

Programme Name: **BCS hons**

Course Code: **CSC 1513**

Course Name: **PROGRAMING FUNDAMENTALS**

**Programming Individual Project Work**

Date of Submission: **6/14/2020**

**Submitted By: Submitted To:**

Student Name: **Dipesh Tha Shrestha** Faculty Name: **PRAKASH CHANDRA**

IUKL ID: **041902900028**  Department: **LMS**

Semester: **Second**

Intake: **Sept 2019**

**Pharmacy administration system**

Pharmacy administration system is a clinical shop system developed in NetBeans utilizing java programs language. This system supplies you the control over your drug store in the manner that what you can add or get rid of from the pharma store.

**INTRODUCTION**

The main aim of the project is the management of the database of the pharmaceutical shop. This project is insight into the design and implementation of a Pharmacy Management System. This is done by creating a database of the available medicines in the shop. The primary aim of pharmacy management system is to improve accuracy and enhance safety and efficiency in the pharmaceutical store. The aim of this project is to develop software for the effective management of a pharmaceutical store. We have developed this software for ensuring effective policing by providing statistics of the drugs in stock. The database is then connected to the main program by using interconnection of the Visual Basic program and the database already created. Pharmacy management system is useful to maintain correct database by providing an option to update the drugs in stock. This is pharmacy management system; it is used to manage most pharmacy related activities in the pharmacy.  
  
Pharmacy management system is a management system that is designed to improve accuracy and to enhance safety and efficiency in the pharmaceutical store. This program can be used in any pharmaceutical shops having a database to maintain. It is a computer based system which helps the Pharmacist to improve inventory management, cost, medical safety etc. The software used can generate reports, as per the user’s requirements. Using this pharmacy management system user is also able to generate report within a specified period of time. The system allows the user to enter a manufacturing and expiry date for a particular product or drug during opening stock and sales transaction. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier. The system will also give report showing the list of products expiry after a specified date before the product eventually expires. The system services and goals are established by consultation with system user. It also involves manual entry upon arrival of new batches of drugs and upon drug movement out of the pharmacy for a certain period. Pharmacy management system is being build. Pharmacy management system is robust, integrated technology. every month, the pharmacist may want to generate report for the movement of drugs in and out of the pharmacy, getting information about the drugs e.g. expiry date, date purchased, number of drug type left, location of a drug in the pharmacy. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. This pharmacy management system is user friendly.

**Description on the topic**

This program can be used in any pharmaceutical shops having a database to maintain. The software used can generate reports, as per the user’s requirements. The software can print invoices, bills, receipts etc. It can also maintain the record of supplies sent in by the supplier. Here, the admin who are handling the organization will be responsible to manage the record of the employee. Each employee will be given with a separate username and password.

**Problem Definition**

The aim of the project is to create an effective software to help the pharmacist to maintain the records of the medicines, handle user details, generate invoice, check and renew validity and provide a scope of communication between users by using inbuilt messaging system. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. This pharmacy management system is user friendly.

**Objectives**

It is the user friendly application for Pharmacist which reduces the burden and helps to manage all sections of Pharmacy like Medicine management and Billing etc., which improve the processing efficiency. This will enhance the efficiency of clinical work and ease patient’s convenience, bearing in mind that in Ethiopia is heading towards pharmaceutical care of patients. It deals with the automating tasks of maintaining of Bills. In Pharmacy, Billing management is the key process. In addition, Pharmacy management system will be able to process drug prescription with ease. Including safe data store about medicine as well as fast searching, delete and update of medicines. The pharmacy management system is built for the sake of ensuring effective and clear data saving and manipulating as well as neat work on the pharmacy medical products. The pharmacy management system is easy for use so the user can do pharmacy actions without ambiguities. This refers the pharmacy management system project highly minimize time and resource by which, searching the medicine data you can get the data in quickest time. The main Objectives of the PMS is making the pharmacy organizations computerized by creating neat work through minimizing or eliminating wasting of time as well as removing the resources such as papers for data saving since know a days is paper based, decrease malfunctioned works on the medical usage by giving correct information on each medicine. This system is also helpful to improve the efficiency of the system by ensuring effective monitoring of services and activities. A summarized list of drugs dispensed to patient can be viewed for monitoring purposes.

**Primary objectiv**e

•To gain practical experience by modeling a software based on real world problem.

•To understand how to work on Java.

