MEDICAL STORE BILLING MANAGEMENT SYSTEM

MINI PROJECT REPORT

By
G. Mahesh

CONTENTS

OBJECTIVE

The main objective of this project is to design and develop a Medical Store Billing System to implement a software application for medical shops and hospitals for maintaining easy billing system.

Medical Store Billing Management System project is designed to focus on the medical store to help them in their billing management with taking care of other details like stock and accounting.

As today in the country most of the medical stores are manual only which means they maintain their record of the person buying drugs and the quantity of it in the paper files, registers and binders.

This practice is very unreliable and prone to mistakes and omissions. The medical store billing system help the owner of the store in keeping the track of its customer and the drugs he/she is buying from store which further help it in the analysis of getting the stock for the next term.

Usually when customer buy the medicines from the store the salesperson has to record his/her details in register then person also has to write the medicines details in separate record which is hectic and time consuming.

This system will automatically update the record in all database as customer buys the medicines which saves the store’s time and the resources.

ABSTRACT

The medical store billing system ease the medical shops and the salesperson by providing the fully computerized data storage facility which help them to generate the bill and update the stock at the same time without any extra manual work.

This also helps the owner to search any record at any point of time to generate the report like how many people are buying what kind of medicines, which kind of drugs' sales is going down, as database is storing all data at one place over the hardware which is globally present.

As the system is online it makes the process of billing faster than before as the staff do not have to check the register for each medicine that it is available in their stock or not, he/she has to just search in system for the drug and it will show the stock detail which makes the billing less time taking process.

If the owner wants to know from where it has bought a particular drug it can search in system with just click as seller record is also well maintained and kept related to the medicine it sold.

Existing System of Medical Store Billing Management System

Considering existing system which is used by the medical store is manual. As all the employees manually working to complete each and every process.

This not just make the task mare time consuming but also makes the employee lethargic to do any productive work. The recording of the data in registers and storing them in file, binders make them prone to errors and mistakes which will harm the store in long term.

This type of system also has less security as one has to take the measure of keeping the stacks of binders and registers safe in the room which will require store to occupy the more space for storage of data instead of products.

This makes the system costlier as maintenance and keeping the files require the human resource and the stationary also. As the data is manually recorded if the theirs is query filed by any customer or the manager it would be difficult to give immediate response to them as its time taken process of finding the information and then analyzing them.

Following this kind of system does not only require lot of human resource but also budget for maintaining them and stationary required. This will incur the weight on the owner's finance. This shows us various drawbacks in this system which are -:

All the process is time consuming.

Every process requires proper arrangements which will need human resource.

Managing the staff entails to spending the money increasing the budget.

Billing process increment the level of complexity as the number of customers coming to book gas increases.

Information is not available globally to both customer and to store.

Staff manually evaluating the product expiration date is susceptible to faults and mistakes.

HARDWARE REQUIREMENTS (minimum)

The hardware required for the development of the project is:

Processor : Intel P-IV system
Processor Speed : 250MHz to 833MHz
RAM : 512MB RAM
Hard Disk : 40GB

INTRODUCTION

Once the planning and analysis of project is completed, the design phase begins. Goal of system design is to transform the information collected about project into the blueprint structure which will serve as a base while constructing the system.

It is considered to be unwieldy process as most of errors are introduce in this phase.

However, if error gets unnoticed in later process it may become difficult to track them down. In this project, we building a system which ease the medical store with the process of billing and maintain the record of the medicines and seller who sold them.

Admin entity maintains the record of the administrator of the store. The employees of the shop will act as the administrator of the system.

Though each admin has restricted access to the system except the owner or the manager of the store who has the full control over system like the salesperson has only access to the bill and product entity.

It cannot check the data of dealer entity whereas manager will be able to see every entity detail. It contains password field which creates a security blanket over the system as person who have authorized username and password will be allowed to login and use the system further prevents from any other anonymous person to take control the system and damage the data. It has following attributes-:

Username stores the name of the admin which act as a unique name given to every employee of the shop with this, they login into system and the transaction they made will be recorded against this name which help the manager if it wants to identify who has made a particular sell or purchase.

