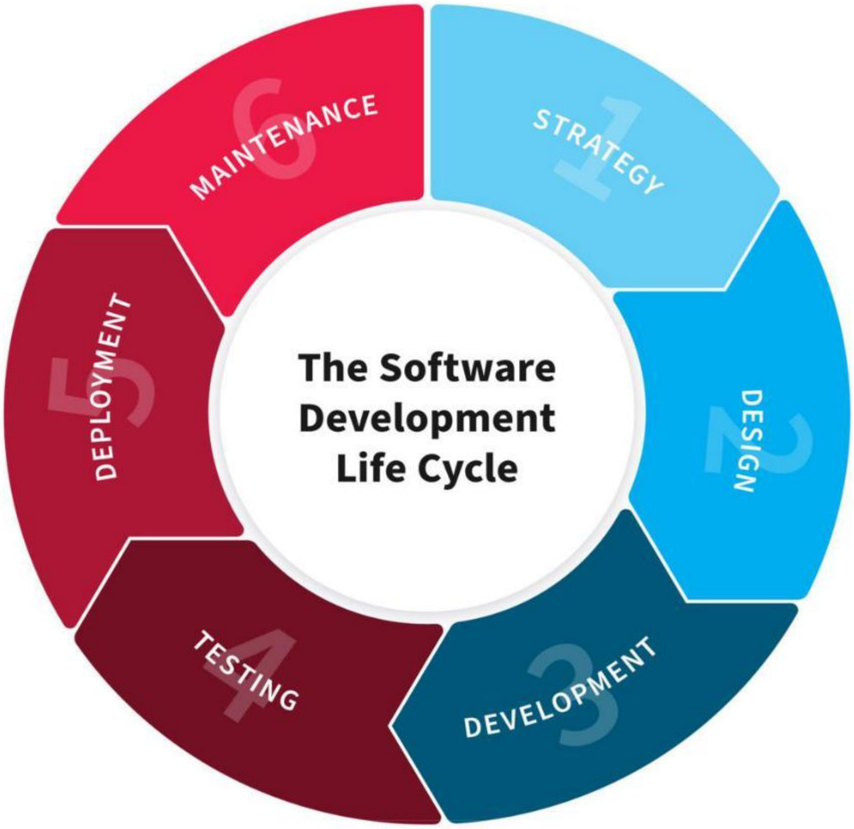
Names: **UMUHOZA Benitha**
Reg No: 221005939
College of Business and Economics
School of Economics
Business Information Technology department (BIT)
Level 2
Class Group: **Group 3**

Pharmacy Management System

This is a description of how our group project complies with all the stages of the software development cycle as presented below:



Stage 1: STRATEGY

.

Our Pharmacy Management system is a java based web application designed to simplify the supply, acquisition and storage of medical drugs all in a neat, bug-free and user friendly system. With this application, the user (a certified licensed pharmacist) will be able to keep track of his/her drug inventory, access supplier details, insure that all drugs are still usable and sell drugs on a regular basis.

Our application was designed to address the following specific problems:

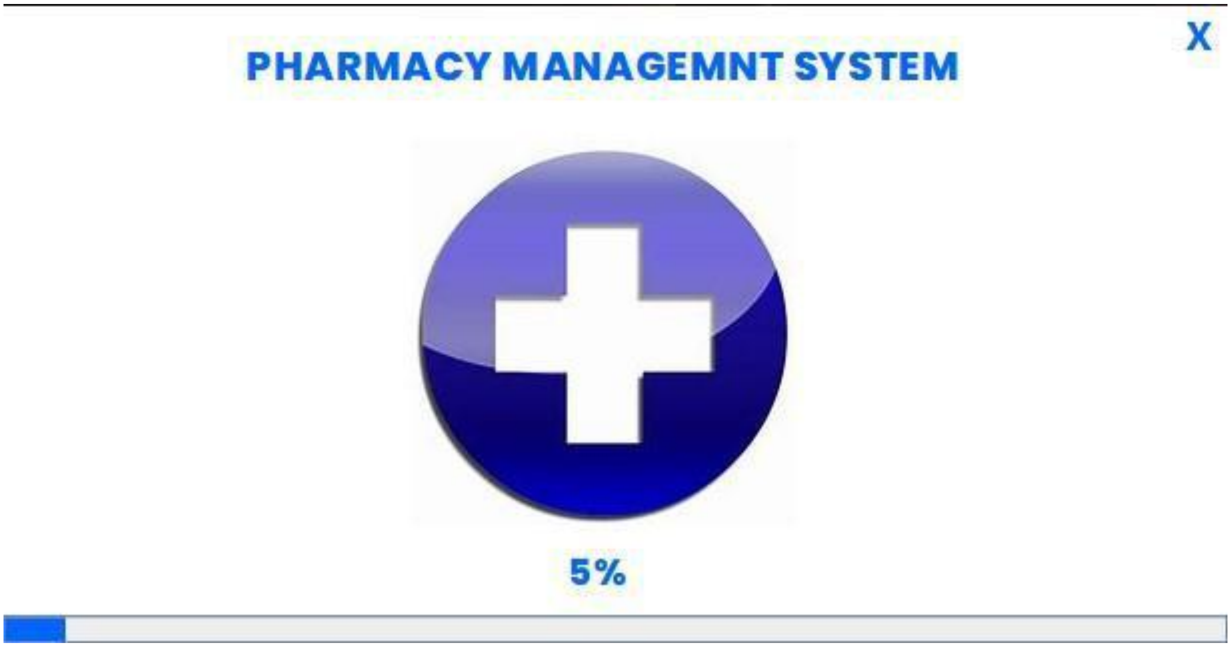
- The lack of a user friendly application that helps to monitor drugs inflow and outflow within any given pharmacy.
- A lack of a multi-user application that supports a shift system which is common in the field of medical care.
- A lack user identification that lets you know who access the application and when at all times.
- The lack of a task-oriented system designed and built to address the problem of drugs acquisition, storage and supply.
- The lack of a reliable system that is not prone to be regularly maintained.

The pharmacy management system serves many purposes, including the safe and effective dispensing of pharmaceutical drugs. During the dispensing process, the system will prompt the pharmacist to verify the medication they have filled is for the correct patient, contains the right quantity and dosage, and displays accurate information on the prescription label. Advanced pharmacy management systems offer clinical decision support and may be configured to alert the pharmacist to perform clinical interventions, such as an opportunity to offer verbal counseling if the patient's prescription requires additional education in the pharmacy.

Stage 2: DESIGN

Our pharmacy management system aimed and simplicity and efficacy during the implementation of its design to insure a user friendly experience. Below is a description of every page with details on the functianilty of every aspect of the design.

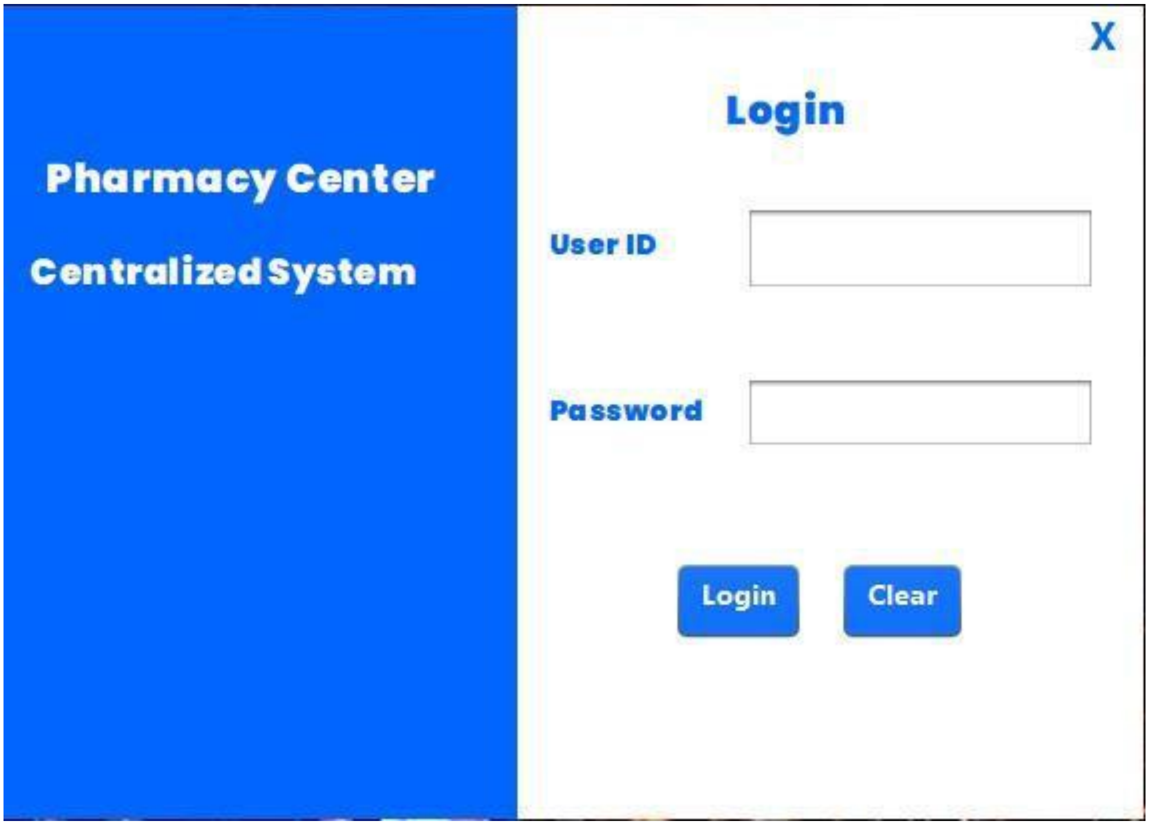
The loading dock splash page



This page illustrates the name of the application. The contains an image that represents the pharmacy logo as well as a progrss bar to display how for the system is loading.

Its key functinality is to provide a starting point for the user to begin to us the application.

