Medical Management System

Date of Submittion:

Submitted by:-

Reg. No. \ Roll No.

Nitin Kumar

Kamesh Shekhar

Keshav Kumar

Omraj Kumar

Under the guidance of :-

Mr. Sanjeet Kumar,
Faculty of Dept. of Computer Application,
Nalanda College, Biharsharif,
Magadh University

Project Report on,

Medical Management System

Submitted in partial fulfillment of requirements for the award of the degree of

BACHELOR IN COMPUTER APPLICATION

by- Nitin Kumar, Kamesh Shekhar,
Keshav Kumar, Omraj Kumar

Under the Guidance of:

Mr. Sanjeet Kumar,
Faculty of Dept. of Computer Application,
Nalanda College, Biharsharif,
Magadh University

Acknowledgement

Working on this project, "Medical Management System" with my team was a source of immense knowledge to me. We would like to express our sincere gratitude to my teacher Mr. Sanjeet Kumar sir for his guidance and valuable support throughout the course of this project work.

We acknowledge with a deep sense of gratitude, the encouragement and inspiration received by our faculty members and colleagues. We would also like to thank our parents for their support.

Nitin Kumar,
Kamesh Shekhar
Keshav Kumar
Omraj Kumar

Table of Contents

1. Introduction

This is a software that aims at easy management of a medical stores. In market, a lot of work is done manually. In this era of computer revolution, where everything is digital, the data management and operation also need to be computerized. The existing manual data base management system is more problematic and tedious. The calculation of data is not always accurate. The computerized system would reduce the complexity in managing the records and making them secure and easy to access, and get any type of information like product detail, patient, etc in easy manner by a quick search.

This system keeps the records of medicines such as medicine name, MRP, MFG, expiry date, quantity in stock, etc. The software also generates patient's or customer's report as print out for later reference. After booking the patient order, our stores provide the medicines by hand or home delivery together with the proper bill generated by this software.

This software focus on high security so that no unauthorized user can make random transaction. Only the person provided with proper user ID and password can do the billing or see the reports. And the for signing up as a new user it requires an Authentication number is required which is only owned by owner of the store . So that no random person can signup to the account.

The software provides two levels of security. As "only billing rights" and "managing rights" so that a person or staff or helper who is appointed for only billing purpose can not manipulate the data stored in the database. But the administrator who has the managing right can hava complete acces to the data.

2. Objectives

The main objective of this software is to provide an easy way to submit information about medicines and some other information as well.

It also facilitates to delete or modify or search all information very easily. The system is also helpful for generating reports and status. The medical shop management system is software which process computerized the information and manages all the record of medicines which is under medical shop management.

The aim of this project is to make possible easy all the manual errors and problems by automating them, which the users are facing at different level of management in medical store as well as provide high level of security of the data. So in this project, we try to do best of our effort and hard working to cover some parts of the system process.

- **Fast Query Process**
- **Entry of new Patient**
- **Maintenance all of records**
- **Issuing reports**
- **Highly secure and portable**

Fast Query Process:-

Computerization of query process is more economic than that of manual processing because query is updated and stored instantly without any calculation mistakes because here we have to just update the product master table or customer master table but in manual process, we have to change all the records.

Entry new Records:-

Maintaining the records of new entry becomes very easy because everyday there are new medicines coming in the market which has a huge demand in the marked. So, its important to stay updated for patients' health sake.

Maintenance all of records:-

Medical shop management helps in updating all the records time to time.

Issuing reports:-

Issuing reports issue from on the completion of the updating of all table or records. Here reports issue status is maintained by which we know to how much patient comes for medicines, we generate reports of stock status after updating the stock records etc.

Highly secure:-

This software is highly secure because in the absence of authorized persons nobody can provide medicines because they have a unique id or password to editing there records which is protected also.

3.Visibility

Project visibility is simply refers to a means through which senior management can have a perspective on the Projects that are being executed in the organization.

Project visibility is knowing how all your projects are performing and the progress that is being made. It's about knowing, when needed, what resources are consumed, any risk or issues impacting the project and the costs incurred on the project to date and planned.

There are a number of factors that can contribute to the lack of visibility on a project, but I'm going to call out three of the over-arching causes as I see them:

1 A Lack of Organizational Project Management Processes

The first obstacle to project visibility is a lack of organizational project management processes. When you have a situation where every project manager is using a different method and tools, it's virtually impossible to roll-up project information in any meaningful way.

2 No Central Repository for Project Information and Documentation

Secondly, when there is no central repository for project information and documentation, you get into the problems like: "Which document is the real one?" or "Where is that document again?" You know what I mean – without the one central document library, you'll find every team member has a different version of a document or they've saved it to some obscure location that is only accessible by them.

3 Poor Communication

And lastly, poor communication is a major problem for every level of the project team. For example, there should always be open lines of communication between the project manager and project sponsor, so the project manager can develop rapport and trust with the sponsor. Without it, the sponsor will not know when they need to step in and handle an issue. Likewise, individual team members' responsibilities must be communicated accurately so nothing gets lost. As I've heard my boss say before: "If it's anyone's job, then no one will do it." And when there is poor communication, that's exactly what will happen.

So from the definitions given earlier in this post, I hope that you can understand the importance of visibility in your organization and the repercussions that might occur if the metaphorical water surrounding your project(s) is muddy as opposed to crystal clear. If this is the case, it pretty much goes without saying that your project is going to experience some difficulties. Here are the top three you're most likely to encounter:

1 Poor Communication Between Team Members and Project Manager

The first repercussion from a project managers point of view is that there will be poor communication between members of the project team. This results in project managers finding it difficult to get in touch with their team members and vice – versa which in turn elongates the process and, as I'm sure you can imagine, annoys many a project manager when they're left chasing information from those who are reporting to them!

2 A Build up of Project Issues for the Senior Executive

The second repercussion in this trio is the late action by senior executives when issues arise, and so often they arise in their multiples! If project visibility is lacking, issues are bound to build up before they get seen to. You can imagine how much of a shock the senior executive gets when they receive an update on a project that suffers from poor visibility only to see half a dozen issues that are in need of urgent attention. Disaster!

3 Confusion Among Project Team Members

A third repercussion that could potentially arise is the difficulties that team members would face when they're not fully informed about the tasks that are expected of them. If a member of the project team doesn’t understand their role fully and the tasks within it, there’s bound to be confusion and a negative knock-on effect into the rest of the project.

It’s all well and good knowing what project visibility is, why you might be lacking it and the inevitable repercussions of lacking it as well of course, but how do you improve it? Well, here are 6 simple ways:

1 Kick-Off Meeting

One often overlooked way to improve project visibility is to have a kick-off meeting with everyone who’s involved in the project. This way everyone starts off on the same page and can also ask any questions they might have about the project before they begin.

2 On-Boarding Documents

Another way to improve project visibility would be to have on-boarding documents for anyone who joins the project late. These documents should explain the details of the project and the different project processes in detail..leave no stone un-turned! This is extremely important for long-term projects that often have high turnover of project workers.

3 Get Project Updates from EVERYONE!

Thirdly, when you’re holding your regular team / company meeting in your organization, ensure that every single person gives an update on the projects that they are working on…be they good or bad. Too often people shy away if their project isn’t going well and this can lead to unresolved issues.

4 Have a Centralized Collaboration Point

Having a centralized collaboration point to hold important documents and other project artifacts (as mentioned above) can save you a LOT of time and effort on your projects. If you find your project team often having unnecessary meetings or if there tends to be a lot of confusion as to who’s doing what, then this is definitely one you should focus on.

5 Hold Weekly Team Meetings

Weekly team meetings to give updates on work – constant communication. My fifth way to improve project visibility is to have regular team meetings. You've probably gathered at this stage that communication is the most important thing when it comes to visibility in your projects, and having a regular team meeting allows for regular discussions whereby issues can be dealt with, tasks delegated or even successes celebrated!

6 Relevant and Continuous Training

And my final point in this list would be for every team member to be trained well (and continuously) in their area. By having a well-trained workforce who know the various project processes like the back of their hands, everyone will be kept informed and everything should run just like clockwork!

So that's it! I wish you the very best of luck with your future projects (and maximizing their visibility in your organization of course.)
And I hope you've found this blog post informative and that you'll be able to apply some of your learning's in real world settings.

4. Reason of selection

We are using JAVA language for making medical shop management because JAVA language is an object oriented INTERNET based programing language and it highly portable and is highly demanded in the marked for commercial application development . So first off all we should describing about JAVA language:-

About Java

In early 1990 there was a need for an Object Oriented language to program and control consumer electronic devices, which could not be done by C++ or any other language. So in 1990 Sun Microsystems decided to develop a language. Which performs above-mentioned functions and in 1991 language "OAK' came into existence. Later in 1995 it was renamed as JAVA which later developed into a full-fledged Object Oriented language both for Internet and stand-alone systems.

Introduction to Object Oriented Programming

Commonly used programming methodologies include:
1. Procedural programming
2. Object-oriented programming

Procedural Programming

The procedural programming methodology involves dividing a large program into a set of sub-procedures or subprograms that performs specific task.

Object-Oriented Programming

A large application consists of component objects, which interact with each other. This is referred to as the object-oriented approach to develop an application.

Advantages of Object-Oriented Programming

- Real-world programming
- Reusability of code
- Modularity of code
- Resilience to change

Object and Classes

An object is a software package consisting of variables and methods. An object is defined as an instance of a class. Classes are designed such that they can be reused.

Features of Object-Oriented Programming

Encapsulation

Encapsulation is the process of hiding all the details of an object that do not contribute to its essential characteristics. Encapsulation is also known called information hiding.

Abstraction

An abstraction denotes the essential characteristics of an object that distinguishes it from all other kind of objects and thus provides crispy defined conceptual boundaries, relative to the perspective of the viewer.

Inheritance

Inheritance enables you to extend the functionality of an existing class. You create a class that inherits the attributes and behavior of another class. In addition, the new class can consist of a few new attributes and behaviors that are specific to the class.

Types of inheritance supported in java:

- **Single inheritance**
- **Multiple inheritance**

JAVA Features

The inventors of Java wanted to design a language that could offer solutions to some of the problems encountered in modern programming. They wanted language to be not only reliable, portable and distributed but also simple, compact and interactive. The following features have made Java the first application language of the World Wide Web.

Compiled and Interpreted

Java language combines both compiling and interpreting approaches first; Java compiler translates source code into byte-code instruction. In the second stage, Java interpreter generates a code that can be directly executed by the machine that is running the Java program.

Platform Independent and Portable

Java programs can be easily moved from one computer system to another anywhere and anytime. Changes and upgrades in the operating systems and system resources will not force any changes in Java programs.

Multithreaded and Interactive

Java supports multithreaded programs (handling multiple tasks simultaneously). The Java runtime comes with tools that support multi process synchronization and construct smoothly running interactive systems.

Dynamic and Extensible

Java is a dynamic language. Java is capable of dynamically linking in new class libraries, methods and objects. Java programs support functions written in other languages such as C and C++.

JAVA Environment

Java environment includes a large number of development tools and hundreds of classes and methods. The development tools are part of the system known as Java Development Kit (JDK) and classes and part of the system known as Java Standard Library (JSL) also known as API.

JAVA Development Kit

The JDK comes with a collection of tools that are used for developing and running Java programs. They include:

Applet-viewer

Enable us to run Java applets.

Java

Java interpreter, which runs applets and applications by reading files and interpreting byte-codes files.

Javac

The Java compiler, which translates Java source code to byte-code files that the interpreter understands.

Javadoc

It Creates HTML-format documentation from Java source code files.

Javah

It Produces header files for use with native methods.

Javap

Java disassembler, which enables us to convert byte-code files into a program description.

Jdb

It is Java debugger, which helps us to find errors in our programs.

Java Architecture

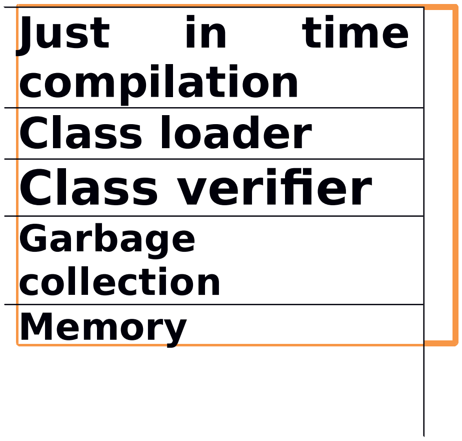
Various components of the java architecture are:

Java *Programming Language and class File*

Java programs are saved with an extension .java. A .java file is compiled to generate the .class file, which contains the byte codes. The JVM converts the byte code contained in the .class file to the machine object code. The JVM needs to be implemented for each platform running on a different operating system.

Java Virtual Machine (JVM)

The JVM forms the base for the java platform and is convenient to use on various hardware-based platform.



Just in time compilation

Class loader

Class verifier

Garbage collection

Memory

Components of the JVM

JVM for different platforms uses different techniques to execute the byte code. The major components of JVM are:

Class loader: The class loader loads the class files, which are required by a program running in the memory. The classes are loaded dynamically when required by the running programs. A

JVM can have following types of class loaders:

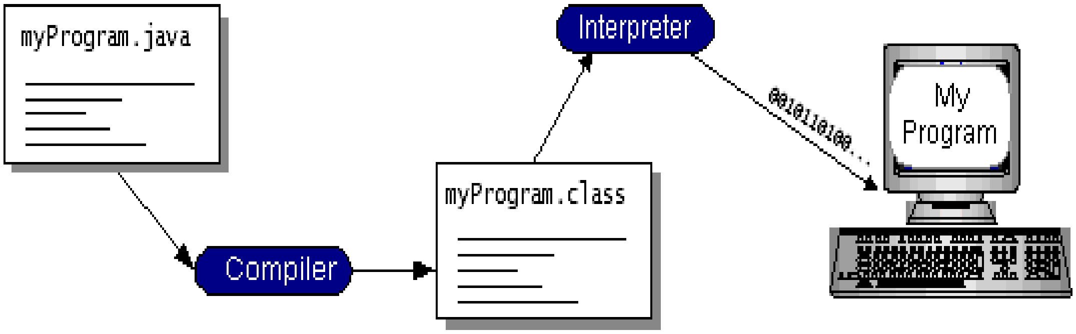
- ***Primordial Class Loader:*** Loads the java API classes required by the running java program.

- ***<u>Class Loader Objects:</u>*** Loads the classes of the java application program. An application can create class loaders at runtime for loading the classes of the application.

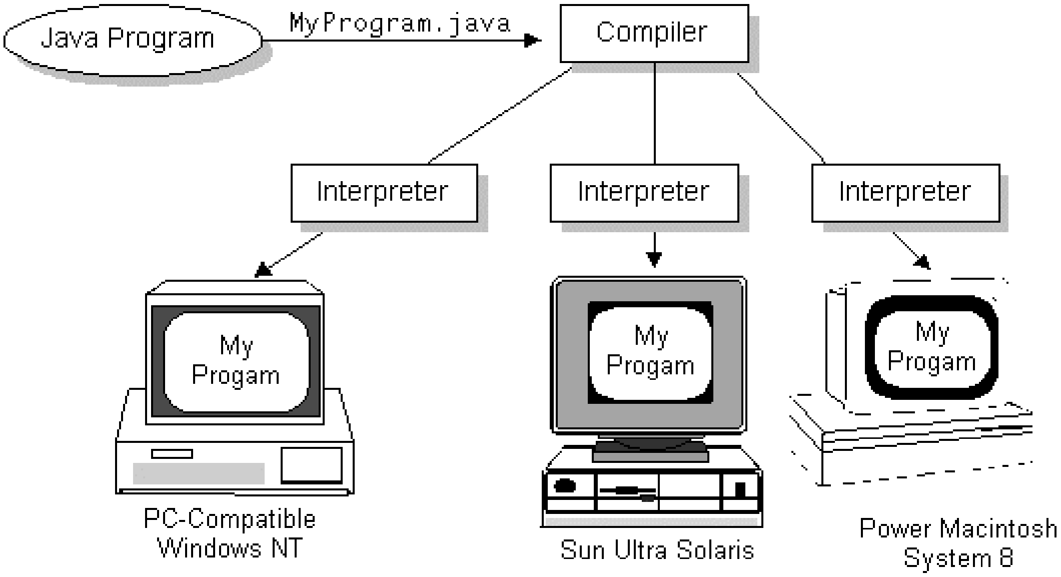
Execution Engine: The java execution engine is the component of the JVM that runs the byte code one line after another. The execution engines are implemented by different vendors to use different techniques to run the byte code.

Just in Time (JIT) compiler: The JIT compiler is used for compiling the byte code into executable code. The JVM runs the JIT compiled code without interpreting because the JIT-compiled code is in the machine code format. Running the JIT-compiled code is faster than running the interpreted code because it is compiled and does not require to be run, line after line.

The java programming language is unusual than other programming languages it first compiles and then interprets the program. Compile first translate the program into intermediate language called intermediate language called java byte code. Java byte code is platform independent code, which is further interpreted by the interpreter on the java platform. Interpreter parses and run each java byte code instruction on the computer. Compilation occurs only once, interpretation occurs each time when the program is executed.



Java byte code helps in making the program "write once, run anywhere". The program can be compiled into byte code by any platform that is having the java compiler; the compiled java byte code program is ready to run on any machine having the java virtual machine (JVM). JVM is an interpreter for byte code.



The Java Platform

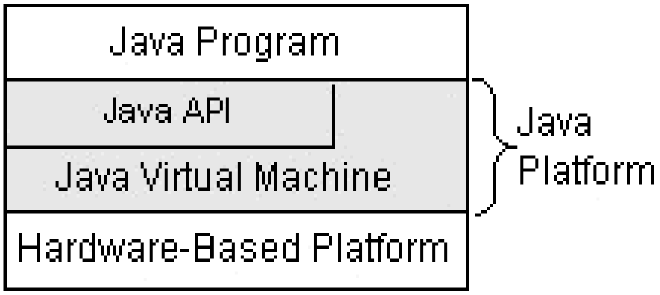
A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components:

- The *Java Virtual Machine* (Java VM)

- The *Java Application Programming Interface* (Java API)

We've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.
The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (*packages*) of related components.
The following figure depicts a Java program, such as an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.



As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring Java's performance close to that of native code without threatening portability.

Applets and Swings

An applet is a program that can be embedded in an HTML web page. An applet is compiled on one computer and can run on another computer through a java enabled web browser or an appletviewer.Applet is developed to support the GUI in java.

To create an applet, you need to follow these steps:-

1. Create a java program for the applet.
2. Compile the java program.
3. Create a web page that contains an applet.
4. Run the applet.

An Applet Life Cycle

An Applet inherits the properties and methods of an applet class.Java provides init(), start(), stop(), paint() and destroy() as the basic applet methods to control the execution of an applet. The different stages of an applet life cycle are

- **Initializing an applet.**
- **Starting the applet.**
- **Stopping the applet.**
- **Destroying the applet**

What Swing Packages Should I Use?

The Swing API is powerful, flexible -- and immense. For example, the 1.1 version of the API has 15 public packages: javax.accessibility, javax.swing, javax.swing.border, javax.swing.colorchooser, javax.swing.event, javax.swing.filechooser, javax.swing.plaf, javax.swing.plaf.basic, javax.swing.plaf.metal, javax.swing.plaf.multi, javax.swing.table, javax.swing.text, javax.swing.text.html, javax.swing.tree, and javax.swing.undo.

