Programming Fundamentals

Program Name: **BCS – IT**

Course Code: **CSC 1510**

Course Name: **Programming Fundamentals**

Assignment: **Fourth – Individual Project (Short Semester)**

Date of Submission: **13th June, 2020**

Submitted By:

Student Name: **Keshav Bhandari**

IUKL ID:

Semester: Second

Intake: September, 2019

Submitted To:

Faculty Name: **Prakash Chandra**

Department: BCS - IT

Programming Fundamentals

JAVA PROJECT ON PHARMACY MANAGEMENT SYSTEM

About:

Pharmacy management system is an information system for the pharmacy.

The project should have following functionalities:

- - Maintain the medicine inventory in a file.

-- Display the complete inventory information.

-- Search the inventory based on a unique medicine ID.

- - Notify when:

a) the stock of any medicine is below given threshold.

b) Medicine is expired.

- - Maintain date wise sales record in a file.

- - Generate detailed billing for a customer in file format.

The generated bill should be named after the given customer name.

- - Show an infographic view of sales in the form of graphs and charts.

NOTE: The inventory file and sales file should have properly defined file headings.

Programming Fundamentals

Main.java

```
import javax.swing.*;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) throws Exception {
        Inventory inventory = new Inventory();

        // To add new products into the inventory
        System.out.println("Do you want to add a product? (Y/N)");
        Scanner sc = new Scanner(System.in);
        String response = sc.nextLine();
        if (response.toUpperCase().equals("Y")) {
            inventory.addToInventory();
        }

        // To display the all details of inventory
        System.out.println("Do you want to display the inventory? (Y/N)");
        if (sc.nextLine().toUpperCase().equals("Y")) {
            Display_inventory.displayInventory();
        }

        // To search the medicine
        System.out.println("Do you want to search a product? (Y/N)");
        if (sc.nextLine().toUpperCase().equals("Y")) {
            Display_inventory displaySearch = new Display_inventory();
            displaySearch.search();
        }

        // To buy the medicine and to maintain the stock into the Pharmacy
        Sales sales = new Sales();
        System.out.println("Do you want to buy products? (Y/N)");
        if (sc.nextLine().toUpperCase().equals("Y")) {
            sales.buyProducts();
        }
        // To check the expired date of products and to check the stock below threshold
        Notification notification = new Notification();
        notification.CompareDate();
        notification.CompareStock();

        // To generate sales record for the user
        sales.generateSalesRecord();
        sales.generateBill();
```

```
        /* To generate bar chart */
        GraphCharts example = new GraphCharts("Bar Chart");
        example.setSize(1000, 500);
        example.setLocationRelativeTo(null);
        example.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        example.setVisible(true);
    }
}
```

Inventory.java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

// All about the inveentory in this section about maintenance of Pharmacy Management System
public class Inventory {
    static ArrayList<String> MedicineID = new ArrayList<>();
    static ArrayList<String> ProductName = new ArrayList<>();
    static ArrayList<String> Stock = new ArrayList<>();
    static ArrayList<String> Price = new ArrayList<>();
    static ArrayList<String> ExpiryDate = new ArrayList<>();
    static File Inventoryfile;

    // TO read the inventory
    public static void readFile() throws FileNotFoundException {
        Inventoryfile = new File("Medicine_Inventory.csv");
        Scanner inputFile = new Scanner(Inventoryfile);
        while (inputFile.hasNext()) {
            String line = inputFile.nextLine();
            String[] arr = line.split(",");
            MedicineID.add(arr[0]);
            ProductName.add(arr[1]);
            Stock.add(arr[2]);
            Price.add(arr[3]);
            ExpiryDate.add(arr[4]);
        }
    }

    // It helps to add medicine into the inventory
    public void addToInventory() throws IOException {

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the no of Products to add to the inventory:");
        int noOfProducts = sc.nextInt();
        String[] addedData = new String[noOfProducts];
        System.out.println("Enter the MedicineID,ProductName,Stock,Price,ExpiryDate:");
        for (int i = 0; i < noOfProducts; i++) {
            addedData[i] = sc.next();
        }
        Inventory.readFile();
        FileWriter out = new FileWriter(Inventoryfile);
```

```
        int cnt = 0;
        for (String i : MedicineID) {

            out.write(i + "," + ProductName.get(cnt) + "," + Stock.get(cnt) + "," + Price.get(cnt) + ","
                    + ExpiryDate.get(cnt) + "\n");
            cnt++;
        }
        for (int i = 0; i < noOfProducts; i++) {
            out.write(addedData[i] + "\n");
        }
        out.close();
    }