The login page

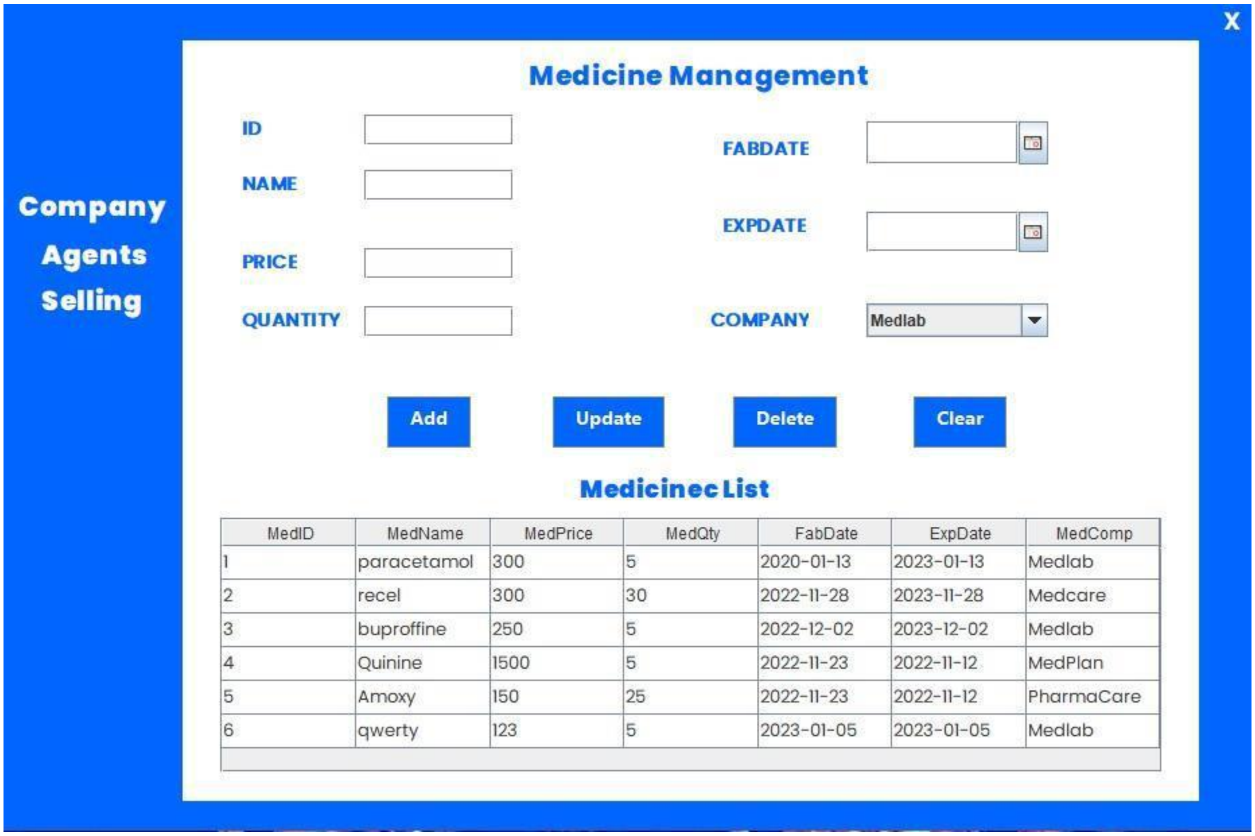


The login page contains

- A text area to hold the user name
- A text area to hold the user password.
- An exit button to close the application.
- A login button to confirm the user credentials and verify if the user is signed in or not.
- A clear button to dispose of any content within both text areas(password & username)

The main function of the login page is to provide the user with space to identify themselves in order to access the system. Incase the user is no recognised the system outputs the message “WRONG PASSWORD”

The medicine management page



COMPANY: Medlab

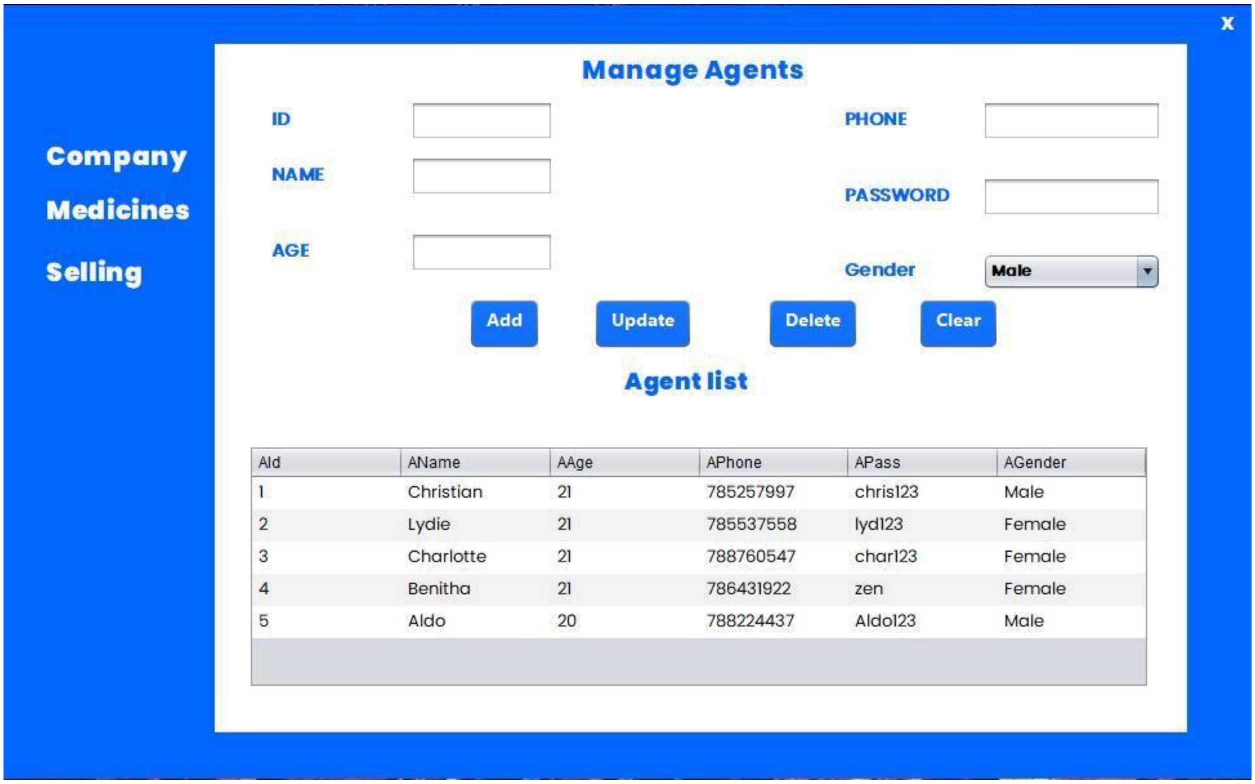
MedID	MedName	MedPrice	MedQty	FabDate	ExpDate	MedComp
1	paracetamol	300	5	2020-01-13	2023-01-13	Medlab
2	recel	300	30	2022-11-28	2023-11-28	Medcare
3	buproffine	250	5	2022-12-02	2023-12-02	Medlab
4	Quinine	1500	5	2022-11-23	2022-11-12	MedPlan
5	Amoxy	150	25	2022-11-23	2022-11-12	PharmaCare
6	qwerty	123	5	2023-01-05	2023-01-05	Medlab

The medicine management page provides the user with the possibility to operate on drugs and access any information concerning it. The following functionalities are provided:

- A text field to hold the medicine ID
- A text field to hold the medicine name.
- A text field to hold the medicine price.
- A text field to hold the medicine quantity available in stock.
- A calendar date chooser to input the medicine fabrication date.
- A calendar date chooser to input the medicine expiration date.
- A combo box to allow the user to choose the company supplying the medicine.
- An add button to add new medicine.
- An update button to modify medicine information.
- A delete button to remove any medicine from the medicine table.
- A clear button to empty the text fields (ID, medname, medprice & medquantity).
- A medicine table to select any medicine from it.
- A company label that links to the company management page.

- An Agents label that links to the Agents management page.
- A selling label that links to the selling management page.

The Agents management page.



Gender: Male

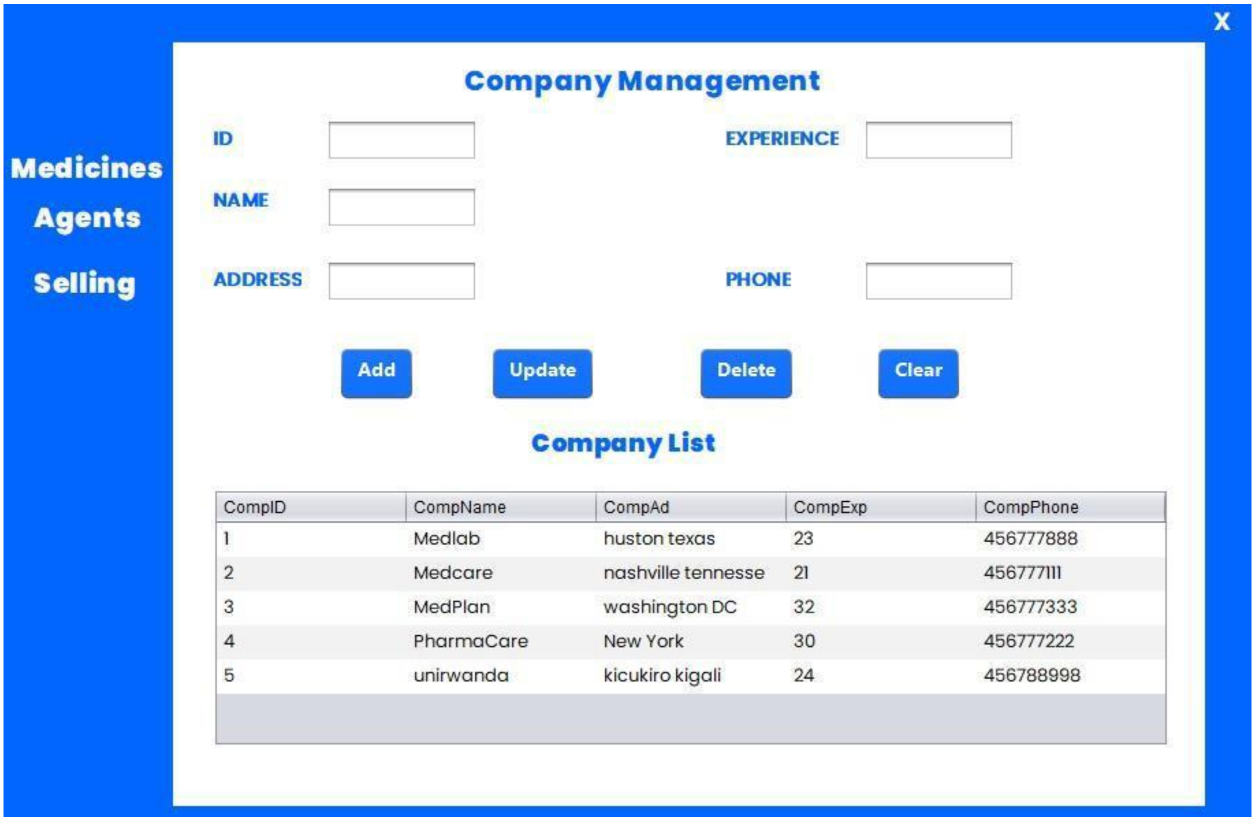
AId	AName	AAge	APhone	APass	AGender
1	Christian	21	785257997	chris123	Male
2	Lydie	21	785537558	lyd123	Female
3	Charlotte	21	788760547	char123	Female
4	Benitha	21	786431922	zen	Female
5	Aldo	20	788224437	Aldo123	Male

The Agents management page provides the user with the possibility to operate on agents and access any information concerning them. The following functionalities are provided:

- A text field to hold the Agents ID
- A text field to hold the Agents name.
- A text field to hold the Agents age.
- A text field to hold the Agents phone number (cell).
- A text field to hold the Agents password.
- A combo box to allow the user to choose the Agents sex.
- An add button to add new Agents.