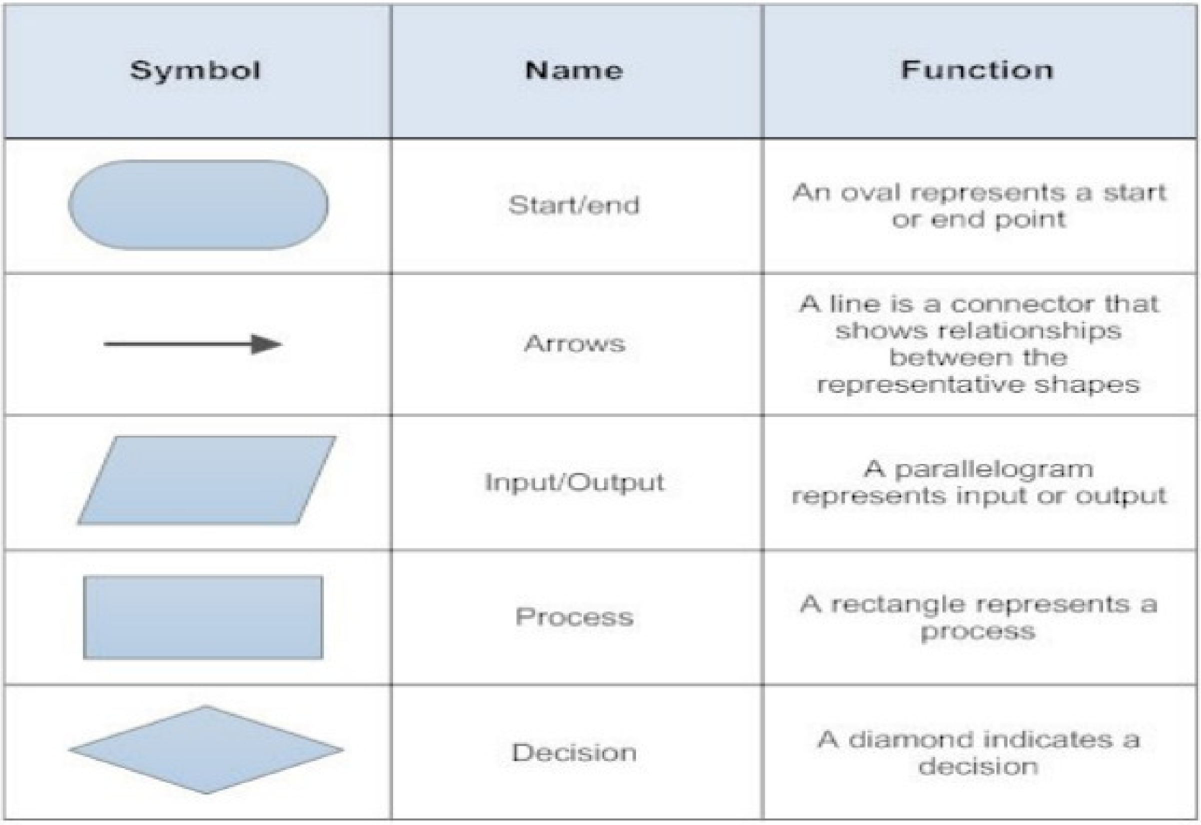
Fortunately, most programs use only a small subset of the API. This trail sorts out the API for you, giving you examples of common code and pointing you to methods and classes you're likely to need. Most of the code in this trail uses only one or two Swing packages:

- javax.swing
- javax.swing.event (not always required)

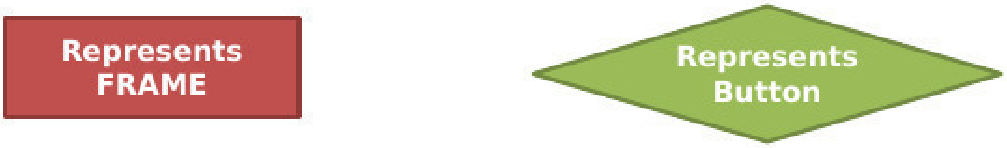
5. Project category

This project is basically minor

6. Flow Chart

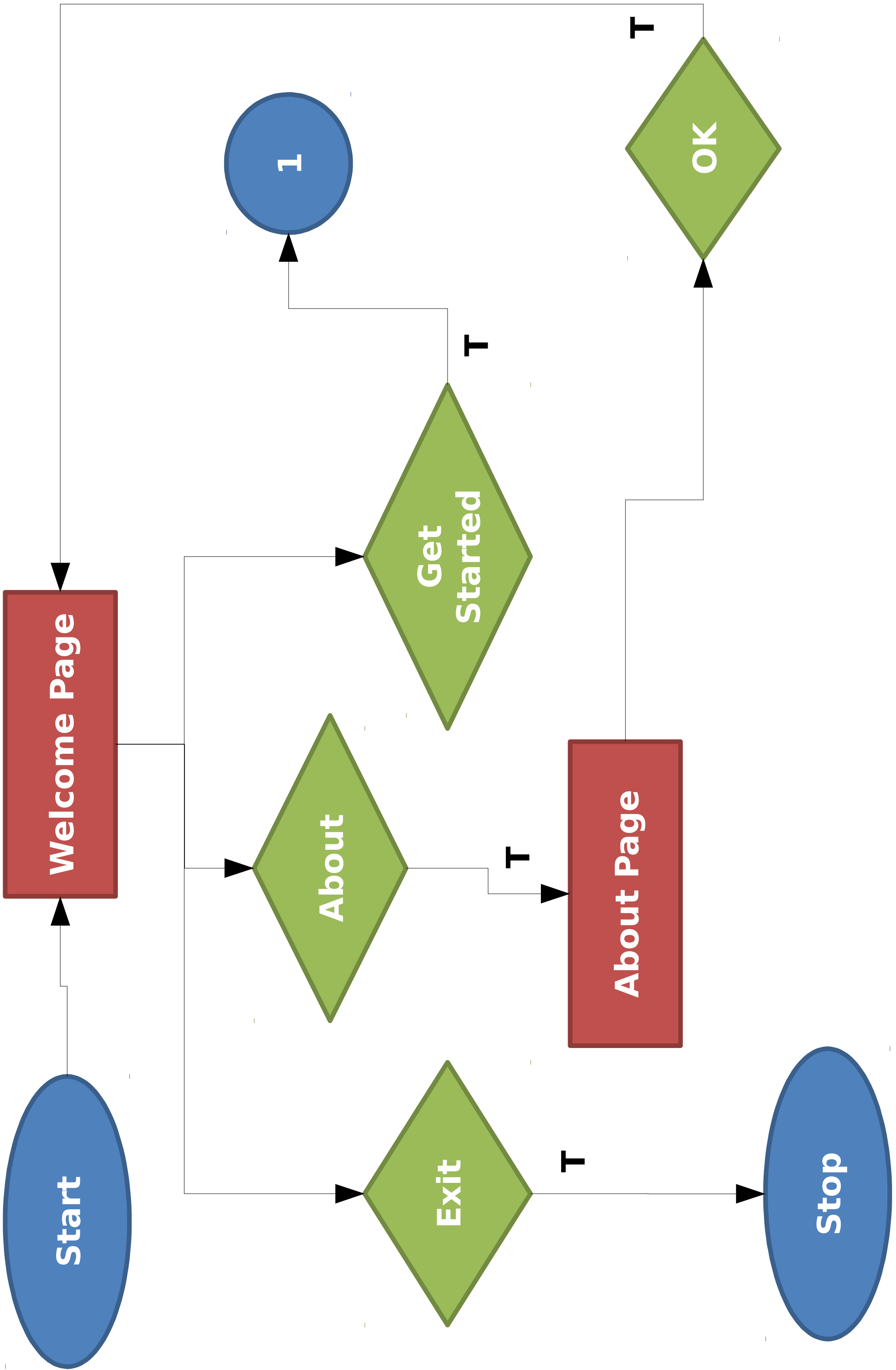


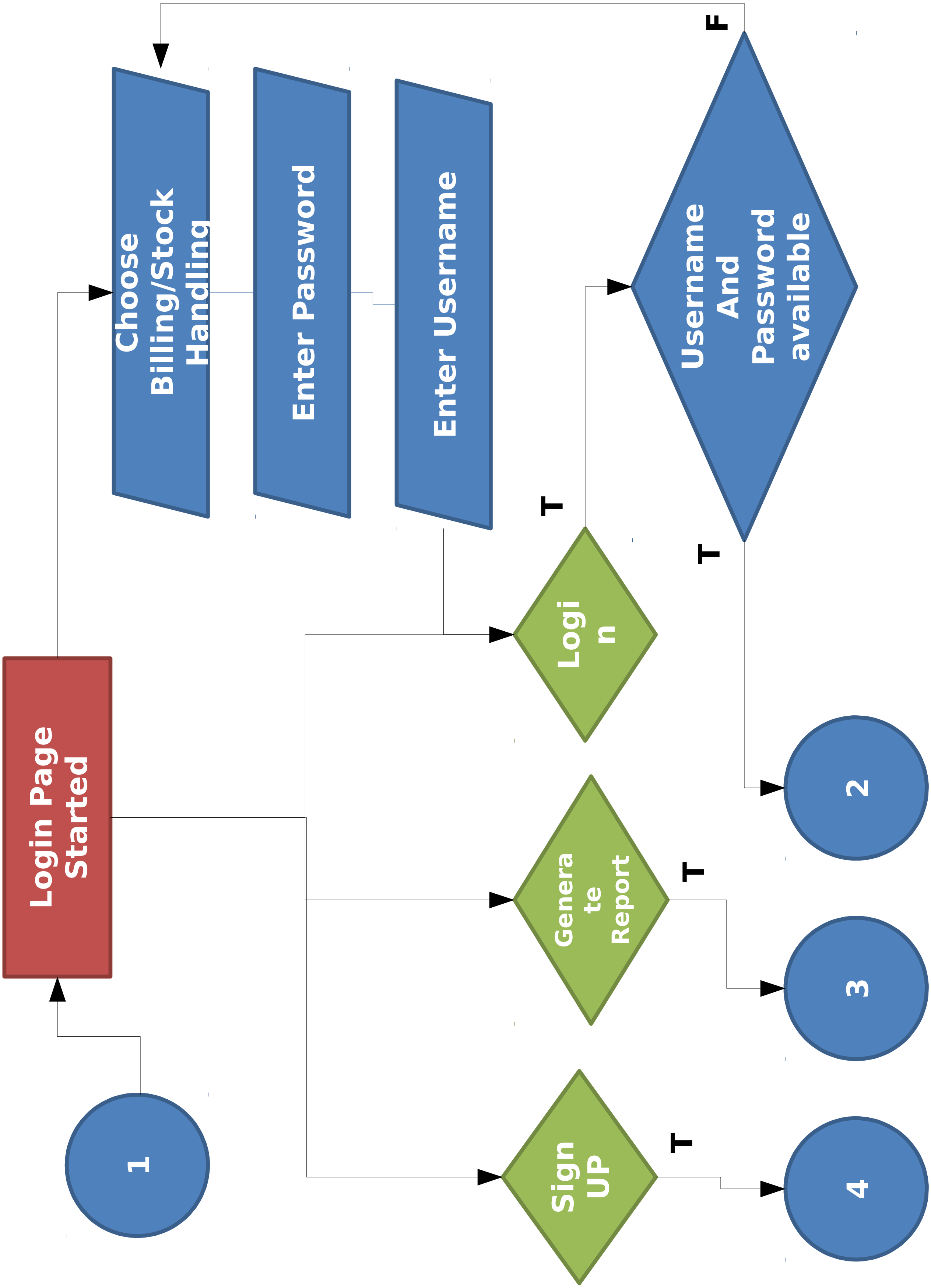
Symbol	Name	Function
⬭	Start/end	An oval represents a start or end point
→	Arrows	A line is a connector that shows relationships between the representative shapes
▱	Input/Output	A parallelogram represents input or output
▭	Process	A rectangle represents a process
◇	Decision	A diamond indicates a decision

