock
{
    public boolean Search(String code)
    {
        String billing=code;
        if(code.length()>4)
            if(code.substring(0,4).equals("bill"))
                code=code.substring(4,code.length());
    }
}

```

```

String name2="";
int qu=0;
double rat=0.0;
double mrp2=0.0;
String exp3="";
boolean ba=false;
try
{
    connect cn=new connect();
    Connection con =cn.aconnect();
    PreparedStatement stmt=con.prepareStatement("Select * from
StockTable where code=?");
    String r=code;
    stmt.setString(1,r);
    ResultSet rs=stmt.executeQuery();
    if(rs.next())
    {
        ba=true;
        name2=rs.getString(2);
        qu=rs.getInt(3);
        rat=rs.getDouble(4);
        mrp2=rs.getDouble(5);
        exp3=rs.getString(6);
    }
    else
        System.out.println("Record is not found");
}
catch(Exception e)
{
    System.out.println(e);
}

if(billing.equals(code))
{
    StockHandlerPage sp=new StockHandlerPage();
    sp.get(name2,qu,rat,mrp2,exp3);
}
else
{
    BillingPage bp=new BillingPage();
    bp.get(name2,qu,rat);
}

return ba;
}
}

```

```

//class to insert stock records in the table
class InsertStock
{
    public boolean Stock(String c,String n,int q,double r,double m,String ex)
        //function to insert record in table
    {
        boolean b=false;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Scanner in=new Scanner(System.in);
            PreparedStatement stmt=con.prepareStatement("Insert into
StockTable values(?,?,?,?,?,?)");
            stmt.setString(1,c);
            stmt.setString(2,n);
            stmt.setInt(3,q);
            stmt.setDouble(4,r);
            stmt.setDouble(5,m);
            stmt.setString(6,ex);
            int rs=stmt.executeUpdate();
            if(rs==1)
                b=true;
            //con.close();
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return b;
    }
}

//***** MainMenu Page *****
class MainMenu extends JFrame implements ActionListener
{
    JMenuItem a1,a2, b1,b2,b3,b4,b5,c1;
    public MainMenu()
    {
        Pharmacy.active="MainMenu";
        //working with menubar
        JMenuBar bar=new JMenuBar();
        JMenu m1=new JMenu("User");
        JMenu m2=new JMenu("Manage");
        JMenu m3=new JMenu("Billing");
        bar.add(m1);
    }
}

```



```

        bar.add(m2);
        bar.add(m3);
        a1=new JMenuItem("New Signup");
        a2=new JMenuItem("Login As Different user");
        b1=new JMenuItem("Search");
        b2=new JMenuItem("Add");
        b3=new JMenuItem("Modify");
        b4=new JMenuItem("Delete");
        b5=new JMenuItem("View Records as Table");
        c1=new JMenuItem("Billing Page");
        a1.setToolTipText("Create an Account for a new User");
        a2.setToolTipText("Re-Login as a different user");
        b1.setToolTipText("Search the Medicines in stock by their product
code");
        b2.setToolTipText("Add new Medicines to the stock");
        b3.setToolTipText("Update a Product Record");
        b4.setToolTipText("Delete Existing Record from the stock");
        b5.setToolTipText("View the Medicines in stock as TABLE");
        c1.setToolTipText("Jump to Billing Page");
        m1.add(a1);
        m1.add(a2);
        m2.add(b1);
        m2.add(b2);
        m2.add(b3);
        m2.add(b4);
        m2.add(b5);
        m3.add(c1);
        // working with rest of the frame
        setLayout(null);
        JLabel l=new JLabel();
        l.setIcon(new ImageIcon("images\\menu.jpg"));
        l.setSize(800,500);
        l.setLocation(0,-15);
        //----- adding components to frame-----
        add(l);
        //-----Setting up frame properties-----
        this.setJMenuBar(bar);
        this.setTitle("Main Menu");
        this.setVisible(true);
        this.setSize(800,500);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        //----- adding listener -----
        a1.addActionListener(this);

```