**Secondary objective**

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

**PROBLEM IN THE EXISTING PHARMACY SYSTEM**  
  
  
At present, manual system is being utilized in the pharmacy. It requires the pharmacist to manually monitor each drug that is available in the pharmacy. Pharmacy management has kept paper record in filing cabinets. Managing a very large pharmacy with records on papers will be tedious and difficult to keep track of inventories with regards to the drugs in the store, expiry date, quantity of drugs available based on the categories and their functions. This implies that these services will be manually completed by the pharmacist. This usually leads to mistakes as the workload of the pharmacist increases. This system also ensures that there exists a level of restricted access based on functionality and role. This system also provides optimal drug inventory management by monitoring the drug movement in the pharmacy. Significant amount of time is allocated for writing the order as the pharmacist needs to go through the stock balance and make rough estimate of the amount to order based on Figures. The system will not be able to handle drug prescription, drug to drug interaction.

CONCLUSIONS AND FUTURE SCOPE

Detailed information gathering has to be done. Without that the purpose for using the software won’t be satisfied properly. However, it can give good profits in the long run. Implementing the software requires change in the business practices. Efficient organization of all knowledge is the analysis company and easy analysis access and retrieval of information is possible. In this project we can also include BAR CODE facility using the bar code reader, which will detect the expiry date and the other information about the related medicines. Company using this software will always be able to plan in future and always be aware of their financial position in the market. It leads to ease in functioning of business processes. The project can be made more robust by including biometric verification. There is also a scope to expand by implementing newer technologies like cloud etcetera.

The project have following functionalities: -

* **Maintaining and displaying the medicine inventory in a file**

Image from an application showing an inventory system which is capable of displaying complete inventory information,

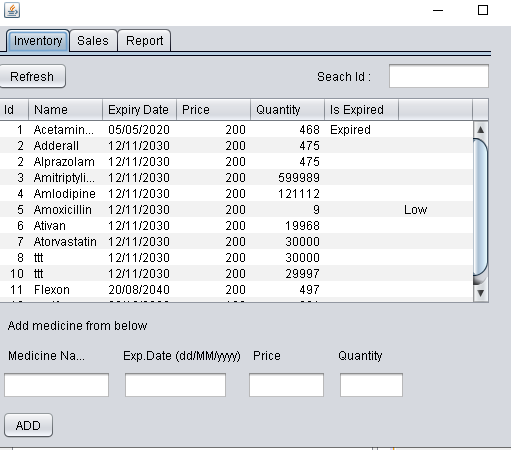
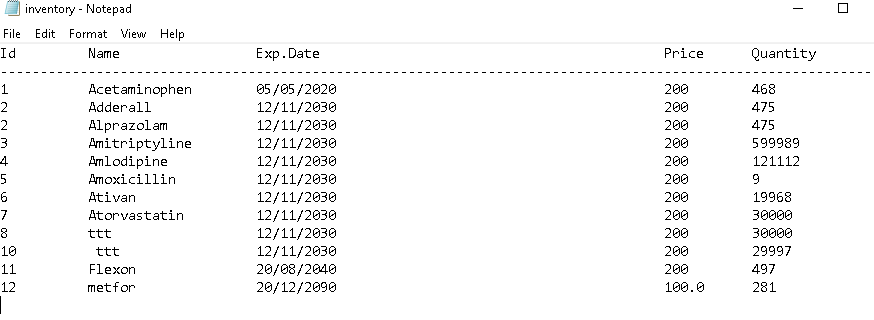
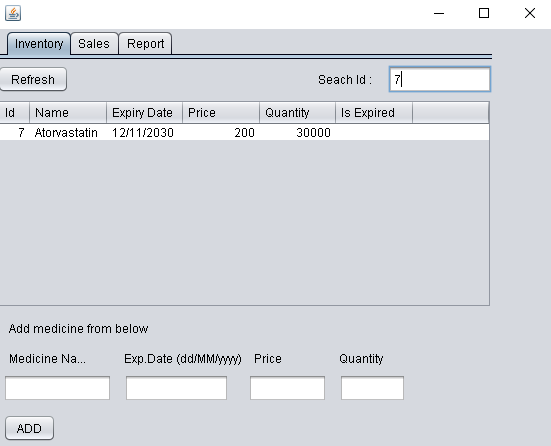


Image of a file where inventory is recorded,



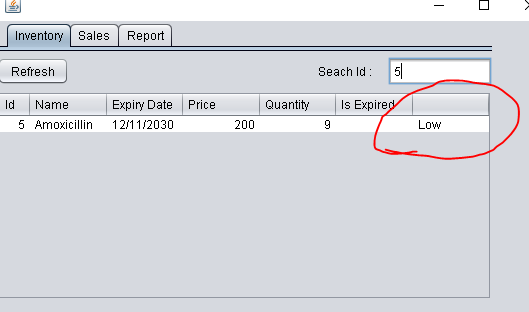
* **Searching inventory based on a unique medicine id**,