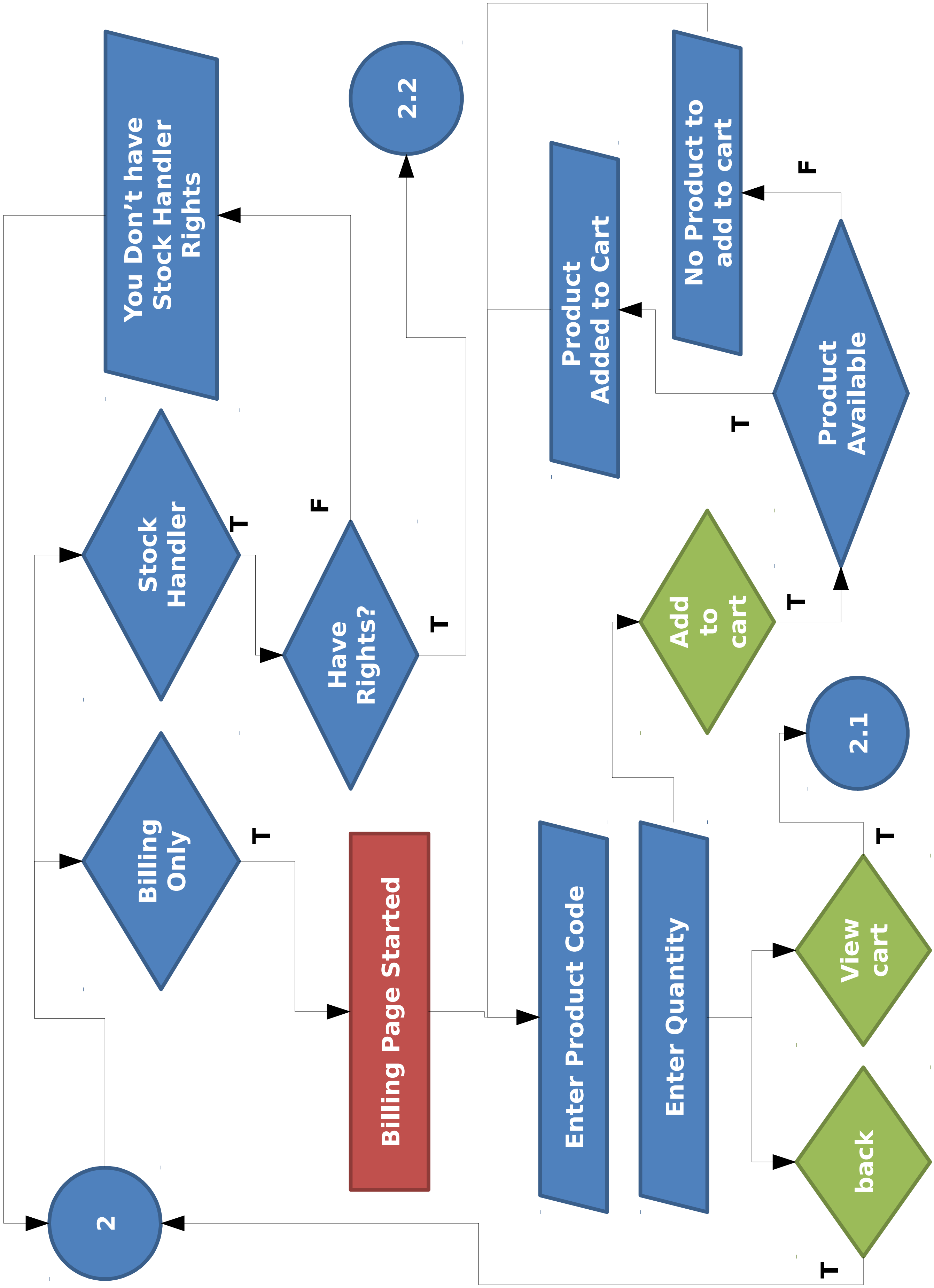


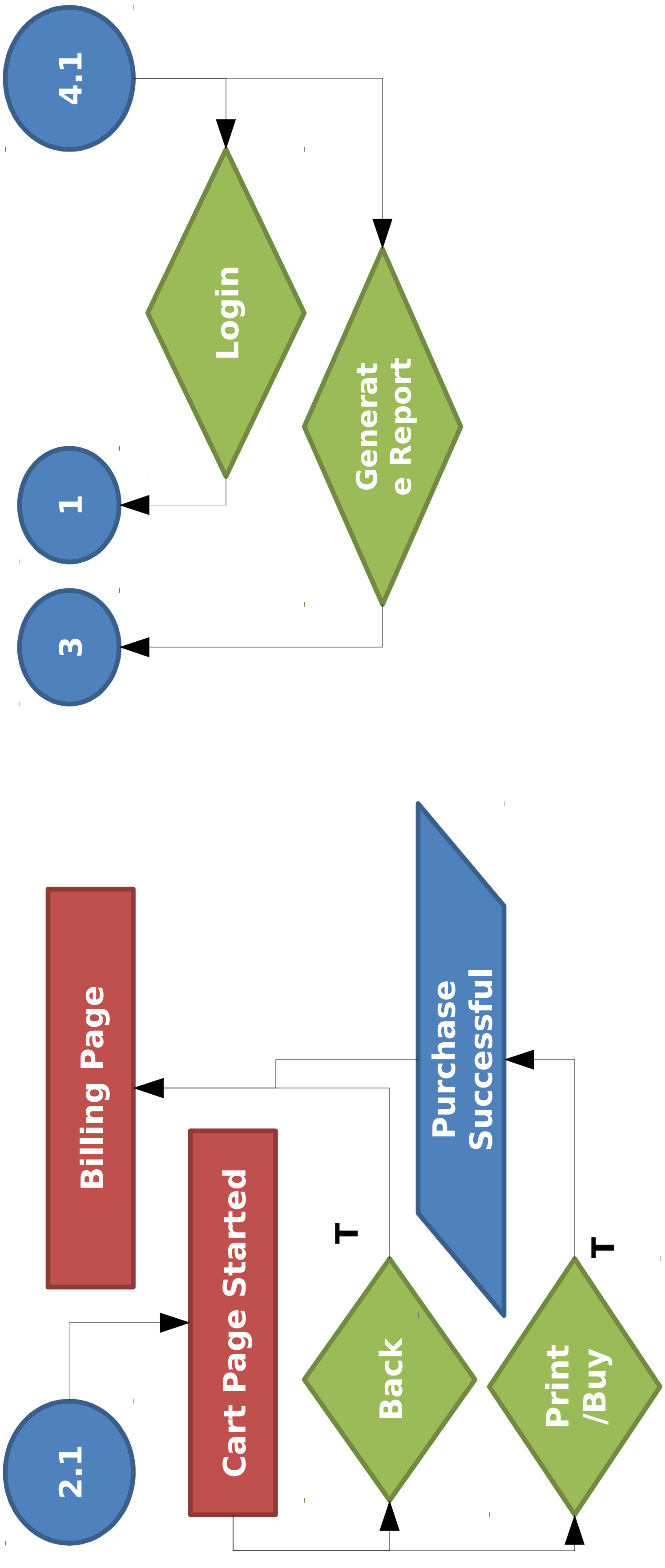
Represents FRAME

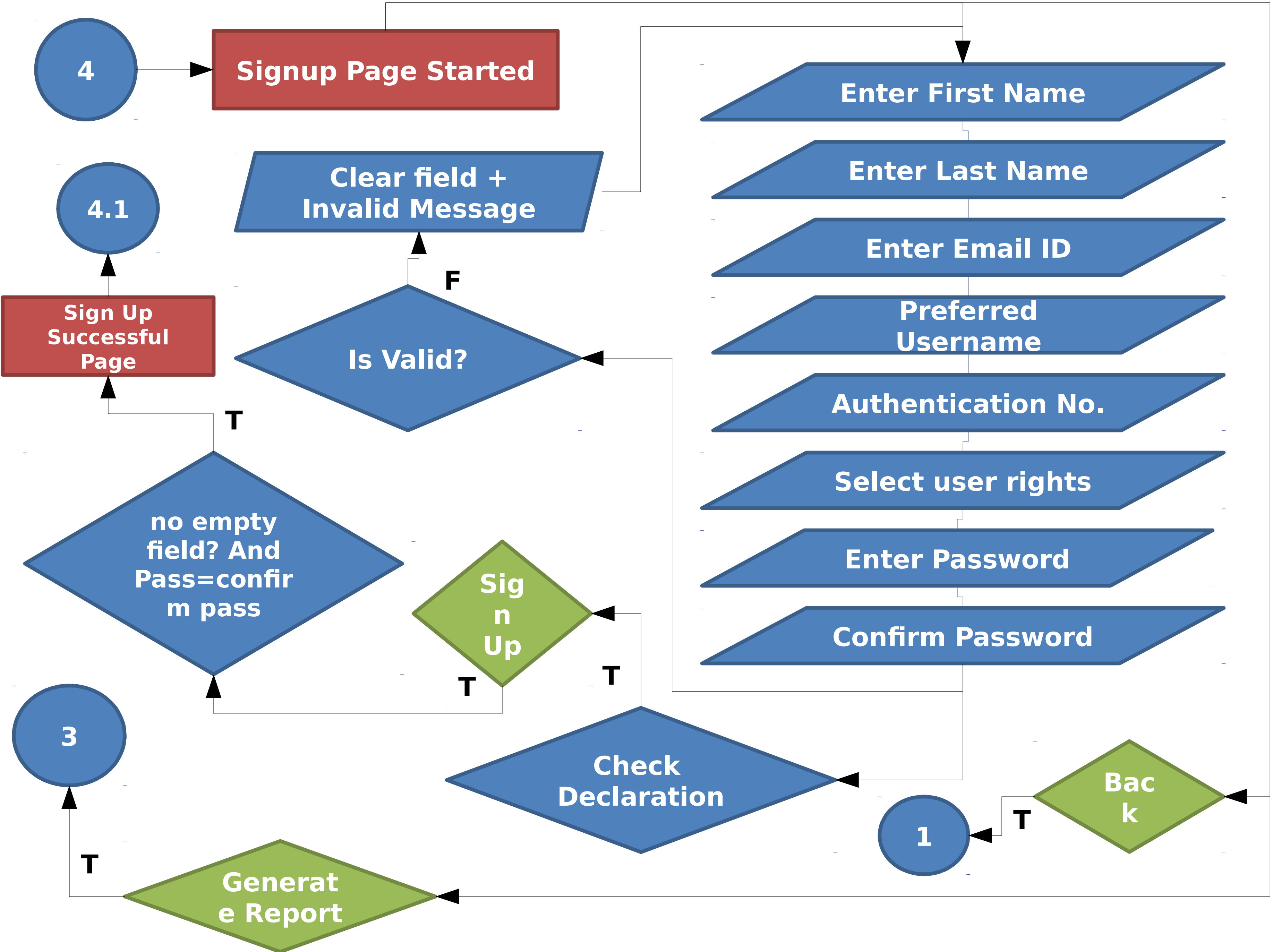
Represents Button

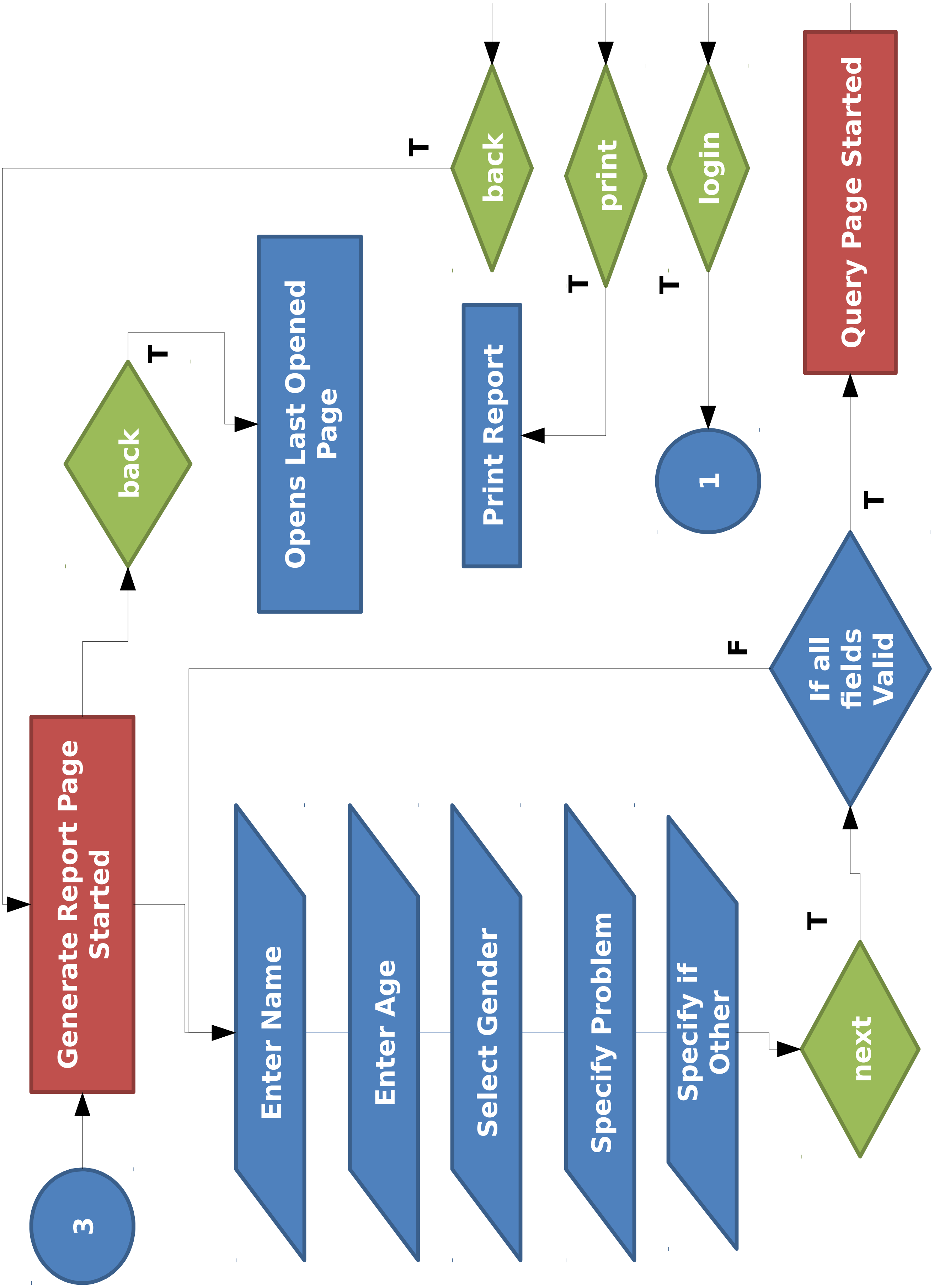

Start
Welcome Page
Get Started
About
Exit
T
1
About Page
T
OK
T
T
Stop

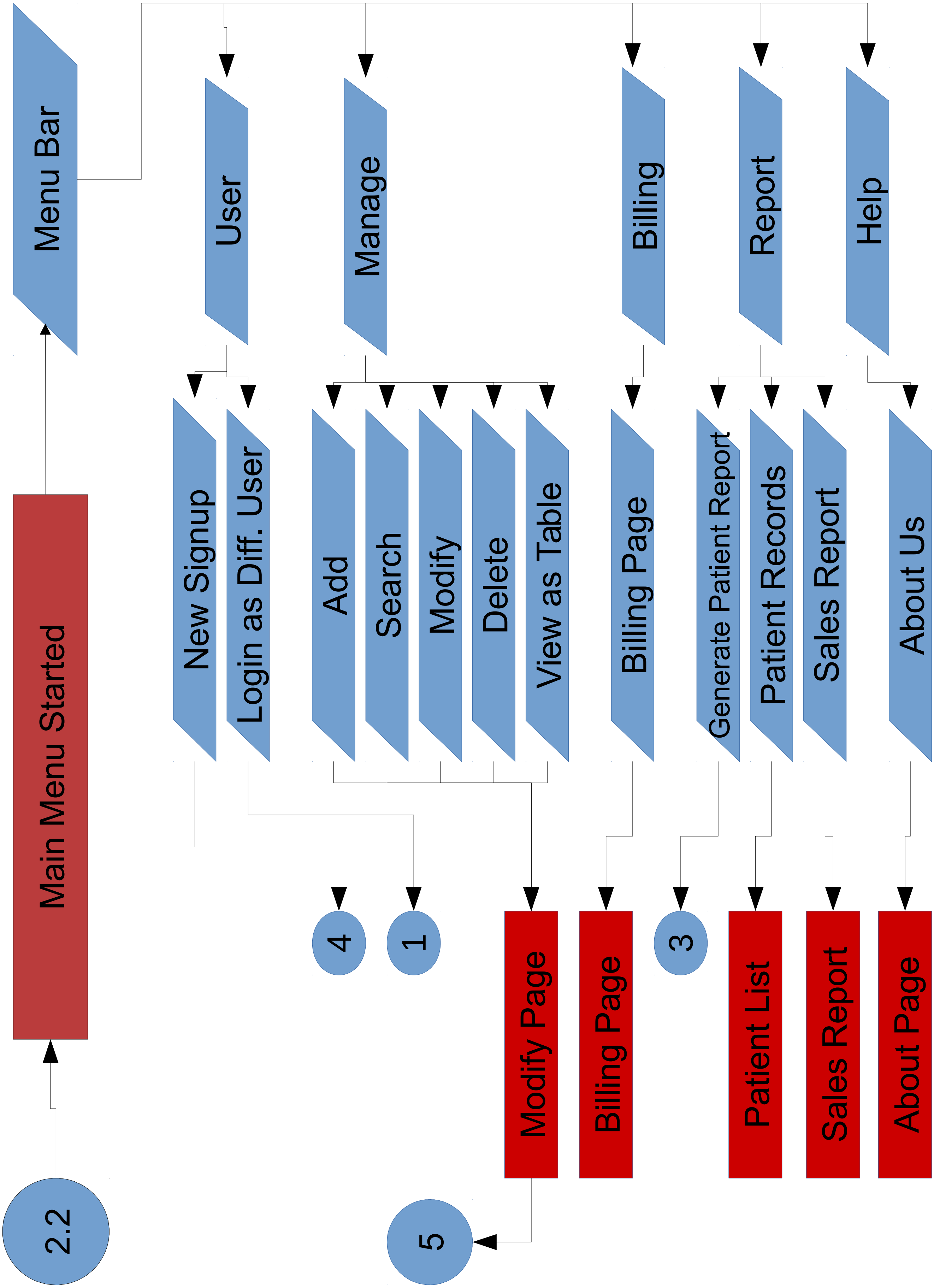

1
Login Page Started
Choose Billing/Stock Handling
Enter Password
Enter Username
Login
Generate Report
Sign UP
Username And Password available
F
T
T
T
T
2
3
4

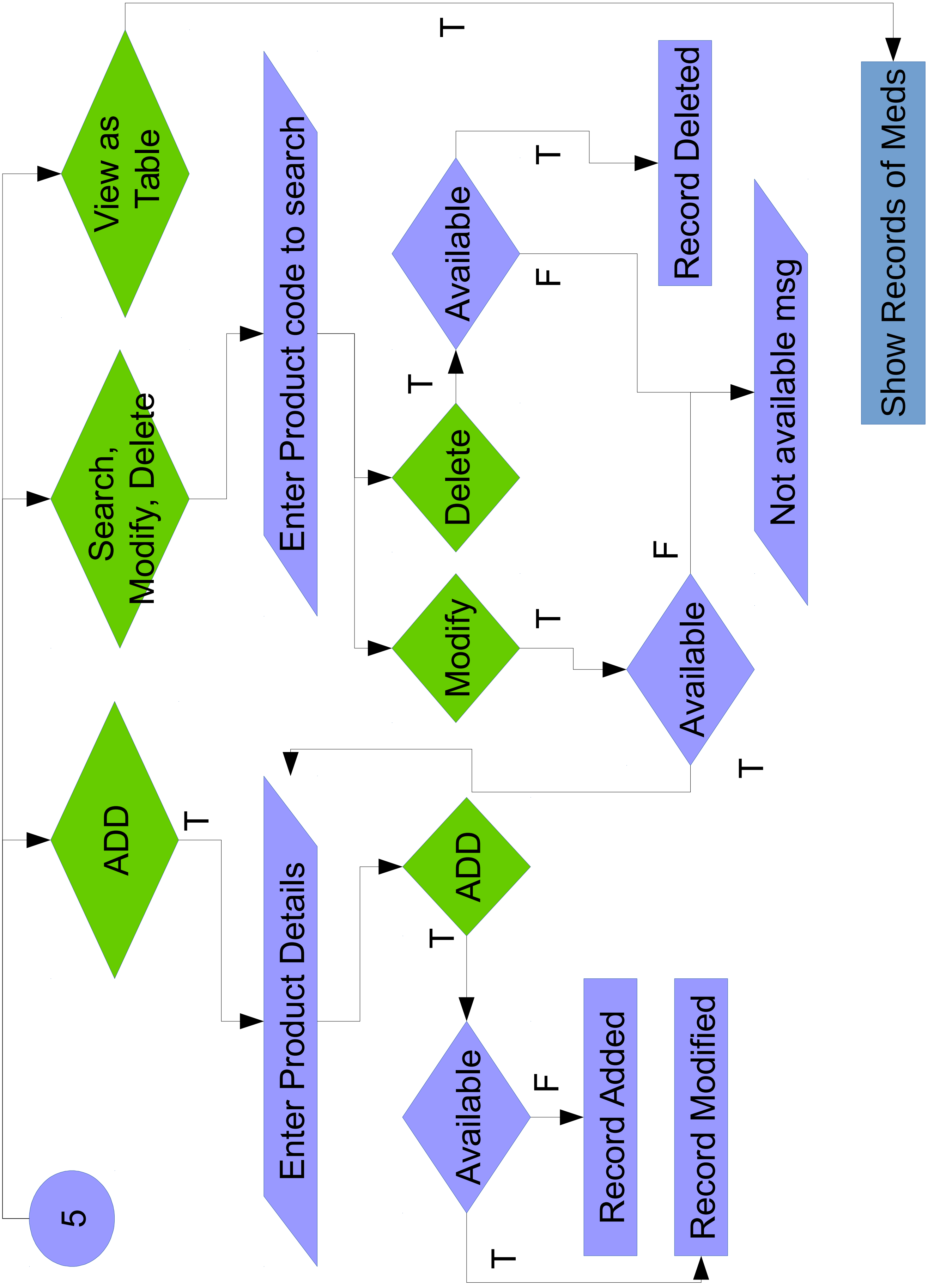

2
Billing Only
T
Stock Handler
T
Have Rights?
F
T
You Don't have Stock Handler Rights
2.2
Billing Page Started
Enter Product Code
Enter Quantity
Add to cart
T
Product Available
T
F
Product Added to Cart
No Product to add to cart
View cart
T
2.1
back
T


4.1
Login
Generat
e Report
1
3
2.1
Cart Page Started
Billing Page
Back
T
Print
/Buy
T
Purchase
Successful


4
Signup Page Started
Enter First Name
Enter Last Name
Enter Email ID
Preferred Username
Authentication No.
Select user rights
Enter Password
Confirm Password
4.1
Clear field + Invalid Message
F
Sign Up Successful Page
Is Valid?
T
no empty field? And Pass=confir m pass
Sig n Up
T
T
3
Check Declaration
1
T
Bac k
T
Generat e Report

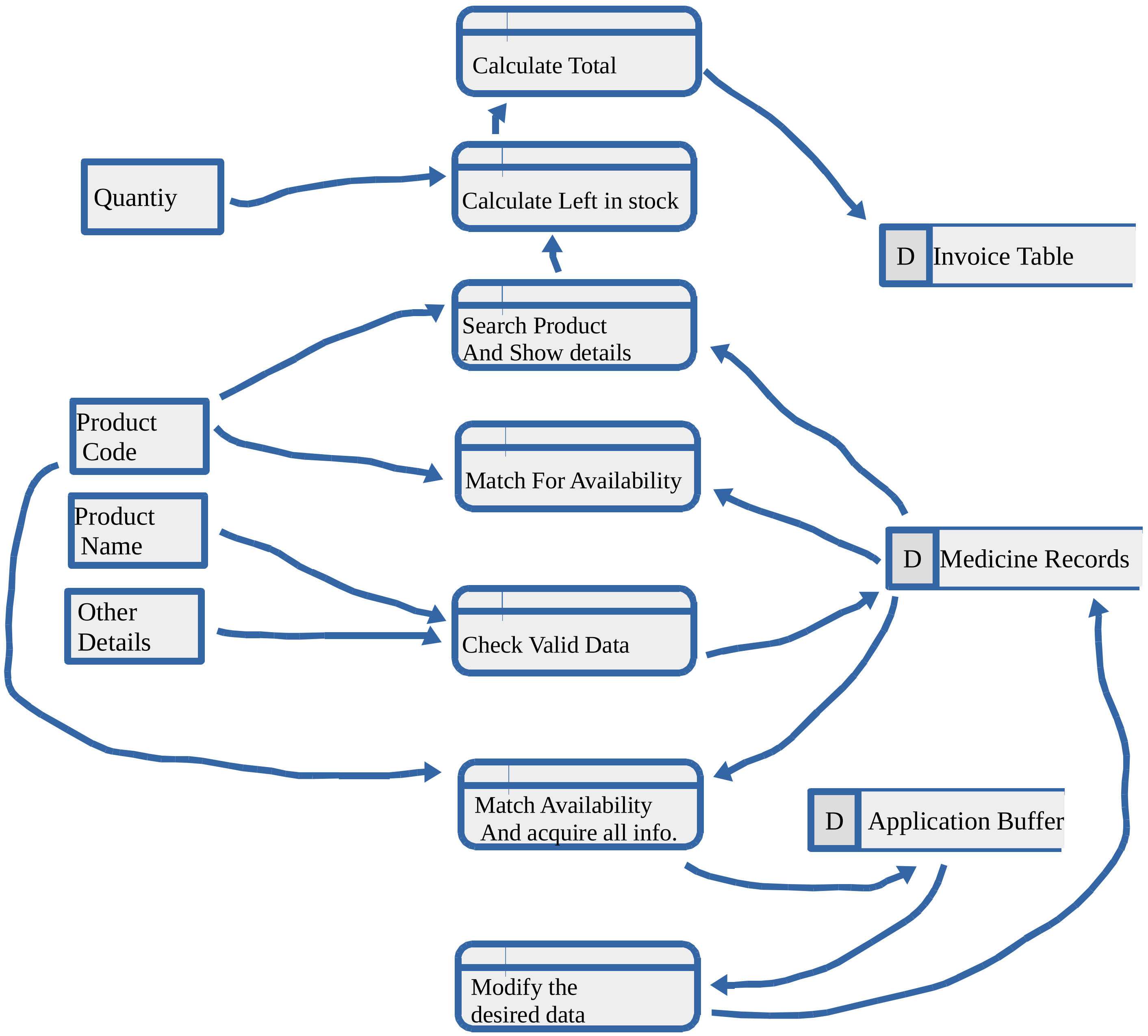

Query Page Started
login
print
back
T
T
T
Print Report
1
Opens Last Opened Page
back
T
If all fields Valid
T
F
Generate Report Page Started
3
Enter Name
Enter Age
Select Gender
Specify Problem
Specify if Other
next
T


