IBM CAREER EDUCATION

MAIN PROJECT

DOMAIN NAME: JAVA

PHARMACY MANAGEMENT SYSTEM

Submitted By,

DHANANJAY KUMAR(18162171001)
KUNAL MALVI (18162171009),MISHANK GEHLOT(18162171004)

II Year – (CS) 'A' Section

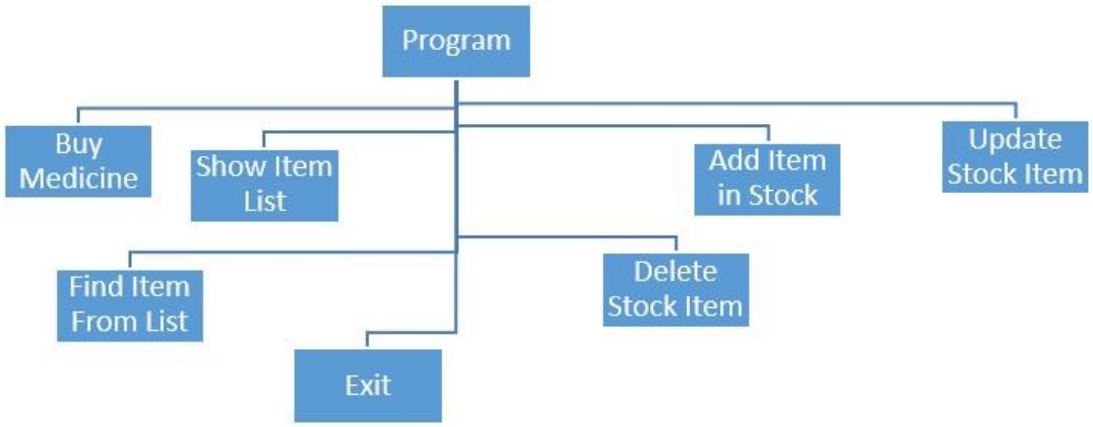
Ganpat University, Ahmedabad.

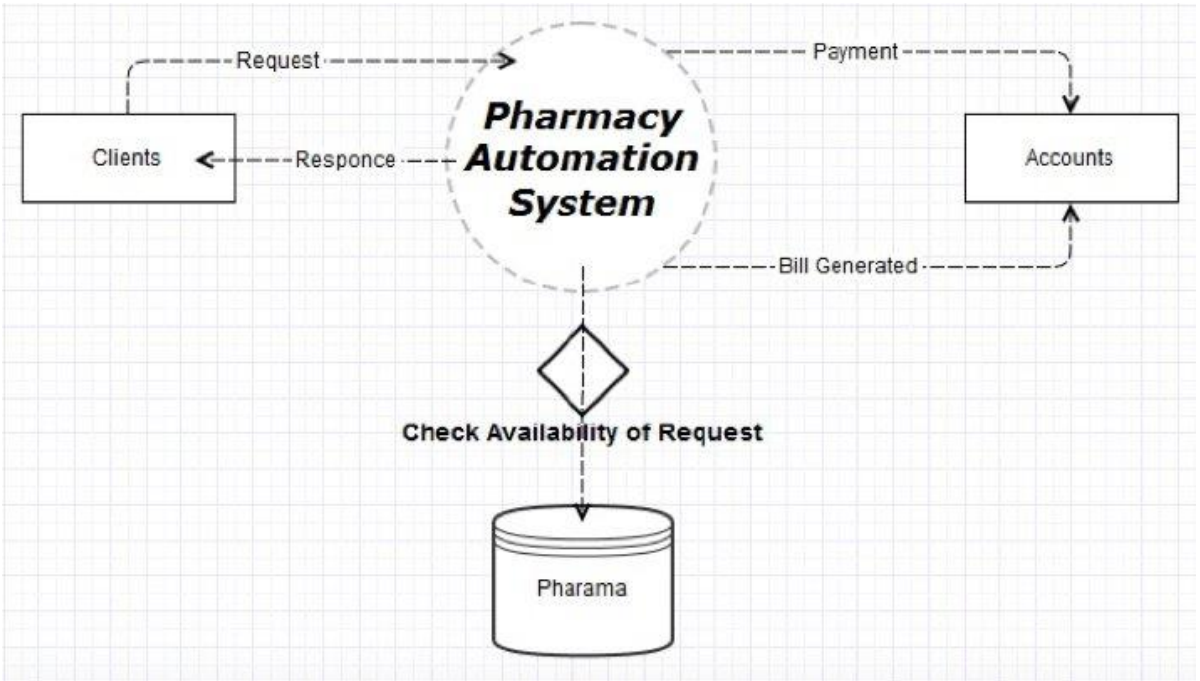
Submitted To:- **A. Saai Sanjeev Achaarya**

IBM Software Technical Trainer

PHARMACY MANAGEMENT SYSTEM

FLOW CHART





PHARMACYMANAGEMENTSYSTEM

SOFTWARE SPECIFICATIONS

- OPERATING SYSTEM : Linux / Windows
- ENVIRONMENT : NET BEANS

HARDWARE SPECIFICATIONS

- PROCESSOR : i3 7th Gen 2.8MHz
- RAM : 8 GB HDD
- MONITOR : 13” COLOR
- HARD DISK : 1TB
- FLOPPY DRIVE : 1.44 MB

PHARMACY MANAGEMENT SYSTEM

DESCRIPTION

INTRODUCTION:

Pharmacy Management System (PMS) is an enterprise resource planning system for a Pharmacy, which is used to add new medicine, to the Pharmacy record, to maintain record of medicine being sold and to maintain record of stock of medicine.