Image showing search functionality

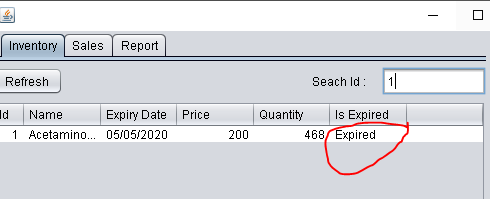


* **Notify when**

1. **The stock is below threshold**

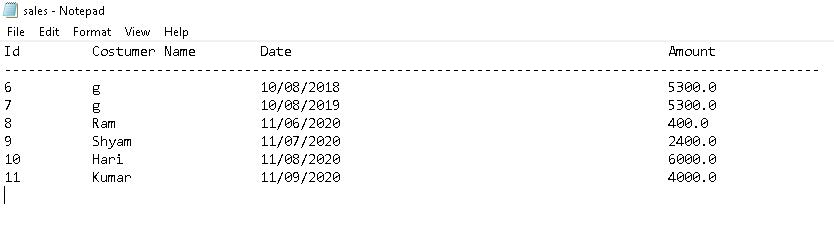


1. **When Mediceine is expired.**



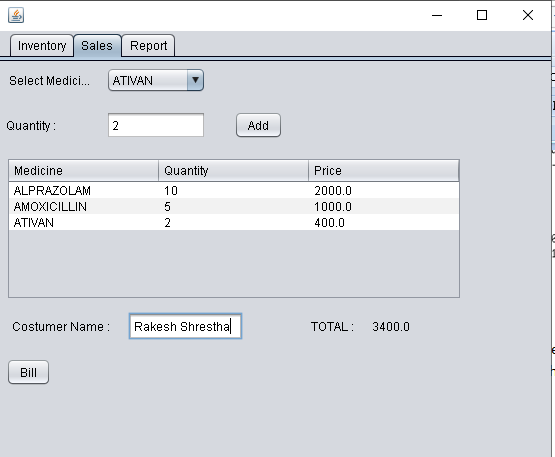
* **Maintaining date wise sales in a file**

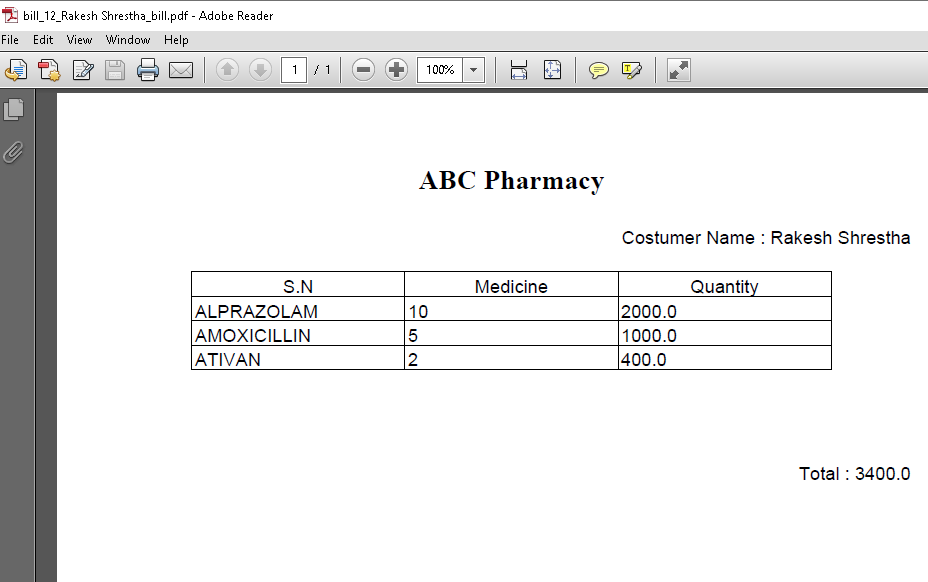
Image showing sales record in a file names sales.txt.



* **Generate detailed billing after the given costumer name.**

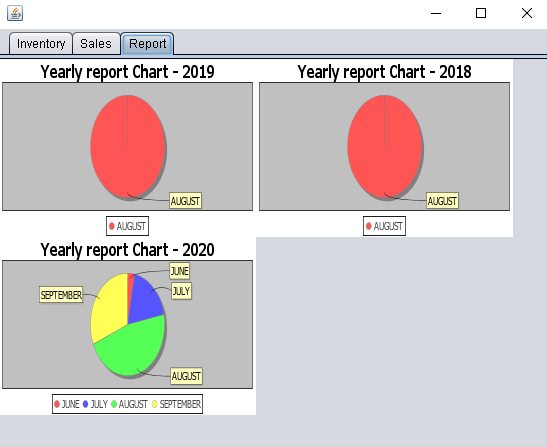
The following image shows the billing or sales section in an application.



Detailed bill printed in the form of .pdf file after clicking bill button.

* **Show Info graphic view of sales in the form of chart.**

A report is generated while clicked in the report tab; this report shows the yearly sales and month wise total sales of the Pharmacy.