2.2
Main Menu Started
Menu Bar
User
Manage
Billing
Report
Help
New Signup
Login as Diff. User
Add
Search
Modify
Delete
View as Table
Billing Page
Generate Patient Report
Patient Records
Sales Report
About Us
4
1
Modify Page
Billing Page
3
Patient List
Sales Report
About Page
5

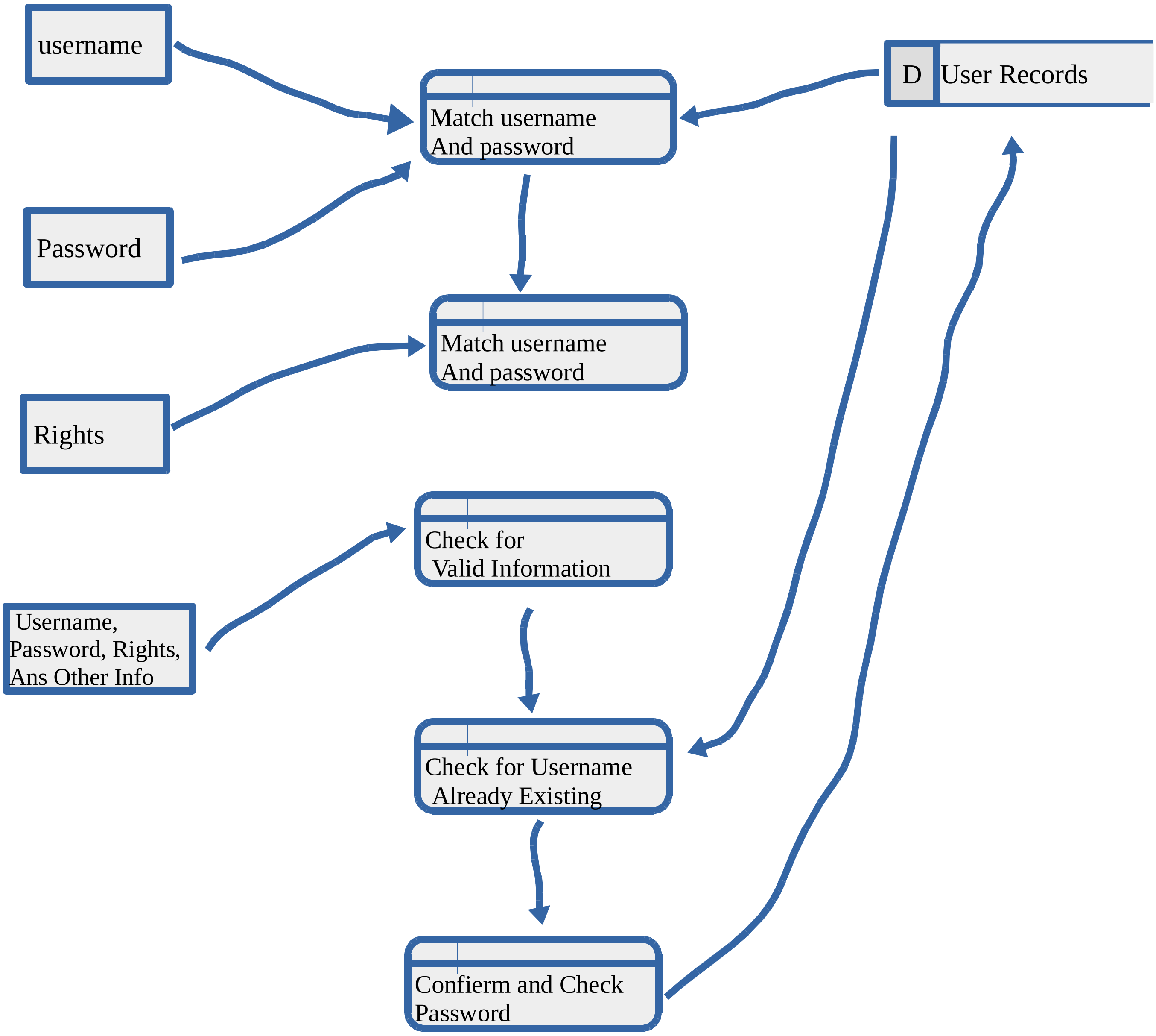

5
ADD
T
Search, Modify, Delete
View as Table
T
Enter Product Details
Enter Product code to search
ADD
T
Modify
T
Delete
T
Available
F
T
Available
T
F
Available
F
T
Record Added
Record Modified
Record Deleted
Not available msg
Show Records of Meds

7. Data Flow Diagram

Stock Managing



User Data Managing



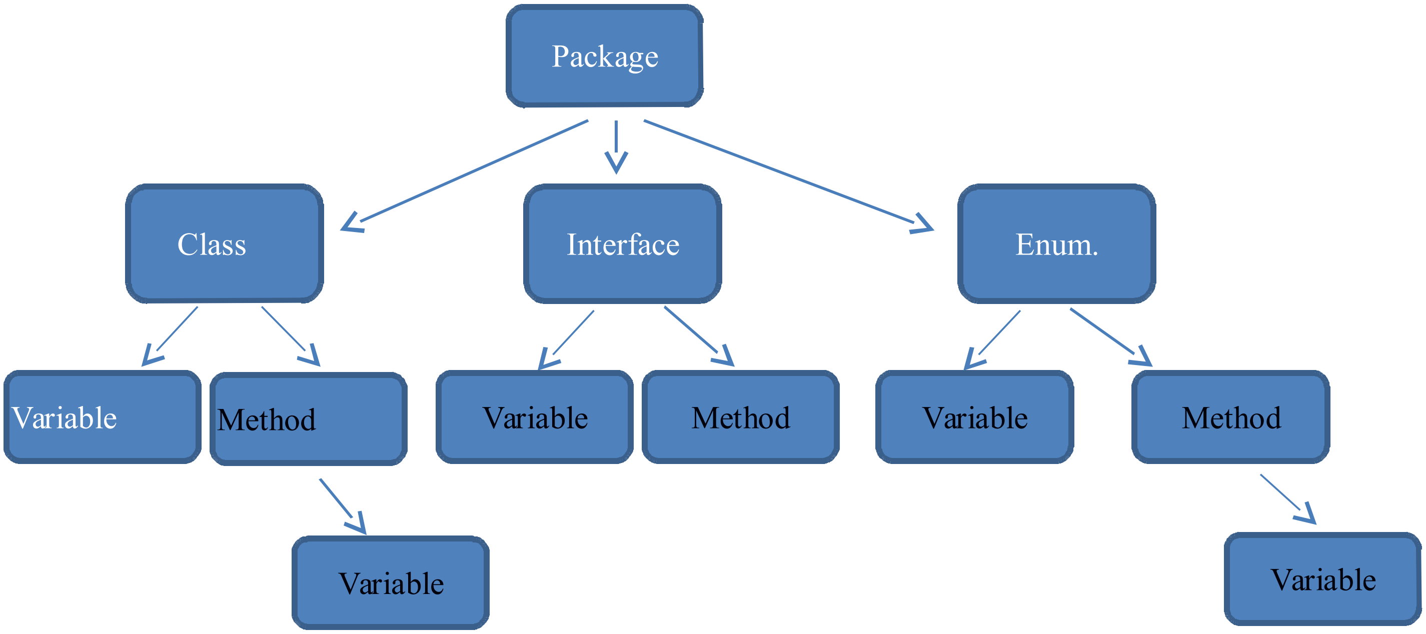
8. Package

In java language as we know that everything is represented in the form of a class, interface of enum.

The concept of class and interface are introduced from begging of java where concept of enum is introduced in java 5.0 and it is used in 5.0 or its higher versions.

Two groups various type of classes and enum java provides the concept of a package whose concept is similar to the header file in C or C++ language.

Package is a logical container or that that contains classes, interfaces, enum or files. The main purpose of introducing the package is to provide unique namespace to its members that means within application whenever name of two classes, interfaces or enums are same. By using the concept of package we can uniquely identify. Package is also use to implement the scope of access specifies in java language.



- we used the following packages to make our projects:-

  1. ***java.util***
  2. ***java.awt***
  3. ***java.awt.event***
  4. ***java.swing***
  5. ***java.swing.event***
  6. ***java.sql***

1. **java.util**
   This package provides class and interface to work with array and object, read a value from keyboard, work with current system date and time.
2. **Java.awt**
   This package provides class and interface to develop windows or GUI based application.
3. **Java.awt.event**
   This package is provides to developer control the event in windows based or GUI based application.
4. **Javax.swing**
   This package is the extension of java.awt that is now a days in development of windows based application or GUI based application.
5. **javax.swing.event**
   This package provides class and interface to develop and control event in windows based or GUI based application.
6. **Java.sql**
   This package provides the facility to connect java applications with a database so that all queries are executed in java.

- **Declaration of Package**

  It is the process of specifying name of a package in a program whose member such as class and interface has to use in a program.

  Syntax to declare a package:-

  a. Import package name.member of a package; (For a particular class or interface of a package).
  b. Import package name.*; (For all classes or interface of a package).

  Here, import is a keyword that is provided by java language to refer class and interface of one package into another package or in java language program.

  Class:-

  It is a keyword that is that is provided by java language to create user defined data store different between type's variables and methods.
  Creating a class means create template and format that denotes common attributes and behaviors of a class. That exist every object of a class.

  Class Name:-

  It denotes name of a class that may be any valid user defined name of java language. It is used further in a program in following situation:-

- To create reference variable of a class.
- To create an object of a class.
- To access static member of a class outside a class.

9. Function and method

We use the function and methods concept for making a successful and easy understandable medical shop management system.
Some system defined function which is used in this project.

- **show()**
  This method is used to display particular container from a deck.
  Syntax: - public void show ()
- **setBackground()**
  We use set Background method for set color of the text background at the run time.
  Syntax: - public void setBackground()
- **setForeground()**
  We use set Foreground method for set the color of the text at the run time.
  Syntax: - public void setForeground()
- **setFont()**
  We use the setFont method to change the styles of specified text.
  Syntax: - public void setFont (string s)
- **setEnabled()**
  We use this method to change the default state of command button in this project.
  Syntax:-public void setEnabled(Boolean b)
- **setToolTipText()**
  We use this method for displaying message for temporary time. In other words we can say, when the cursor moves on the any button that time display that button names in temporary time.
  Syntax:- public void setToolTipText(String string)
- **setsize()**
  Set the size of component is size of pixel by using set as method of container class.
  Syntax: - public void setSize (int width, int height)

- **set Visible()**
  Make a frame visible by using the setVisible() of a container class.
  Syntax: - public void setVisible (Boolean b)
- **setLocation()**
  Set the location of the specified button.
  Public void setLocation(int i, int i1)
- **setLocationRelativeTo()**
  This function accepts null value in this project.
  Public void setLocationRelativeTo(Component cmpnt)
- **setTitle()**
  Set the tittle message by using setTitle () of a frame class.
  Syntax: - public void setTitle (String s)
- **setResizeable()**
  This function is used to prevent end user from resizing the window size by passing value false.
  Syntax:- public void setResizeable(Boolean b)
- **getActionCommand()**
  It returns a message of a source on which action event is generated.
  Syntax: - public void getActionCommand()
- **setString()**
  This function is used to set value to insert record in sql table by passing index no and value.
  Syntax:- public void setString(int I, String string) throws sqlException

- **setLabel()**
  It is a method of Button class. Which is use to change message of a button.
  Syntax: - public void setLabel(string s)

- **getSource()**
  It returns a reference of source object on which action event is generated.
  Syntax: - public string getSource()

- **getLabel()**
  It is a method of Button class. Which is use to obtain message of a button.
  Syntax: - public string getLabel ()

- **getText()**
  It is method of label class which is returns a text associated with label.
  Syntax: - public string getText ()

- **setText()**
  It is also a method of label class. Which is use to change a text associated with a label at the run time.
  Syntax: - public void setText (String s)
- **setLayout()**
  This method is used to set layout value to null.
  Syntax:- public void setLayout(LayoutManager lmg)
- **toUpperCase()**
  We use this method for converting string lowercase to uppercase.
  Syntax:- public String toUpperCase()
- **toLowerCase()**
  We use this method for converting string uppercase to lowercase.
  Syntax:- public String toLowerCase()
- **isSelected()**
  This function is used to check which of the radio button is selected.
  Syntax:- public Boolean isSelected()
- **dispose()**
  We use this method for hide current executed class.
  Syntax:- public void dispose()
- **add()**
  Add all components to a container frame by using add method contains class.
  Syntax: - public void add (Component c)
- **getComponent()**

This function returns component value and accepts nothing.
Syntax:- public component getComponent()

- **ceil()**

  This function is defined in math class. This function accepts double value and returns also double value.
  Syntax:- public static double ceil(double d)

Some user defined function and classes which is used in this project.

*** Medical**

This class contains main method. Welcome page object is created in this class.

- **connect**

  This class contains acconect() method. A static connection type variable con is declared in this class.

- ***aconnect()***

  We create an acconect method to connect java with oracle. This method not accepts any value but it returns connection. This method is defined within connect class. A try and catch block is used within this function.
  Driver path and oracle userid, password, port number and other necessary things are done in this
  Particular function.

- **Validity**

  This class is used to validate value while inserting in signuppage.
  This contains a string type variable Mystring and contains 7 methods named name(),email(),

Username(),authentication(),password(),setVarifiedString(),noEmpty Fields().

- ***name()***

  We create name method within validity class. This method accepts string type value and return Boolean value.
  It validates first name value and last name value

entered while entering data in the signuppage. First me trim the string and then

Change the string to upper case. We defined a Boolean type variable valid whose default value is true.

At any condition check if it is found that condition is not satisfied then value of valid is changed to false.

First we have checked the length of the string that whether the string length is between 2 to 15 characters.

Then we have used a for loop which contains if conditional control statement that checks that whether

All characters in string is alphabets. Then if valid value is true then we copy value of s to mystring.

Then at last we have returned the value of valid.

- ***email()***

We email name method within validity class. This method accepts also string type value and return Boolean

Value. This checks the email value entered in signuppage. A valid Boolean type variable is declared in this

Method and it's default value is true. First of all string is trim and converted to upper case.

Then if conditional control statement is checked that whether the string contains two consecutive dots

Or the string contains .@ or @. . If these found true then valid is set to be false. Then length of string is

Checked that whether It is greater than 30 characters. Then it is checked that whether the email value ends

With .com and does not starts with a dot symbol. It is also checked that the string value must contain

Only one @ symbol. At last if value of valid is true then string value is passed to mystring and finally

Valid value is returned to calling function.

- ***username()***

We create username method within validity class. This method also accepts string type value and return Boolean value. A valid Boolean type variable is also defined in this method as defined in all methods of validity class.

A For loop is used to check character by character value that if any special symbol is present in the username.

Then length of string is checked that it must be between 6 to 30 characters. Then a condition is checked that

No user can sign up with the name superuser as it is administrator and predefined name by the programmer

To login the project. Then finally if the value of valid Boolean type variable is true then email value is copied

To mystring and finally valid value is returned to calling function.

- ***authentication()***

  We create authentication method within validity class. This method also accepts string type value and return Boolean value. This function checks only that authentication number value is only "ncBCA201417".

  This is used so that no unauthorized person can sign up and login the project without the permission

  Of programmer.

- ***password()***

  We create password method within validity class. This method accepts string type value and return Boolean value. In this method it is checked that whether data inputed in password text box and confirm password text

  Box matches or not. It is also checked that both the string length must be between 6 to 15 characters.

- ***setVarifiedString()***

We create setVarifiedString method within validity class. This method accepts nothing but return string type value. It returns mystring value to the calling function.

- **noEmptyFields()**

This function is defined in validity class. This function accepts 7 values and returns Boolean type variable.

This method basically contains all values entered in the signuppage and checks that whether any value

Is empty or null.

- **AboutPage**

This class is used to show the details of the project including information about the contributor in this

Project. This class contains showFrame() method which is used to create a frame of awt and set image

And command button. It contains an ok button. When a user clicks ok button it simply dispose that is

It's widow is automatically closed.

- **WelcomePage**

This class is used to make welcomepage window in this project. It contains three command buttons that is

Getstarted, exit, About. If a user clicks on get started button then login widow will be displayed.

If a user clicks on exit then project will be closed and if a user clicks on about button then aboutpage object

Is created that is details of the project is shown.

- ***ReportPage***

This class is used to make Report page form. The basic details of patient like name, sex, age and problem

Of patient is input in the respective text box. This form contains two buttons next and back. If a user clicks

On next button then report is generated. If a user clicks on back button then last window is displayed whether

The user left of.

- **ValidateHome**

  This class is used to validate data input in patient record that is report table. It contains a method named valid().

- **Valid()**

  This function is defined in valid class to validate values inserted in report table. This function accepts two

  Two String type values and returns Boolean type value. An if conditional control statement is used to check

  The length of the string is between 3 and 30 characters or not. Then it is checked that special symbols or

  Numeric digits must be not present in the name. Age length is also checked that it's length should be less than

  Three characters and must contain only numeric digits.

- **InsertTableHome**

  This class is used to insert value input in patient detail form to insert in oracle table named ReportTable.

  It contains a method named home() .

- ***home()***

  We create home method within InsertTableHome class. This method accepts four string type values and return Boolean value. All four values is inserted in ReportTable named sql table. These values are name, age, sex,

  And problem of patient inserted in report table window.

- **LoginPage**

  This class is created to make login window that accepts user name and password in the respective text boxes.

  It contains showFrame() method that contains all required coding of awt to hold different labels and textboxes.

  It also contains a radio button which contains which type of login user wants that is stock handler or billing.

- **UserCheck**

  This class is used to check existing users from oracle table. It contains a method named check().

- ***check()***

  We create check method within UserCheck class. This method accepts string type value and return int type value.
  This function is used to take data that is user name and password from login page. Then it checks in signUpTable
  Created in oracle that whether this user and password exists or not.
  It also checks the rights of the user.