- An update button to modify Agents information.
- A delete button to remove any Agents from the Agents table.
- A clear button to empty the text fields (ID, Agent name, Agent age, Agent phone & Agent password).
- An Agent table to select any Agent from it.
- A company label that links to the company management page.
- A medicine label that links to the medicine management page.
- A selling label that links to the selling management page.

The company management page.



CompID	CompName	CompAd	CompExp	CompPhone
1	Medlab	huston texas	23	456777888
2	Medcare	nashville tennesse	21	456777111
3	MedPlan	washington DC	32	456777333
4	PharmaCare	New York	30	456777222
5	unirwanda	kicukiro kigali	24	456788998

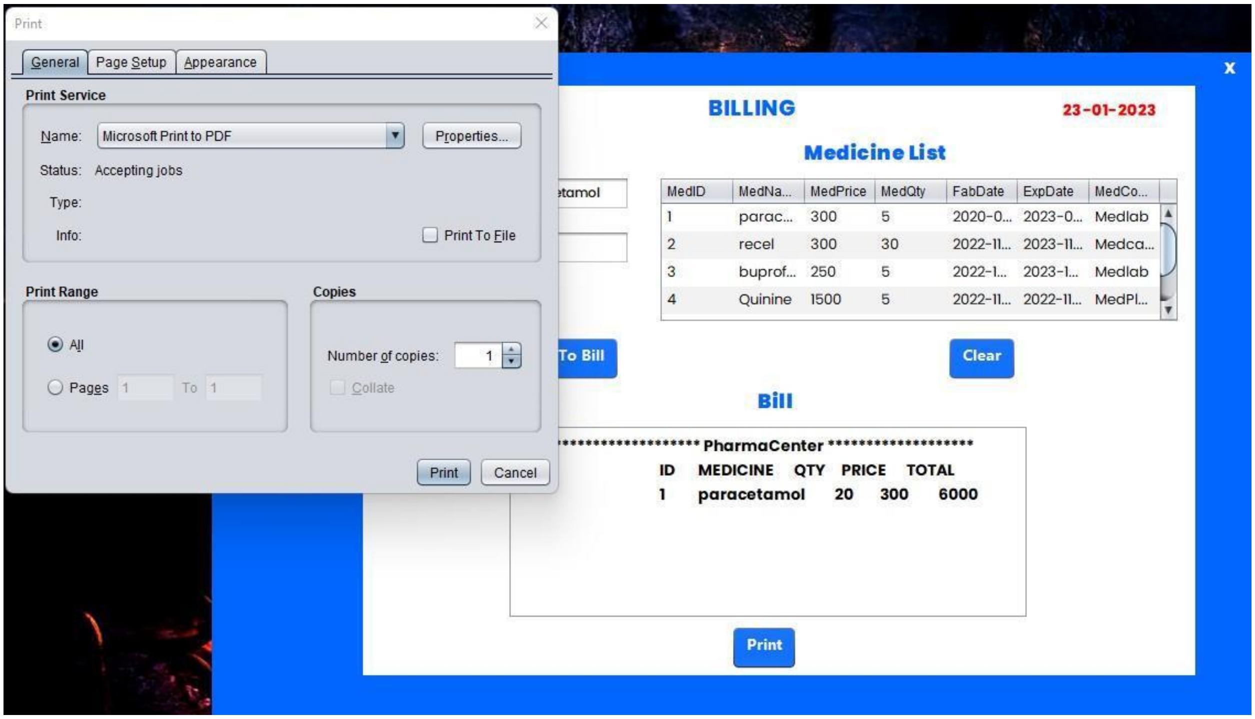
The company management page provides the user with the possibility to operate on company supplying medicine to the pharmacy and access any information concerning them.

The following functionalities are provided:

- A text field to hold the company ID
- A text field to hold the company name.
- Text fields to hold the company address.
- A text field to hold the company phone number (cell).

- A text field to hold the company experience. This field holds how much time the company has been operating and manufacturing medical drugs and supplies.
- An add button to add new company.
- An update button to modify company information.
- A delete button to remove any company from the company table.
- A clear button to empty the text fields (ID, company name, company address, company phone & company experience).
- A company table to select any company from it.
- A medicine label that links to the medicine management page.
- An Agents label that links to the Agents management page.
- A selling label that links to the selling management page.

The selling management page.



Print

Print Service

Name: Microsoft Print to PDF

Status: Accepting jobs

Type:

Info:

□ Print To File

Print Range

● All

○ Pages 1 To 1

Copies

Number of copies: 1

□ Collate

BILLING

23-01-2023

Medicine List

etamol

MedID	MedNa...	MedPrice	MedQty	FabDate	ExpDate	MedCo...
1	parac...	300	5	2020-0...	2023-0...	Medlab
2	recel	300	30	2022-11...	2023-11...	Medca...
3	buprof...	250	5	2022-1...	2023-1...	Medlab
4	Quinine	1500	5	2022-11...	2022-11...	MedPl...

Bill

```
******************** PharmaCenter ********************
ID   MEDICINE     QTY   PRICE   TOTAL
1    paracetamol   20    300     6000
```

The selling management page provides the user with the possibility to operate on sells of medicine to the clients, create bills and access any information concerning sells.

The following functionalities are provided:

- A text field to hold the medicine name to be sold.
- A text field to hold the medicine quantity to be sold.

- A text area to hold the medicine to be sold that is added to the bill.
- An Add to bill button the medicine to the bill.
- A clear button to empty the text fields (ID, medicine name & medicine quantity).
- A print button to print the bill.
- A company table to select any company from it.
- A medicine label that links to the medicine management page.
- A company label that links to the company management page.
- An Agents label that links to the Agents management page.

Stage 3: DEVELOPMENT

As the name suggests, a pharmacy management system is a platform that digitizes workflows and functionalities in pharmacies. People who visit pharmacies often rely on pharmacists to help them understand medication parameters and even interpret doctors' recommendations.

Additionally, pharmacists have to dispense medication, complete insurance formalities, manage sales and day-to-day administrative activities. This is where a pharmacy software system comes in handy.

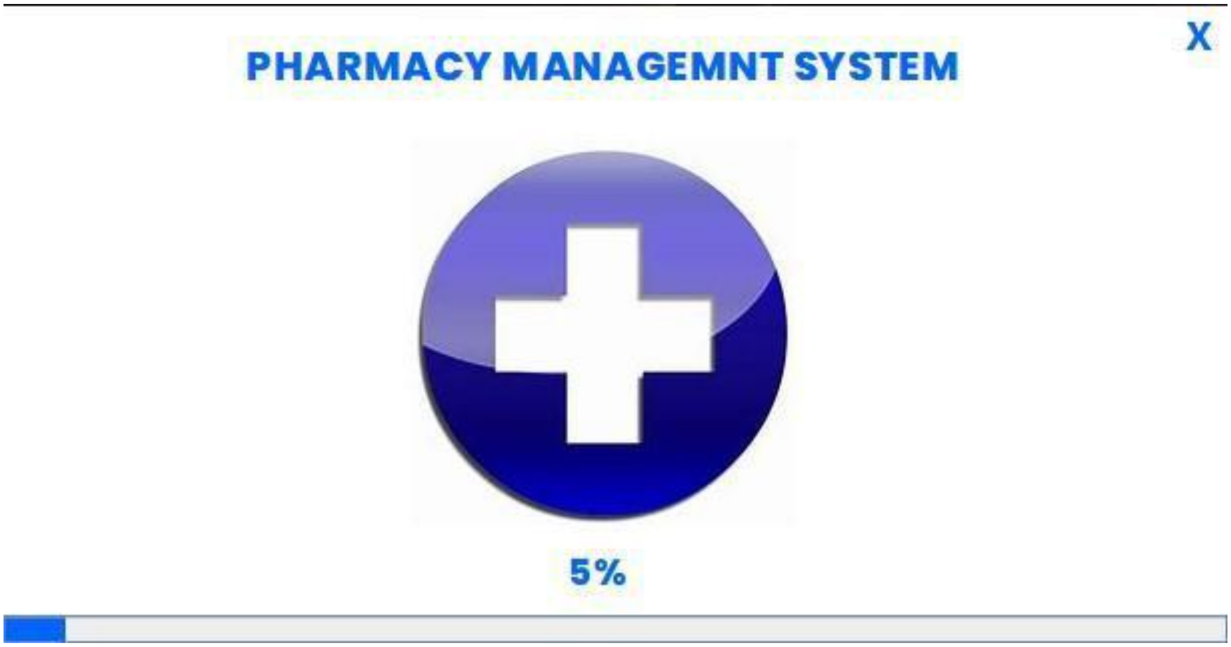
The following libraries were used:

- **Mysql-connector-j-8.0.31.jar** to connect to the database.
- **PostgreSQL JDBC Driver – psorgresql-442.2.16.jar** to access the database in the application (Apache NetBeans IDE 15).
- **Commons-dbutils-1.5-sources.jar** to link up the tables created in the design to the tables in the MySQL database.
    - **rs2xml.jar** to manipulate the data input into the tables and allow selection.
- **jcalendar-1.4.jar** to create a jcalendar field on the medicine management page that allows the user to select fabrication date as well as expiration date.
- **JDK 19** to allow the computer to read jar based files.