Password attribute holds the secured keyword given to every employee who need the access of the system. This should not be shared with any other member as it

would make any being to enter the system and see the information regarding sale and purchase of the store.

It has relationship with every entity as it supervises every data incoming or outgoing the system.

It generates the bill when customer buys the product from the store which have all the detail about buyer and product, he/she bought.

Considering existing system which is used by the medical store is manual. As all the employees manually working to complete each and every process.

This not just make the task mare time consuming but also makes the employee lethargic to do any productive work. The recording of the data in registers and storing them in file, binders make them prone to errors and mistakes which will harm the store in long term.

This type of system also has less security as one has to take the measure of keeping the stacks of binders and registers safe in the room which will require store to occupy the more space for storage of data instead of products.

This makes the system costlier as maintenance and keeping the files require the human resource and the stationary also. As the data is manually recorded if the theirs is query filed by any customer or the manager it would be difficult to give immediate response to them as its time taken process of finding the information and then analysing them.

Following this kind of system does not only require lot of human resource but also budget for maintaining them and stationary required. This will incur the weight on the owner’s finance.

This will also manage the stock present in the store with date and quantity with that it also sees the dealer details as what product they have sold to store at what price.

The following key points is considered for system-

: Interface for customer to buy the medicines

A Payment details should be handled

Store and maintain the detail information about medicines

System to check the stock and keep a record

Easy updating and adding of stocks

Record of seller from whom stock is bought

USER INTERFACE DESIGN of Medical Store Billing Management System

The Medical Store Billing System hassles out all the complication of conventional method which is combination of the folders, spreadsheets, the emails and for the customer and the employees.

It has a unified system in which all the process from registration, billing process and purchase-sales of products in one system. Here's how system manages to keep it simple for customers and the store:

Admin Interface: Every employee who must interact with the system will be given a username and password. He has the access to system by logging in. It provides the security to the store data and to the system.

Billing Interface: The employee should just input the name of the product and check its availability then make the bill through the system only without checking the records manually.

SOURCe coDE:

The source code has been divided into the three parts.

The first part is used for username and password verification of the user. If the user enters the valid username and password then he will be allowed or it is treated as an illegal entry.

Username stores the name of the admin which act as a unique name given to every employee of the shop with this, they login into system and the transaction they made will be recorded against this name which helps the manager if it wants to identify who has made a particular sell or purchase.

Password attribute holds the secured keyword given to every employee who needs the access of the system. This should not be shared with any other member as it would make any being to enter the system and see the information regarding sale and purchase of the store.

```
import Mypack.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MedicalBill extends JFrame
{
        public static void main(String[] args)
       {
         MedicalBill frameTabel = new MedicalBill();
```

```
}

    JButton blogin = new JButton("Login");
    JPanel panel = new JPanel();
     JLabel username = new JLabel("Username:");    JLabel password =
     new JLabel("Password:");
    JTextField txtadmin = new JTextField(15);
    JPasswordField pass = new JPasswordField(15);

    MedicalBill()
        {
       super("Login Authentication");
       setSize(400,300);
       setLocation(500,280);
       setResizable(false);
       panel.setLayout(null);
       username.setBounds(80, 65, 100, 20);
       password.setBounds(80, 110, 100, 20);
       txtadmin.setBounds(155, 65, 150, 20);
       pass.setBounds(155, 110, 150, 20);
       blogin.setBounds(160, 180, 80, 20);
       panel.add(blogin);
       panel.add(username);
       panel.add(password);

       panel.add(txtadmin);
```

```
        panel.add(pass);
        getContentPane().add(panel);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
        actionlogin();
    }

    public void actionlogin()
{
        blogin.addActionListener(new ActionListener()
{
            public void actionPerformed(ActionEvent ae)
{
                String paname = txtadmin.getText();
                String ppaswd = pass.getText();
                if((paname.equals("litcse_cse")&&ppaswd.equals("litcse392")))
{
                    NewFrame regFace = new
                    NewFrame(); regFace.setVisible(true);
                    dispose();
                }
Else
```

```
            {
            JOptionPane.showMessageDialog(null, "Invalid Password / "
                  + "Username");
            txtadmin.setText("");
            pass.setText("");
            txtadmin.requestFocus();
            }
          }
      }
     );
   }
}
```

When the user enters the correct username and password then the user is allowed to continue.