```

        a2.addActionListener(this);
        b1.addActionListener(this);
        b2.addActionListener(this);
        b3.addActionListener(this);
        b4.addActionListener(this);
        b5.addActionListener(this);
        c1.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==a1)
        {
            SignupPage p=new SignupPage();
            p.showFrame();
            dispose();
        }
        else if(e.getSource()==a2)
        {
            LoginPage p=new LoginPage();
            p.showFrame();
            dispose();
        }
        else if(e.getSource()==b1)
        {
            StockHandlerPage p=new StockHandlerPage();
            p.showFrame();
            p.del.setSelected(true);
            JOptionPane.showMessageDialog(this,"Search the Record to be
deleted");
            dispose();
        }
        else if(e.getSource()==b2)
        {
            StockHandlerPage p=new StockHandlerPage();
            p.showFrame();
            p.add.setSelected(true);
            dispose();
        }
        else if(e.getSource()==b3)
        {
            StockHandlerPage p=new StockHandlerPage();
            p.showFrame();
            p.del.setSelected(true);
            JOptionPane.showMessageDialog(this,"Search the Record to be
Modified");

```

```

        dispose();
    }
    else if(e.getSource()==b4)
    {
        StockHandlerPage p=new StockHandlerPage();
        p.showFrame();
        p.del.setSelected(true);
        JOptionPane.showMessageDialog(this,"Search the Record to be
Deleted");
        dispose();
    }
    else if(e.getSource()==b5)
    {
        StockHandlerPage p=new StockHandlerPage();
        p.showFrame();
        p.view.setSelected(true);
        dispose();
    }
    else if(e.getSource()==c1)
    {
        BillingPage p=new BillingPage();
        p.showFrame();
        dispose();
    }
}
}
class ViewReport
{
    public int getrow()
    {
        int row=0;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from
PharmacyReport");
            while(rs.next())
                row++;
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}

```

```

        return row;
    }
    /* this function */
    public String[][] views()
    {
        int row=getrow();
        String arr[][]=new String [row][6];
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from
PharmacyReport");

            int c=0;
            while(rs.next())
            {
                arr[c][0]=rs.getString(1);
                arr[c][1]=rs.getString(2);
                arr[c][2]=rs.getString(3);
                arr[c][3]=rs.getString(4);
                arr[c][4]=rs.getString(5);
                arr[c][5]=rs.getString(6);
                c++;
            }
        }
        catch(Exception e)
        {
            System.out.print(e);
        }

        return arr;
    }
}
class Report extends JFrame implements ActionListener
{
    static JButton back;
    static JTable table;
    int row=0;
    static String s[][];
    public void showFrame()
    {
        Container c=this.getContentPane();
        //creating header Components
        back=new JButton("BACK");
    }
}

```

```

        JPanel p1=new JPanel();
        p1.setLayout(new BorderLayout());
        p1.add(back,BorderLayout.WEST);
        //Creating Body Components
        String
head[]={"Code","Name","Quantity","MRP","Price","Date","Total"};
        /* tablestock class object ts is created
        to use method to display record in stock
        getrow method returns number of records
        in the form of integer so that
        jTable row can be adjusted in that format */
        ViewReport ts=new ViewReport();
        row=ts.getrow();
        /* string s double dimension array object
        is created to pass and accept values from
        views method of ViewReport class */
        s=new String[row][6];
        if(row!=0)
            s=ts.views();
        else
            JOptionPane.showMessageDialog(this,"No purchase is
done");

        double total=0.0;
        String tab[][]=new String[row][7];
        String date="",da="";
        for(int i=0;i<row;i++)
        {
            for(int j=0;j<6;j++)
                tab[i][j]=s[i][j];
            date=s[i][5];
            double t=Double.parseDouble(tab[i][4]);
            total+=t;
            if(i!=0)
            {
                if(da.equals(date))
                    tab[i][6]="";
                else
                {
                    double tr=total-t;
                    tab[i-1][6]=String.valueOf(tr);
                    total=t;
                }
            }
            da=date;
            if(i==row-1)

```

```

        tab[i][6]=String.valueOf(total);
    }
    String body[][]=tab;
    table=new JTable(body,head);
    this.add(new JScrollPane(table));
    //Adding Components to the Container c
    c.add(p1,BorderLayout.NORTH);
    //Setting Frame Properties
    this.setTitle("MedicineReport");
    this.setSize(700,550);
    this.setVisible(true);
    this.setLocationRelativeTo(null);
    this.setResizable(false);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
//Adding Listeners to the buttons and other components
    back.addActionListener(this);
}
public void actionPerformed(ActionEvent e)
{
    if(e.getSource()==back)
    {
        //new MainMenu();
        //p.showFrame();
        dispose();
    }
}
}
//***** CART PAGE *****
class CartPage extends JFrame implements ActionListener
{
    static JButton back,print;
    static JTextField total;
    static JTable tabl;
    static String code,name,quant,mrp,price;
    String s[][];
    int row=0;
    static Double tot=0.0;
    public void get(int r)
    {
        row=r;
    }
    public void showFrame()
    {
        Container c=this.getContentPane();
        //creating header Components

```