**Source Code**

1. **PharmacyUI.java**

import java.awt.FlowLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.DefaultComboBoxModel;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JLabel;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableColumnModel;

import org.jfree.chart.ChartFactory;

import org.jfree.chart.ChartPanel;

import org.jfree.chart.JFreeChart;

import org.jfree.data.general.DefaultPieDataset;

import java.util.Date;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Element;

import com.itextpdf.text.Font;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.Phrase;

import com.itextpdf.text.pdf.PdfPCell;

import com.itextpdf.text.pdf.PdfPTable;

import com.itextpdf.text.pdf.PdfWriter;

import java.awt.GridLayout;

import java.io.FileOutputStream;

import java.time.Month;

import java.util.HashMap;

import java.util.Set;

import pharmacy.Sales;

import pharmacy.Sales.Bill;

public final class PharmacyUI extends javax.swing.JFrame {

public int threshold = 20;

public PharmacyUI() {

initComponents();

try {

startApplication();

} catch (ParseException | FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jTabbedPane1 = new javax.swing.JTabbedPane();

jPanel2 = new javax.swing.JPanel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

jTextField1 = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jTextField2 = new javax.swing.JTextField();

jTextField3 = new javax.swing.JTextField();

jTextField4 = new javax.swing.JTextField();

jTextField5 = new javax.swing.JTextField();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jButton2 = new javax.swing.JButton();

jPanel3 = new javax.swing.JPanel();

jTextField6 = new javax.swing.JTextField();

jLabel7 = new javax.swing.JLabel();

jComboBox1 = new javax.swing.JComboBox<>();

jLabel8 = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jTextField7 = new javax.swing.JTextField();

jScrollPane2 = new javax.swing.JScrollPane();

jTable2 = new javax.swing.JTable();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jLabel10 = new javax.swing.JLabel();

jLabel11 = new javax.swing.JLabel();

jLabel12 = new javax.swing.JLabel();

jPanel4 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

panel1 = new java.awt.Panel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jTabbedPane1.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusGained(java.awt.event.FocusEvent evt) {

jTabbedPane1FocusGained(evt);

}

});

jTabbedPane1.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

jTabbedPane1MouseClicked(evt);

}

});

jTable1.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Id", "Name", "Expiry Date", "Price", "Quantity", "Is Expired", ""

}

) {

Class[] types = new Class [] {

java.lang.Integer.class, java.lang.String.class, java.lang.String.class, java.lang.Double.class, java.lang.Integer.class, java.lang.Object.class, java.lang.Object.class

};

public Class getColumnClass(int columnIndex) {

return types [columnIndex];

}

});

jScrollPane1.setViewportView(jTable1);

jTextField1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jTextField1ActionPerformed(evt);

}

});

jButton1.setText("ADD");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jLabel1.setText(" Add medicine from below");

jLabel2.setText(" Medicine Name");

jLabel3.setText("Exp.Date (dd/MM/yyyy)");

jLabel4.setText(" Price");

jLabel5.setText("Quantity");

jLabel6.setText("Seach Id :");

jButton2.setText("Refresh");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()

.addComponent(jButton2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 64, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 104, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT\_SIZE, 551, Short.MAX\_VALUE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jButton1)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 165, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 109, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 89, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 105, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, 79, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 58, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel5)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, 67, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel2Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel6)

.addComponent(jButton2))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 209, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 28, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel3)

.addComponent(jLabel4)

.addComponent(jLabel5))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField5, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(jButton1)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

jTabbedPane1.addTab("Inventory", jPanel2);

jLabel7.setText(" Costumer Name :");

jLabel8.setText(" Select Medicine");

jLabel9.setText("Quantity :");

jTextField7.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusLost(java.awt.event.FocusEvent evt) {

jTextField7FocusLost(evt);

}

});

jTable2.setModel(new javax.swing.table.DefaultTableModel(

new Object [][] {

},

new String [] {

"Medicine", "Quantity", "Price", ""

}

) {

boolean[] canEdit = new boolean [] {

false, false, true, true

};

public boolean isCellEditable(int rowIndex, int columnIndex) {

return canEdit [columnIndex];

}

});

jScrollPane2.setViewportView(jTable2);

jButton3.setText("Bill");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jButton4.setText("Add");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jLabel10.setText("TOTAL :");

javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED\_SIZE, 88, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED\_SIZE, 71, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jComboBox1, 0, 100, Short.MAX\_VALUE)

.addComponent(jTextField7))

.addGap(28, 28, 28)

.addComponent(jButton4))

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 116, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, 115, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(68, 68, 68)

.addComponent(jLabel10)

.addGap(18, 18, 18)

.addComponent(jLabel11, javax.swing.GroupLayout.PREFERRED\_SIZE, 73, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(jPanel3Layout.createSequentialGroup()