DESCRIPTION:

It is a project which aims in developing a computerized system to maintain all the daily work of Pharmacy. It is used by the Pharmacist to manage the pharmacy using a computerised system where he/she can record various transactions such addition of medicine, stock of medicine, records of medicine, etc.

ADVANTAGES:

- Reduces the human efforts.
- As the system is computerised so making it fast, this leads to time saving.
- Saves the cost of store.
- Risk of losing record is reduced.
- Most importantly searching becomes quite easy

DISADVANTAGES:

- Software needs to be updated regularly.
- The existing way of managing Pharmacy is quite flexible.

AIM:

To Create a Computerised Pharmacy Management System :-

PROGRAM :

```
public class AddData {

   public void setInitialValues(PharmacyDirectory
pharmacyDirectory)

   {
      PharmacyDirectory pharmaDirectory =
pharmacyDirectory;

      Pharmacy pharma1 = new Pharmacy();

      Pharmacy pharma2 = new Pharmacy();


      pharma1.setStoreName("pharma1");
      pharma2.setStoreName("pharma2");

      pharmaDirectory.addPharmacy(pharma1);
     pharmaDirectory.addPharmacy(pharma2);
       Drug drug1 = new Drug();


       //drug1.setDrugID(1);
       drug1.setDrugName("Crocin");
       drug1.setDrugType("paracetemol");
       drug1.setComposition("paracetemol ");
       drug1.setDrugAvailibility(40);
       drug1.setDrugPrice(20);
       drug1.setDrugDescription("used for fever");
       drug1.setExpirationDate("20-05-2018");
```

```
        drug1.setManufacturedDate("10-09-2015");
        pharma1.getDrugCatalog().addDrugs(drug1);

        Drug drug2 = new Drug();
        //drug2.setDrugID(2);
        drug2.setDrugName("Anacin");
        drug2.setDrugType("antacid");
        drug2.setComposition("antacid ");
        drug2.setDrugAvailibility(80);
        drug2.setDrugPrice(30);
        drug2.setDrugDescription("used for acidity");
        drug2.setExpirationDate("20-10-2018");
        drug2.setManufacturedDate("10-10-2015");
        pharma1.getDrugCatalog().addDrugs(drug2);

        Drug drug12 = new Drug();
        //drug12.setDrugID(3);
        drug12.setDrugName("Cetaphil");
        drug12.setDrugType("cetaphil");
        drug12.setComposition("antacid ");
        drug12.setDrugAvailibility(80);
        drug12.setDrugPrice(30);
        drug12.setDrugDescription("used for acne");
        drug12.setExpirationDate("20-10-2018");
        drug12.setManufacturedDate("10-10-2015");
        pharma2.getDrugCatalog().addDrugs(drug12);

    }
    public void setStoreInitialValues(StoreDirectory
storeDirectory){
        StoreDirectory sd = storeDirectory;

        Store store1 = new Store();

        Store store2 = new Store();

        store1.setStoreName("Store 1");
        store1.setStoreLocation("Boston");
        sd.addStore(store1);
```

```
        store2.setStoreName("Store 2");
        store2.setStoreLocation("New york");
        sd.addStore(store2);


    }

}


public class Drug {

    private int drugID;
    private String drugName;
    private String expirationDate;
    private String manufacturedDate;
    private String composition;
    private String drugType;
    private String drugDescription;
    private int drugAvailibility;
    private int drugPrice;

    private static int count;

    public Drug(){
        count++;
        drugID = count;
    }

    public String getExpirationDate() {
        return expirationDate;
    }

    public void setExpirationDate(String expirationDate) {
        this.expirationDate = expirationDate;
    }

    public String getManufacturedDate() {
```

```
        return manufacturedDate;
    }

    public void setManufacturedDate(String manufacturedDate)
{
        this.manufacturedDate = manufacturedDate;
    }

    public String getDrugDescription() {
        return drugDescription;
    }

    public void setDrugDescription(String drugDescription) {
        this.drugDescription = drugDescription;
    }

    public int getDrugAvailibility() {
        return drugAvailibility;
    }

    public void setDrugAvailibility(int drugAvailibility) {
        this.drugAvailibility = drugAvailibility;
    }

    public int getDrugPrice() {
        return drugPrice;
    }

    public void setDrugPrice(int drugPrice) {
        this.drugPrice = drugPrice;
    }

    public int getDrugID() {
        return drugID;
    }

    public void setDrugID(int drugID) {
        this.drugID = drugID;
    }
```

```
    public String getDrugName() {
        return drugName;
    }

    public void setDrugName(String drugName) {
        this.drugName = drugName;
    }

    public String getComposition() {
        return composition;
    }

    public void setComposition(String composition) {
        this.composition = composition;
    }

    public String getDrugType() {
        return drugType;
    }

    public void setDrugType(String drugType) {
        this.drugType = drugType;
    }

      @Override
    public String toString() {
       // return "Drug{" + "drugID=" + drugID + '}';
        return drugName;
    }

}

public class DrugCatalog {

private ArrayList<Drug> drugList;


    public DrugCatalog(){
```

```
    this.drugList = new ArrayList<Drug>();
}
    public ArrayList<Drug> getDrugList() {
        return drugList;
    }

    public void setDrugList(ArrayList<Drug> drugList) {
        this.drugList = drugList;
    }

    public Drug addDrugs(Drug d) {
       // Drug d = new Drug();
        drugList.add(d);
        return d;

    }
    public void removeDrug(Drug d)
    {
        drugList.remove(d);
    }


}


public class MasterOrderCatalog {
    private ArrayList<Order> orderCatalog;

    public MasterOrderCatalog(){
        orderCatalog = new ArrayList<Order>();
    }

    public Order addOrder(Order o){

        orderCatalog.add(o);
        return o;
    }

    public ArrayList<Order> getOrderCatalog() {
```

```
        return orderCatalog;
    }

    public void setOrderCatalog(ArrayList<Order> orderCatalog)
{
        this.orderCatalog = orderCatalog;
    }

}


public class Order {
    private ArrayList<OrderItem> orderItemList;
    private int orderNumber;
    private static int count;

    public int getOrderNumber() {
        return orderNumber;
    }

    public void setOrderNumber(int orderNumber) {
        this.orderNumber = orderNumber;
    }

    public Order()
    {
        orderItemList =  new ArrayList<>();
        count++;
        orderNumber=count;

    }

    public OrderItem addOrderItem(Drug d, int quantity, int