```

JLabel title=new JLabel("CART");
JPanel p0=new JPanel();
p0.add(title);
title.setFont(new Font("Times New Roman",Font.BOLD,20));
back=new JButton("BACK");
JPanel p1=new JPanel();
p1.setLayout(new BorderLayout());
p1.add(back,BorderLayout.WEST);
p1.add(p0,BorderLayout.CENTER);
//creating footer Components
print=new JButton("Print/Buy");
JLabel lbl=new JLabel("TOTAL = ");
total=new JTextField(15);
total.setEditable(false);
JPanel p2=new JPanel();
p2.add(lbl);
p2.add(total);
p2.add(print);
//Creating Body Components
String head[]={"Product Code","Product
Name","Quantity","MRP","Price"};
int ro=row-1;
s=new String[row][5];
//String arr=new String[row][5];
viewcart vc=new viewcart();
if(row>0)
    s=vc.view(s,row);
else
    JOptionPane.showMessageDialog(this,"No Product in cart");
tot=0.0;
//System.out.println("value in total"+tot);
//System.out.println("value of row="+row);
for(int r=0;r<row;r++)
{
    String sr=s[r][4];
    if(sr.equals("null"))
        tot=0.0;
    else
    {
        //System.out.println("value"+s[r][4]);
        tot+=Double.parseDouble(sr);
        //System.out.println("Total"+tot);
    }
}
total.setText(String.valueOf(tot));

```

```

        String body[][]=s;
        tabl=new JTable(body,head);
        this.add(new JScrollPane(tabl));
        //Adding Components to the Container c
        c.add(p1, BorderLayout.NORTH);
        c.add(p2, BorderLayout.SOUTH);
        //Setting Frame Properties
        this.setTitle("Cart");
        this.setSize(700,550);
        this.setVisible(true);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        //Adding Listeners to the buttons and other components
        back.addActionListener(this);
        print.addActionListener(this);
    }
    public String date()
    {
        Calendar c=Calendar.getInstance();
        int d=c.get(Calendar.DATE);
        int m=c.get(Calendar.MONTH)+1;
        int y=c.get(Calendar.YEAR);
        String da;
        if(d<10)
        {
            String s="0";
            da=String.valueOf(d);
            da=s+da;
        }
        else
            da=String.valueOf(d);
        String mon;
        if(m<10)
        {
            String mo="0";
            mon=String.valueOf(m);
            mon=mo+mon;
        }
        else
            mon=String.valueOf(m);
        String dat=da+"-"+mon+"-"+String.valueOf(y);
        return dat;
    }
    public void actionPerformed(ActionEvent e)

```



```

        {
            if(e.getSource()==back)
            {
                BillingPage p=new BillingPage();
                p.showFrame();
                dispose();
            }
            else if(e.getSource()==print)
            {
                if(row<1)
                    JOptionPane.showMessageDialog(this,"No Product
Available");
                else
                {
                    JOptionPane.showMessageDialog(this,"Purchase
Successfull.");
                    Updatequant uq=new Updatequant();
                    uq.upstock();
                    uq.trunc();
                    String report[][]=new String[row][6];
                    for(int i=0;i<row;i++)
                    {
                        for(int j=0;j<5;j++)
                        {
                            report[i][j]=s[i][j];
                            //System.out.println(report[i][j]);
                        }
                        report[i][5]=date();
                        //System.out.println(report[i][5]);
                    }
                    InsertReport ir=new InsertReport();
                    ir.pass(report,row);
                    dispose();
                    new MainMenu();
                } //p.showFrame();
            }
        }
    }
}
class InsertReport
{
    public void pass(String arr[],int row)
    {
        for(int i=0;i<row;i++)
        {

```

```

        boolean
b=Report(arr[i][0],arr[i][1],arr[i][2],arr[i][3],arr[i][4],arr[i][5]);
        //System.out.println(b);
    }
}
public boolean Report(String code,String name,String quantity,String
mrp,String price,String mdate) //function to insert record in table
{
    boolean b=false;
    try
    {
        connect cn=new connect();
        Connection con =cn.aconnect();
        Scanner in=new Scanner(System.in);
        PreparedStatement stmt=con.prepareStatement("Insert into
PharmacyReport values(?,?,?,?,?,?)");
        stmt.setString(1,code);
        stmt.setString(2,name);
        stmt.setString(3,quantity);
        stmt.setString(4,mrp);
        stmt.setString(5,price);
        stmt.setString(6,mdate);
        int rs=stmt.executeUpdate();
        if(rs==1)
            b=true;
        //con.close();
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    return b;
}
}
class Updatequant
{
    public void upstock()
    {
        String code="";
        int quantity=0;
        int quant=0;
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();

```