.addComponent(jButton3)

.addGap(52, 52, 52)

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 166, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(89, Short.MAX\_VALUE))

);

jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel8))

.addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel9)

.addComponent(jTextField7, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton4))

.addGap(18, 18, 18)

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 143, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jTextField6, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jLabel10)

.addComponent(jLabel11))

.addGap(18, 18, 18)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton3)

.addComponent(jLabel12, javax.swing.GroupLayout.PREFERRED\_SIZE, 23, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(75, Short.MAX\_VALUE))

);

jTabbedPane1.addTab("Sales", jPanel3);

jPanel4.addFocusListener(new java.awt.event.FocusAdapter() {

public void focusGained(java.awt.event.FocusEvent evt) {

jPanel4FocusGained(evt);

}

});

javax.swing.GroupLayout panel1Layout = new javax.swing.GroupLayout(panel1);

panel1.setLayout(panel1Layout);

panel1Layout.setHorizontalGroup(

panel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 514, Short.MAX\_VALUE)

);

panel1Layout.setVerticalGroup(

panel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 356, Short.MAX\_VALUE)

);

jScrollPane3.setViewportView(panel1);

javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);

jPanel4.setLayout(jPanel4Layout);

jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 516, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 35, Short.MAX\_VALUE))

);

jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel4Layout.createSequentialGroup()

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 358, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(0, 22, Short.MAX\_VALUE))

);

jTabbedPane1.addTab("Report", jPanel4);

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTabbedPane1)

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jTabbedPane1)

);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

);

pack();

}// </editor-fold>

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

try {

File obj = new File("inventory.txt");

Scanner data = new Scanner(obj);

int lineCount = 0;

String searchString = evt.getActionCommand();

Object[] row = new Object[7];

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

while (data.hasNextLine()) {

String displayData = data.nextLine();

if (lineCount > 1) {

if ("".equals(searchString)) {

row[0] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[0]);

row[1] = displayData.trim().replaceAll("\\s+", " ").split(" ")[1];

row[2] = displayData.trim().replaceAll("\\s+", " ").split(" ")[2];

row[3] = Float.parseFloat(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]);

row[4] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]);

if (new SimpleDateFormat("dd/MM/yyyy").parse(displayData.trim().replaceAll("\\s+", " ").split(" ")[2]).before(new Date())) {

row[5] = "Expired";

} else {

row[5] = "";

}

if (Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]) < threshold) {

row[6] = "Low";

} else {

row[6] = "";

}

model.addRow(row);

} else if (Integer.parseInt(searchString) == Integer.parseInt(displayData.split(" ")[0])) {

row[0] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[0]);

row[1] = displayData.trim().replaceAll("\\s+", " ").split(" ")[1];

row[2] = displayData.trim().replaceAll("\\s+", " ").split(" ")[2];

row[3] = Float.parseFloat(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]);

row[4] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]);

if (new SimpleDateFormat("dd/MM/yyyy").parse(displayData.trim().replaceAll("\\s+", " ").split(" ")[2]).before(new Date())) {

row[5] = "Expired";

} else {

row[5] = "";

}

if (Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]) < threshold) {

row[6] = "Low";

} else {

row[6] = "";

}

model.addRow(row);

}

}

lineCount++;

}

} catch (FileNotFoundException | ParseException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

Pharmacy.Inventory inv = new Pharmacy.Inventory();

Pharmacy.Inventory.Medicine med = inv.new Medicine();

String last = null, line;

BufferedReader input;

try {

input = new BufferedReader(new FileReader("inventory.txt"));

while ((line = input.readLine()) != null) {

last = line;

}

med.id = Integer.parseInt(last.split(" ")[0]) + 1;

} catch (FileNotFoundException e) {

File file = new File("inventory.txt");

if (!file.exists()) {

FileWriter writer = null;

try {

file.createNewFile();

writer = new FileWriter("inventory.txt");

writer.write(String.format("%-10s %-20s %-50s %-10s %-10s \n", "Id", "Name", "Exp.Date", "Price",

"Quantity"));

writer.write(String.format("%-150s \n",

"------------------------------------------------------------------------------------------------------"));

writer.close();

} catch (IOException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

writer.close();

} catch (IOException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

med.id = 1;

} catch (IOException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

med.medcineName = jTextField2.getText();

med.expiryDate = jTextField3.getText();

med.price = Float.parseFloat(jTextField4.getText());

med.quantity = Integer.parseInt(jTextField5.getText());

try {

inv.saveData(med);

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

Object[] row = {med.id, med.medcineName, med.expiryDate, med.price, med.quantity};

model.addRow(row);

} catch (IOException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

try {

startApplication();

} catch (ParseException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

} catch (FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Document document = new Document();

try {

Sales sales = new Sales();