The storage database used in the development of this application is **MySQL** with the use of a XAMPP control panel to start the server

The following highlight the codes used to create the application's functionality:

The loading dock splash page

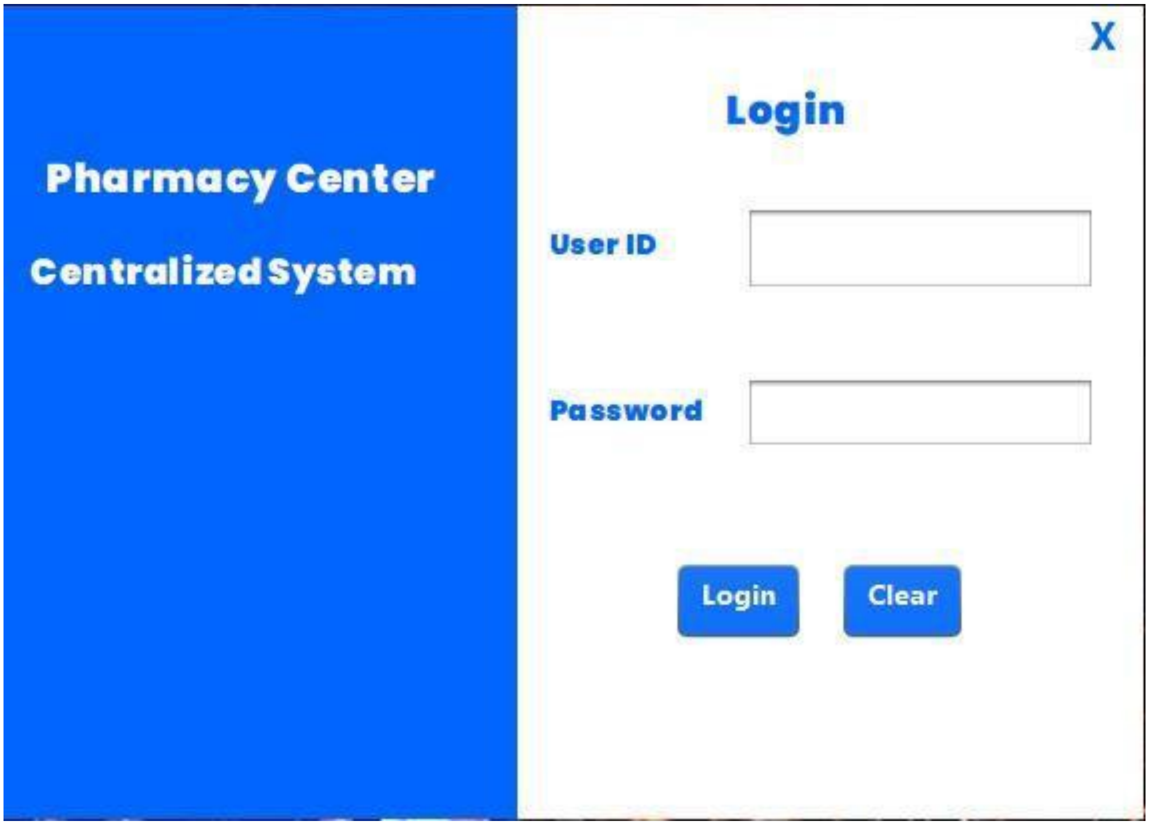


CODES

```
public class SPLASH extends javax.swing.JFrame {

  public SPLASH() {
     initComponents();
  private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
     System.exit(0);
   }


  public static void main(String args[]) {

   SPLASH Mysplash = new SPLASH();
   Mysplash.setVisible(true);
   try
   {
   for (int i = 0; i<= 100; i++)
   {
   Thread.sleep(10000);
   Mysplash.Myprogress.setValue(i);
   Mysplash.Percentage.setText(Integer.toString(i)+"%");
   }
   }
```

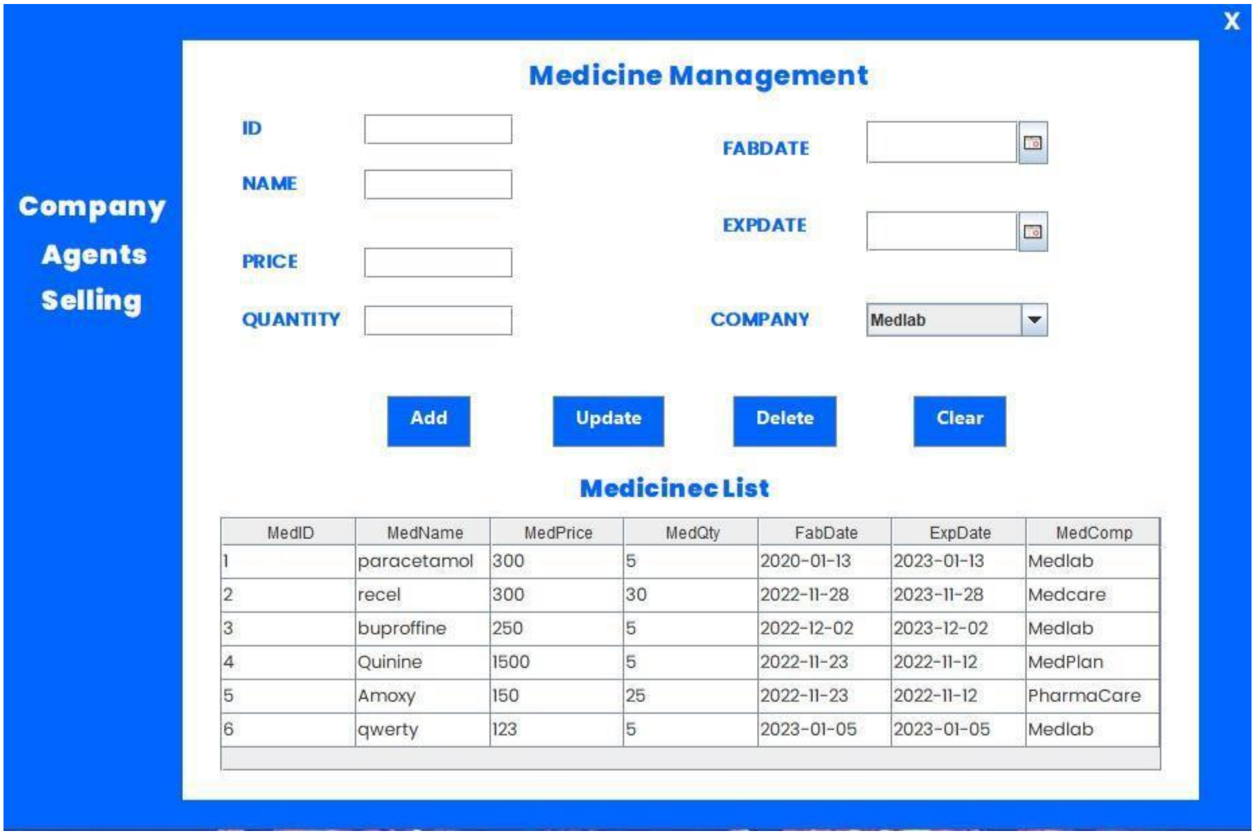
```
    catch (Exception e)
    {
    }
    Mysplash.dispose();
    new Login().setVisible(true);
   }
}
```

The login page



```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;
import java.sql.SQLException;
import java.sql.ResultSet;

/**
 *
 * @author RWIGEMA
 P.Christian */
public class Login extends javax.swing.JFrame {
 public Login() {
      initComponents();
   }

Connection Con = null;
Statement St = null;
ResultSet Rs = null;
   private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {
      String Query = "select * from agenttbl where AName = '" +Uid.getText()+"' and APass =
'"+Pass.getText()+"'";
      try
      {
       Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
       St = Con.createStatement();
       Rs = St.executeQuery(Query);
       if (Rs.next())
       {
          new Medicine().setVisible(true);
          this.dispose();
       }
       else
       {
       JOptionPane.showMessageDialog(this, "WRONG PASSWORD");
       }
      }
      catch(SQLException e)
      {
      e.printStackTrace();
      }
   }

    private void jLabel7MouseClicked(java.awt.event.MouseEvent evt) {
       System.exit(0);
   }

    private void ClearBtnMouseClicked(java.awt.event.MouseEvent evt) {
       Uid.setText("");
       Pass.setText("");
   }

    /**
     * @param args the command line arguments
 */
public static void main(String args[]) {
  java.awt.EventQueue.invokeLater(new Runnable() {
       public void run() {
          new Login().setVisible(true);
       }
    });
  }
}
```

The medicine management page



Medicine Management

COMPANY: Medlab

Medicinec List

MedID	MedName	MedPrice	MedQty	FabDate	ExpDate	MedComp
1	paracetamol	300	5	2020-01-13	2023-01-13	Medlab
2	recel	300	30	2022-11-28	2023-11-28	Medcare
3	buproffine	250	5	2022-12-02	2023-12-02	Medlab
4	Quinine	1500	5	2022-11-23	2022-11-12	MedPlan
5	Amoxy	150	25	2022-11-23	2022-11-12	PharmaCare
6	qwerty	123	5	2023-01-05	2023-01-05	Medlab