If the user enters the invalid username or password then a dialog box is shown saying that the user has entered an invalid password or username.

The second part shows the welcome screen.

2.

```
package Mypack;
import
javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class NewFrame extends JFrame
{
```

```
   public static void main(String[] args)
{
      NewFrame frameTabel = new NewFrame();
   }
   JLabel welcome = new JLabel("******WELCOME TO APOLLO MEDICALS
STORE******",
         SwingConstants.CENTER);
   JLabel dms = new JLabel(" Genuine Products with 10% offers on Beauty,
Nutrition, etc");
     JButton b = new JButton("continue");
   JPanel panel = new JPanel();


  public NewFrame()
{
      super("Welcome!");
      setSize(400,300);
      setLocation(500,280);
      setResizable(false);
      panel.setLayout(null);


      welcome.setBounds(60, 70, 300, 60);
      dms.setBounds(30, 100, 300, 60);
      b.setBounds(150,150,90,50);

      panel.add(welcome);
      panel.add(dms); panel.add(b);
      getContentPane().add(panel);
```

```
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
}
```

As soon as the user login then he sees a welcome screen with the store name and other things such as a continue button.

When the user clicks the continue button he will go the billing part.

The third part of the code deals with the billing process.

It generates the bill when customer buys the product from the store which have all the detail about buyer and product he/she bought.

Following this kind of system does not only require lot of human resource but also budget for maintaining them and stationary required. This will incur the weight on the owner's finance.

As the user clicks the continue button he gets a frame as shown below where the billing process is done.

It has a table which stores the medicine name,code ,service description,quantity and cost.

And also has three buttons:

Add: this button is used for the adding the medicine or selecting the medicine in the table.

Update: this button is used to update the added data in the table.

Delete : delete is used to remove the data from the table.

There are five text boxes where the user enters the medicine details and then as the user clicks the add button these are placed in the table and also can be update through the update button. Then the delete button is used to delete the data from the table.

We have another three buttons:

Select item: it is used to select the medicine from the table.

Reset: it is used to reset the table data.

Print: print button is used to print the selected items in the bill with the total amount.

```
import java.awt.Color;
import java.awt.Font;
```

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.JTextField;
importjavax.swing.JTextArea;
import javax.swing.table.DefaultTableModel;
import java.util.*;
 public class JTableRow extends JFrame
 {
public static void main (String args[])
{
JFrame frame = new JFrame();
JTable table = new JTable();
Object[] columns =
{"Medicalcode","Medicalname","Service_description","Quantity_available","Cost"
};
DefaultTableModel model = new
DefaultTableModel();
model.setColumnIdentifiers(columns);
```

```
table.setModel(model);


table.setBackground(Color.LIGHT_GRAY);
table.setForeground(Color.black);
 Font font = new Font("",1,22);
 table.setFont(font);
 table.setRowHeight(30);
 JTextField textMedicalcode = new JTextField();


 JTextField textMedicalname = new JTextField();
 JTextField textService_description = new JTextField();
 JTextField textQuantity_available = new JTextField();
 JTextField textCost = new JTextField();
 JTextArea billarea = new JTextArea(" ********** APOLLOMEDICALS
********** MEDICALBILL");
 JButton btnAdd = new JButton("Add");
 JButton btnDelete = new JButton("Delete");
 JButton btnUpdate = new JButton("Update");
 JButton sel_item = new JButton("Select_item");
 JButton res = new JButton("reset");
 JButton prin = new JButton("print");
 textMedicalcode.setBounds(20, 220, 100, 25); textMedicalname.setBounds(20,
250, 100, 25);
 textService_description.setBounds(20, 280, 100,
 25);
```