}
```

Display_inventory.java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

public class Display_inventory {
    static ArrayList<String> MedicineID = new ArrayList<>();
    static ArrayList<String> ProductName = new ArrayList<>();
    static ArrayList<String> Stock = new ArrayList<>();
    static ArrayList<String> Price = new ArrayList<>();
    static ArrayList<String> ExpiryDate = new ArrayList<>();

    // reads the inventory file
    public static void readFile() throws FileNotFoundException {
        File Inventoryfile = new File("Medicine_Inventory.csv");
        Scanner inputFile = new Scanner(Inventoryfile);
        while (inputFile.hasNext()) {
            String line = inputFile.nextLine();
            String[] arr = line.split(",");
            MedicineID.add(arr[0]);
            ProductName.add(arr[1]);
            Stock.add(arr[2]);
            Price.add(arr[3]);
            ExpiryDate.add(arr[4]);
        }
    }

    // Displaying the inventory
    public static void displayInventory() throws FileNotFoundException {
        Display_inventory.readFile();
        System.out.println("\t\t\t\tHere are the details of medicine in the Pharmacy:");

System.out.println("==========================================================
====================");
        int cnt = 0;
        for (String i : MedicineID) {

            System.out.println(i + ((i.length() > 5) ? "\t\t" : "\t\t\t") + ProductName.get(cnt)
                    + ((ProductName.get(cnt).length() < 8) ? "\t\t\t" : "\t\t") + Stock.get(cnt)
                    + ((Stock.get(cnt).length() > 8) ? "\t" : "\t\t\t\t\t") + Price.get(cnt)
                    + ((ExpiryDate.get(cnt).contains("ExpiryDate")) ? "\t\t" : "\t\t\t") +
ExpiryDate.get(cnt));
            cnt++;
        }
```

```
System.out.println("==========================================================
====================");
    }

    // Search the medicine with Unique ID
    public void search() {
        Scanner sc = new Scanner(System.in);
        System.out.println("=================Searching Here================");
        System.out.println("Enter the MedicineID of the medicine you want to search:");
        String MED_ID = sc.nextLine();

        int cnt = 0;
        int check = 0;
        for (String i : MedicineID) {
            if (i.equals(MED_ID)) {
                System.out.println("Medicine ID:\t\t" + i + "\nProduct Name:\t" +
ProductName.get(cnt)
                        + "\nStock amount:\t" + Stock.get(cnt) + "\nPrice:\t\t\t" + Price.get(cnt) +
"\nExpiry Date:\t"
                        + ExpiryDate.get(cnt));
                check *= 0;
                break;
            } else {
                check += 1;
            }
            cnt++;
        }

        if (check > 0) {
            System.out.println("Sorry, we don't have that item!!");
        }
    }

    public static ArrayList<String> getMedicineID() {
        return MedicineID;
    }

    public static ArrayList<String> getProductName() {
        return ProductName;
    }

    public static ArrayList<String> getStock() {
        return Stock;
    }
```

```
    public static ArrayList<String> getPrice() {
        return Price;
    }

    public static ArrayList<String> getExpiryDate() {
        return ExpiryDate;
    }
}
```

Notification.java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.text.*;
import java.util.ArrayList;
import java.util.Date;
import java.util.Scanner;

public class Notification {
    ArrayList<String> Stock = new ArrayList<>();
    ArrayList<String> ExpiryDate = new ArrayList<>();
    static int lineCountInventory, thresholdForStock;
    static ArrayList<String> ProductName = new ArrayList<>();

    public void readFile() throws FileNotFoundException {
        File Inventory_file = new File("Medicine_Inventory.csv");
        Scanner inputFile = new Scanner(Inventory_file);
        while (inputFile.hasNext()) {
            String line = inputFile.nextLine();
            lineCountInventory++;
            String[] arr = line.split(",");
            ProductName.add(arr[1]);
            Stock.add(arr[2]);
            if (arr[2].contains("limit")) {
                String[] array = arr[2].split("=");
                thresholdForStock = Integer.parseInt(array[1].substring(0, array[1].length() - 1));
            }
            ExpiryDate.add(arr[4]);
        }
    }