- **SignUpPage**

  This class is used to create frame of signup page. If a user wants to sign up to create a username and password to
  Become an authorized user. Various validations on different text boxes like first name, last name, email, password and other required fields through validity class. It contains a showFrame method which contains
  All the required components to be added to signuppage.

- ***insertuser()***

We create insertuser method within InsertSignUp class. This method accepts string type value and return null value. This method is used to insert all values of users who sign up to create new username and password to
Login the window. All values are inserted in SignupTable values.

- **ExaminPage**
  This class is used to generate report of patient which is filled in the report page. It also asks some questions to
  The patient so that doctors can easily recognize the disease. A ShowFrame method is defined in this class
  To add all the necessary components to create this window using swing and awt. Get() method is also defined
  To take value from report table window and setText to this examin page window.

- **UserExist**
  This class is used to check username while filling in signup table that whether user already exists in oracle table.
  It contains a method named user().

- ***user()***
  We create user method within UserExist class. This method is accepted string type value and return Boolean value. It checks in signuptable exist in oracle that whether this particular user already exists or not.

- **SignedUpPage**
  This class is used to generate successful message after complete filling of sign up page.

- **BillingPage**
  This class is used to create the window the billing page of this project. When user inputs product code and press
  Tab then name price is automatically filled by implementing focus listener on product code text box. There is a drop down menu to choose quantity of product. After selecting quantity and press tab left in stock is automatically filled. Then add to cart button is to be pressed if a user wants to add the product to his shopping list. User can also view cart to see added item to the list. The showFrame() method is defined in this class to
  Set all the required item to create this window. A get() method is also defined in this class.

- ***get()***
  We create get method within BillingPage class. This method accepts null type value and return null value.

- **InsertCart**
  This class is used to insert values to be purchased in the oracle table so that it can be easily retrieved when
  Clicked on view cart button. An insert method is defined in this class to perform operations to insert record
  In the table.

- **StockHandlerPage**
  This class is used to manage medicine stock. showFrame() method is defined in this class to add all the
  Necessary components in this frame in order to view, add, delete, modify all the records. Get and clear
  Method is also defined in this class.

- ***insert()***
  We create insert method within InsertCart Class. This method accepts null type value and return Boolean value.

A simple try catch block is used to insert record in the oracle table.

- ***clear()***
  We create clear method within StockHandler class. This method accepts null type value and return null value.
  This method is used to clear all the text box after deleting, successfully adding or modifying, deleting etc.

- **StockJTable**
  This class is used to view record of stock medicine. A showFrame() method is defined in this class
  To make frame and define jtable and store all value from oracle table to jtable body.

- **TableStock**
  This class is used to access value from oracle table named StockTable. getRow() and views() method is
  Defined in this class to perform these operations.

- ***getRow()***
  We create getRow method within TableStock class. This method accepts nothing but return int type value.
  This function is used to count number of rows in StockTable. A simple try and catch block is used to Perform these operations
- **Views()**
  This method is defined in TableStock class. It accepts a double dimension string array and also returns a
  Double dimension string array. All the values are accessed from table and one by one stored in double
  Dimension string array.

- **UpdateStock**
  This class is used to update record of TableStock class. Upstock() method is defined in this class.

- ***upstock()***
  We create upstock method within UpdateStock class. This method accepts string type value and return Boolean value. Update operation is performed in this class on the basis of product code. If update is successful then it
  Returns true to the calling function otherwise returns false.

- **ValidateStock**
  This class is used to validate values while adding record to medicine stock. Valid() method is defined in
  This class to perform various check operations to validate the values. Check() method is also defined in
  This class to check numeric type values.

- ***valid()***
  We create valid method within ValidateStock class. This method accepts string type value and return Boolean value. No characters other than alphabets is allowed in name and only numeric digits is allowed in quantity.

- **Check()**
  This function accepts a string and returns Boolean type variable. This function is used to validate mrp
  and quantity.

- **CheckCode**
  This class is used to check code value that whether it is already existing in oracle table or not. Check()
  Function is defined in this class to perform these operations.

- **DeleteStock**
  This class is used to delete a particular record on the basis of code. Delete() method is defined in this
  Class to perform these operations.

- ***delete()***
  We create delete method within DeleteStock class. This method accepts nothing but return Boolean value.
  Code is accepted by this method. On the basis of code this method delete that particular record.

- **SearchStock**
  This class is defined to search record when user clicks on search command button on stock handler page.
  To perform this operation search() method is defined in this class.

- ***search()***
  We create search method within SearchStock class. This method accepts string type value and return Boolean value. This function search record on the basis of code of medicine.

- **InsertStock**
  This class is used to insert a fresh record into the oracle table. Stock() method is defined in this class.

- ***stock()***
  We create stock method within InsertStock class. This method accepts string type value and return Boolean value.
  This method accepts 6 values from calling function that is code, name, mrp, quantity, exp and price. Then all

Values are inserted in stock table class.

- ***MainMenu()***
  This default constructor is defined in MainMenu class. This is used to make menu based operation on this project.
  All operations related to this project can be handled from this menu.

- **ViewReport**
  This class is used to view report of medicine. Two methods are defined in this class. First one is getrow() method.
  This method accepts nothing but returns integer type variable that is number of rows.
  This method is used to count the number of records in the MedicalReport table.
  Second method is views() method that is defined to access data from table and store in a double dimension array.
  This method accepts nothing but returns a 2-d string array.

- **Report**
  This class is used to make frame to view report of medicine in the form of jtable. showFrame() method is defined
  In this class to perform these operations. After accessing a 2-d string array from views() method of ViewReport
  Class we check the date that whether date change. If date change then total calculated price is shown to previous
  Record in total row. At last last date is also calculated and shown to total row.

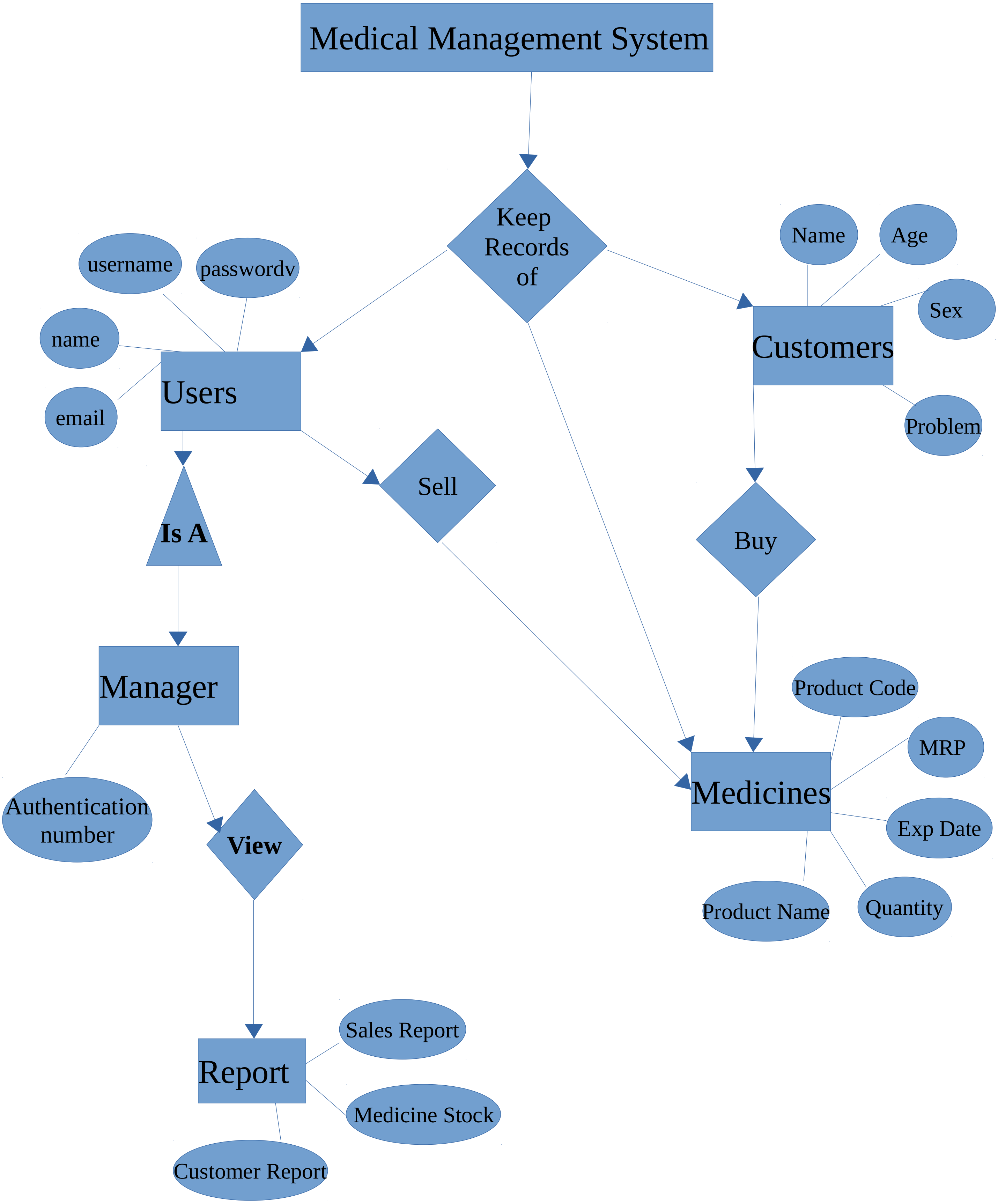
- **CartPage**
  This class is used to make frame of billing page receipt. It opens whenever user clicks on view cart button On billing page.

showFrame() method is defined to add all the necessary components to the this frame.
Get() and date() method is also defined in this class.
Get() method accepts integer type variable and does
Not return any value. Date() method accepts nothing and return one string type variable.

- **InsertReport**
  This class is used to insert records in medical report table created in oracle. Pass() and report() method is defined in this class.
- **Pass()**
  This method is defined in insert report class. This method accepts one 2-d string array and returns nothing. This method simply calls Report() method as many times as the number of rows. It passes 6 values to report class.
- **Report()**
  This method is defined in insert Report class to insert records in medical report table created in oracle. It accepts
- ***update()***
  We create update method within UpdateQuant class. This method accepts string type value but return null value.
- ***upstock()***
  We create upstock method within UpdateQuant class. This method accepts null value and return null value.
- ***trunc()***
  We create trunc method within UpdateQuant class. This method accepts null value and return null value.

- ***quantst()***
  We create quantst method within UpdateQuant class. This method accept string type value and return int value.

10. E-R Diagram



11. File Description or Table Column

1. SignupTable

ColumnName	Datatype	Size	Attributes
fname	varchar2	15	Not null
lname	varchar2	15	
email	varchar2	30	Not null
userid	varchar2	20	Not null
auth	varchar2	15	Not null
rights	varchar2	40	
pass	varchar2	15	Not null

2. ReportTable

name	varchar2	30	Not noll
age	number	5	
sex	varchar2	7	
problem	varchar2	20	

3. StockTable

code	varchar2	20	Primary key
name	varchar2	40	
quant	number	5	
rate	number	5,2	
mrp	number	5,2	
exp	varchar2	20	

4. InsertCart

pcode	varchar2	20	
pname	varchar2	40	
quant	varchar2	2	
price	varchar2	6	
netprice	varchar2	6	

5. MedicalReport

code	varchar2	20	
name	varchar2	40	
quantity	varchar2	2	
mrp	varchar2	7	
price	varchar2	7	
mdate	varchar2	12	

12. Coding

```
/*
Medical Store Management System
Admin: superuser
password: nkk50kkk
authentication number: ncBCA201417
*/
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import java.sql.*;
import java.util.*;
/* to create table in sql
    user the file Table.lst which can be found in
the same directory as the Medical.java file

    @D:\Project\Table.lst;
    path may change according to
    availability of Table.lst file in your
computer*/


public class Medical
{
    static public String active="Medical"; //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();

    }
}

//************** Misc Classes
**************************
```

```
// class connect is used to connect java
// with oracle by giving driver path and oracle
// id and password according to currend userid
// and password change the userid and password
// and also path if required


class connect
{
    static Connection con;//used form storing
connection path in a variable

    /*  aconnect method is used to provide path to
    // connect java with oracle
    // this method returns Connetion and
    // this not accept any value*/
    public Connection aconnect()
    {
        try
        {


Class.forName("oracle.jdbc.driver.OracleDriver");


con=DriverManager.getConnection("jdbc:oracle:thin:
@localhost:1521:xe","system","manager");
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
        return con;
    }
}


//Validity class contains all the methods related
to varifying the different fields
class Validity
{
    //all the methods below validiates an Value as
per the method's name
```

```
    //it returns false if it is not valid else
returns true

    String myString;

    public boolean name(String s)
    //NAME
    {
        /*
        length should be more than 2
        and it should not contain any symbol other
tha A-Z
        */

        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;
        //checks if there is no any consecutive
occurance of '.' or occurance of '.' invalid
position
        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");
        }
        //checks if ends with '.com' and does not
start with '.'
        if(s.endsWith(".com") && !
(s.startsWith(".")))
        {
        //limits the character range to a-z,0-
9,@,_,-,.
            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
    {
        /*
        contains no space
        and limits characters to only 0-9,A-Z,a-z
        */

        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;
    }


    //return the varified string
    public String setVarifiedString()
    {
        return myString;
    }

    //checks if there is any empty field or not
    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.equals("")||s7.equals(""))
        {
            valid=false;
        }

        return valid;
    }

}

//********************** THE ABOUT
PAGE*****************
class AboutPage extends JFrame implements
ActionListener
{

    static JButton ok;
    public void showFrame()
    {

        Container c=this.getContentPane();

        //label
        JLabel title=new JLabel("Image");
        title.setIcon(new
ImageIcon("images\\about.jpg"));

        //text area
        JTextArea about=new JTextArea(200,150);
        about.setEditable(false);
        about.setText("Welcome to Medical
Management System \n\nDeveloped by: Nitin Kumar,
Kamesh Shekhar, Omraj Kumar and Keshav
Kumar\nCollege: Nalanda College, BCA Department
(Session 2014-17)\n\nObjectivity: The Project is
Aimed at providing ease in maintaing and handling
the stocks\n of medicines.Along with it, The
application provides facility to generate
patient's report\n that will allow the doctor to
treat the patient in best possible way");
        about.setFont(new Font("Times New
Roman",Font.ITALIC,15));
        about.setBackground(Color.BLACK);
        about.setForeground(Color.WHITE);



        ok=new JButton("OK");

        JPanel p=new JPanel();
        p.add(ok);


        c.add(title,BorderLayout.NORTH);
        c.add(about,BorderLayout.CENTER);
        c.add(p,BorderLayout.SOUTH);

        this.setTitle("About");
        this.setVisible(true);
        this.setSize(600,400);
        this.setLocationRelativeTo(null);
        this.setResizable(false);
        //this.setDefaultCloseOperation(JFrame.EXI
T_ON_CLOSE);

        ok.addActionListener(this);

    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==ok)
        {
            dispose();
        }
    }
}

//**************** THE WELCOME PAGE
**********************
class WelcomePage extends JFrame implements
ActionListener
```

```
{
    static JButton start, exit, about;
    public void showFrame()
    {

        this.setLayout(null);
        Container c=this.getContentPane();

        JLabel lbl=new JLabel();

        start=new JButton("Get Started");
        exit=new JButton("Exit");
        about=new JButton("About");

        //*************** 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");

        about.setBackground(Color.YELLOW);
        about.setForeground(Color.BLACK);
        about.setFont(new Font("Times New
Roman",Font.BOLD,15));
        about.setToolTipText(" It contains
Developer and college information");

        start.setSize(150,50);
        start.setLocation(310,350);
        about.setSize(80,40);
        about.setLocation(50,350);

        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(about);
        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
);

        about.addActionListener(this);
        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);
        }
        else if(e.getSource()==about)
        {
            AboutPage a=new AboutPage();
            a.showFrame();
        }
    }
}

//**************** 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 illnes
is related to

        type=new Choice();
        type.add("Cough and Cold");
        type.add("Mental Illness");
        type.add("Eyes");
        type.add("Bones");
        type.add("Stomac 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);

        //this.setDefaultCloseOperation(JFrame.EXI
T_ON_CLOSE);
```

```
        //************* Adding listeners to all
the three buttons ************************
        next.addActionListener(this);
        back.addActionListener(this);

        //name.addFocusListener(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);
               //passing value of
name,age,sex,type that is problem
               //to class Insert table Home so
that value can be inserted in table
               //variable b is used to check
whether value is successfully inserted or not

               if(b==false)

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


JOptionPane.showMessageDialog(this,"Successfully
inserted");

               ExaminPage h=new ExaminPage();
               dispose();
               h.showFrame();

               /*get function is defined in
               ExaminPage to set value
               of name,age,sex and problem
               so we are passing four values*/

               h.get(n,age.getText(),s,t);
           }
       }
       else if(e.getSource()==back)
       {
            if(Medical.active.equals("LoginPage"))
            {
                LoginPage h=new LoginPage();
                h.showFrame();
            }
            else
if(Medical.active.equals("MainMenu"))
                new MainMenu();

                dispose();
        }
    }
}

//ValidateHome class is used to validate
//values while inserting detils of patient like
name ,age etc

class ValidateHome
{
    //valid method is used to validate name and
age
    //it returns boolean type variable
    //it accepts two string type variable
    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;

        //this for loop is used to check that
        // if name variable is containing
        // only alphabets and space after
        for(int i=0;i<name.length();i++)
        {
            if(name.charAt(i)>'Z' ||
name.charAt(i)<'A' )
            {
                //na variable is used to get ascci
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();

            //acconect 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 report, 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()
    {
        Medical.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");
        report=new JButton("Generate Report");


            //******* 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);
                p1.add(report);

        //********************** 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);
        report.addActionListener(this);

    }

    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==login)
        {
            String usr=user.getText();
            String psw=pass.getText();

            /*-------------------------------------
--------------------------------
            code for getting username and password
from the database goes here. */

            userchek u=new userchek();//creating
object of class to check user
            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("nkk50kkk")
||b==1||b==2)
           {
               if(rb1.isSelected())
               {
                   BillingPage p=new
BillingPage();
                   p.showFrame();
                   dispose();
               }
               else if(rb2.isSelected())
               {
                   if(superRights)
                   {
                      //StockHandlerPage p=new 
StockHandlerPage();
                      //p.showFrame();
                      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();
      }
      else if(e.getSource()==report)
      {
          ReportPage h=new ReportPage();
              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 unauthorazed
     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);

        //**** declaration of correctness of data
            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
authintication 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.getSelectedItem();

    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 Mandotary!");
        }
        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(Medical.active.equals("LoginPage"))
                {
                LoginPage p=new LoginPage();
                p.showFrame();
                }
                else
if(Medical.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 SignedUpPage 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(Medical.active.equals("LoginPage"))
               {
                   LoginPage p=new LoginPage();
                   p.showFrame();
               }
               else
if(Medical.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
                   mandatory for customer to
select atleast one value
                   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);

		delete.setLocation(630,240);
		modify.setLocation(630,310);

		//------------------------------
Setting up size of other labels and textFields
		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.g
etText(),mrp1.getText());
                      if(v==true)
                      {
                          String
n=name1.getText();
                          int
q=Integer.parseInt(quan1.getText());
                          double
r=Double.parseDouble(rate1.getText());
                          double
m=Double.parseDouble(mrp1.getText());

                          String ex;//used to
concate date value

                          String
month=exp1.getSelectedItem();
                          String
year=exp2.getSelectedItem();

                          ex=month+year;


                          InsertStock Is=new
InsertStock();//creating object of class that
insert stock
                          //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
                          values in stock method
of insertstock
                          class if it returns true
then successfully inserted
                          message is shown
otherwise error message is displayed*/

                          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 updatestock 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","Exp
iry"};

        /* 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
```