```
textQuantity_available.setBounds(20, 310, 100, 25);
textCost.setBounds(20,340,100,25);
sel_item.setBounds(100,490,120,25);
res.setBounds(220,490,100,25);
prin.setBounds(300,490,100,25);
btnAdd.setBounds(150, 220, 100, 25);
btnUpdate.setBounds(150, 265, 100, 25);
btnDelete.setBounds(150, 310, 100, 25);
billarea.setBounds(400,200,385,300);
// create JScrollPane
JScrollPane pane = new JScrollPane(table);
pane.setBounds(0, 0, 880, 200);
frame.setLayout(null);

frame.add(pane);
// add JTextFields to the jframe
frame.add(textMedicalcode);
frame.add(textMedicalname);
frame.add(textService_description);
frame.add(textQuantity_available);
frame.add(textCost);
// add JButtons to the jframe
frame.add(btnAdd);
frame.add(btnDelete);
```

```
frame.add(btnUpdate);
frame.add(sel_item); frame.add(res);
frame.add(prin);
frame.add(billarea);
Date obj = new Date();
String date = obj.toString();
billarea.setText(billarea.getText()+"\n"+"\t\t"+ "Date::"+date+"\n");
billarea.setText(billarea.getText()+" MEDICAL_NAME      "+
"SERVICE_DESCRIPTION      "+ "          COST"+"\n");

Object[] row = new Object[5];

btnAdd.addActionListener(new ActionListener()
{
public void actionPerformed(ActionEvent e)
{
row[0] = textMedicalcode.getText();
row[1] = textMedicalname.getText();
row[2] =
textService_description.getText(); row[3]
=textQuantity_available.getText();

row[4]=textCost.getText();
model.addRow(row);
}
```

```
}
)
;
btnDelete.addActionListener(new ActionListener()
{
@Override
public void actionPerformed(ActionEvent e) {

int i = table.getSelectedRow();
 if(i >= 0)
{

model.removeRow(i);
}
Else
{
System.out.println("Delete Error");
}
}
}
)
;
// get selected row data From table to textfields
table.addMouseListener(new MouseAdapter()
```

```
{
@Override
public void mouseClicked(MouseEvent e)
{
// i = the index of the selected row int
i = table.getSelectedRow();
textMedicalcode.setText(model.getValueAt(i,
0).toString());
textMedicalname.setText(model.getValueAt(i,
1).toString());
textService_description.setText(model.getValueAt(i, 2).toString());
textQuantity_available.setText(model.getValueAt(i, 3).toString());
textCost.setText(model.getValueAt(i, 4).toString());
}
}
)
;

btnUpdate.addActionListener(new ActionListener(){
@Override
public void actionPerformed(ActionEvent e)
{
int i = table.getSelectedRow();
if(i >= 0)
```

```
{
model.setValueAt(textMedicalcode.getText(
), i, 0);
model.setValueAt(textMedicalname.getText(
), i, 1);
model.setValueAt(textService_description.getText(), i, 2);
model.setValueAt(textQuantity_available.getText(), i, 3);
model.setValueAt(textCost.getText(), i, 4);
} else
{
System.out.println("Update Error");
}
}
}
)
;
sel_item.addActionListener(new ActionListener(){
   @Override
public void actionPerformed(ActionEvent ae)