```

        Statement stmt=con.createStatement();
        ResultSet rs=stmt.executeQuery("select * from
InsertCart");
        while(rs.next())
        {
            code=rs.getString(1);
            quantity=Integer.parseInt(rs.getString(3));
            quant=quantst(code);
            update(code,quant-quantity);
        }
    }
    catch(Exception exx)
    {
        System.out.println(exx);
    }
}

public void update(String code,int left)
{
    if(left<1)
        left=0;
    code=code.toUpperCase();
    try
    {
        connect cn=new connect();
        Connection con =cn.aconnect();
        PreparedStatement stmt=con.prepareStatement("update StockTable
set quant=? where code=?");
        stmt.setInt(1,left);
        stmt.setString(2,code);
        int rn=stmt.executeUpdate();
        if(rn==1)
            System.out.println("Successfully updated");
        else
            System.out.println("Error");
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}

public void trunc()
{
    try
    {
        connect cn=new connect();

```

```

        Connection con =cn.aconnect();
        Statement stmt=con.createStatement();
        stmt.execute("truncate table InsertCart");
        System.out.println("success");
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
}
public int quantst(String code)
{
    code=code.toLowerCase();
    int q=0;
    try
    {
        connect cn=new connect();
        Connection con =cn.aconnect();
        PreparedStatement stmta=con.prepareStatement("Select
* from StockTable where code=?");
        stmta.setString(1,code);
        ResultSet rsa=stmta.executeQuery();
        if(rsa.next())
            q=rsa.getInt(3);
    }
    catch(Exception e)
    {
        System.out.println(e);
    }
    return q;
}
}
class viewcart
{
    static Double total=0.0;
    public String[][] view(String arr[],int row)
    {
        try
        {
            connect cn=new connect();
            Connection con =cn.aconnect();
            Statement stmt=con.createStatement();
            ResultSet rs=stmt.executeQuery("Select * from
InsertCart");