CODES

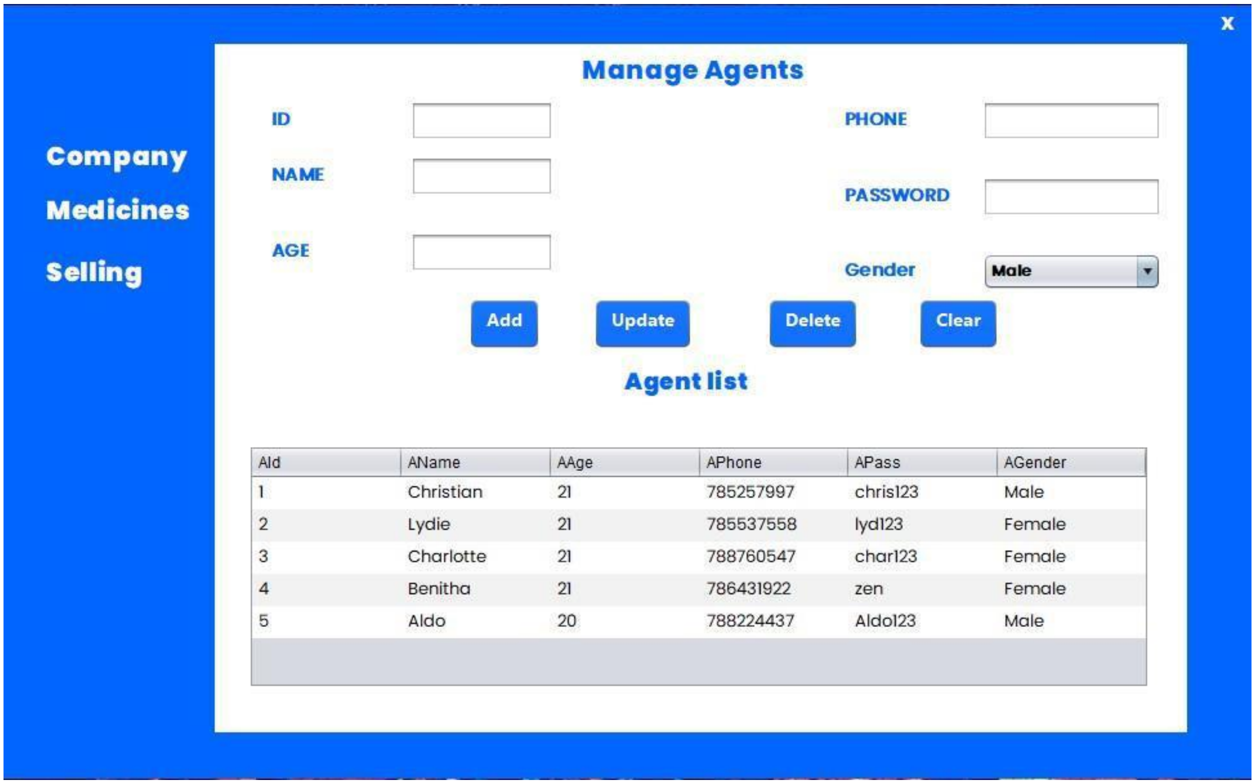
```
import java.sql.Statement;
import java.sql.Connection;
import java.sql.SQLException;
import java.sql.ResultSet;
import java.sql.PreparedStatement;
import java.sql.DriverManager;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;
public class Medicine extends javax.swing.JFrame {

  public Medicine() {
    initComponents();
    SelectMed();
    GetCompany();
  }
Connection Con = null;
Statement St = null;
ResultSet Rs = null;

java.util.Date FDate,EDate;
java.sql.Date MyFabDate, MyExpDate;
public void SelectMed()
{
  try {
  Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
  St = Con.createStatement();
  Rs = St.executeQuery("Select * from medicinetbl");
  MedicineTable.setModel(DbUtils.resultSetToTableModel(Rs));

  }
  catch(SQLException e)
      {
        e.printStackTrace();
      }
}
  public void GetCompany()
  {
  try
  {
    Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
    St = Con.createStatement();
    String query = "Select * from companytbl";
    Rs = St.executeQuery(query);
    while(Rs.next())
    {
      String MyComp = Rs.getString("CompName");
      CompCb.addItem(MyComp);
      }
    }
    catch(SQLException e)
    {
      e.printStackTrace();
    }
    }

    private void AddBtn1MouseClicked(java.awt.event.MouseEvent evt) {
 FDate = FabDate.getDate();
 MyFabDate = new java.sql.Date(FDate.getTime());
  EDate = ExpDate.getDate();
 MyExpDate = new java.sql.Date(EDate.getTime());

      try {
    Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
    PreparedStatement add = Con.prepareStatement("insert into medicinetbl values
(?,?,?,?,?,?,?)");
    add.setInt(1, Integer.valueOf(MedId.getText()));
   add.setString(2,MedName.getText());
   add.setInt(3,Integer.valueOf(MedPrice.getText()));
   add.setInt(4,Integer.valueOf(MedQty.getText()));
   add.setDate(5, MyFabDate);
    add.setDate(6, MyExpDate);
    add.setString(7, CompCb.getSelectedItem().toString());
    int row = add. executeUpdate();
    JOptionPane.showMessageDialog(this, "Medicine Added Successfully");
    Con.close();
    SelectMed();
 }
 catch(SQLException e)
       {
         e.printStackTrace();
       }
    }

    private void DeleteBtnMouseClicked(java.awt.event.MouseEvent evt) { if
(MedId.getText().isEmpty())
{
   JOptionPane.showMessageDialog(this, "Enter Medicine to be Deleted");
}
else
```

```
{
 try
 {
    Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
    String Id = MedId.getText();
    String Query = "Delete from medicinetbl where MedId="+Id;
    Statement Add = Con.createStatement();
    Add.executeUpdate(Query);
    SelectMed();
    JOptionPane.showMessageDialog(this, "Medicine Deleted successfully");
 }
 catch(SQLException e)
 {
    e.printStackTrace();
 }
}
   }

   private void MedicineTableMouseClicked(java.awt.event.MouseEvent evt) {
DefaultTableModel model = (DefaultTableModel)MedicineTable.getModel(); int
Myindex = MedicineTable.getSelectedRow();
MedId.setText(model.getValueAt(Myindex, 0).toString());
MedName.setText(model.getValueAt(Myindex, 1).toString());
MedPrice.setText(model.getValueAt(Myindex, 2).toString());
MedQty.setText(model.getValueAt(Myindex, 3).toString());
   }

   private void UpdateBtnMouseClicked(java.awt.event.MouseEvent evt) {
      if(MedId .getText().isEmpty() || MedName.getText().isEmpty() ||
MedPrice.getText().isEmpty() || MedQty.getText().isEmpty())
      {
        JOptionPane.showMessageDialog(this, "Missing Innformation");
      }
      else
      {
        try
        {
         FDate = FabDate.getDate();
    MyFabDate = new java.sql.Date(FDate.getTime());
    EDate = ExpDate.getDate();
    MyExpDate = new java.sql.Date(EDate.getTime());
```

```
        Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
        String Id = MedId.getText();
        String UpdateQuery = "Update medicinetbl set MedName =
'"+MedName.getText()+"',MedPrice = '"+ MedPrice.getText()+"',MedQty =
'"+MedQty.getText()+"',FabDate = '"+MyFabDate+"',ExpDate = '"+MyExpDate+"',MedComp
= '"+ CompCb.getSelectedItem().toString()+"' where MedID ="+Id; Statement
        Add = Con.createStatement(); Add.executeUpdate(UpdateQuery);
        JOptionPane.showMessageDialog(this, "Medicine Update successfully");

          }
          catch (SQLException e)
          {
             e.printStackTrace();
          }
          SelectMed();
        }
    }

    private void ClearBtnMouseClicked(java.awt.event.MouseEvent evt) {
        MedId.setText("");
        MedName.setText("");
        MedPrice.setText("");
        MedQty.setText("");
    }

    private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
        new Company().setVisible(true);
        this.dispose();
    }

    private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
         new Agents().setVisible(true);
        this.dispose();
    }

    private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
        new Selling ().setVisible(true);
        this.dispose();
    }

    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
        System.exit(0);
```

```
  }

  /**
   * @param args the command line arguments */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
      public void run() {
        new Medicine().setVisible(true);
      }
    });
  }
```

The Agents management page.



Gender Male

AId	AName	AAge	APhone	APass	AGender
1	Christian	21	785257997	chris123	Male
2	Lydie	21	785537558	lyd123	Female
3	Charlotte	21	788760547	char123	Female
4	Benitha	21	786431922	zen	Female
5	Aldo	20	788224437	Aldo123	Male

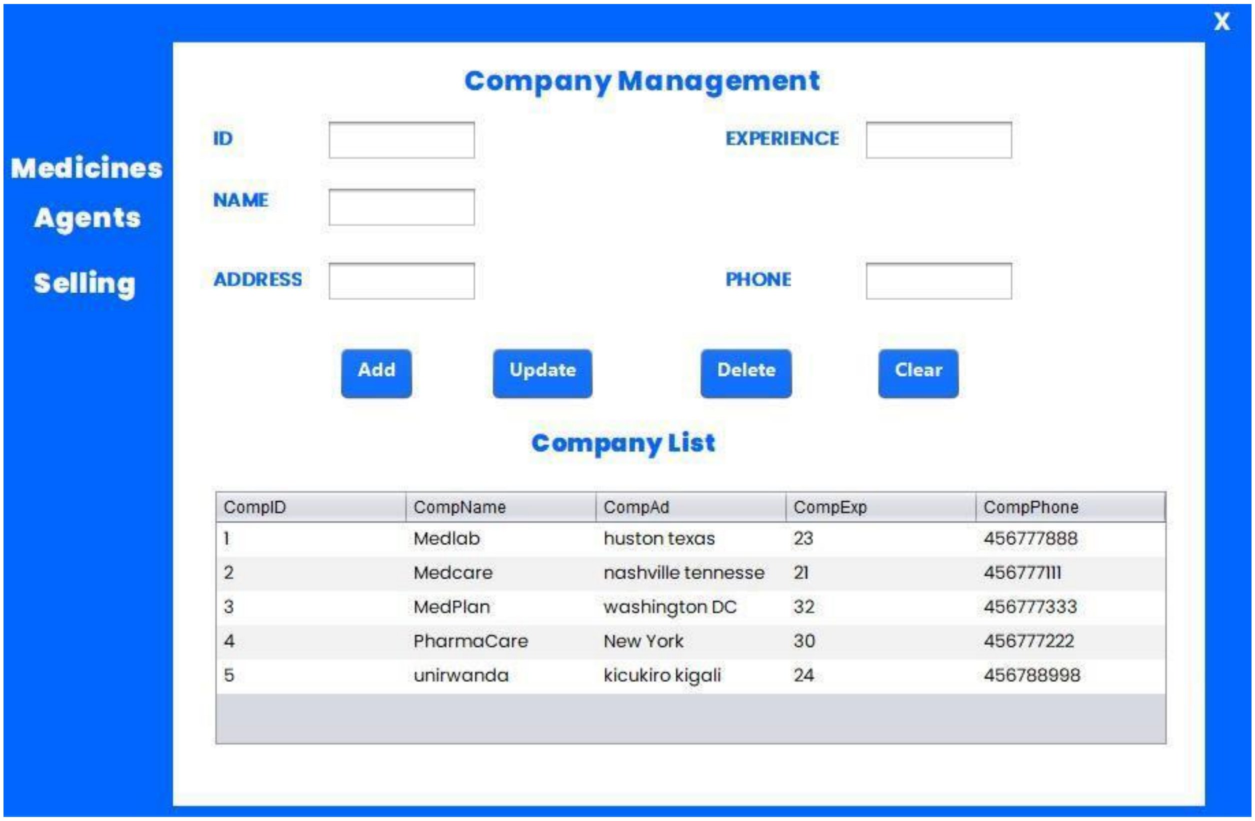
CODES

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author RWIGEMA
 P.Christian */
public class Agents extends javax.swing.JFrame {

