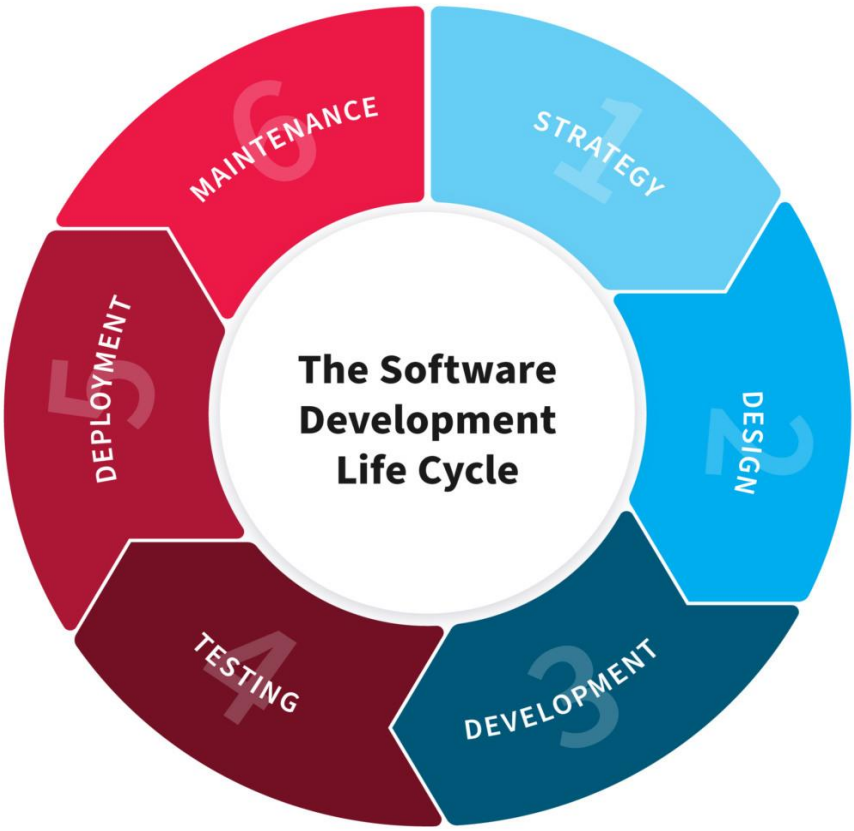
Names: **RWIGEMA Pierre Christian**
Reg No: **221015356**
College of Business and Economics
School of Economics
Business Information Technology department (BIT)
Level 2
Class Group: **Group 3**

23rd/01/2023

Project Name: **Pharmacy Management System**

Below is a description of how our team 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 main goal of the system is to insure that pharmacist/user will be able to access data concerning drugs at any time and apply changes swiftly and effortlessly.

The main objective of the system is to provide a safe platform for pharmacist to perform their everyday work with simplicity and more effectively which would improve the overall process of service delivery in the healthcare system.