price)
    {
        OrderItem orderItem = new OrderItem();
        orderItem.setDrug(d);
        orderItem.setQuantity(quantity);
        orderItem.setSalesPrice(price);
```

```
        orderItemList.add(orderItem);
        return orderItem;
    }

    @Override
    public String toString() {
        return String.valueOf(orderNumber);
    }

    public void removeOrderItem(OrderItem o)
    {
        orderItemList.remove(o);
    }

    public ArrayList<OrderItem> getOrderItemList() {
        return orderItemList;
    }

    public void setOrderItemList(ArrayList<OrderItem>
orderItemList) {
        this.orderItemList = orderItemList;
    }

}

public class OrderItem {

    private Drug drug;
    private int quantity;
    private int salesPrice;

    @Override
    public String toString() {
        return drug.getDrugName();
    }

    public Drug getDrug() {
        return drug;
    }
```

```
    public void setDrug(Drug drug) {
        this.drug = drug;
    }

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public int getSalesPrice() {
        return salesPrice;
    }

    public void setSalesPrice(int salesPrice) {
        this.salesPrice = salesPrice;
    }

}

public class Pharmacy {

    private String pharmaName;
    private int pharmaID;
    private String pharmaLocation;
    private DrugCatalog drugCatalog;

    @Override
    public String toString() {
        //return "Pharmacy{" + "pharmaName=" + pharmaName +
'}';
        return pharmaName;
    }

    public Pharmacy(){
        drugCatalog = new DrugCatalog();
```

```
    }
    public DrugCatalog getDrugCatalog() {
        return drugCatalog;
    }

    public void setDrugCatalog(DrugCatalog drugCatalog) {
        this.drugCatalog = drugCatalog;
    }

    public String getStoreName() {
        return pharmaName;
    }

    public void setStoreName(String storeName) {
        this.pharmaName = storeName;
    }

    public int getStoreID() {
        return pharmaID;
    }

    public void setStoreID(int storeID) {
        this.pharmaID = storeID;
    }

    public String getStoreLocation() {
        return pharmaLocation;
    }

    public void setStoreLocation(String storeLocation) {
        this.pharmaLocation = storeLocation;
    }


}

public class PharmacyDirectory {

    ArrayList<Pharmacy> pharmaList;
```

```
    public PharmacyDirectory(){
        this.pharmaList =  new ArrayList<Pharmacy>();
    }
    public ArrayList<Pharmacy> getStoreList() {
        return pharmaList;
    }

    public void setStoreList(ArrayList<Pharmacy> storeList) {
        this.pharmaList = storeList;
    }

    public Pharmacy addPharmacy(Pharmacy pharma) {

        //Pharmacy pharma = new Pharmacy();
        pharmaList.add(pharma);
        return pharma;
    }

    public void removeSupplier(Pharmacy pharmacy) {
     pharmaList.remove(pharmacy);
    }

    public Drug searchDrug(int drugID)
    {
        for(Pharmacy p : pharmaList){
        for(Drug drug : p.getDrugCatalog().getDrugList()){
            if(drugID == drug.getDrugID())
                return drug;
        }
         }
        return null;
    }
}

public class Store {

    private String storeName;
    private String storeLocation;
```

```
    private MasterOrderCatalog masterOrderCatalog;

    public Store(){
        masterOrderCatalog = new MasterOrderCatalog();

    }
    @Override
    public String toString() {
        //return "Store{" + "storeName=" + storeName + '}';
        return getStoreName();
    }

    public String getStoreName() {
        return storeName;
    }

    public void setStoreName(String storeName) {
        this.storeName = storeName;
    }

    public String getStoreLocation() {
        return storeLocation;
    }

    public void setStoreLocation(String storeLocation) {
        this.storeLocation = storeLocation;
    }

    public MasterOrderCatalog getMasterOrderCatalog() {
        return masterOrderCatalog;
    }

    public void setMasterOrderCatalog(MasterOrderCatalog
masterOrderCatalog) {
        this.masterOrderCatalog = masterOrderCatalog;
    }

}
```

```
public class StoreDirectory {

    ArrayList<Store> storeList;

    public StoreDirectory(){
        this.storeList = new ArrayList<>();
    }

    public ArrayList<Store> getStoreList() {
        return storeList;
    }

    public void setStoreList(ArrayList<Store> storeList) {
        this.storeList = storeList;
    }

    public Store addStore(Store store){
        //Store store = new Store();
        storeList.add(store);
        return store;
    }
    public void removeStore(Store store){
        storeList.remove(store);
    }

   }

public class Validator {

    public boolean isValidInteger(String str)
    {
        int a;
        try {
              a = Integer.parseInt(str);
              return true;
           }
           catch(NumberFormatException nfe){
                JOptionPane.showMessageDialog(null, "Please
enter a number","warning",JOptionPane.ERROR_MESSAGE);
```

```
                return false;
            }
    }

    public boolean isValidFloat(String str)
    {
        float a;
        try {
                a = Float.parseFloat(str);
                return true;
            }
            catch(NumberFormatException nfe){
                JOptionPane.showMessageDialog(null, "Please
enter a number","warning",JOptionPane.ERROR_MESSAGE);
                return false;
            }
    }

    public boolean isStringValid(String str)
    {
        if(str.trim().isEmpty()){
            return false;
        }
        else{
            return true;
        }

    }

}


public static void main(String args[]) {

            try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
```

```
                break;
            }
        }
    } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(MainJFrame.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(MainJFrame.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(MainJFrame.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException
ex) {

java.util.logging.Logger.getLogger(MainJFrame.class.getName(
)).log(java.util.logging.Level.SEVERE, null, ex);
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new MainJFrame().setVisible(true);
        }
    });
}
private javax.swing.JButton CVSAdminJButton;
private javax.swing.JPanel jPanel1;
private javax.swing.JSplitPane jSplitPane1;
private javax.swing.JButton pharmaAdminJButton;
private javax.swing.JButton storeAdminJButton;
private javax.swing.JPanel userProcessContainer;
}
```