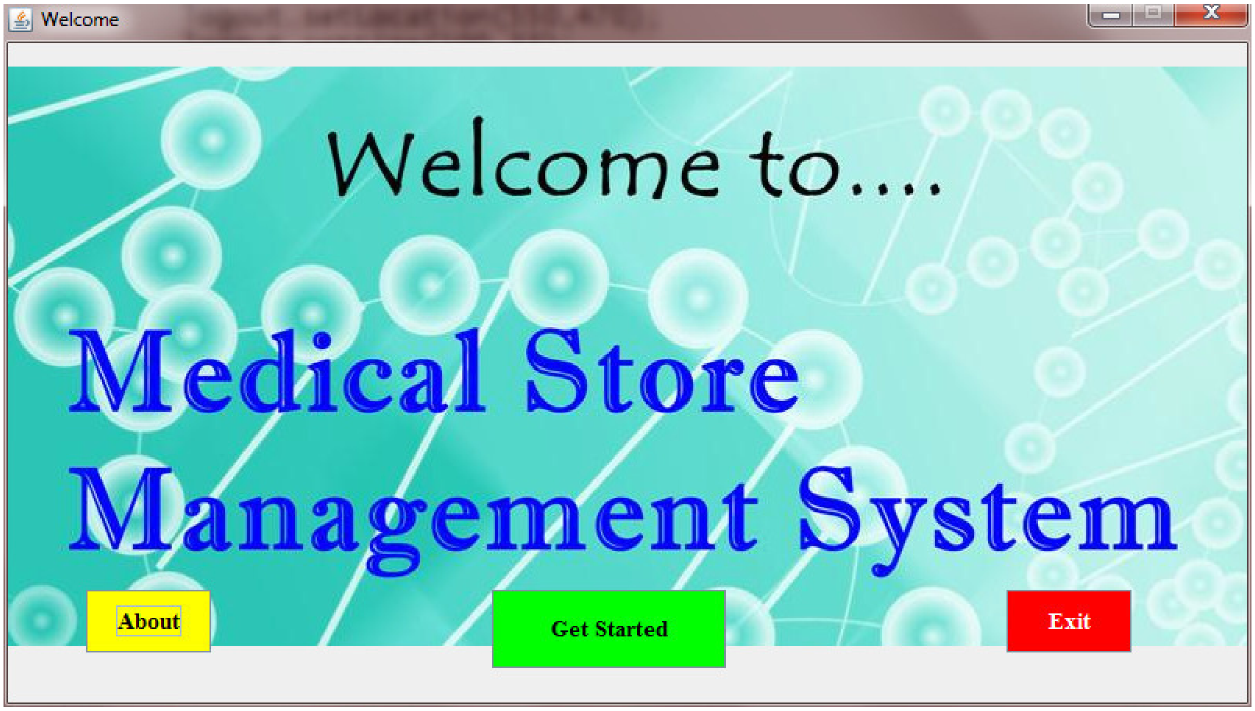
```
        back.addActionListener(this);
    }

    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;
    }
}
```

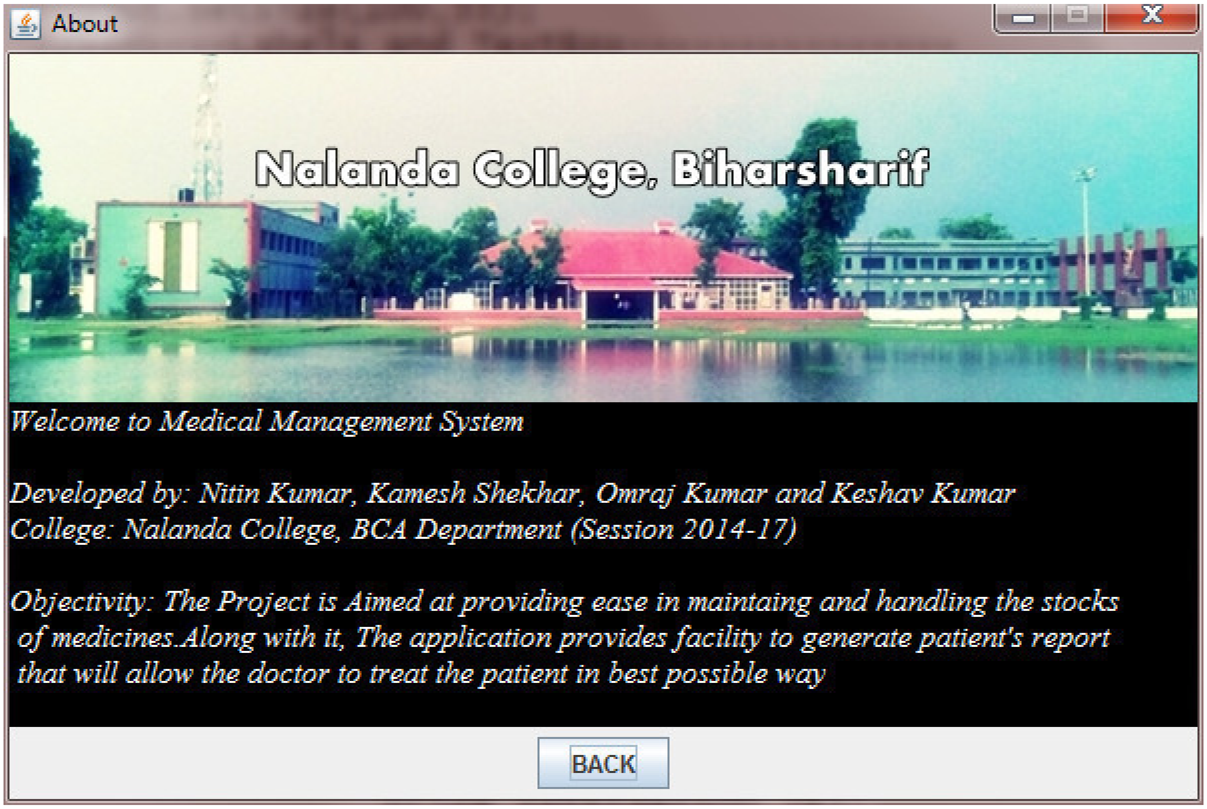
```
/* updatestock class is used to update
the record of stocktable
after searching operation and user press modify
button
*/

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,d1,d2,e1;

    public MainMenu()
    {
        Medical.active="MainMenu";
        //working with menubar

        JMenuBar bar=new JMenuBar();
        JMenu m1=new JMenu("User");
        JMenu m2=new JMenu("Manage");
        JMenu m3=new JMenu("Billing");
        JMenu m4=new JMenu("Records");
        JMenu m5=new JMenu("Help");
        bar.add(m1);
        bar.add(m2);
        bar.add(m3);
        bar.add(m4);
        bar.add(m5);

         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");

         d1=new JMenuItem("Generate Patient 
Record");
         d2=new JMenuItem("Invoice Report");

         e1=new JMenuItem("About Medical 
Management System");


        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 poduct 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");
        d1.setToolTipText("Add a new Patient");
        d2.setToolTipText("View the Sales Report
as Table");
        e1.setToolTipText("About Medical
Management System");


        m1.add(a1);
        m1.add(a2);
        m2.add(b1);
        m2.add(b2);
        m2.add(b3);
        m2.add(b4);
        m2.add(b5);
        m3.add(c1);
        m4.add(d1);
        m4.add(d2);
        m5.add(e1);


        // 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);
        d1.addActionListener(this);
        d2.addActionListener(this);
        e1.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();
        }
        else if(e.getSource()==d1)
        {
            ReportPage p=new ReportPage();
            p.showFrame();
        }
        else if(e.getSource()==d2)
        {
            Report r=new Report();
            r.showFrame();
        }
        else if(e.getSource()==e1)
        {
            AboutPage p=new AboutPage();
            p.showFrame();
        }

    }


}


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 MedicalReport");
            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
MedicalReport");
                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","Da
te","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
MedicalReport 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;
    }
}
```

13. Input or output screen design

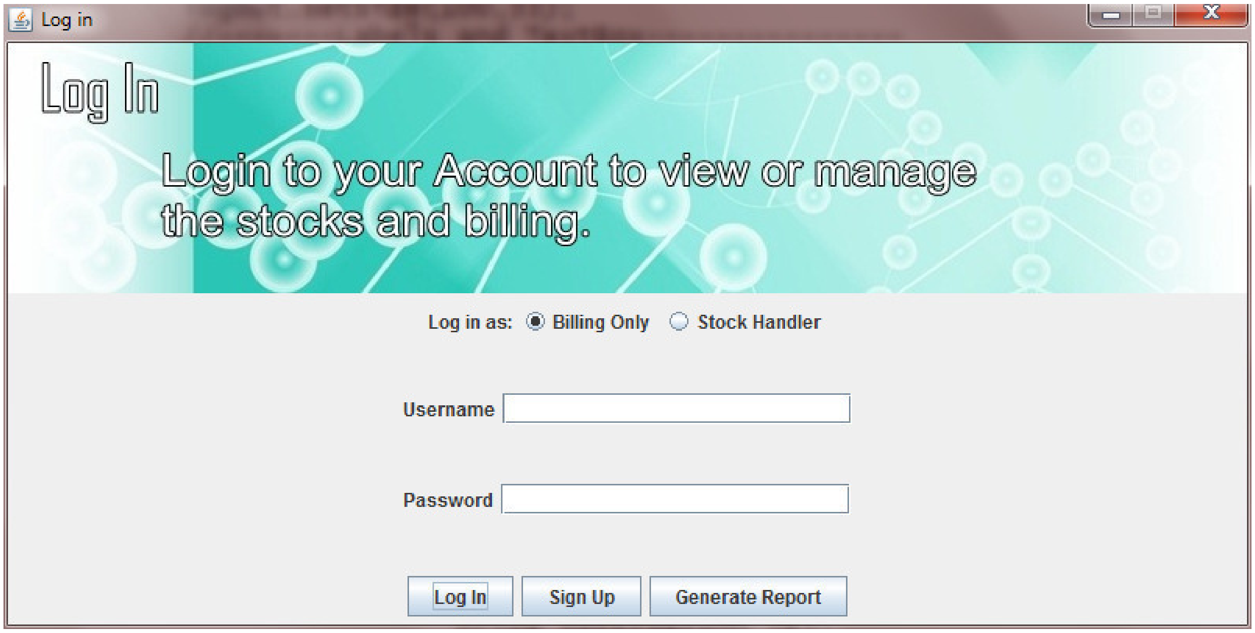




Welcome to Medical Management System

Developed by: Nitin Kumar, Kamesh Shekhar, Omraj Kumar and Keshav Kumar
College: Nalanda College, BCA Department (Session 2014-17)

Objectivity: The Project is Aimed at providing ease in maintaing and handling the stocks of medicines.Along with it, The application provides facility to generate patient's report that will allow the doctor to treat the patient in best possible way



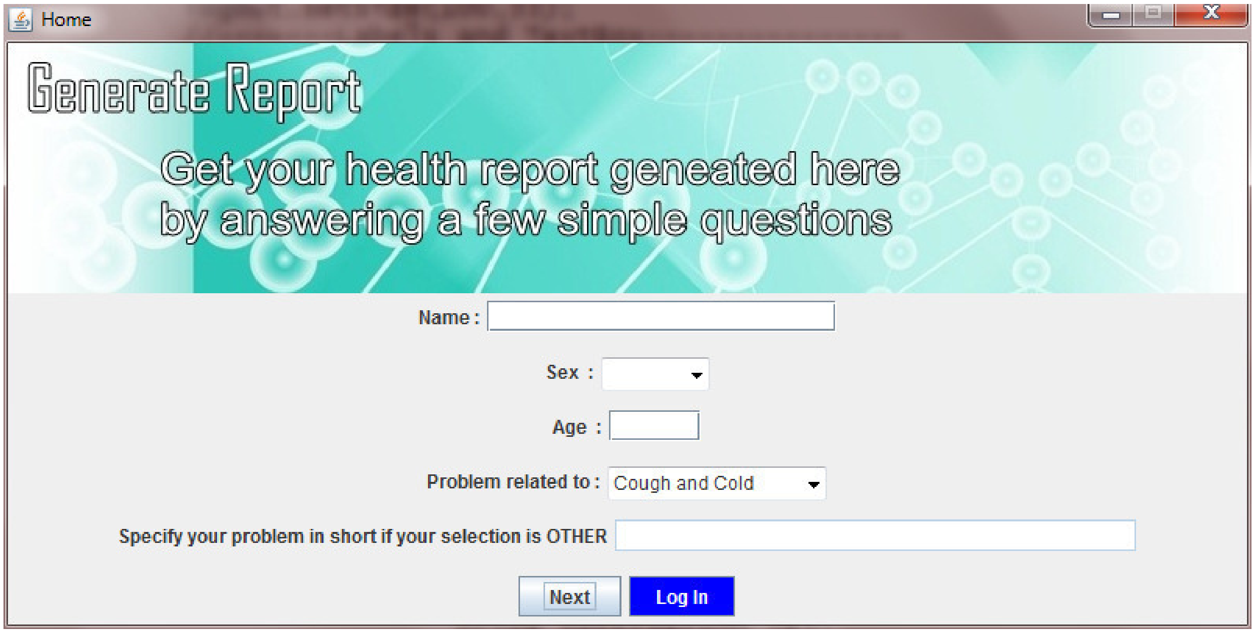
Login to your Account to view or manage the stocks and billing.

Log in as: ● Billing Only ○ Stock Handler

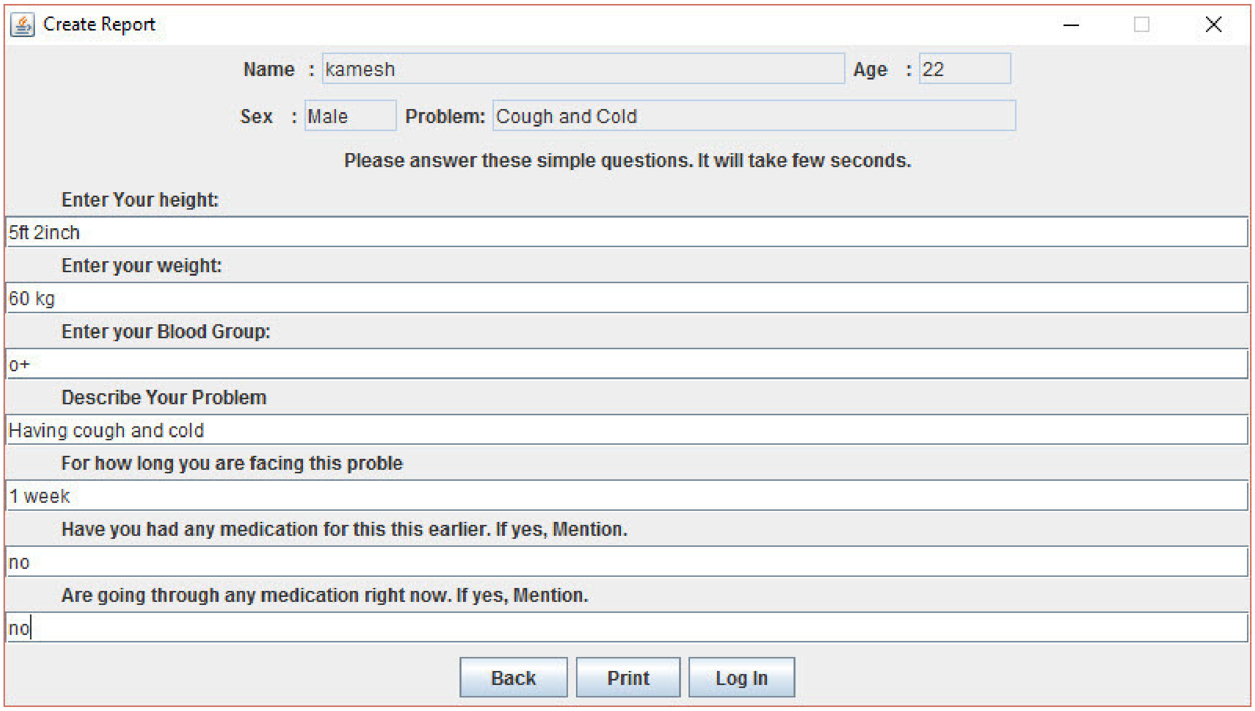
Username

Password

If user clicks on generate report then report page is visible otherwise log in is done according to user’s rights provided.


Home
Generate Report
Get your health report geneated here
by answering a few simple questions
Name :
Sex :
Age :
Problem related to : Cough and Cold
Specify your problem in short if your selection is OTHER
Next
Log In

Generate Report: Provides Report of the customer or patient for future reference and allows us to keep it as an Hard Copy.



Name : kamesh Age : 22

Sex : Male Problem: Cough and Cold

Please answer these simple questions. It will take few seconds.