Sales.Bill bill = sales.new Bill();

bill.cos\_name = jTextField6.getText();

bill.date = new SimpleDateFormat("dd/MM/yyyy").format(new Date());

bill.amount = Float.parseFloat(jLabel11.getText());

HashMap<Integer, Integer> medList = new HashMap<>();

DefaultTableModel model = (DefaultTableModel) jTable2.getModel();

for (int i = 0; i < model.getRowCount(); i++) {

medList.put(Integer.parseInt(model.getValueAt(i, 3).toString()), Integer.parseInt(model.getValueAt(i, 1).toString()));

}

int billId = sales.getBillId(bill, medList);

PdfWriter.getInstance(document, new FileOutputStream("bill/bill\_" + billId + "\_" + jTextField6.getText() + "\_bill.pdf"));

document.open();

document.addTitle("bill/bill\_" + billId + "\_" + jTextField6.getText() + "\_bill.pdf");

Paragraph pharmacyName = new Paragraph("ABC Pharmacy", new Font(Font.FontFamily.TIMES\_ROMAN, 18,

Font.BOLD));

pharmacyName.setAlignment(Element.ALIGN\_CENTER);

document.newPage();

document.add(pharmacyName);

document.add(new Paragraph(" "));

Paragraph costumerName = new Paragraph("Costumer Name : " + jTextField6.getText());

costumerName.setAlignment(Element.ALIGN\_RIGHT);

document.add(costumerName);

document.add(new Paragraph(" "));

PdfPTable table = new PdfPTable(3);

PdfPCell c1 = new PdfPCell(new Phrase("S.N"));

c1.setHorizontalAlignment(Element.ALIGN\_CENTER);

table.addCell(c1);

c1 = new PdfPCell(new Phrase("Medicine"));

c1.setHorizontalAlignment(Element.ALIGN\_CENTER);

table.addCell(c1);

c1 = new PdfPCell(new Phrase("Quantity"));

c1.setHorizontalAlignment(Element.ALIGN\_CENTER);

table.addCell(c1);

table.setHeaderRows(1);

for (int i = 0; i < model.getRowCount(); i++) {

for (int j = 0; j < 3; j++) {

table.addCell(model.getValueAt(i, j).toString());

}

}

document.add(table);

document.add(new Paragraph(" "));

document.add(new Paragraph(" "));

document.add(new Paragraph(" "));

Paragraph total = new Paragraph("Total : " + jLabel11.getText());

total.setAlignment(Element.ALIGN\_RIGHT);

document.add(total);

document.close();

model.setRowCount(0);

jTextField7.setText("");

jTextField6.setText("");

jLabel12.setText("Bill Printed!!!!");

jLabel1.setText("");

} catch (DocumentException | FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jTabbedPane1FocusGained(java.awt.event.FocusEvent evt) {

// TODO add your handling code here:

}

private void jPanel4FocusGained(java.awt.event.FocusEvent evt) {

try {

createChart();

} catch (FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

} catch (ParseException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

private void jTabbedPane1MouseClicked(java.awt.event.MouseEvent evt) {

if (jTabbedPane1.getSelectedIndex() == 2) {

try {

createChart();

} catch (FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

} catch (ParseException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

} else if (jTabbedPane1.getSelectedIndex() == 1) {

try {

loadComboBox();

} catch (FileNotFoundException ex) {

Logger.getLogger(PharmacyUI.class.getName()).log(Level.SEVERE, null, ex);

}

}

}

float total = 0;

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Item item = (Item) jComboBox1.getSelectedItem();

int quantity = Integer.parseInt(jTextField7.getText());

if (item.quantity < quantity) {

JDialog d = new JDialog(this, "Dialog Example", true);

d.setLayout(new FlowLayout());

JButton b = new JButton("OK");

b.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

d.setVisible(false);

}

});

d.add(new JLabel("Only " + item.quantity + " pieces available!!!"));

jTextField7.setText(Integer.toString(item.quantity));

d.add(b);

d.setSize(300, 100);

d.setVisible(true);

} else {

DefaultTableModel model = (DefaultTableModel) jTable2.getModel();

Object[] row = new Object[4];

row[0] = item.description;

row[1] = quantity;

row[2] = (item.price) \* quantity;

row[3] = item.id;

model.addRow(row);

TableColumnModel tc = jTable2.getColumnModel();

tc.getColumn(3).setMinWidth(0);

tc.getColumn(3).setMaxWidth(0);

item.quantity = item.quantity - quantity;

if (item.quantity < 1) {

jComboBox1.removeItem(item);

}

total = (quantity \* (item.price)) + total;

jLabel11.setText(Float.toString(total));

}

}

private void jTextField7FocusLost(java.awt.event.FocusEvent evt) {

String text = jTextField7.getText();

int num;

try {

num = Integer.parseInt(text);