ABOUT PROJECT:

This project basically has modules:

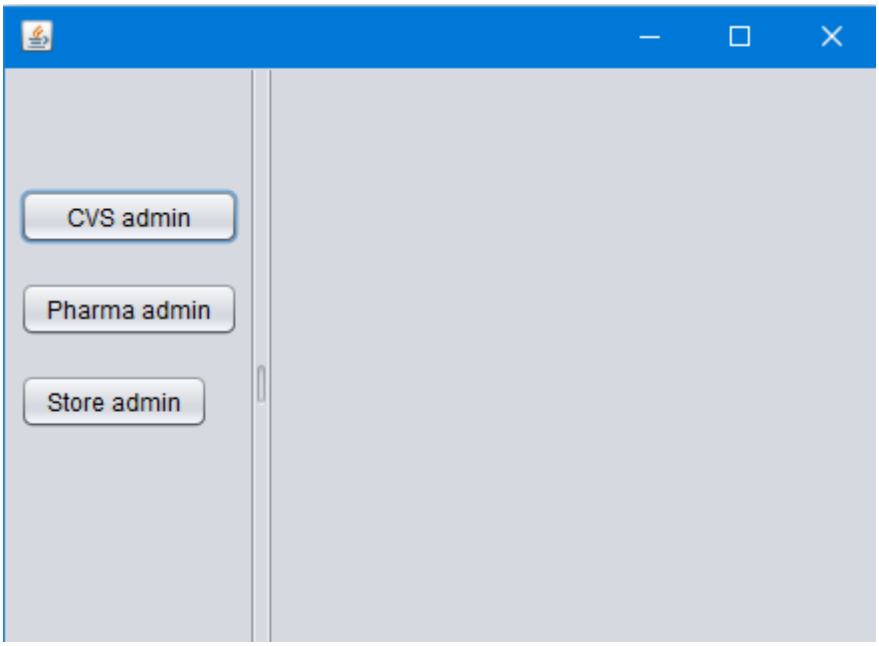
1.Adding medicine:- Adding of new medicine to the medicine record. Adding details of the medicine which is to added.

2.Order Status:-Order can give the details of ordered medicine.