Enter Your height:
5ft 2inch

Enter your weight:
60 kg

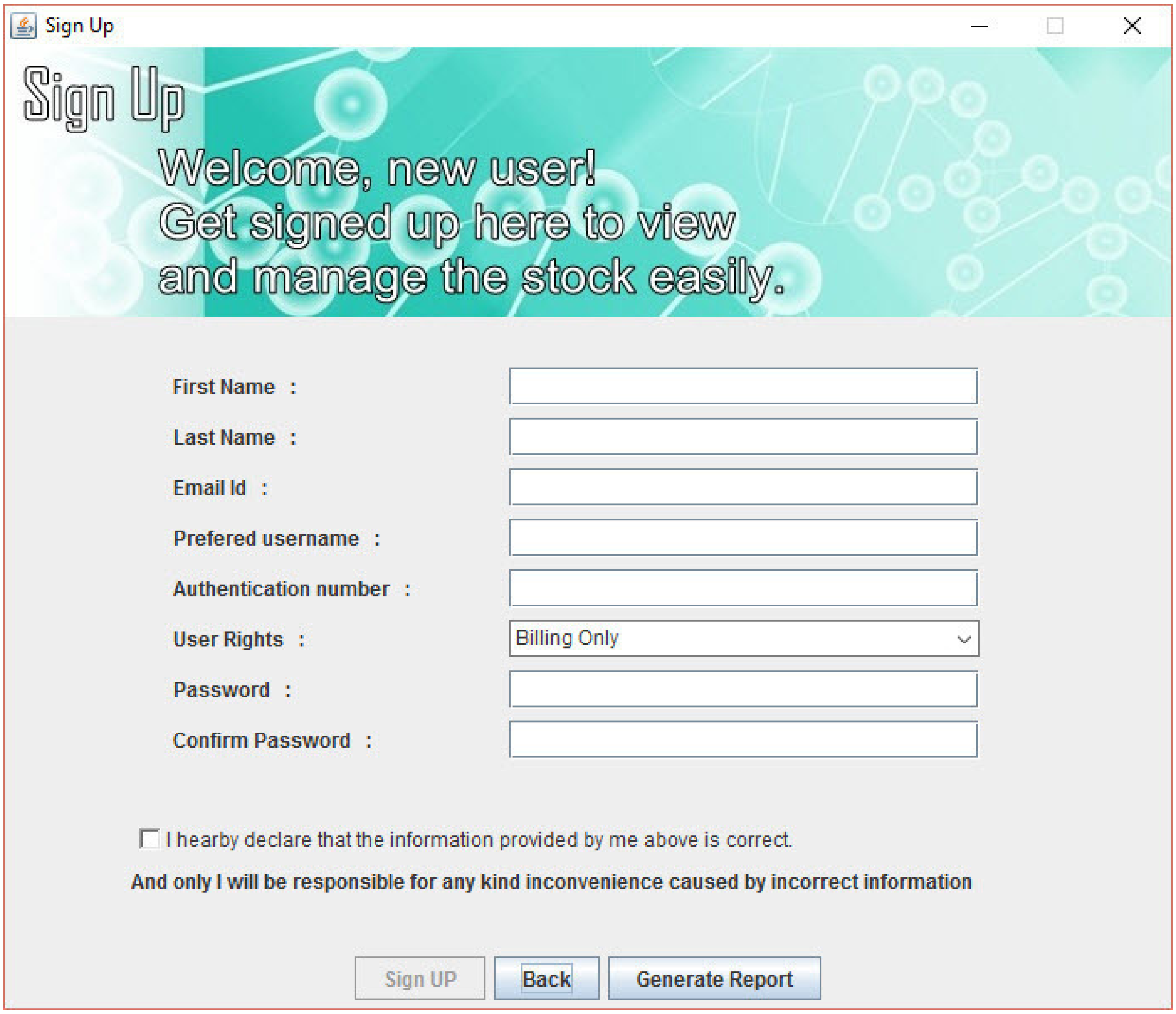
Enter your Blood Group:
o+

Describe Your Problem
Having cough and cold

For how long you are facing this proble
1 week

Have you had any medication for this this earlier. If yes, Mention.
no

Are going through any medication right now. If yes, Mention.
no

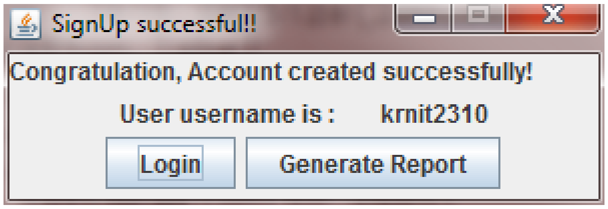


Welcome, new user!
Get signed up here to view
and manage the stock easily.

First Name :	
Last Name :	
Email Id :	
Prefered username :	
Authentication number :	
User Rights :	Billing Only
Password :	
Confirm Password :	

□ I hearby declare that the information provided by me above is correct.

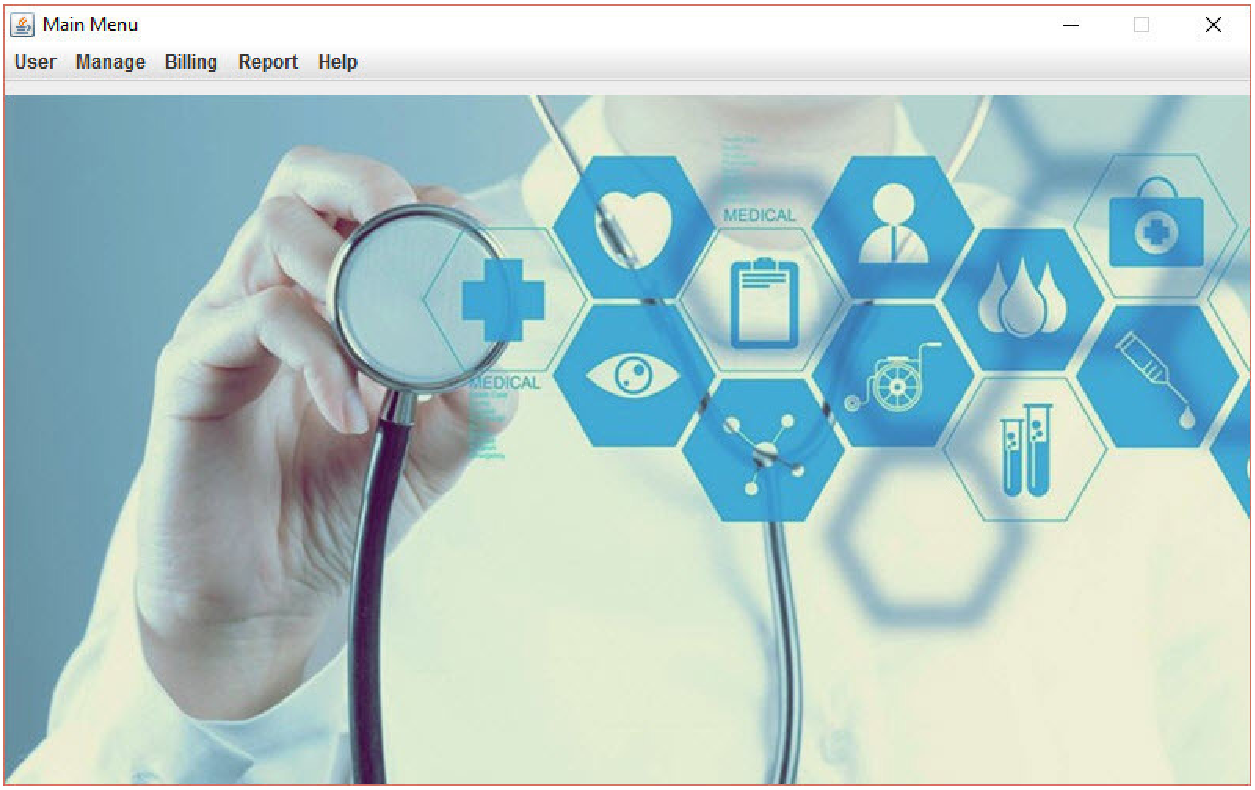
And only I will be responsible for any kind inconvenience caused by incorrect information



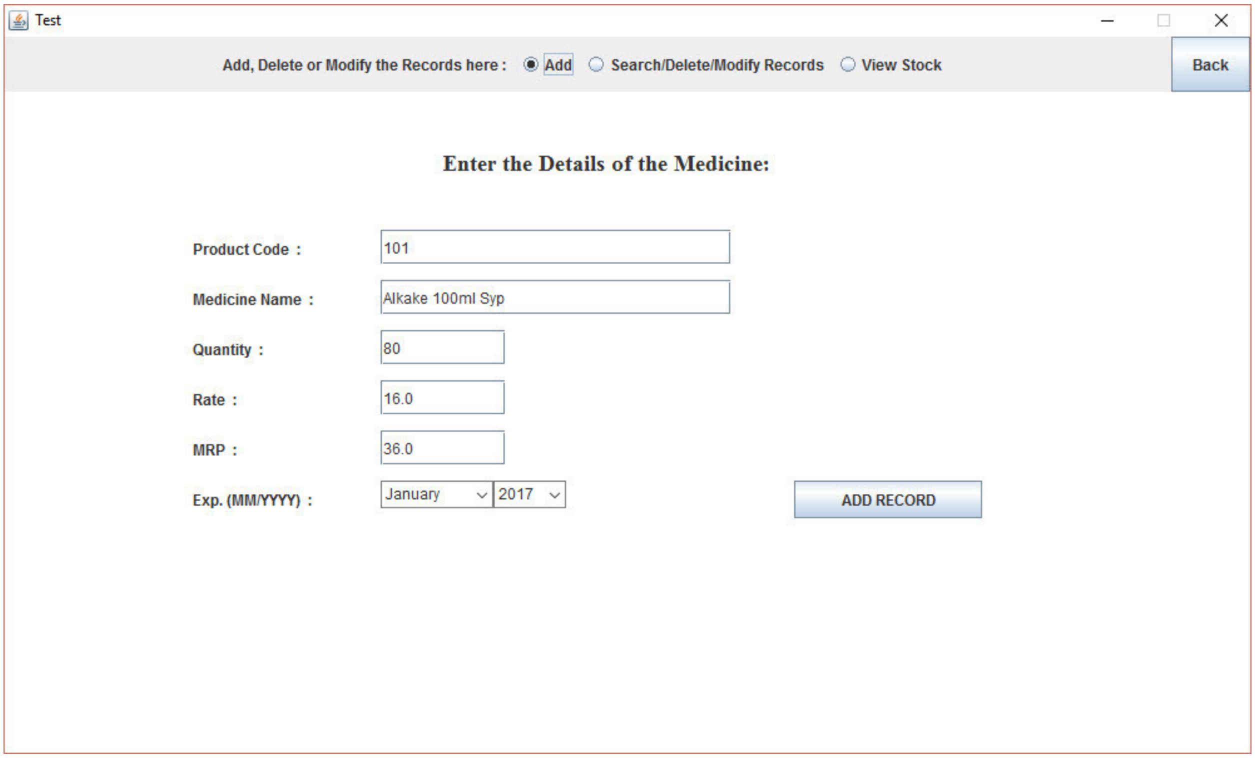
Congratulation, Account created successfully!

User username is : krnit2310

A new user can get signed up by an authorized person for using this software.



The Main menu contains all quick link. In the menu bar so that the user can jump to the desired location, saving his/ her time.



Add, Delete or Modify the Records here : ● Add ○ Search/Delete/Modify Records ○ View Stock

Enter the Details of the Medicine:

Product Code : 101

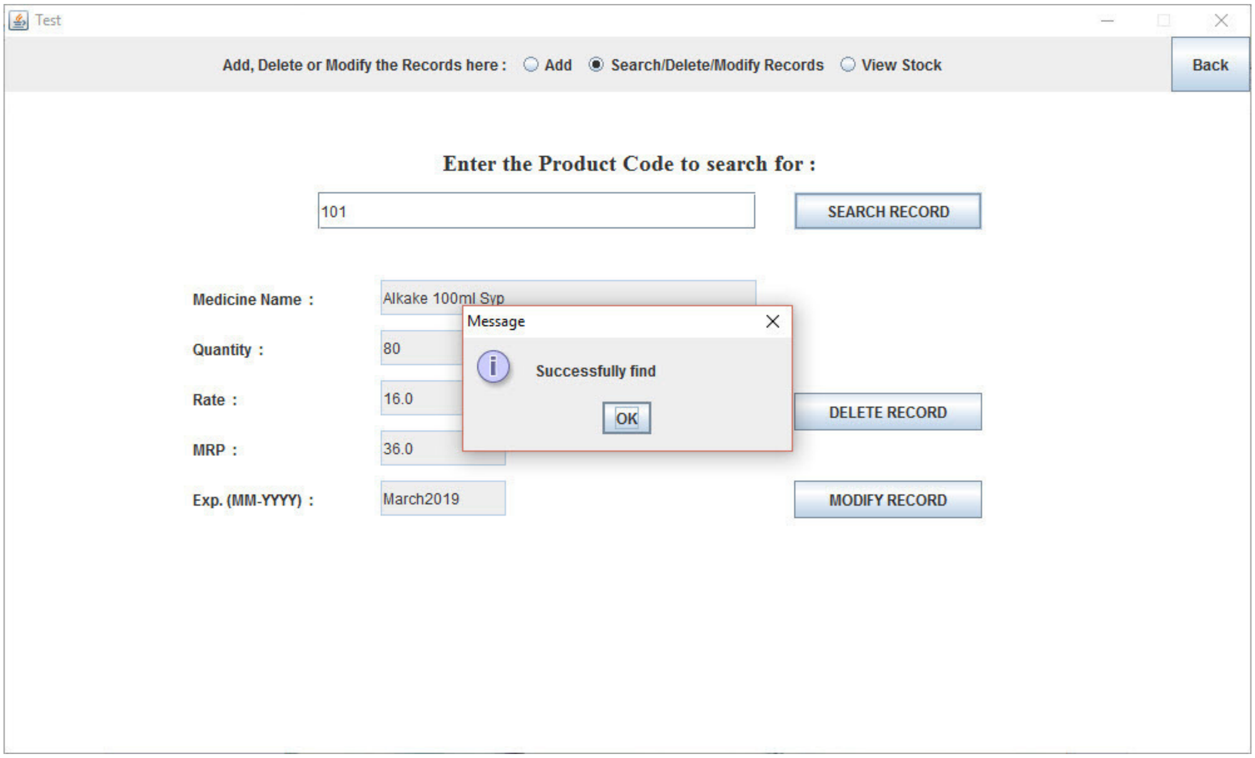
Medicine Name : Alkake 100ml Syp

Quantity : 80

Rate : 16.0

MRP : 36.0

Exp. (MM/YYYY) : January 2017



Add, Delete or Modify the Records here : ○ Add ● Search/Delete/Modify Records ○ View Stock

Enter the Product Code to search for : 101

Medicine Name : Alkake 100ml Syp

Quantity : 80

Rate : 16.0

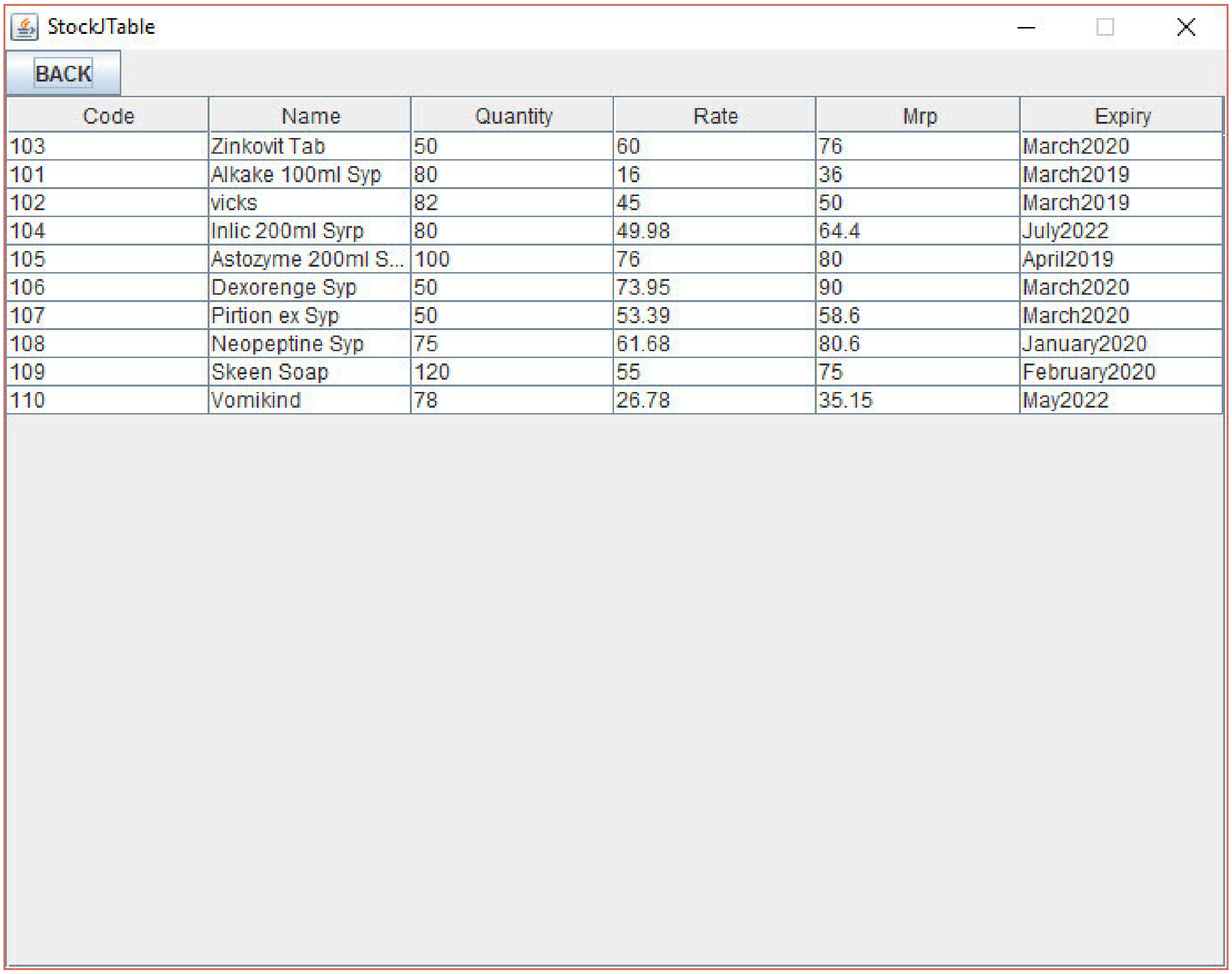
MRP : 36.0

Exp. (MM-YYYY) : March2019

Message

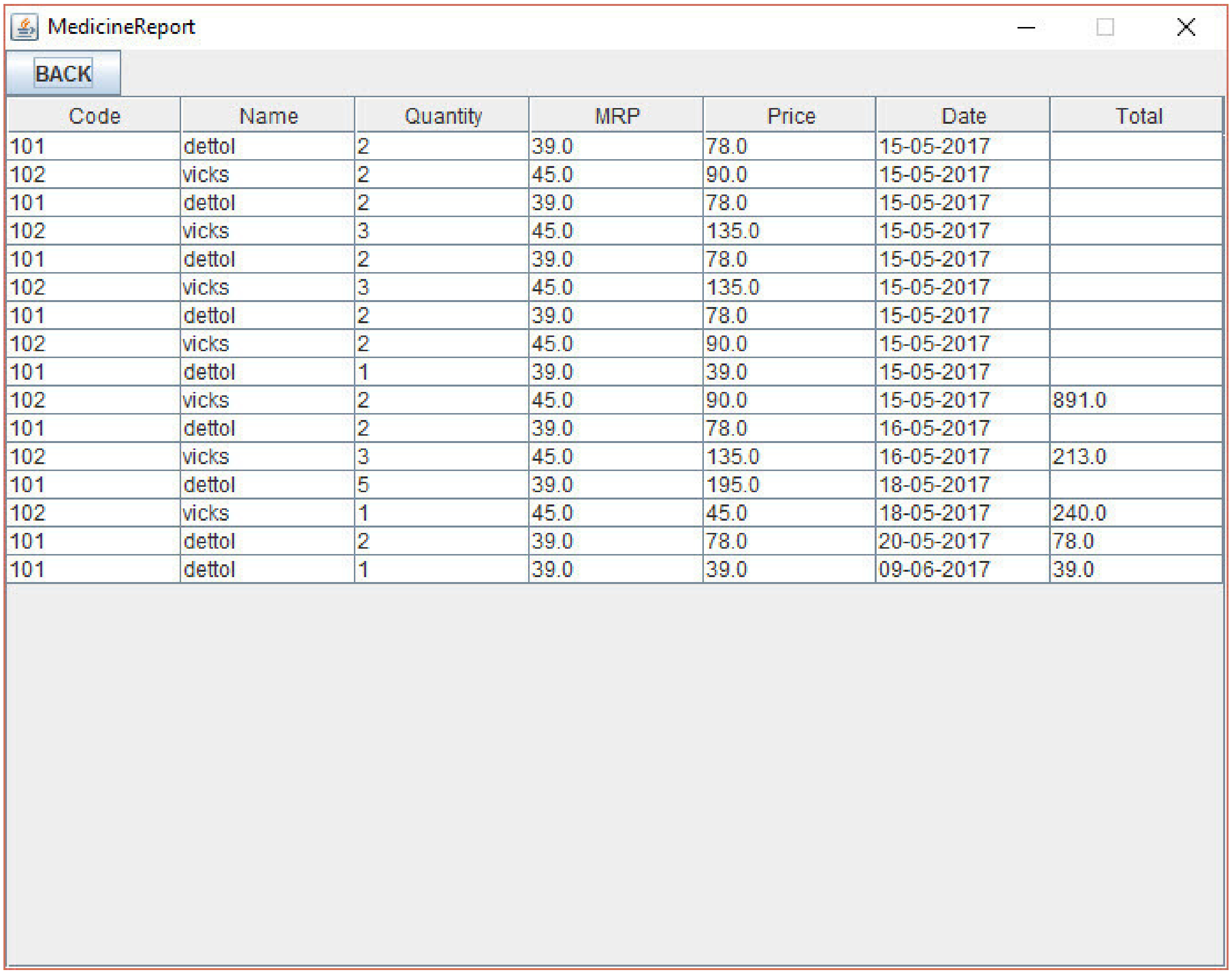
Successfully find

Search for the desired medicine quickly for modification or deletion.