{
int i=0,total_amount=0,sum=525;
int []a =new int[args.length];
if(i<args.length)
```

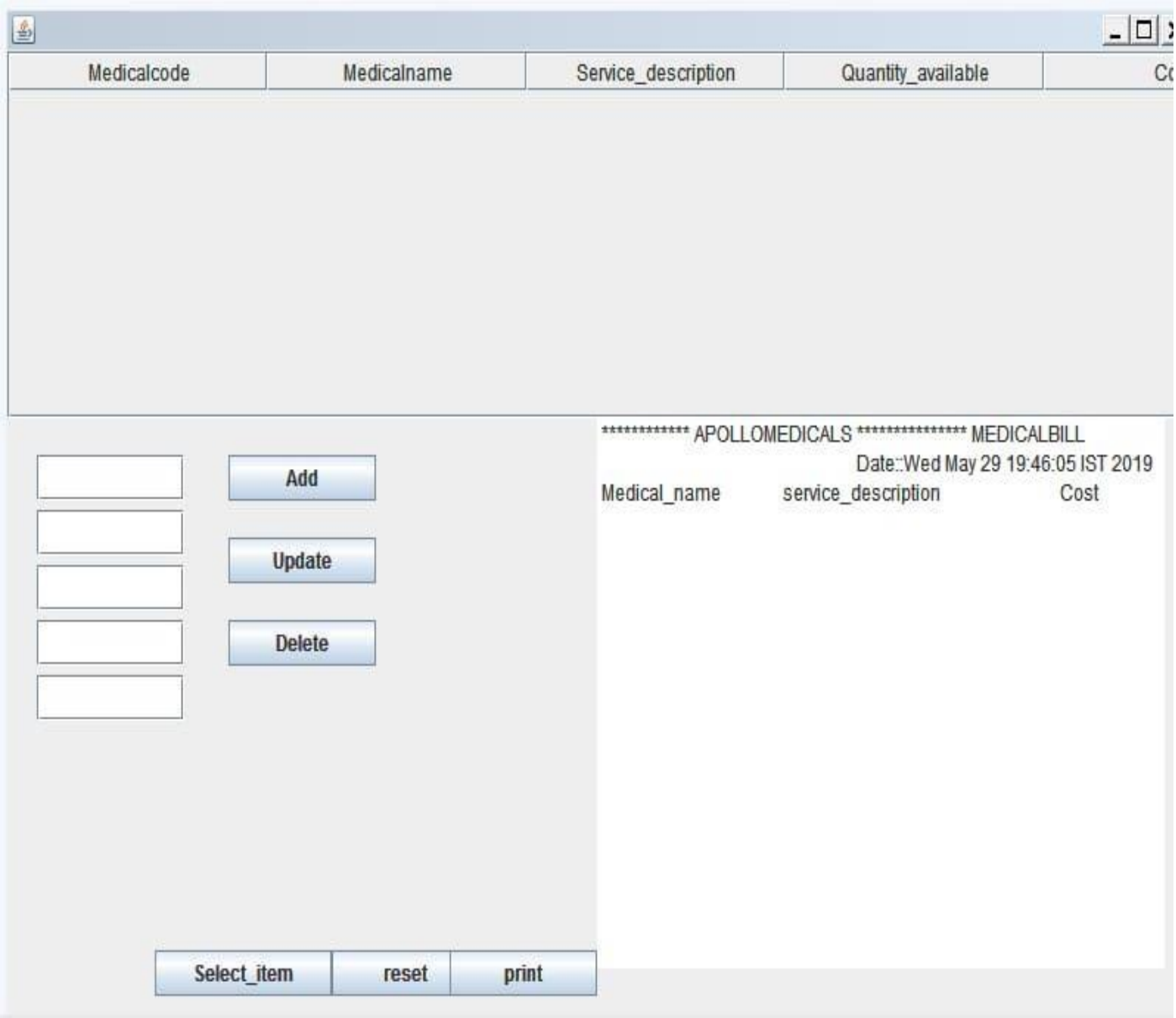
```
{
int t=Integer.parseInt(args[0]);
int q=Integer.parseInt(textCost.getText());
a[i]=t*q;
billarea.setText(billarea.getText()+"\n\n"+" 2   "+ textMedicalname.getText() +"
"+
textService_description.getText() +"                    " +
a[i]); i++;
}
if(i==args.length-1)
{
for(int p=0;p<args.length;p++)
{
sum=sum+a[p];
}
billarea.setText(billarea.getText()+"\n\n\n"+"\t\t"+"        TOTAL_AMOUNT :"+
sum
+"\n");
}
}
}
)
;
res.addActionListener(new ActionListener()
{
```

```
public void actionPerformed(ActionEvent e)
{
billarea.setText(" ********** APOLLOMEDICALS ********** MEDICAL BILL");
billarea.setText(billarea.getText()+"\n"+"\t\t"+ "Date::"+date+"\n");
billarea.setText(billarea.getText()+" MEDICAL_NAME
                                    "+ "SERVICE_DESCRIPTION
                                    "+ " COST"+"\n");
}