            int c=0;

```

```

        while(rs.next())
        {
            arr[c][0]=rs.getString(1);
            arr[c][1]=rs.getString(2);
            arr[c][2]=rs.getString(3);
            arr[c][3]=rs.getString(4);
            arr[c][4]=rs.getString(5);
            total=total+Double.parseDouble(arr[c][4]);
            c++;
        }
    }
    catch(Exception e)
    {
        System.out.print(e);
    }
    return arr;
}
}

```

OUTPUT

Sign Up
Welcome! Get Signed Up

| | |
|-------------------------|-----------------------------|
| First Name : | SAEE |
| Last Name : | GAVALI |
| Email Id : | saeegavali28@gmail.com |
| Preferred username : | sae28 |
| Authentication number : | ncBCA201417 |
| User Rights : | Billing and Managing Stocks |
| Password : | ***** |
| Confirm Password : | ***** |

☒ I hereby declare that the information provided by me above is correct.
And only I will be responsible for any kind inconvenience caused by incorrect information

[Sign UP](#) [Back](#) [Generate Report](#)

SignUp successful!!

Congratulations, SAEЕ GAVALI. Your Account created successfully!

User username is : sae28

[Login](#) [Generate Report](#)

Log in

LogIn to get access to the Stocks and Short List the Medicines...

Log in as: ☐ Billing Only ☒ Stock Handler

Username


Password

Cart

CART

| Product Code | Product Name | Quantity | MRP | Price |
|--------------|--------------|----------|------|-------|
| 0001 | Dolo650 | 5 | 10.0 | 50.0 |

Message

 Purchase Successfull.

TOTAL =

Billing Page

Product Billing

Product Code : 0001

Product Name : Dolo650

Quantity : 5

Left in Stock: 95

Price(Rs) : 10.0

Net Price(Rs): 50.0

View Cart Add To Cart BACK

Message

Added to Cart.

OK

Log in

LogIn to get access to the Stocks and Short List the Medicines...

Log in as: ☐ Billing Only ☒ Stock Handler

Username superuser

Password

Log In Sign Up Generate Report

Test

Add, Delete or Modify the Records here : ☐ Add ☒ Search/Delete/Modify Records ☐ View Stock

Back

Enter the Product Code to search for :

0001

SEARCH RECORD

Medicine Name : Dolo650

Quantity : 100

Rate : 10.0

MRP : 15.0

Exp. (MM-YYYY) : April2023

Message
Successfully find
OK

DELETE RECORD

MODIFY RECORD

Stock/Table

BACK

| Code | Name | Quantity | Rate | Mrp | Expiry |
|------|--------------|----------|------|-----|-------------|
| 1111 | dolo650 | 500 | 10 | 15 | April2023 |
| 2345 | drgy | 50 | 2 | 5 | January2023 |
| 1112 | crocina100mg | 85 | 5 | 8 | January2023 |
| 1113 | cofsil | 62 | 6 | 10 | March2023 |
| 1114 | meptasplus | 75 | 10 | 15 | April2024 |
| 0001 | Dolo650 | 100 | 10 | 15 | April2023 |

TESTING

Testing Approach

Computer s/w has become more complex. The need for specialized testing approaches has also grown. The "white box" and "black box testing" methods are applicable across all environment GUI presents integrity challenges for software engineers. Because of reusable components provided, the creation of the user interface has become less time consuming and more process. But at the same time the complexity of GUI's has grown, leading to more difficulty in the design and execution of test cases. Finite step modeling graphs may be used to drive a series of test cases that address specific data and program objects that are relevant to the GUI. Due to the large no. of permutations associated with GUI operations testing should be approached using automated tools.

Verification and validation:-

Verification refers to the set of activities that insure that software correctly implements a specific function. Validation refers to a different set of activities that insure that the software that has been built is traceable to customer requirements.

In other words validation provides final assurance that software meets all functional, behavioral and performance requirements.

"Boehm" states another way:-

- Verification: - "Are we building the product right".
- Validation: - "Are we building the right product".

Unit testing:-

Unit testing focus verification afford on the smallest unit of the software design-the software component/module. Using component level design description as a guide, important control paths are tested o uncover within the boundary of the module. The unite test is white box printed. The test that occurs as parts of unit tests is illustrated schematically

Integration testing:-

Integration testing is a systemic technique for contracting the program structure while at the same time conducting tests to uncover error associated with interfacing.

Validation testing:-

Software validation is achieved through a series of "black box" tests that demonstrates conformity with requirements. Each validation test case has been conducted, one of two possible conditions exists:-

- The functions or performance characteristics conform to specification and are accepted.
- A deviation from specification is uncovered and a deficiency list is created. Deviation or error discovered at this stage in a project can recovery be corrected prior to schedule delivery.

System testing:-

System testing is actually a series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work of verify that system elements have been properly integrated and perform allocated functions.

➤ *Recovery Testing:-*

Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed. If recovery is automatic (performed by the

system itself) reutilization, checkpointing mechanism, data recovery and restart are evaluated for correctness. If repair (MTTR) is evaluated to determine whether it is within acceptance limits.

➤ *Security Testing:-*

Security testing attempts to verify that protection mechanism built into a system will protect it from improper penetration. During security testing the tester plays the role of the individual who desires to penetrate the system. The role of the system designer is to make penetration cost more than the value of the information that will be obtained.

CONCLUSION

Pharmacy management system is actually a software which handle the essential data and save the data and actually about the database of a pharmacy and its management. This software helps in effectively management of the pharmaceutical store or shop. It provides the statistics about medicine or drugs which are in stocks which data can also be updated and edited. It works as per the requirement of the user and have options accordingly. It allow user to enter manufacturing as well as the expiry date of medicine placing in stock and for sales transaction. This software also has ability to print reports and receipts etc.

The presented system manages and controls the medicine issuing cycle of a pharmacy inside a hospital. The introduced system includes database built using SQL server and GUI design utilizing Visual Studio . The purpose of using the database was to present different benefits, such as reducing data redundancy, reducing updating errors, increasing consistency, great data integrity, and improving data access to users through the use of a host and query languages and improved data security. At the other hand, the designed GUI frames facilitated the working on the system by allowing the user to interact with the system through graphical icons. In other meaning, the system can be managed with unskilled users. The introduced system has been tested by applying many case studies, such as registering a patient (inpatient/ outpatient), issuing medicines to this patient, and requesting medicines from a warehouse.

BIBLIOGRAPHY

Books –

Let us java – Yashvant Kanetkar

Java the Complete Reference, Seventh Ed – Herbert Schild

Online References –

www.codeacademy.com/learn/learn-java

<http://stackoverflow.com>

www.youtube.com/