Stage 2: DESIGN

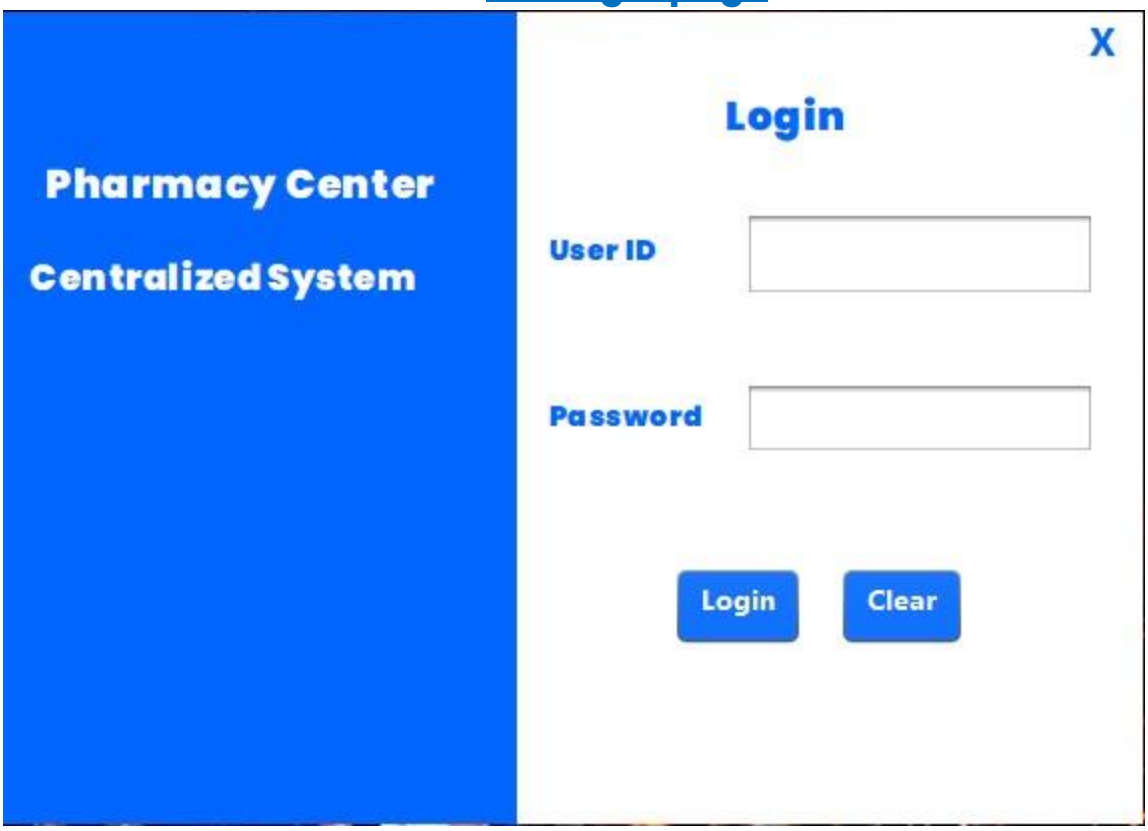
Our pharmacy management system aimed and simplicity and efficacy during the implementation of its design to insure a user friendly experience. Below is a description of every page with details on the functianilty of every aspect of the design.