    public void CompareDate() throws Exception {
        readFile();
        int check = 0;
        String pattern = "yyyy/MM/dd";
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat(pattern);
        String date = simpleDateFormat.format(new Date());
        Date currentDate = simpleDateFormat.parse(date);
        Date[] dates = new Date[lineCountInventory - 1];
        for (int i = 1; i < lineCountInventory; i++) {
            dates[i - 1] = simpleDateFormat.parse(ExpiryDate.get(i));
        }
        System.out.println("**********Checking the Expiry
Date********** ");

        for (int i = 1; i < lineCountInventory; i++) {
```

```
            if (dates[i - 1].compareTo(currentDate) < 0) {
                System.out.println(ProductName.get(i) + " is expired!!");
                check++;
            }
        }
        if (check == 0) {
            System.out.println("None are expried");
        }

    }

    public void CompareStock() {
        int check = 0;
        System.out.println("**********************Checking Stock
Availability************************** ");
        for (int i = 1; i < lineCountInventory; i++) {
            if (Integer.parseInt(Stock.get(i)) < thresholdForStock) {
                System.out.println(ProductName.get(i) + " to be added!!");
                check++;
            }
        }
        if (check == 0) {
            System.out.println("Stock is fine.");
        }
    }

}
```

Sales.java

```
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

// It is all about sales.....................
public class Sales {
    static ArrayList<String> MedicineID = new ArrayList<>();
    static ArrayList<String> ProductName = new ArrayList<>();
    static ArrayList<String> Stock = new ArrayList<>();
    static ArrayList<String> Price = new ArrayList<>();
    static ArrayList<String> ExpiryDate = new ArrayList<>();
    static String CustName, TodayDate;
    static int TotalSales, noOfProducts;
    static String[] nameOfProduct;
    static int[] amount, individualPrice;

    public String readFile() throws IOException {
        File Sales = new File("Sales.csv");
        Scanner inputFile = new Scanner(Sales);
        String saveFile = "";
        while (inputFile.hasNext()) {
            saveFile += inputFile.nextLine() + "\n";
        }
        return saveFile;
    }

    public void readInventoryData() {
        MedicineID = Display_inventory.getMedicineID();
        ProductName = Display_inventory.getProductName();
        Stock = Display_inventory.getStock();
        Price = Display_inventory.getPrice();
        ExpiryDate = Display_inventory.getExpiryDate();
    }

    public void buyProducts() throws IOException {
        readInventoryData();
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter your name:");
        CustName = sc.next();
        System.out.println("Enter today's date:");
        TodayDate = sc.next();
        System.out.println("Enter the number of products to buy:");
```

```
noOfProducts = sc.nextInt();
nameOfProduct = new String[noOfProducts];
amount = new int[noOfProducts];
individualPrice = new int[noOfProducts];
for (int i = 0; i < noOfProducts; i++) {
    System.out.println("Enter the name of product " + (i + 1) + ":");
    nameOfProduct[i] = sc.next();
}
int cnt = 0;
int checkForStock = 0;

for (String i : ProductName) {
    int stock;
    for (int j = 0; j < noOfProducts; j++) {
        if (i.equalsIgnoreCase(nameOfProduct[j])) {
            System.out.println("Enter the amount of medicine you want for " + i + ":");
            amount[j] = sc.nextInt();
            stock = Integer.parseInt(Stock.get(cnt));
            stock -= amount[j];
            individualPrice[j] = Integer.parseInt(Price.get(cnt));
            TotalSales += amount[j] * individualPrice[j];
            if (stock > 0) {

                Stock.set(cnt, "" + stock);
                checkForStock *= 0;
            } else {
                checkForStock += 1;

            }

        }

    }
    cnt++;
}

if ((checkForStock > 0)) {
    System.out.println("Out of the stock for this product!!");
} else {
    System.out.println("Go for buying!!");
}

cnt = 0;
FileWriter out = new FileWriter("Medicine_Inventory.csv");
for (String i : MedicineID) {
```

```
            out.write(i + "," + ProductName.get(cnt) + "," + Stock.get(cnt) + "," + Price.get(cnt) + ","
                    + ExpiryDate.get(cnt) + "\n");
            cnt++;
        }

        out.close();
    }

    public void generateBill() throws IOException {
        FileWriter out = new FileWriter("Bills of "+CustName);
        out.write("Name:" + CustName + "\nDate:" + TodayDate +
"\n====================BILL==================");
        out.write("\nNo. ProductName\tQuantity\tPrice\tTotal\n");
        for (int i = 0; i < noOfProducts; i++) {
            out.write((i + 1) + "  " + nameOfProduct[i] + ((nameOfProduct[i].length() > 7) ? "\t" :
"\t\t") + amount[i]
                    + "\t\t\t" + individualPrice[i] + "\t\t" + (amount[i] * individualPrice[i]) + "\n");
        }
        out.write("\n=========================================");
        out.write("\nGrand Total\t\t\t\t\t" + TotalSales);
        out.close();