} catch (NumberFormatException e) {

jTextField7.setText("");

}

}

public static void main(String args[]) throws ParseException, FileNotFoundException {

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

public void run() {

new PharmacyUI().setVisible(true);

}

});

}

public void startApplication() throws ParseException, FileNotFoundException {

File obj = null;

obj = new File("inventory.txt");

Scanner data = new Scanner(obj);

int lineCount = 0;

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

TableColumnModel columnModel = jTable1.getColumnModel();

columnModel.getColumn(0).setMaxWidth(30);

Object[] row = new Object[7];

while (data.hasNextLine()) {

String displayData = data.nextLine();

if (lineCount > 1) {

row[0] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[0]);

row[1] = displayData.trim().replaceAll("\\s+", " ").split(" ")[1];

row[2] = displayData.trim().replaceAll("\\s+", " ").split(" ")[2];

row[3] = Float.parseFloat(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]);

row[4] = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]);

if (new SimpleDateFormat("dd/MM/yyyy").parse(displayData.trim().replaceAll("\\s+", " ").split(" ")[2]).before(new Date())) {

row[5] = "Expired";

} else {

row[5] = "";

}

if (Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]) < threshold) {

row[6] = "Low";

} else {

row[6] = "";

}

model.addRow(row);

}

lineCount++;

}

}

public void createChart() throws FileNotFoundException, ParseException {

panel1.removeAll();

File obj = null;

obj = new File("sales.txt");

Scanner data = new Scanner(obj);

int lineCount = 0;

HashMap<String, HashMap<String, Double>> yearlyData = new HashMap<>();

while (data.hasNextLine()) {

String displayData = data.nextLine();

if (lineCount > 1) {

String date = displayData.trim().replaceAll("\\s+", " ").split(" ")[2];

String year = date.split("/")[2];

String month = date.split("/")[1];

HashMap<String, Double> chartData = new HashMap<>();

if (yearlyData.containsKey(year)) {

chartData = yearlyData.get(year);

if (!chartData.containsKey(month)) {

chartData.put(month, Double.parseDouble(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]));

} else {

Double amount = chartData.get(month) + Double.parseDouble(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]);

chartData.put(month, amount);

}

yearlyData.put(year, chartData);

} else {

Double amount = Double.parseDouble(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]);

chartData.put(month, amount);

yearlyData.put(year, chartData);

}

}

lineCount++;

}

Set<String> key = yearlyData.keySet();

panel1.setLayout(new GridLayout(2,2));

int W =400, H = 200;

for (String kdata : key) {

DefaultPieDataset dataset = new DefaultPieDataset();

Set<String> monthKeySet = yearlyData.get(kdata).keySet();

for (String mkdata : monthKeySet) {

Month month = Month.of(Integer.parseInt(mkdata));

dataset.setValue(month.name(), yearlyData.get(kdata).get(mkdata));

}

JFreeChart chart = ChartFactory.createPieChart("Yearly report Chart - " + kdata, dataset, true,

true, false);

ChartPanel chartPanel = new ChartPanel(chart, W, H, W, H, W, H,

false, true, true, true, true, true);

panel1.add(chartPanel);

jScrollPane3.add(panel1);

}

}

public void loadComboBox() throws FileNotFoundException {

File obj = null;

obj = new File("inventory.txt");

Scanner data = new Scanner(obj);

int lineCount = 0;

DefaultComboBoxModel model = (DefaultComboBoxModel) jComboBox1.getModel();

model.removeAllElements();

while (data.hasNextLine()) {

String displayData = data.nextLine();

if (lineCount > 1) {

Item item = new Item(Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[0]), displayData.trim().replaceAll("\\s+", " ").split(" ")[1], Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]), Float.parseFloat(displayData.trim().replaceAll("\\s+", " ").split(" ")[3]));

model.addElement(item);

}

lineCount++;

}

}

class Item {

private int id;

private String description;

private int quantity;

private float price;

public Item(int id, String description, int quantity, float price) {

this.id = id;

this.description = description.toUpperCase();

this.quantity = quantity;

this.price = price;

}

public int getId() {

return id;

}

public String getDescription() {

return description;

}

public int getQuantity() {

return quantity;

}

@Override

public String toString() {

return description;

}

}

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JButton jButton3;

private javax.swing.JButton jButton4;

private javax.swing.JComboBox<String> jComboBox1;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel10;

private javax.swing.JLabel jLabel11;

private javax.swing.JLabel jLabel12;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JLabel jLabel4;

private javax.swing.JLabel jLabel5;

private javax.swing.JLabel jLabel6;

private javax.swing.JLabel jLabel7;

private javax.swing.JLabel jLabel8;

private javax.swing.JLabel jLabel9;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JTabbedPane jTabbedPane1;

private javax.swing.JTable jTable1;

private javax.swing.JTable jTable2;

private javax.swing.JTextField jTextField1;

private javax.swing.JTextField jTextField2;

private javax.swing.JTextField jTextField3;

private javax.swing.JTextField jTextField4;

private javax.swing.JTextField jTextField5;

private javax.swing.JTextField jTextField6;

private javax.swing.JTextField jTextField7;

private java.awt.Panel panel1;