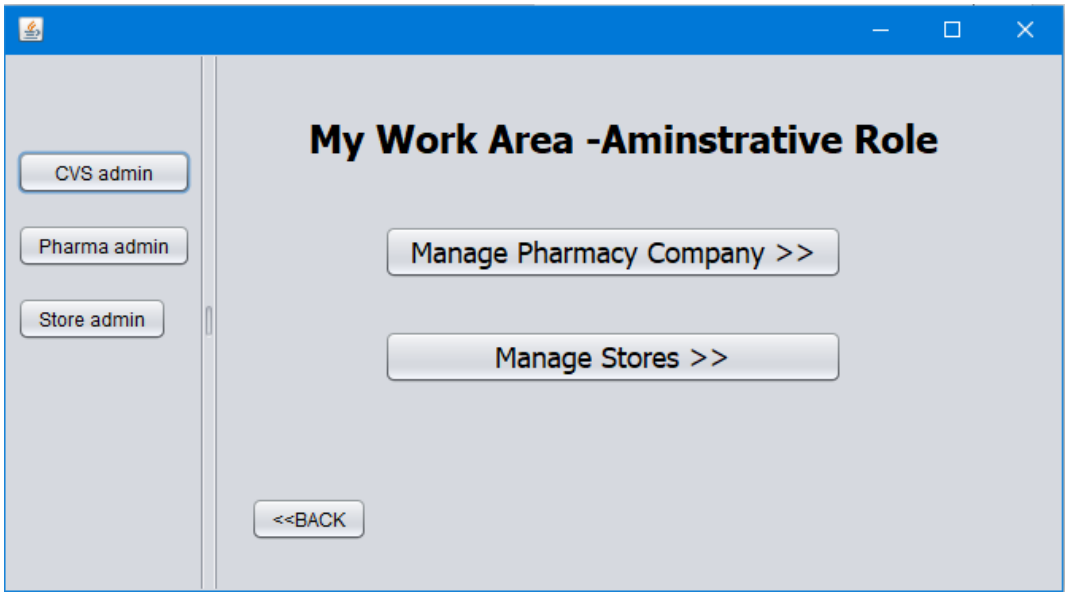
3.Medicine List:-For checking the list of medicine available in store.

4.Pharma Admin:-To managing multiple pharmacy at a time so it will save our(store manager) time.

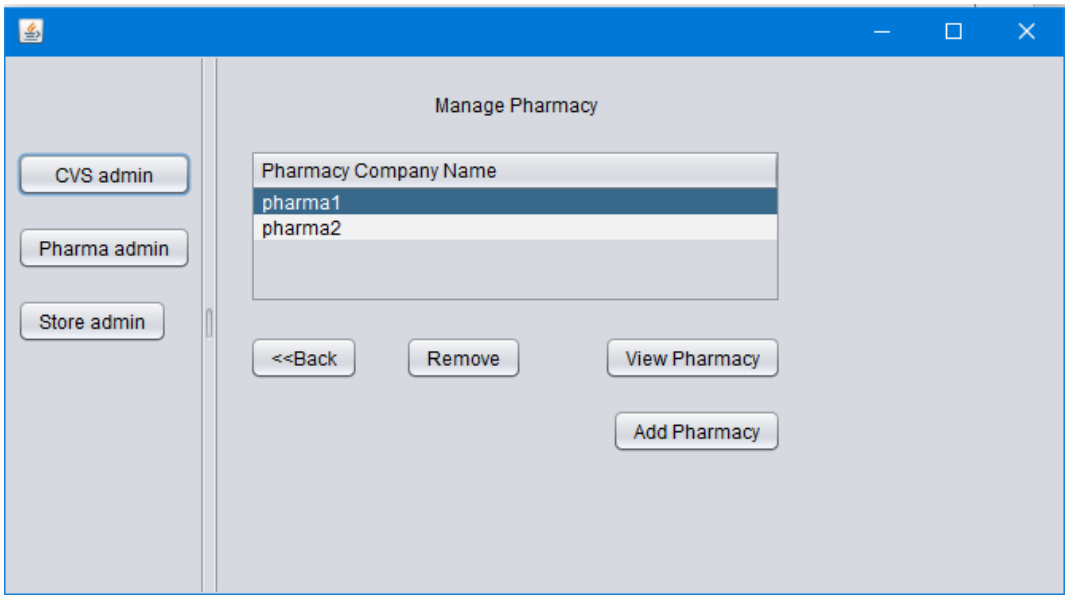
OUTPUT SCREENSHOTS:



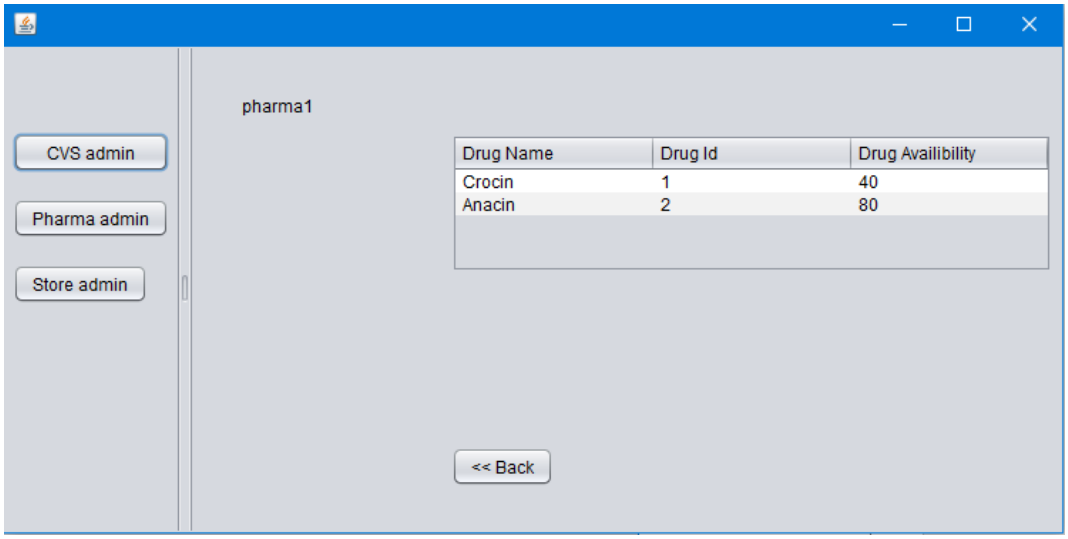
Main frame after login



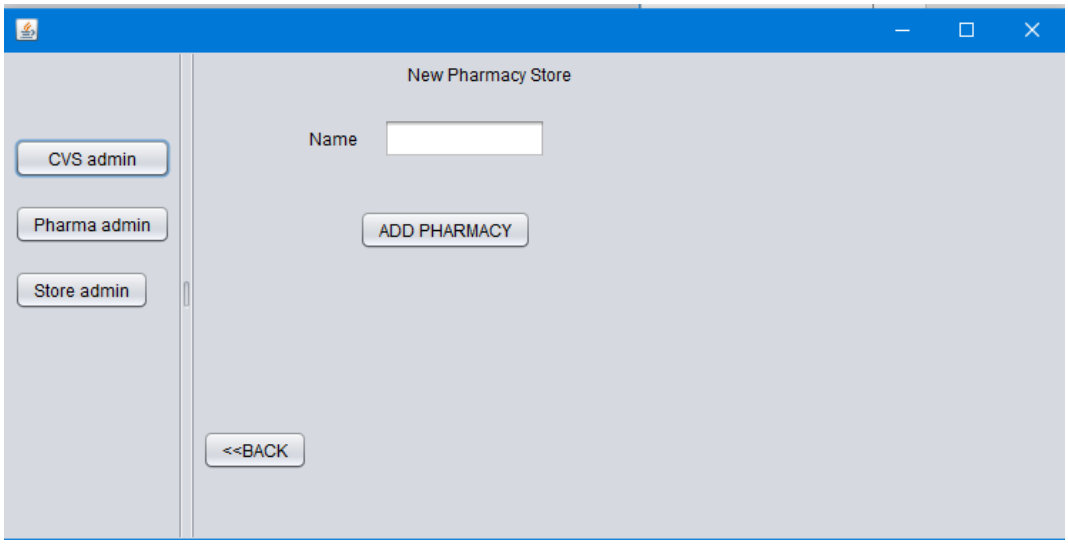
Under CVS admin



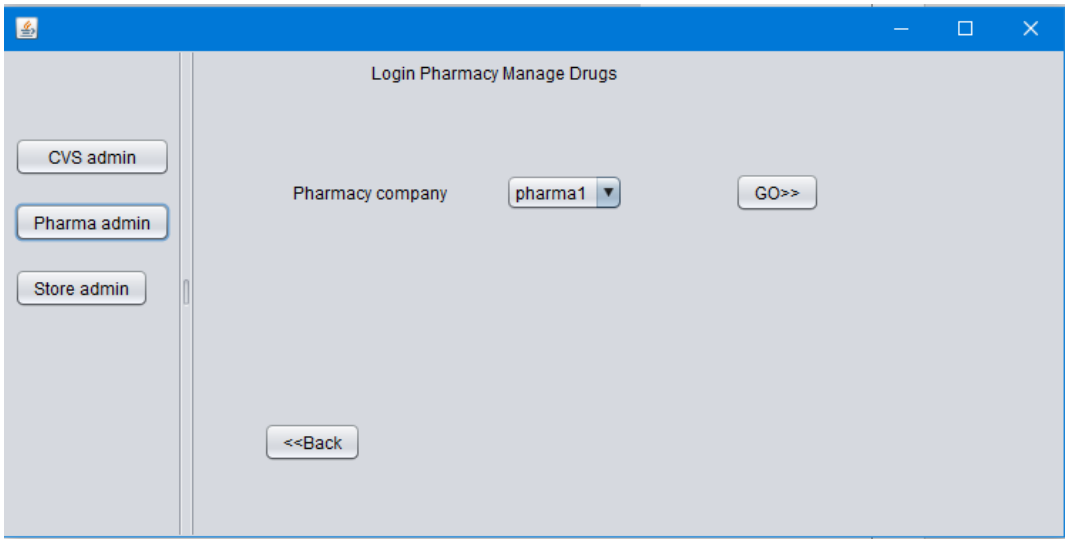
CVS admin->manage Pharmacy company