   /**
    * Creates new form Agents
    */
   public Agents() {
      initComponents();
      SelectAgent();
   }

Connection Con = null;
Statement St = null;
java.sql.ResultSet Rs = null;
public void SelectAgent()
{
   try {
   Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
   St = Con.createStatement();
   Rs = St.executeQuery("Select * from agenttbl");
   AgentTable.setModel(DbUtils.resultSetToTableModel(Rs));

   }
   catch(SQLException e)
         {
            e.printStackTrace();
         }
}
   private void AddBtnMouseClicked(java.awt.event.MouseEvent evt) {
      try {
    Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
    PreparedStatement add = Con.prepareStatement("insert into agenttbl values
(?,?,?,?,?,?)");
    add.setInt(1, Integer.valueOf(AId.getText()));
    add.setString(2,AName.getText());
    add.setInt(3,Integer.valueOf(Aage.getText()));
    add.setString(4,Aphone.getText());
    add.setString(5,Apass.getText());
    add.setString(6, GenderCb.getSelectedItem().toString());
    int row = add. executeUpdate();
    JOptionPane.showMessageDialog(this, "Agent Added Successfully");
    Con.close();
   SelectAgent();
 }
 catch(SQLException e)
       {
          e.printStackTrace();
       }
   }

   private void ClearBtnMouseClicked(java.awt.event.MouseEvent evt) {
     AId.setText("");
     AName.setText("");
     Aage.setText("");
     Aphone.setText("");
     Apass.setText("");

   }

   private void DeleteBtnMouseClicked(java.awt.event.MouseEvent evt) { if
      (AId.getText().isEmpty())
{
   JOptionPane.showMessageDialog(this, "Enter Agent to be Deleted");
}
else
{
  try
  {
     Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
     String Id = AId.getText();
     String Query = "Delete from agenttbl where AId="+Id;
     Statement Add = Con.createStatement();
     Add.executeUpdate(Query);
     SelectAgent();
     JOptionPane.showMessageDialog(this, "Agent Deleted successfully");
  }
  catch(SQLException e)
 {
   e.printStackTrace();
 }
}
  }

  private void AgentTableMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel)AgentTable.getModel();
int Myindex = AgentTable.getSelectedRow();
AId.setText(model.getValueAt(Myindex, 0).toString());
AName.setText(model.getValueAt(Myindex, 1).toString());
Aage.setText(model.getValueAt(Myindex, 2).toString());
Aphone.setText(model.getValueAt(Myindex, 3).toString());
Apass.setText(model.getValueAt(Myindex, 4).toString());
  }

  private void UpdateBtnMouseClicked(java.awt.event.MouseEvent evt) {
          if(AId .getText().isEmpty() || AName.getText().isEmpty() || Aage.getText().isEmpty()
|| Aphone.getText().isEmpty() || Apass.getText().isEmpty())
    {
      JOptionPane.showMessageDialog(this, "Missing Innformation");
    }
    else
    {
      try
      {
     Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
     String Id = AId.getText();
     String UpdateQuery = "Update agenttbl set AName = '"+AName.getText()+"',AAge= '"+
Aage.getText()+"',APhone = '"+Aphone.getText()+"',APass= '"+Apass.getText()+"',AGender
= '"+ GenderCb.getSelectedItem().toString()+"' where AId ="+Id; Statement
     Add = Con.createStatement(); Add.executeUpdate(UpdateQuery);
     JOptionPane.showMessageDialog(this, "Agent Update successfully");

      }
      catch (SQLException e)
      {
        e.printStackTrace();
      }
      SelectAgent();
    }
  }
```

```
private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
  new Company().setVisible(true);
  this.dispose();
}

private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
  new Medicine().setVisible(true);
  this.dispose();
}

private void GenderCbActionPerformed(java.awt.event.ActionEvent evt) { //
  TODO add your handling code here:
}

private void UpdateBtnActionPerformed(java.awt.event.ActionEvent evt) { //
  TODO add your handling code here:
}

private void ClearBtnActionPerformed(java.awt.event.ActionEvent evt) { //
  TODO add your handling code here:
}

private void AIdActionPerformed(java.awt.event.ActionEvent evt) { //
  TODO add your handling code here:
}

private void jLabel14MouseClicked(java.awt.event.MouseEvent evt) {
  new Selling ().setVisible(true);
  this.dispose();
}

private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
  System.exit(0);
}

/**
 * @param args the command line arguments */
public static void main(String args[]) {


java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
      new Agents().setVisible(true);
```

```
            }
        });
    }
}
```

The company management page.



CompID	CompName	CompAd	CompExp	CompPhone
1	Medlab	huston texas	23	456777888
2	Medcare	nashville tennesse	21	456777111
3	MedPlan	washington DC	32	456777333
4	PharmaCare	New York	30	456777222
5	unirwanda	kicukiro kigali	24	456788998

CODES

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author RWIGEMA
 P.Christian */
```

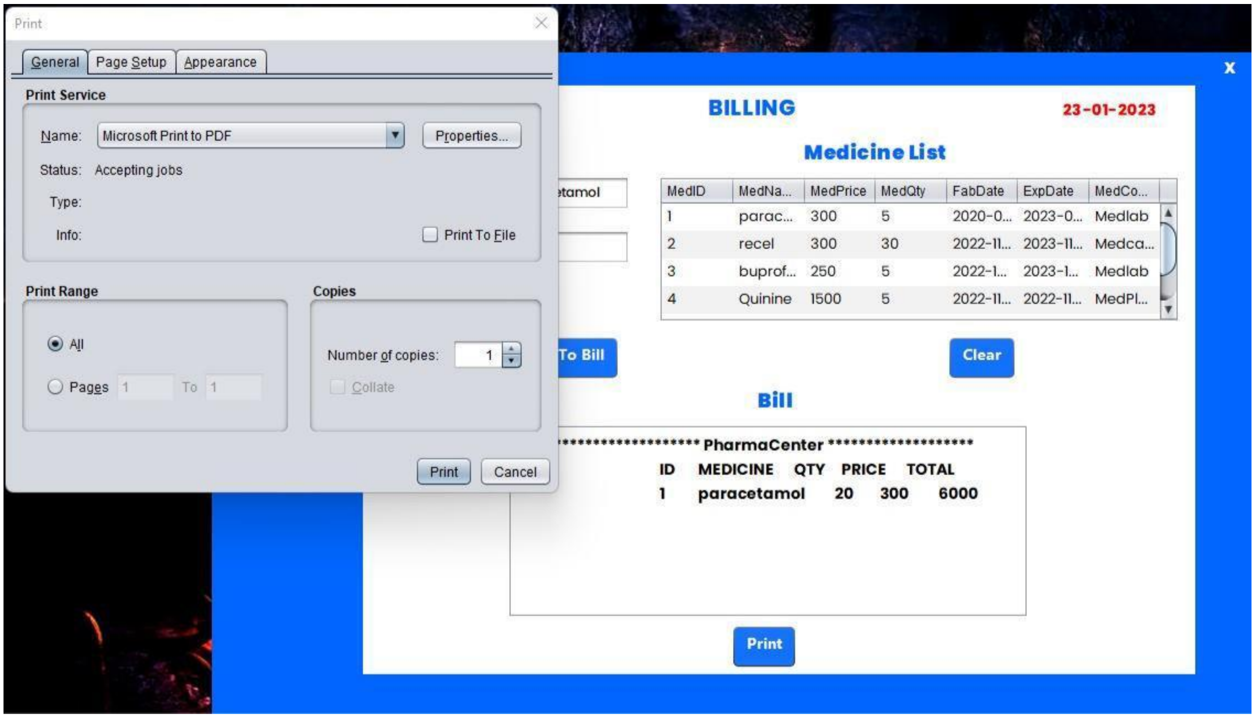
```
public class Company extends javax.swing.JFrame {

   /**
    * Creates new form Company
    */
   public  Company()  {
      initComponents();
      SelectCompany();
   }

Connection Con = null;
Statement St = null;
java.sql.ResultSet Rs = null;
public void SelectCompany()
{
   try {
   Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
   St = Con.createStatement();
   Rs = St.executeQuery("Select * from companytbl");
   CompanyTable.setModel(DbUtils.resultSetToTableModel(Rs));

   }
   catch(SQLException e)
         {
            e.printStackTrace();
         }
}
   private void AddBtnMouseClicked(java.awt.event.MouseEvent evt) {
          try {
    Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
    PreparedStatement add = Con.prepareStatement("insert into companytbl values
(?,?,?,?,?)");
    add.setInt(1, Integer.valueOf(CompId.getText()));
    add.setString(2,Compname.getText());
    add.setString(3,Compad.getText());
    add.setInt(4, Integer.valueOf(Compexp.getText()));
    add.setString(5,Compphone.getText());
    int row = add. executeUpdate();
    JOptionPane.showMessageDialog(this, "Company Added Successfully");
    Con.close();
```

```
  SelectCompany();
 }
 catch(SQLException e)
     {
       e.printStackTrace();
     }
  }

  private void DeleteBtnMouseClicked(java.awt.event.MouseEvent evt) { if
        (CompId.getText().isEmpty())
{
  JOptionPane.showMessageDialog(this, "Enter company to be Deleted");
}
else
{
 try
 {
   Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
   String Id = CompId.getText();
   String Query = "Delete from companytbl where CompID="+Id;
   Statement Add = Con.createStatement();
   Add.executeUpdate(Query);
   SelectCompany();
   JOptionPane.showMessageDialog(this, "Company Deleted successfully");
 }
 catch(SQLException e)
 {
   e.printStackTrace();
 }
}
  }

  private void UpdateBtnMouseClicked(java.awt.event.MouseEvent evt) {
            if(CompId .getText().isEmpty() || Compname.getText().isEmpty() ||
Compad.getText().isEmpty() || Compexp.getText().isEmpty() ||
Compphone.getText().isEmpty())
    {
      JOptionPane.showMessageDialog(this, "Missing Innformation");
    }
    else
    {
      try
```

```
         {
       Con = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
       String Id = CompId.getText();
       String UpdateQuery = "Update companytbl set CompName = 
'"+Compname.getText()+"',CompAd = '"+ Compad.getText()+"',CompExp = 
'"+Compexp.getText()+"',CompPhone= '"+Compphone.getText()+"' where CompID ="+Id;
       Statement Add = Con.createStatement();
       Add.executeUpdate(UpdateQuery);
       JOptionPane.showMessageDialog(this, "Company Update successfully");
         }
         catch (SQLException e)
         {
            e.printStackTrace();
         }
         SelectCompany();
      }
    }

    private void ClearBtnMouseClicked(java.awt.event.MouseEvent evt) {
       CompId.setText("");
       Compname.setText("");
       Compad.setText("");
       Compexp.setText("");
       Compphone.setText("");

    }

    private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
       new Medicine().setVisible(true);
       this.dispose();
    }

    private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
       new Agents().setVisible(true);
       this.dispose();
    }

    private void CompanyTableMouseClicked(java.awt.event.MouseEvent evt) {
       DefaultTableModel model = (DefaultTableModel)CompanyTable.getModel();
       int Myindex = CompanyTable.getSelectedRow();
       CompId.setText(model.getValueAt(Myindex, 0).toString());
       Compname.setText(model.getValueAt(Myindex, 1).toString());
```

```
        Compad.setText(model.getValueAt(Myindex, 2).toString());
        Compexp.setText(model.getValueAt(Myindex, 3).toString());
        Compphone.setText(model.getValueAt(Myindex, 4).toString());
    }

    private void CompnameActionPerformed(java.awt.event.ActionEvent evt) { //
        TODO add your handling code here:
    }

    private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
        new Selling ().setVisible(true);
        this.dispose();
    }

    private void jLabel13MouseClicked(java.awt.event.MouseEvent evt) {
        System.exit(0);
    }

    /**
     * @param args the command line arguments */
    public static void main(String args[]) {
     java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Company().setVisible(true);
            }
        });
    }
}
```