    }

    public void generateSalesRecord() throws IOException {
        File sales = new File("Sales.csv");
        String saveFile = readFile();
        FileWriter out = new FileWriter(sales);
        out.write(saveFile);
        out.write(TodayDate + "," + CustName + "," + TotalSales);
        out.close();

    }

}
```

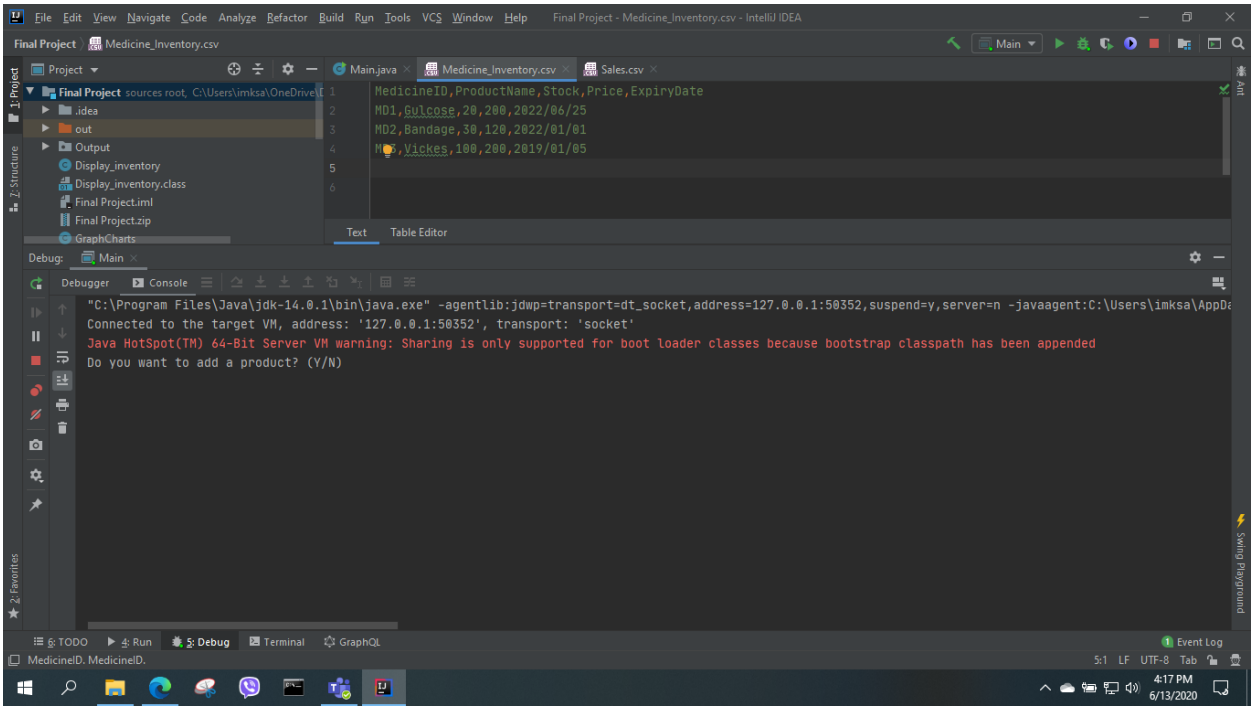
GraphCharts.java

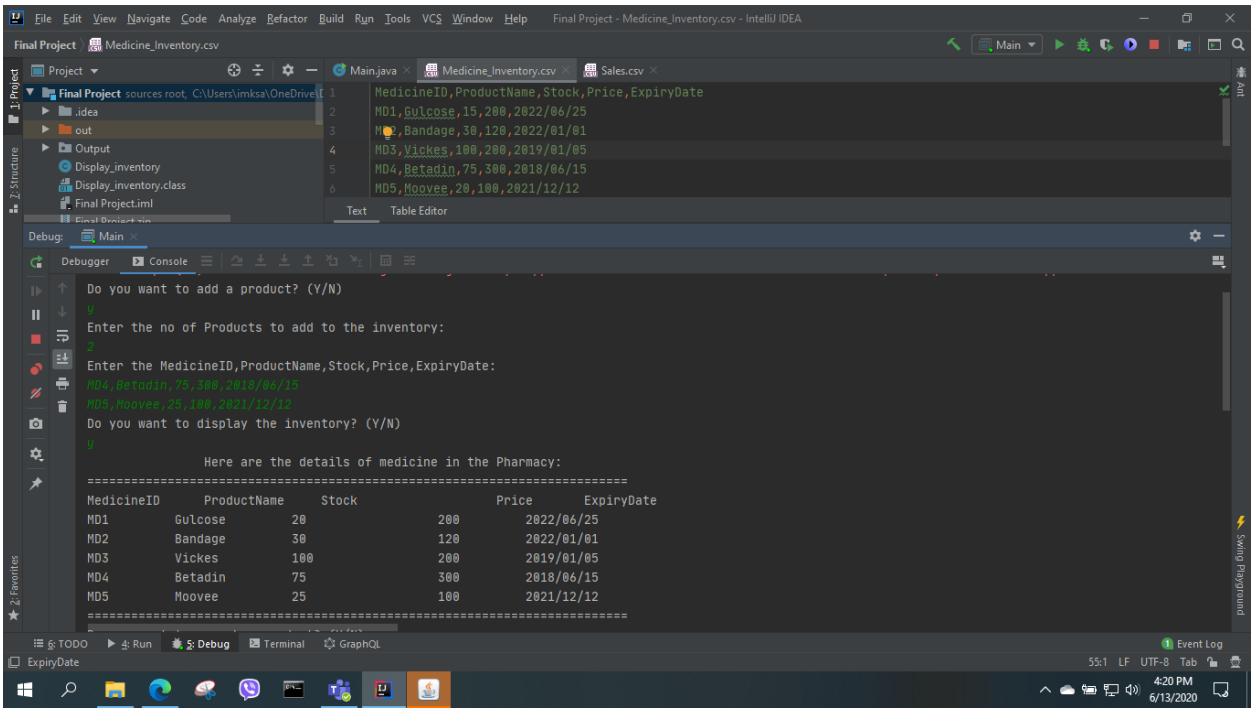
```
import javax.swing.*;