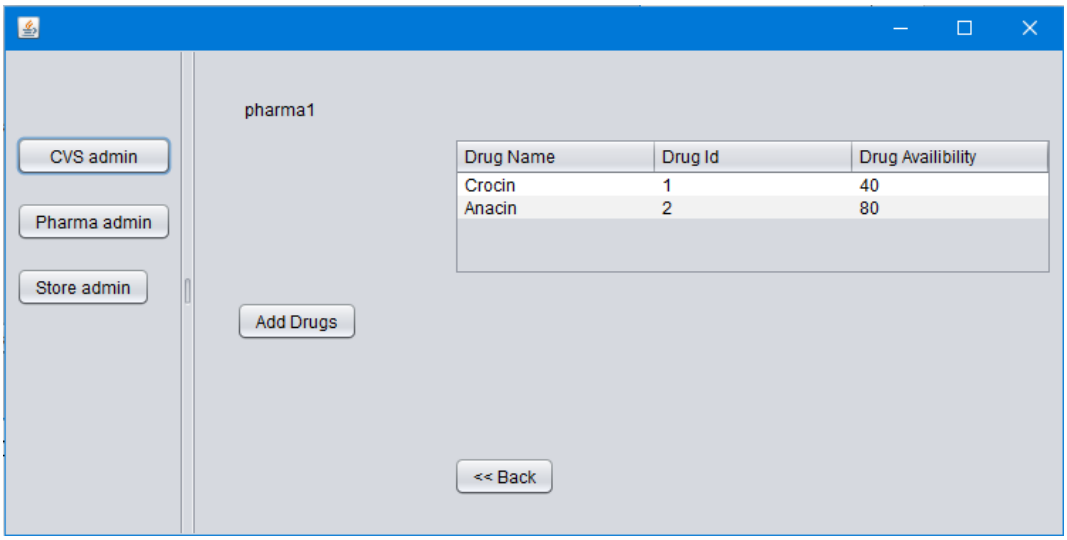
CVS admin->manage Pharmacy company->view Pharmacy



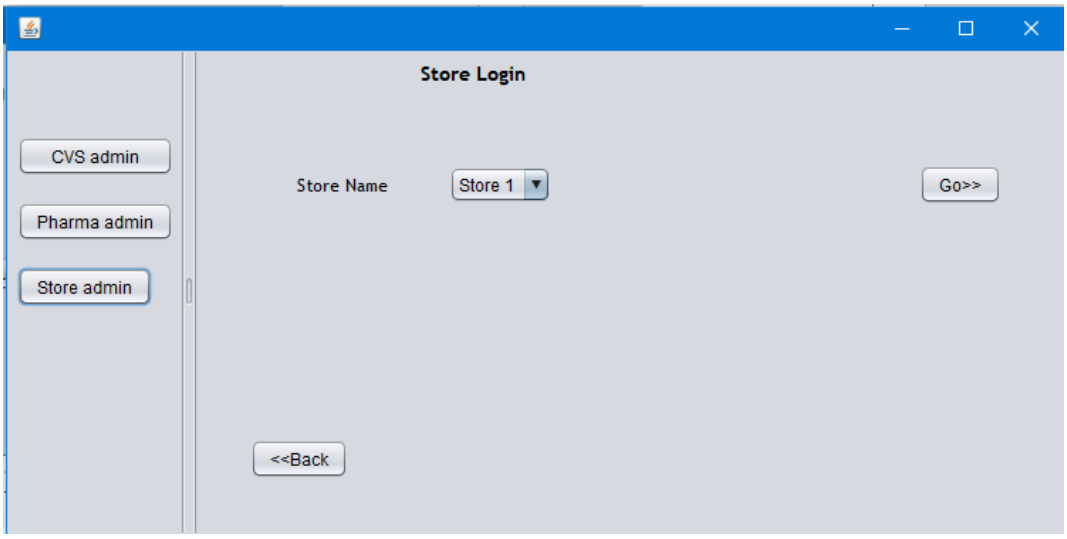
CVS admin->manage Pharmacy company->add Pharmacy