}

1. **Inventory.java**

package Pharmacy;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStream;

import java.text.ParseException;

import java.util.HashMap;

import java.util.Scanner;

public class Inventory {

public class Medicine {

public int id;

public String medcineName;

public String expiryDate;

public float price;

public int quantity;

}

public String saveData(Medicine med) throws IOException {

String returnMessage = "Data Saved Successfully";

String last = null, line;

BufferedReader input;

try {

input = new BufferedReader(new FileReader("inventory.txt"));

while ((line = input.readLine()) != null) {

last = line;

}

med.id = Integer.parseInt(last.split(" ")[0]) + 1;

} catch (FileNotFoundException e) {

File file = new File("inventory.txt");

if (!file.exists()) {

file.createNewFile();

FileWriter writer = new FileWriter("inventory.txt");

writer.write(String.format("%-10s %-20s %-50s %-10s %-10s \n", "Id", "Name", "Exp.Date", "Price",

"Quantity"));

writer.write(String.format("%-150s \n",

"------------------------------------------------------------------------------------------------------"));

writer.close();

}

med.id = 1;

}

String data = String.format("%-10s %-20s %-50s %-10s %-10s \n", med.id, med.medcineName, med.expiryDate, med.price,

med.quantity);

OutputStream os = null;

try {

os = new FileOutputStream(new File("inventory.txt"), true);

os.write(data.getBytes(), 0, data.length());

} finally {

try {

os.close();

} catch (IOException e) {

returnMessage = "Couldn't save data!!";

}

}

return returnMessage;

}

public void decreaseInventory(HashMap<Integer, Integer> medList) throws FileNotFoundException, ParseException, IOException {

File obj = new File("inventory.txt");

File tempFile = new File("inventoryx.txt");

BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));

Scanner data = new Scanner(obj);

int lineCount = 0;

Object[] row = new Object[7];

while (data.hasNextLine()) {

String displayData = data.nextLine();

if (lineCount > 1) {

row[0] = Integer.parseInt(displayData.split(" ")[0]);

if (medList.containsKey(row[0])) {

int quantity = Integer.parseInt(displayData.trim().replaceAll("\\s+", " ").split(" ")[4]);

int newQuantity = quantity - medList.get(Integer.parseInt(displayData.split(" ")[0]));

System.out.println("Pharmacy.Inventory.decreaseInventory()--------" + newQuantity);

displayData = displayData.replace(Integer.toString(quantity), Integer.toString(newQuantity));

System.out.println(displayData);

}

}

writer.write(displayData + "\n");

lineCount++;

}

data.close();

writer.close();

obj.delete();

tempFile.renameTo(obj);

}

}

1. **Sales.java**

package pharmacy;

import Pharmacy.Inventory;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.OutputStream;

import java.text.ParseException;

import java.util.HashMap;

import java.util.logging.Level;

import java.util.logging.Logger;

public class Sales {

public class Bill {

public int id;

public String cos\_name;

public String date;

public float amount;

}

public int getBillId(Bill bill,HashMap<Integer,Integer> medList) throws IOException, FileNotFoundException {

String last = null, line;

BufferedReader input;

try {

input = new BufferedReader(new FileReader("sales.txt"));

while ((line = input.readLine()) != null) {

last = line;

}

bill.id = Integer.parseInt(last.split(" ")[0]) + 1;

} catch (FileNotFoundException e) {

File file = new File("sales.txt");

if (!file.exists()) {

file.createNewFile();

FileWriter writer = new FileWriter("sales.txt");

writer.write(String.format("%-10s %-20s %-50s %-10s \n", "Id", "Costumer Name", "Date", "Amount"));

writer.write(String.format("%-150s \n",

"------------------------------------------------------------------------------------------------------"));

writer.close();

}

bill.id = 1;

}

String data = String.format("%-10s %-20s %-50s %-10s \n", bill.id, bill.cos\_name, bill.date, bill.amount);

OutputStream os = null;

try {

os = new FileOutputStream(new File("sales.txt"), true);

os.write(data.getBytes(), 0, data.length());

Inventory inventory = new Inventory();

inventory.decreaseInventory(medList);

}catch (ParseException ex) {

Logger.getLogger(Sales.class.getName()).log(Level.SEVERE, null, ex);

} finally {

try {

os.close();

} catch (IOException e) {

e.printStackTrace();

}

}

return bill.id;

}

}