import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartPanel;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class GraphCharts extends JFrame {
   static int noOfSalesData;

   public GraphCharts(String appTitle) throws FileNotFoundException {
      super(appTitle);

      // Create Dataset
      CategoryDataset dataset = createDataset();

      // Create chart
      JFreeChart chart = ChartFactory.createBarChart("Sales Record ", // Chart Title
            "Fields", // Category axis
            "Sales in Rs", // Value axis
            dataset, PlotOrientation.VERTICAL, true, true, false);

      ChartPanel panel = new ChartPanel(chart);
      setContentPane(panel);
   }

   public String[][] scanningData_Sales() throws FileNotFoundException {
      File Sales = new File("Sales.csv");
      Scanner inputFile = new Scanner(Sales);
      Scanner input = new Scanner(Sales);
      while (inputFile.hasNext()) {
         inputFile.nextLine();
         noOfSalesData++;
      }
      String[][] SalesData = new String[noOfSalesData - 1][3];

      for (int i = 0; i < noOfSalesData; i++) {
         String line = input.nextLine();
         String[] arr = line.split(",");
```

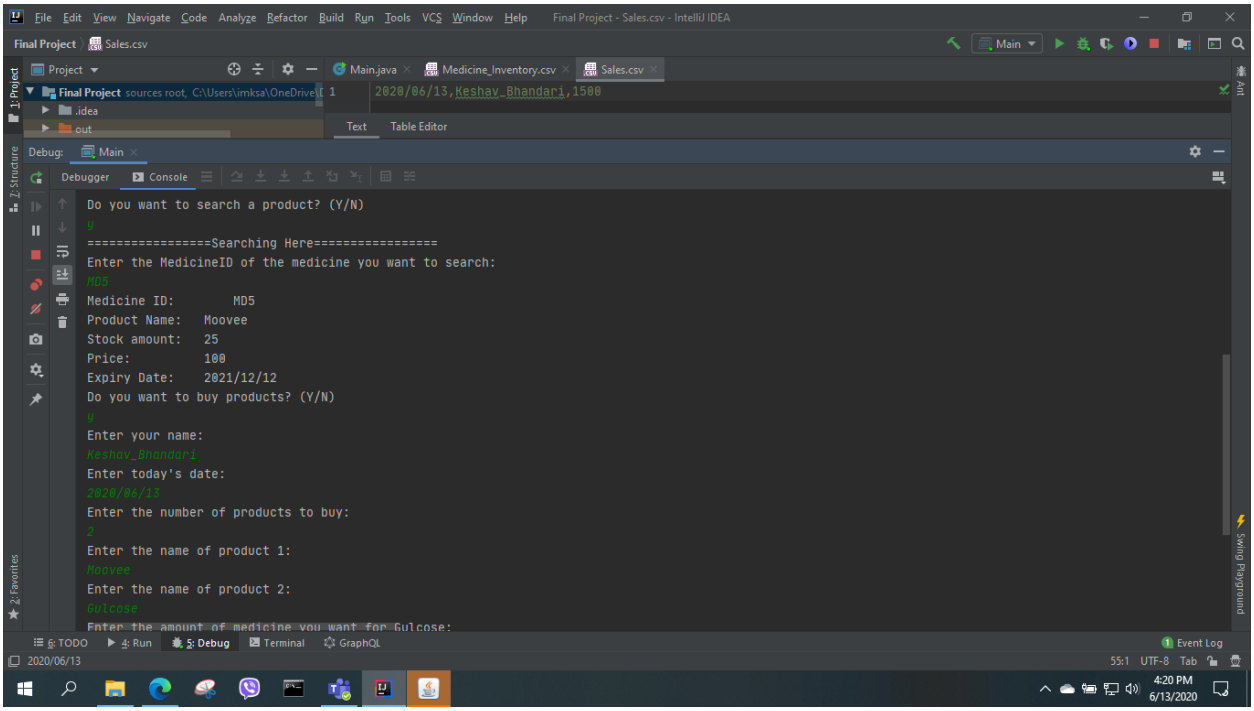
```
            if (i == 0)
                continue;

            for (int j = 0; j < 3; j++) {
                SalesData[i - 1][j] = arr[j];

            }
        }

        inputFile.close();
        input.close();
        return SalesData;
    }

    private CategoryDataset createDataset() throws FileNotFoundException {
        String[][] data = scanningData_Sales();
        DefaultCategoryDataset dataset = new DefaultCategoryDataset();

        for (String[] salesDatum : data) {
            dataset.addValue(Double.parseDouble(salesDatum[2]), "Sales", (salesDatum[1] + ":" +
salesDatum[0]));
        }