}
)
;
prin.addActionListener(new
ActionListener(){ public void
actionPerformed(ActionEvent ae){ try
{
   billarea.setText(billarea.getText()+"\n\n\n\n"+""+"       The good physician treats the disease the great physician treats \n\t the patient who has the disease \n\t**********THANK YOU**********");
billarea.print();
}
catch(Exception e)
{
}
}
```

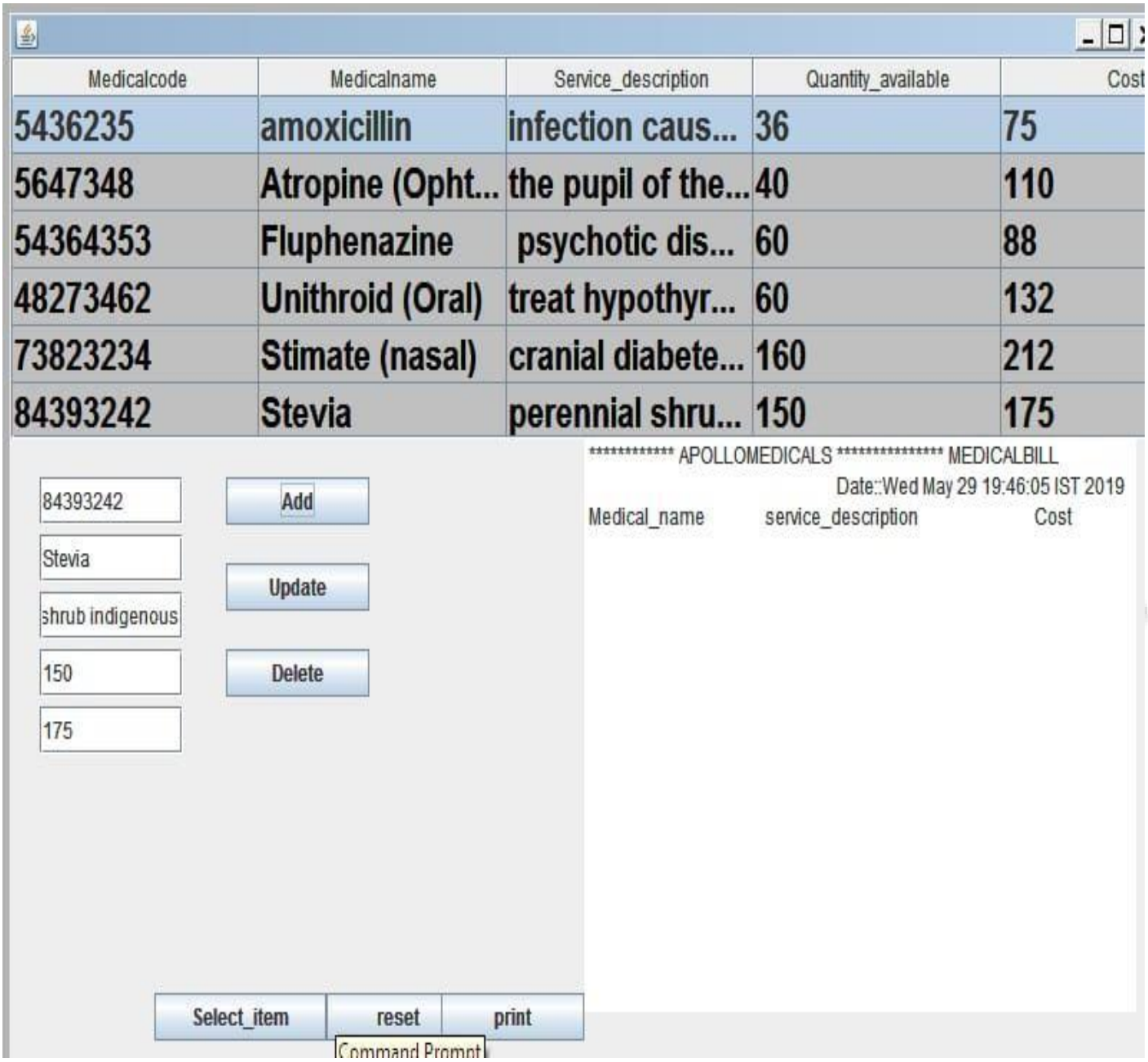
```
}
)
;
frame.setSize(1000,1000);
frame.setLocationRelativeTo(null);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOS
E); frame.setVisible(true);
}
}
```