The selling management page.



Print Service

Name: Microsoft Print to PDF

Status: Accepting jobs

Type:

Info:

☐ Print To File

Print Range

● All

○ Pages 1 To 1

Copies

Number of copies: 1

☐ Collate

BILLING

23-01-2023

Medicine List

etamol

MedID	MedNa...	MedPrice	MedQty	FabDate	ExpDate	MedCo...
1	parac...	300	5	2020-0...	2023-0...	Medlab
2	recel	300	30	2022-11...	2023-11...	Medca...
3	buprof...	250	5	2022-1...	2023-1...	Medlab
4	Quinine	1500	5	2022-11...	2022-11...	MedPl...

Bill

```
******************* PharmaCenter *******************
ID   MEDICINE     QTY   PRICE   TOTAL
1    paracetamol   20    300     6000
```

CODES

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.SimpleDateFormat;
import java.util.Date;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import net.proteanit.sql.DbUtils;

/**
 *
 * @author RWIGEMA
 P.Christian */
public class Selling extends javax.swing.JFrame {

   /**
    * Creates new form Selling
    */
   public Selling() {
      initComponents();
      SelectMed();
      ShowDate();

   }
   public void ShowDate()
   {
   Date d = new Date();
   SimpleDateFormat s = new SimpleDateFormat("dd-MM-yyyy");
   DateLbl.setText(s.format(d));
   }
Connection Con = null;
Statement St = null;
java.sql.ResultSet Rs = null;
public void SelectMed()
{
   try {
   Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
   St = Con.createStatement();
   Rs = St.executeQuery("Select * from medicinetbl");
  MedicineTable.setModel(DbUtils.resultSetToTableModel(Rs));

   }
   catch(SQLException e)
         {
            e.printStackTrace();
         }
}
   private void PrintBtnMouseClicked(java.awt.event.MouseEvent evt) {
     /* if(AId .getText().isEmpty() || AName.getText().isEmpty() || Aage.getText().isEmpty() ||
Aphone.getText().isEmpty() || Apass.getText().isEmpty())
      {
         JOptionPane.showMessageDialog(this, "Missing Innformation");
      }
      else
      {
         try
         {
            Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
            String Id = AId.getText();
            String UpdateQuery = "Update agenttbl set AName = '"+AName.getText()+"',AAge=
'"+ Aage.getText()+"',APhone = '"+Aphone.getText()+"',APass= '"+Apass.getText()+"',AGender
= '"+ GenderCb.getSelectedItem().toString()+"' where AId ="+Id;
            Statement Add = Con.createStatement();
            Add.executeUpdate(UpdateQuery);
            JOptionPane.showMessageDialog(this, "Agent Update successfully");
         }
         catch (SQLException e)
         {
            e.printStackTrace();
         }
         SelectSells();
      }*/
     try
     {
     BillTxt.print();
     }
     catch(Exception e)
     {
     e.printStackTrace();
     }

   }

   private void PrintBtnActionPerformed(java.awt.event.ActionEvent evt) { //
      TODO add your handling code here:
   }

   private void ClearBtnMouseClicked(java.awt.event.MouseEvent evt) {
      MedText.setText("");
      Qty.setText("");

   }

   private void ClearBtnActionPerformed(java.awt.event.ActionEvent evt) { //
      TODO add your handling code here:
   }
public void Update ()
{
   int newQty;
   newQty = Q1dQty - Integer.valueOf(Qty.getText());
   try
   {
```

```
Con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/pharmacydb?zeroDateTimeBeh
avior=CONVERT_TO_NULL" ,"root", "");
       String UpdateQuery = "Update medicinetbl set MedQty = "+ newQty +" where MedID
="+ Medid;
       Statement Add = Con.createStatement();
       Add.executeUpdate(UpdateQuery);
       JOptionPane.showMessageDialog(this, "Medicine Update successfully");
         }
         catch (SQLException e)
         {
            e.printStackTrace();
         }
         SelectMed();
}
   int i = 0, price, Medid , Q1dQty;
   private void AddBtnMouseClicked(java.awt.event.MouseEvent evt) {

      if (MedText.getText().isEmpty() || Qty.getText().isEmpty())
      {
      JOptionPane.showMessageDialog(this, "Missing Information");
      }

      else {
      i++;
      Update();
      if(i == 1)
      {
      BillTxt.setText(BillTxt.getText() +
            "\n\t ID       MEDICINE   QTY   PRICE    TOTAL\n \t"
             + " "+
            i +" "+ MedText.getText() + "
            " +
            Qty.getText() + "       " + price +          " + Integer.valueOf(Qty.getText())*price +
            "
"\n");
      }
      else
      {
      BillTxt.setText(BillTxt.getText() +"\t  "+ i + "         " + MedText.getText() +"         "
+Qty.getText() + "       " + price + "       " + Integer.valueOf(Qty.getText())*price+"\n");
      }
      }
```

```
 }

 private void jLabel2MouseClicked(java.awt.event.MouseEvent evt) {
    new Company().setVisible(true);
    this.dispose();
 }

 private void jLabel4MouseClicked(java.awt.event.MouseEvent evt) {
    new Medicine().setVisible(true);
    this.dispose();
 }

 private void MedTextActionPerformed(java.awt.event.ActionEvent evt) { //
    TODO add your handling code here:
 }

 private void QtyActionPerformed(java.awt.event.ActionEvent evt) { //
    TODO add your handling code here:
 }

 private void MedicineTableMouseClicked(java.awt.event.MouseEvent evt) {
    DefaultTableModel model = (DefaultTableModel)MedicineTable.getModel();
    int Myindex = MedicineTable.getSelectedRow();
   // MedId.setText(model.getValueAt(Myindex, 0).toString());
    MedText.setText(model.getValueAt(Myindex, 1).toString()); Medid =
    Integer.valueOf(model.getValueAt(Myindex, 0).toString()); price =
    Integer.valueOf(model.getValueAt(Myindex, 2).toString()); Q1dQty =
    Integer.valueOf(model.getValueAt(Myindex, 3).toString());
 }

 private void jLabel1MouseClicked(java.awt.event.MouseEvent evt) {
    System.exit(0);
 }

 private void jLabel14MouseClicked(java.awt.event.MouseEvent evt) {
    new Agents().setVisible(true);
    this.dispose();
 }