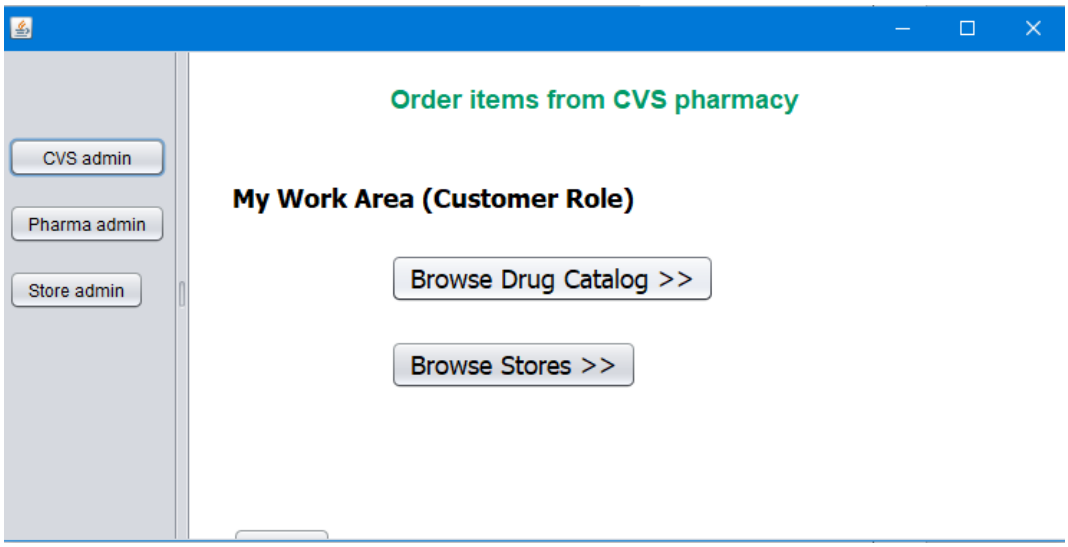
Under Pharma admin



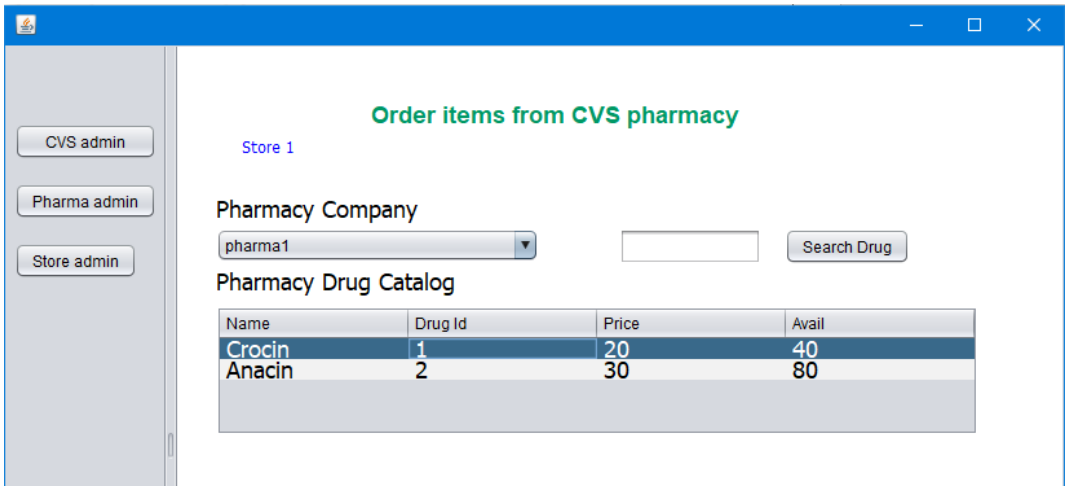
Pharma admin->(Select Pharmacy company)Go



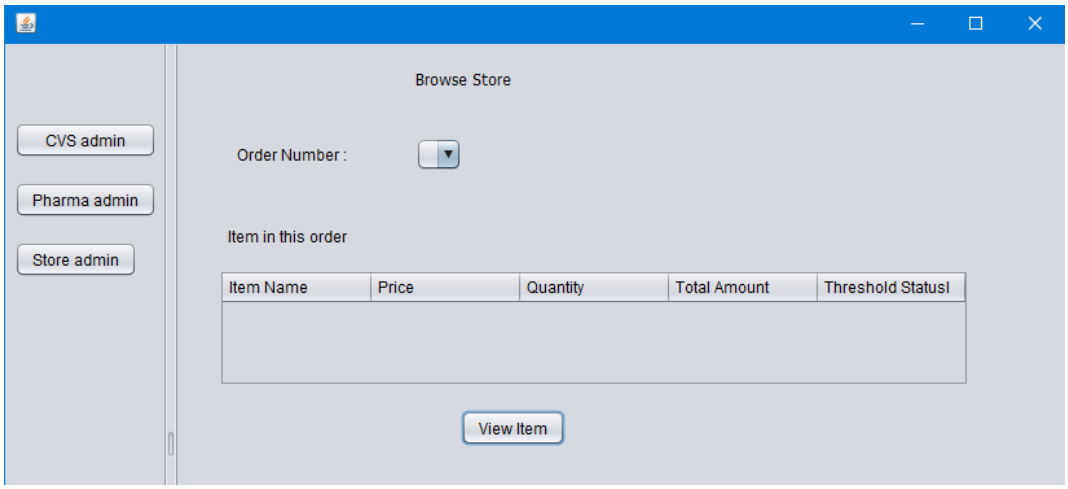
Under Store admin



Store admin->(Select store name)Go



Store admin->(Select store name)Go->Browse Drug catalog



Store admin->(Select store name)Go->Browse Store

REFERENCES:

1. Wikipedia
2. Google

CONCLUSION:

Finally concluded some important things that include designing a good program architecture and converting real life situations into an efficient code, and how to write an good looking easily readable and understandable as well as time and memory efficient code.