        return dataset;
    }

}
```

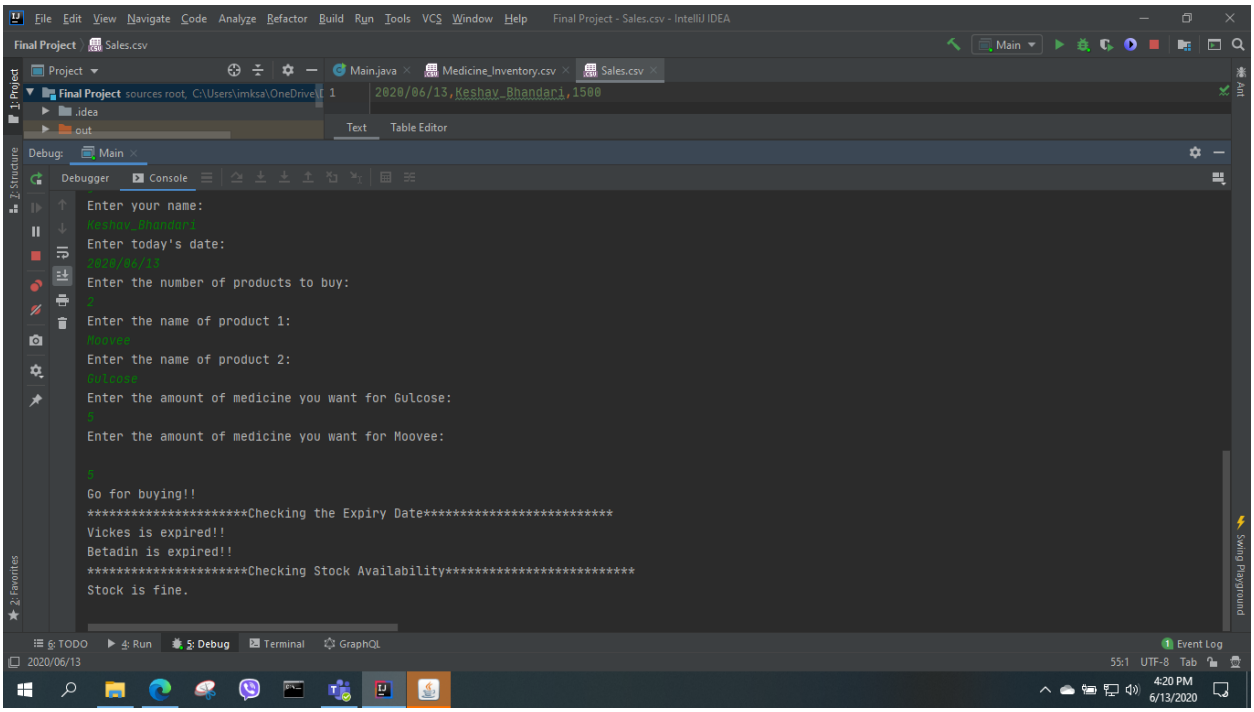
Programming Fundamentals

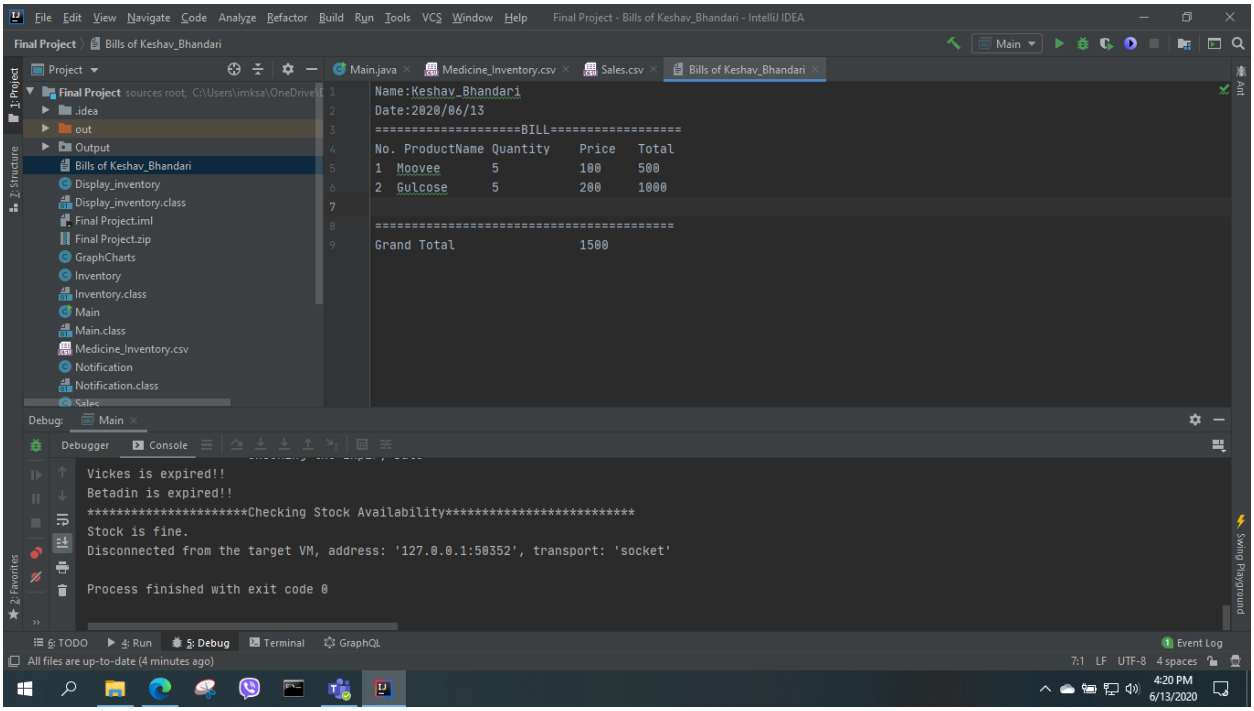
Output are:

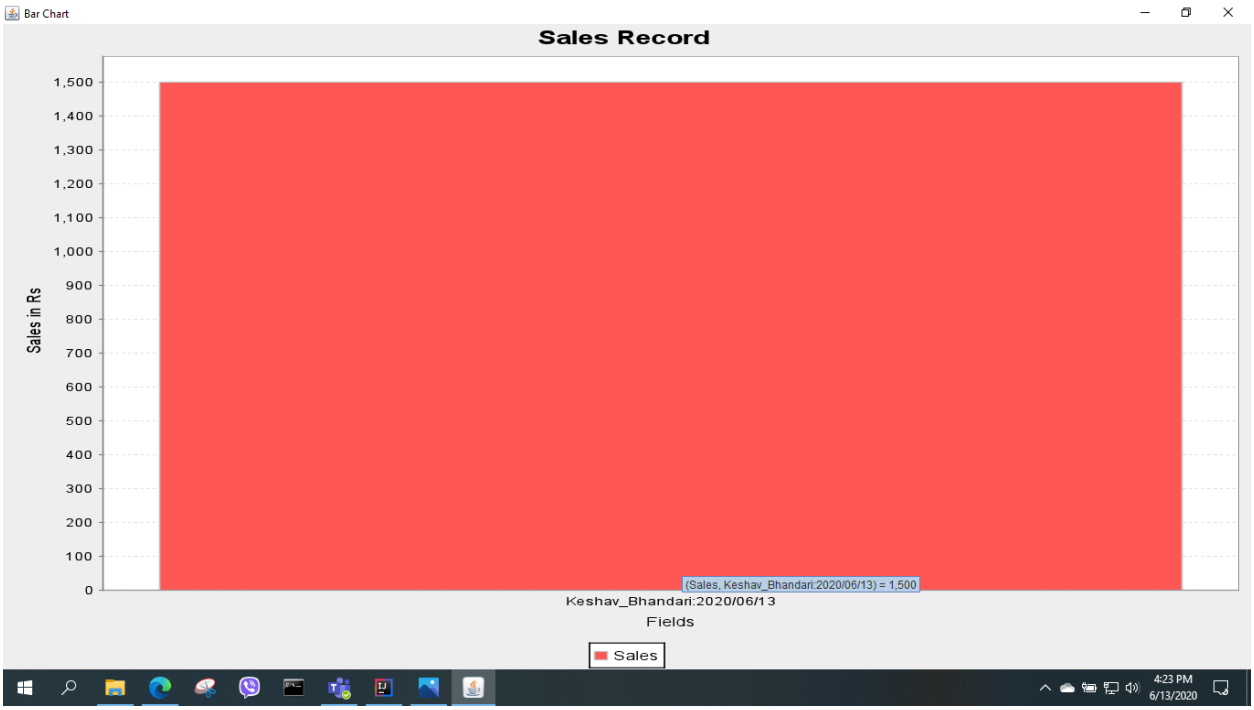





Do you want to search a product? (Y/N)
Y
==========Searching Here==========
Enter the MedicineID of the medicine you want to search:
MD5
Medicine ID: MD5
Product Name: Moovee
Stock amount: 25
Price: 100
Expiry Date: 2021/12/12
Do you want to buy products? (Y/N)
Y
Enter your name:
Keshav_Bhandari
Enter today's date:
2020/06/13
Enter the number of products to buy:
2