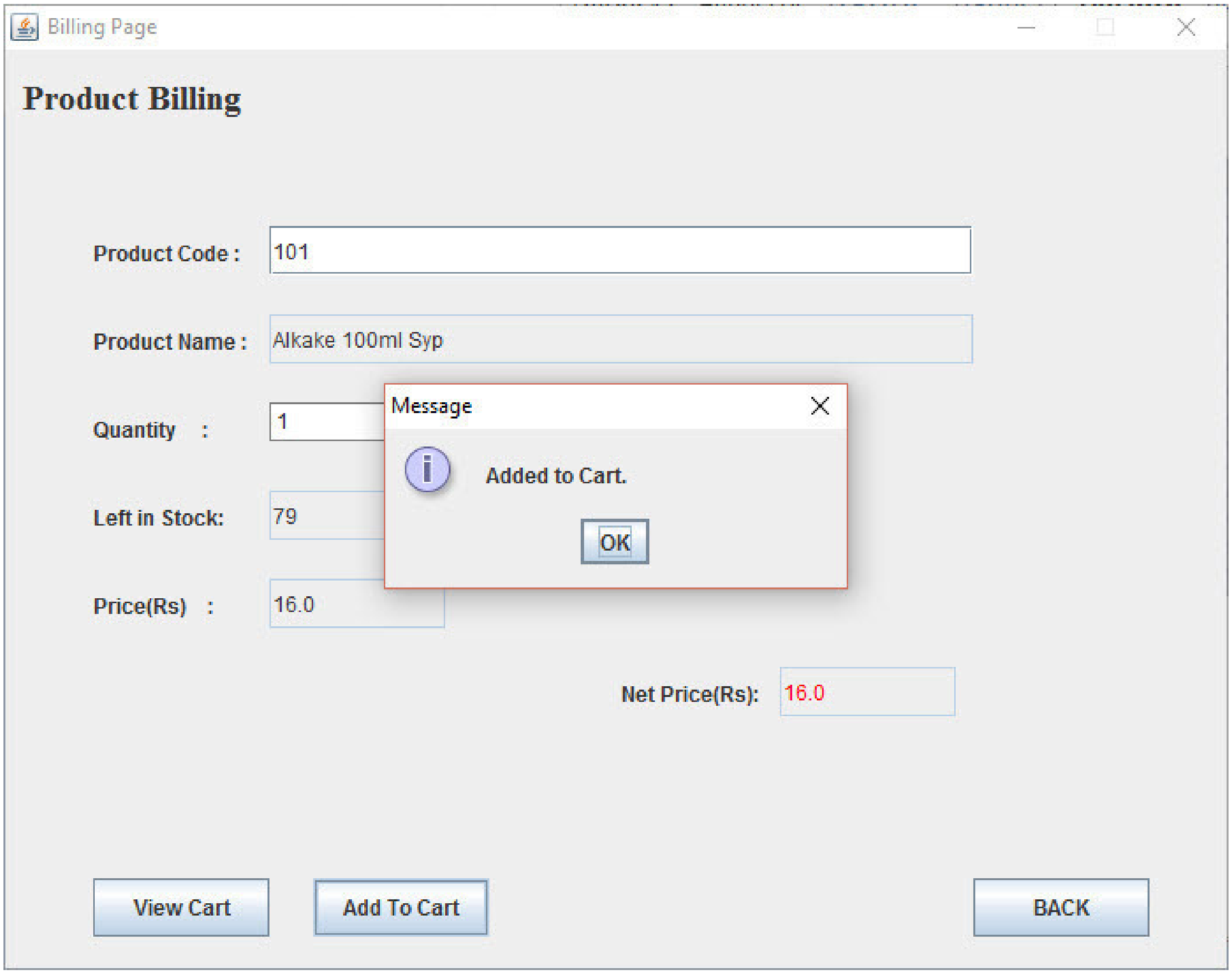
Code	Name	Quantity	Rate	Mrp	Expiry
103	Zinkovit Tab	50	60	76	March2020
101	Alkake 100ml Syp	80	16	36	March2019
102	vicks	82	45	50	March2019
104	Inlic 200ml Syrp	80	49.98	64.4	July2022
105	Astozyme 200ml S...	100	76	80	April2019
106	Dexorenge Syp	50	73.95	90	March2020
107	Pirtion ex Syp	50	53.39	58.6	March2020
108	Neopeptine Syp	75	61.68	80.6	January2020
109	Skeen Soap	120	55	75	February2020
110	Vomikind	78	26.78	35.15	May2022

View the stock as Table



Code	Name	Quantity	MRP	Price	Date	Total
101	dettol	2	39.0	78.0	15-05-2017	
102	vicks	2	45.0	90.0	15-05-2017	
101	dettol	2	39.0	78.0	15-05-2017	
102	vicks	3	45.0	135.0	15-05-2017	
101	dettol	2	39.0	78.0	15-05-2017	
102	vicks	3	45.0	135.0	15-05-2017	
101	dettol	2	39.0	78.0	15-05-2017	
102	vicks	2	45.0	90.0	15-05-2017	
101	dettol	1	39.0	39.0	15-05-2017	
102	vicks	2	45.0	90.0	15-05-2017	891.0
101	dettol	2	39.0	78.0	16-05-2017	
102	vicks	3	45.0	135.0	16-05-2017	213.0
101	dettol	5	39.0	195.0	18-05-2017	
102	vicks	1	45.0	45.0	18-05-2017	240.0
101	dettol	2	39.0	78.0	20-05-2017	78.0
101	dettol	1	39.0	39.0	09-06-2017	39.0

Keep track to all the the transaction by easily checking out Sales Report



Product Billing

Product Code : 101

Product Name : Alkake 100ml Syp

Quantity : 1

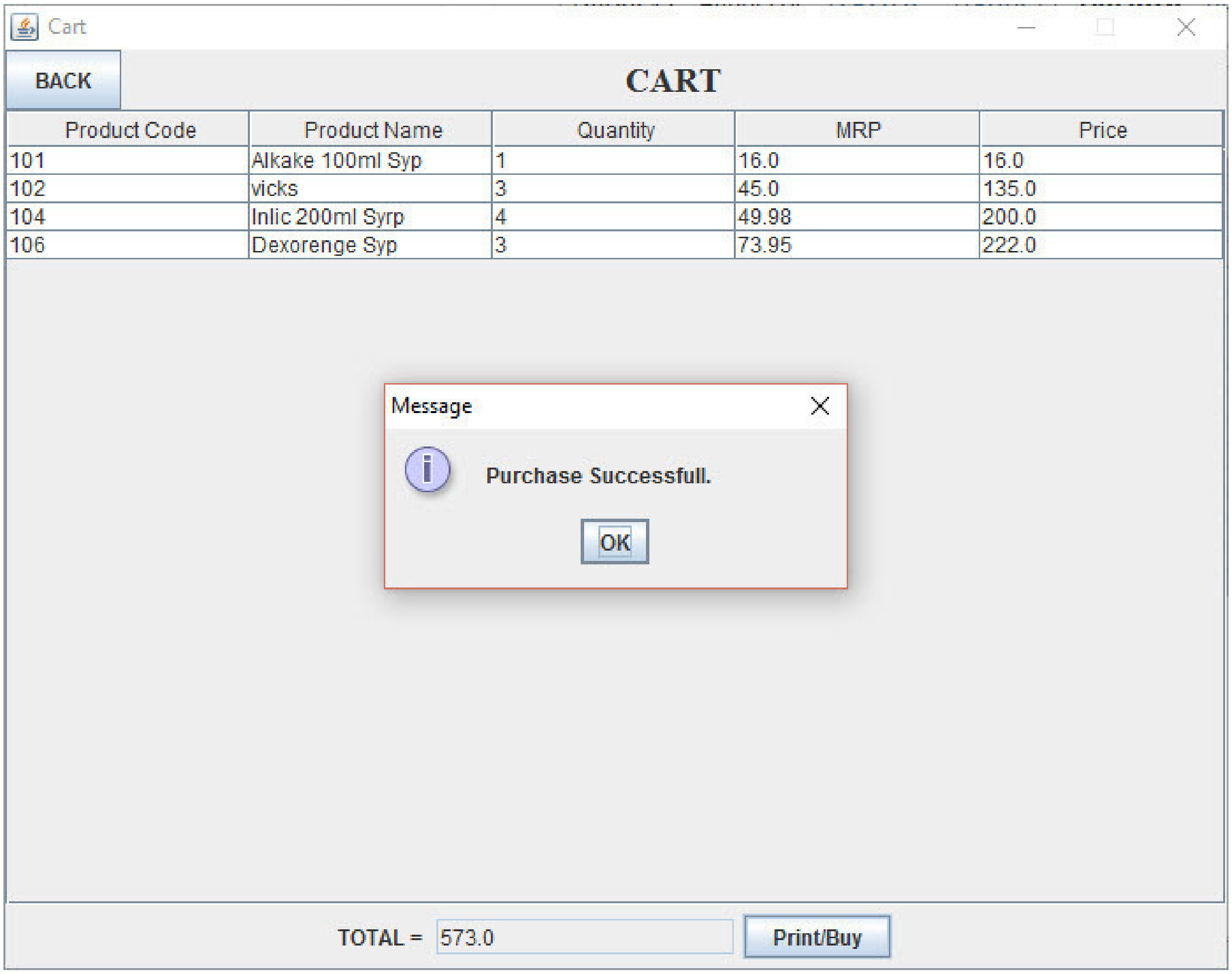
Left in Stock: 79

Price(Rs) : 16.0

Net Price(Rs): 16.0

Added to Cart.

Hassle Free Billing.



Product Code	Product Name	Quantity	MRP	Price
101	Alkake 100ml Syp	1	16.0	16.0
102	vicks	3	45.0	135.0
104	Inlic 200ml Syrp	4	49.98	200.0
106	Dexorenge Syp	3	73.95	222.0

Message

Purchase Successfull.

TOTAL = 573.0

Easily Customize the purchases.

14. 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 challengers 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 completely of GUI"s has grown, leading to more difficulty is the design and execution of test crises. Finites step modeling graphs man is used to drive a series of test that address specific data and programs objects that are relevant to the GUI. Due to the large no. of permutation associated with GUI operations testing should be approached using automated fools.

Verification and validation:-

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

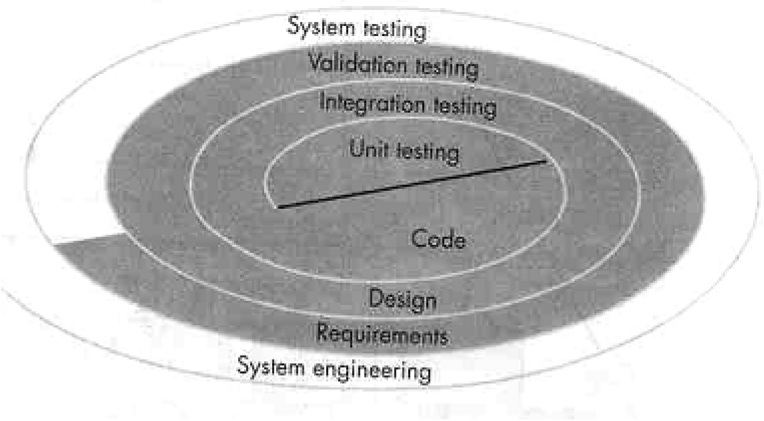
In other words validation provides final assurance that software meets all functional, behavioral and performance requires.

"Boebam" states another way:-

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

Verification and validation encompasses a wide array of SQA (software quality assurance) activities that include formal technical review, quality and configuration audits, performance monitoring, simulation, fusibility study, documentation review, development testing, qualification testing and installation testing. Software testing strategy:-

A strategy for software testing may also be viewed in the context



of the spiral.

Unit testing is begins at the vortex of the spiral and concentration on each units (component) of the software as implemented in source code. Testing programs by moving outward along the spiral to integrated testing, where the focus is on design and construction of the software architecture. Taking another term out word on the spiral is in countered validation testing, where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive other at system testing where the s/w, we special out along stream lines that we spiral the scope of testing with each turned.

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.

The module interface is tested to insure that information properly follows into and out of the program unit under test. The local data structure is examined to insure that data stored temporarily maintains its integrity during all steps in an algorithms execution. Boundary condition is tested to insure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are executed at least once. And finally all error handling paths are tested.

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. A number of different incremental integration strategies are:-

Components are combining to form clusters 0, 1, 2, & 3.Each of the cluster 1&2 subordinate to "Ma". Similarly driver "D3" for cluster "3" is removed prior to integration with module "Mb". Both "Ma" and "Mb" will ultimately be integrated with component "M.C." and forth. Integration moves upward.

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.

There are two types of validation testing are:-

- **Alpha testing**
- **Beta testing**

A. **alpha testing:-**

The alpha testis conducted at the developer's site by a customer. The software is used in natural setting with the developer “looking over and shoulder" of the user and recording errors and using problems. Alpha test are conducted in a control environment.

B. **Beta testing:-**

Beta testing is conducted at one or more costumer site by the end user of the software. Unlike alpha testing, the developer is generally not present. Therefore the beta test is a "live" application of the software in an environment that cannot be controlled by the developer. The customer records all problems that are countered during beta testing and reports these to the developer at regular intervals. As a result of problems reported during beta tests, software engineers make modifications and then prepare for relation of the software product to the enter customer based.

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 property integrated and perform allocated functions.

- **Recovery Testing:-**

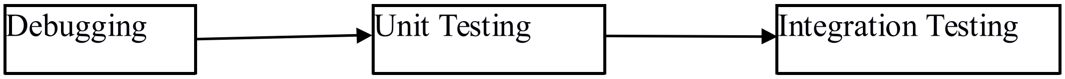
  Recovery testing is a system test that forces the software to fail in a variety of ways and verifies that recovery is properly performed. If recovery is automatic (perform by the system itself) reutilization, check pointing mechanism, data recovery and restart are evaluated for correctness. If repair (MTTR) is evaluated to determine whether it is within acceptation limits.

- **Security Testing:-**

  Security testing attempts to verify that protection mechanism built into a system will infect, protect it from improper penetration. During security testing the tester plays the role of the individual who desire 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.

Debugging Process:-

Debugging occurs as a consequence of successful testing. That is, when a test case uncovers and error, debugging is the process that results in the removal of the error.



The debugging process will always have one of two outcomes:-

1. The cause will be found and corrected.
2. The cause will not be found.

The person performing debugging may suspect a cause, ensign a test case to help validate the suspicion, and work toward error correction in an iterative fashion.

15. Scope of project

The project can be used in any industry with some minor modification. These modifications may be of both types- either hardware or software limitation or additional requirement made by the staffs of the industry.

There is always room for further enhancement of this system in the following areas.

- Hierarchy modification / additional capabilities are inbuilt in the system.
- Dynamic screens according to requirement can be introduced any time.
- System is very flexible for further modifications.
- With very little modification, it can be deployed for use in other industry system.

16. Limitation of project

The Software is developed for only window based environment. Actually not only Windows environment is sufficient rather Windows 98 onward version is needed, because here I am going to use JAVA as front end and Oracle edition as Backend, which doesn't support earlier version of windows.

Minimum Hardware Requirement

1. **Computer system**: Pentium Based system with at least 2.4 GHz speed or Higher

2. **RAM**: At least 256 MB, it is recommended to have 512 MB of RAM for fast processing on data.
3. **Hard Disk Space**: At least 10 GB of free space, it is recommended to have 20 GB of free space for data storage.

Minimum Software Requirement

1. **Operating System**	: Windows 2007 or Higher.
2. **DATABASE Package**	: oracle
3. **FRONT**-END	: java

Others

This version will not work on a **network system** as this is developed for a single system because in the industry there is only one computer available for handling those tasks. If, this software is to be implemented on a network based organization then some modified will be necessary.

17. Implementation & Maintenance

The software of **Medical shop management system** completely new system i.e. the existing system is manual.

The main aspects of implementation are as follows:-

- Training personnel
- Conversion procedure.
- Demonstration.

For the proposed system **Medical shop management system** will be beneficial. Under this approach, users continue to operate the old system in the usual manner but they also start using the new system. This method is safest one because it ensures that in case of any problem in using the new system, the organization can still fall back to the old system without loss of time and money.

Maintenance

Software maintenance is of course for more than "fixing mistakes". We may define maintains by describing for a activates that are under ken after a program is released for use. For different maintained activates are:-

1. **Corrective maintains:-**

   Even with the best quality assurance activities, it is likely that the customer will uncover defects in the software. Corrective main tense changes the software to correct defects.

2. **Adaptive Maintains:-**

   Overtime, the original environment (CPU, OS, business rules, external product characteristics) for which the software's was developed in likely to change. Adaptive maintains result in modification to the software to accommodate change to its external environment.

3. Perfective maintains or enhancement maintence:-

As software is used, the customer/user will recognize additional function that will provide benefit. Perfective maintained extends the software beyond its original function requirement.

4. Perfective maintains/Reengineering:-

Computers software deteriorates due to change, and because of this, preventive maintained, often called software reengineering, and must be conducted to enable the software to serves the needs of its end users. In essence, preventive maintains makes change to computer programs so that they can be more easily corrected, adapted and enhanced.

5. Installation testing:-

Implementation means to take into practice. A crucial phase in the system line cycle is the successful implementation of the new system design. Implementation includes all those activities that take place to convert or system or automated system.

18. Bibliography

Books as Reference:

Let us Java – Yashvant Kanetkar
Java the Complete Reference, Seventh Ed. - Herbert Schild

Websites as Reference:

www.tutorialspoint.com/java
www.codeacademy.com/learn/learn-java
http://stackoverflow.com
www.youtube.com/