Initially as the user clicks the continue button he encounters with the below frame.



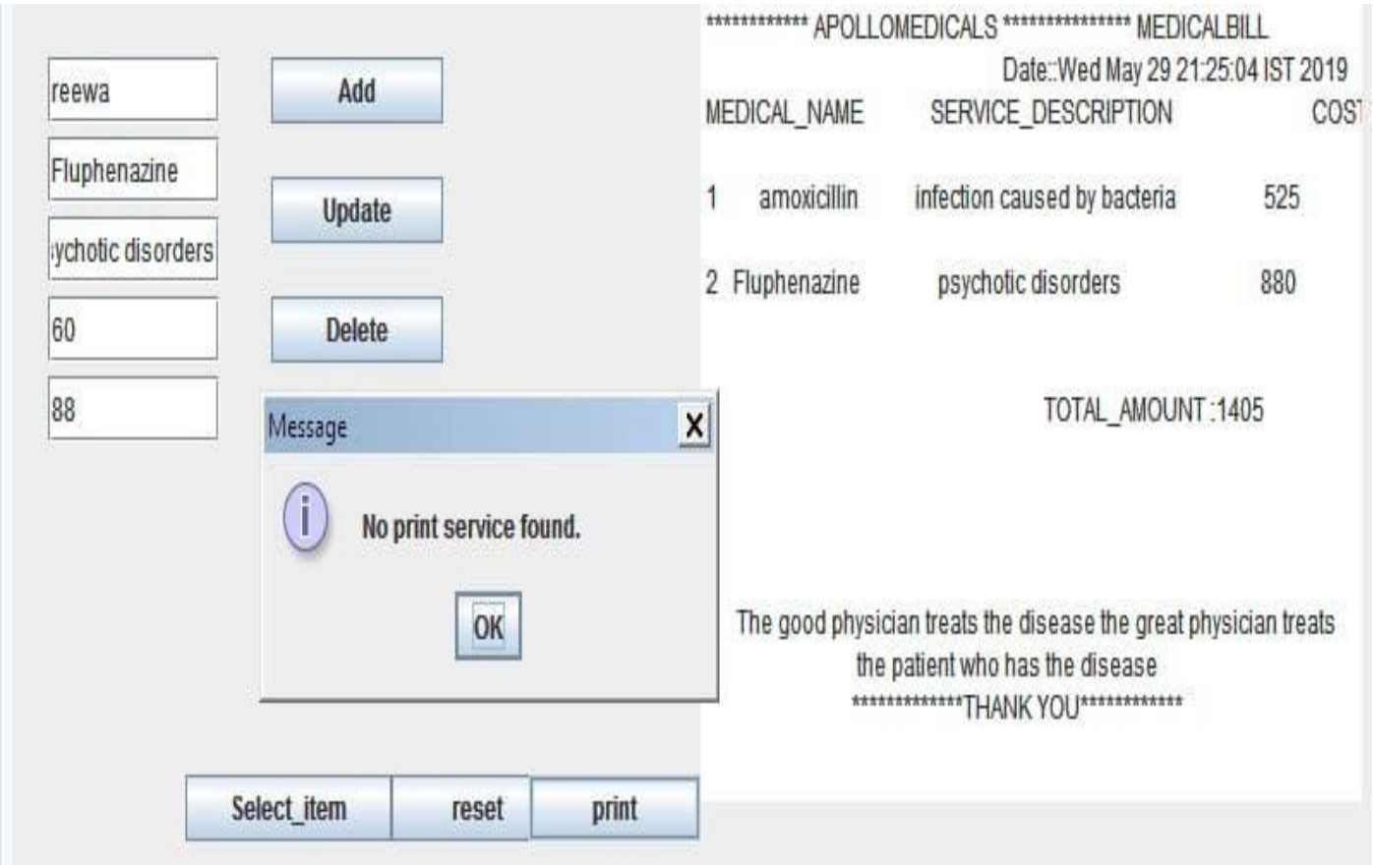
Then the adds the medicine details using the add button.

Then the frame looks like



As user selects the items that he wants to bill using the select item and is placed in the text area which represents the bill and that bill is printed using the print button.

As the user selects the item and clicks on the print button it starts to print the bill if a printer is connected or else it is saves as a pdf.



If there is no printer connected then it is saved in the pdf form.