Enter the name of product 1:
Moovee
Enter the name of product 2:
Gulcose
Enter the amount of medicine you want for Gulcose:


Enter your name:
Keshav_Bhandari
Enter today's date:
2020/06/13
Enter the number of products to buy:
2
Enter the name of product 1:
Moovee
Enter the name of product 2:
Gulcose
Enter the amount of medicine you want for Gulcose:
5
Enter the amount of medicine you want for Moovee:
5
Go for buying!!
**********Checking the Expiry Date**********
Vickes is expired!!
Betadin is expired!!
**********Checking Stock Availability**********
Stock is fine.


Name:Keshav_Bhandari
Date:2020/06/13
==========BILL==========
No. ProductName Quantity Price Total
1 Moovee 5 100 500
2 Gulcose 5 200 1000
==========
Grand Total 1500
Vickes is expired!!
Betadin is expired!!
**********Checking Stock Availability**********
Stock is fine.
Disconnected from the target VM, address: '127.0.0.1:50352', transport: 'socket'
Process finished with exit code 0


Bar Chart
Sales Record
Sales in Rs
1,500
1,400
1,300
1,200
1,100
1,000
900
800
700
600
500
400
300
200
100
0
(Sales, Keshav_Bhandari:2020/06/13) = 1,500
Keshav_Bhandari:2020/06/13
Fields
Sales