 /**
  * @param args the command line arguments */
 public static void main(String args[]) {
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
                new Selling().setVisible(true);
            }
        });
    }
```

Stage 4: TESTING

System testing

System Testing can be done at two stages namely:-

1. Unit Testing, the system is tested in modules before integration is done. This is important as faults are discovered before the systems complexity increases through system integration.
2. System Testing, the system is tested for conformity with requirements after all modules have been put together and the system as a whole is tested to authenticate that general system requirements have been met.

In the development of the System, various criterions were used as testing yardsticks of the system. These yardsticks include Graphic User Interface, usability testing, database and exception handling (Martin, 1991). Both the black-box and white-box methods were used in the unit testing.

Graphical user interface testing

The Graphical User Interface (GUI) testing involves testing the systems graphic components to ensure that it covers the entire domain i.e. the complete functionality of the system, with respect to the different modules. For example, in the testing of the System, both customer service and administrator modules were tested to ensure that they all contained links and tables that directly links to their requirements as designed with use cases in the preceding section. There also comes a stage of sequencing testing with GUI testing. This ensures that domains of requirements have an efficient follow up order, that is, users are less likely going to be confused or going to find it difficult to navigate around the system in search of how to perform some sort of operation. For example, how the user can view details of a product and the manufacturer of the product.

One of the most important steps when you test a Pharmacy management system is determining the right solution—with the right features—for your pharmacy. Ensure the system you choose can grow and scale with you and has all of the features and integrations you need to be successful. For long-term care pharmacies, that means looking for a platform that offers:

- Workflow automation capabilities
- Real-time data and reporting
- Facility-centric customization
- Sophisticated billing functionality
- Quality checks and balances
- Scalability to support growth
- Seamless system integrations
- Regulatory and auditing support

Stage 5: DEPLOYMENT

Our application (Pharmacy Management System) is locally run and as such it is deployed on the localhost server (**localhost 3306**) of the device. To insure that everything runs smoothly, the following applications were installed:

- **MySQL database server.**
- **XAMPP control panel.**
- **JDK-19.**
- **Apache NetBeans IDE 15.**
- **Mysql-connector-j-8.0.31.jar.**
- **Commons-dbutils-1.5-sources.jar**
- **jcalendar-1.4.jar.**
- **rs2xml.jar.**

Each one of these applications and files had to be installed on every computer/device to ensure the pharmacy Management System application could successfully.

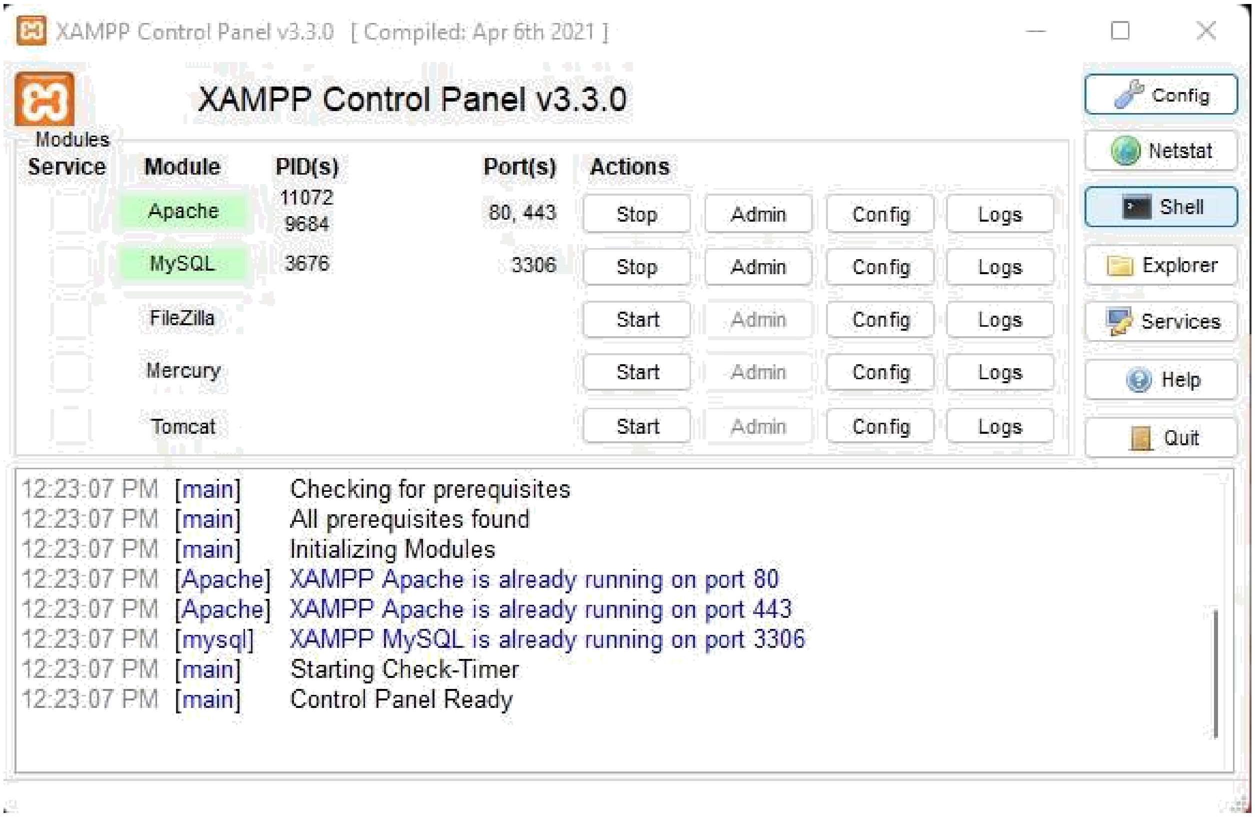
Stage 6: MAINTAINACE

. The application was originally built in my computer as such, everything wither it be installation, testing, deployment and performance monitoring, was first operated within my device.

We begin with the installation of all the necessary applications i.e. **MySQL database server, XAMPP control panel,JDK-19 & Apache NetBeans IDE 15** necessary to build the application. Next, we had to install other additional files to ensure the application run smoothly. This are: **Mysql-connector-j-8.0.31.jar, Commons-dbutils-1.5-sources.jar, jcalendar-1.4.jar & rs2xml.jar.**

Next we have the deployment phase which constitutes the actual building of the application's frontend and back-end using the applications listed above.

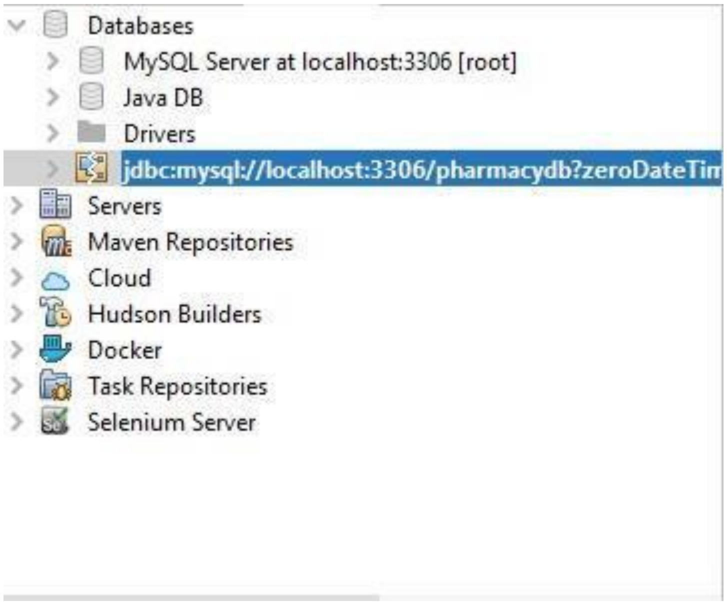
Next we have the testing phase which is meant to ensure that the applications works and performs tasks as intended. During this phase we had to ensure that Mysql server was up and running (we had to start the server and Apache) on the XAMPP Control Panel and make sure that the project was connected to the server. The following images highlight how to do that


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

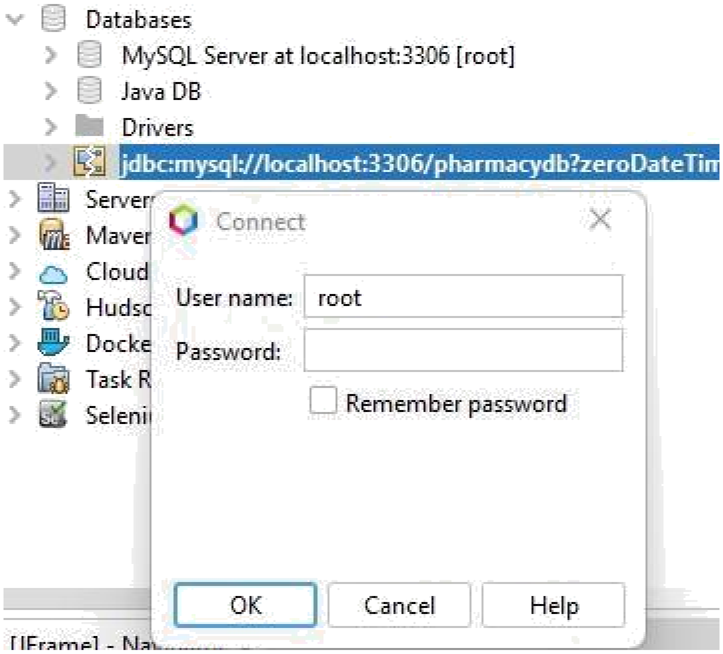
Service	Module	PID(s)	Port(s)	Actions			
	Apache	11072 9684	80, 443	Stop	Admin	Config	Logs
	MySQL	3676	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:23:07 PM  [main]     Checking for prerequisites
12:23:07 PM  [main]     All prerequisites found
12:23:07 PM  [main]     Initializing Modules
12:23:07 PM  [Apache]   XAMPP Apache is already running on port 80
12:23:07 PM  [Apache]   XAMPP Apache is already running on port 443
12:23:07 PM  [mysql]    XAMPP MySQL is already running on port 3306
12:23:07 PM  [main]     Starting Check-Timer
12:23:07 PM  [main]     Control Panel Ready
```

In this image we see that both Apache and MySQL are started and running.



In this image we see that the application is not connected to the server.

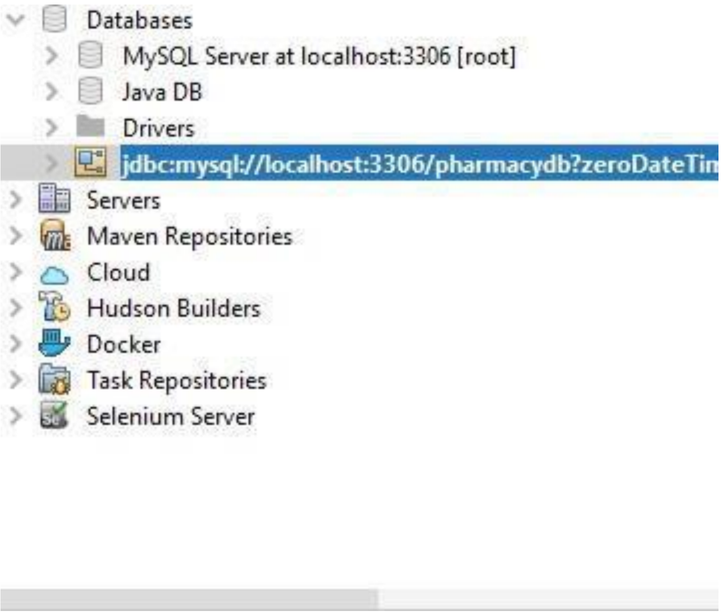


User name: root

Password:

□ Remember password

To connect to the server, right click on the database name, click on connect, enter the user name and password and press ok to connect. Make sure the server is up and running (started) first.



This image shows a successful connection.

Lastly, the application was run for 25 consecutive with no signs of disruptions, bugs or glitches days to ensure the performance were optimal and that the application was working smoothly.