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

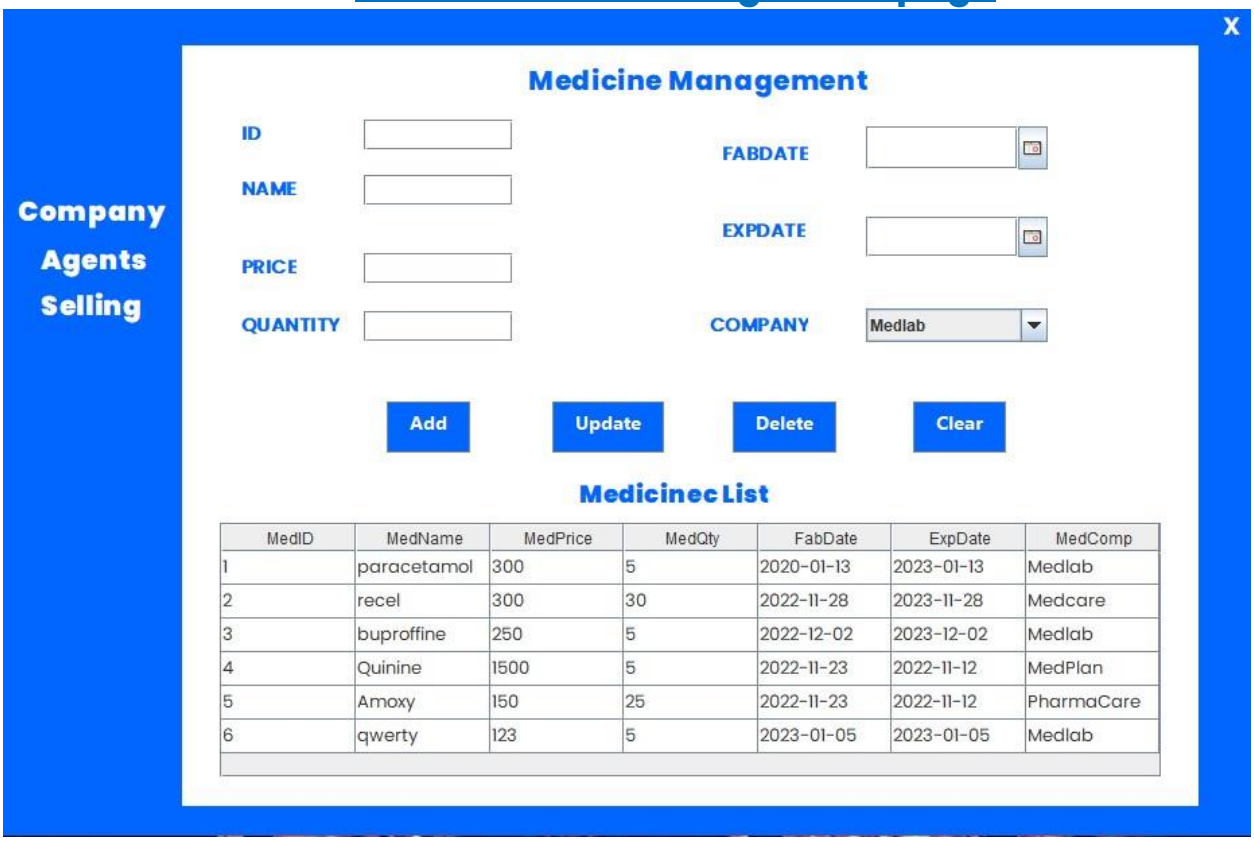


The login page contains

- 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

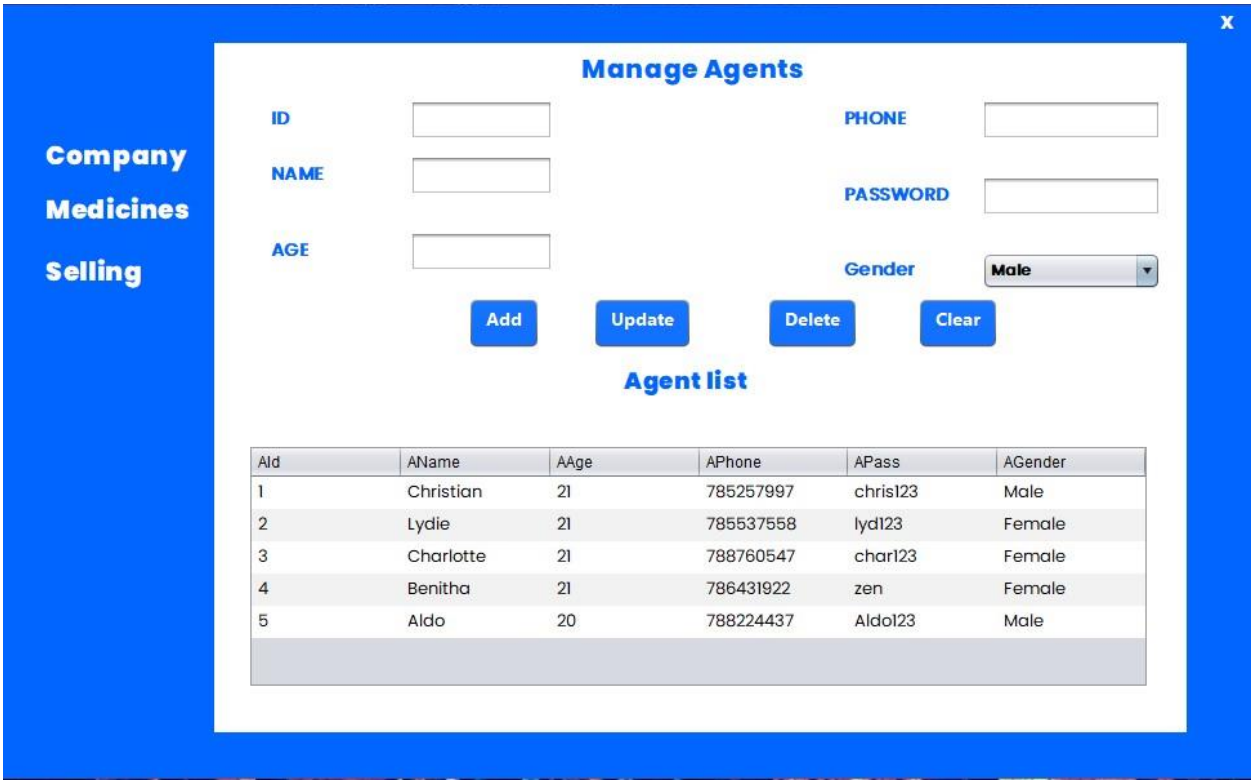


The medicine management page provides the user with the possibility to operate on drugs and access any information concerning it.
The following functionalities are provided:

- 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.



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.

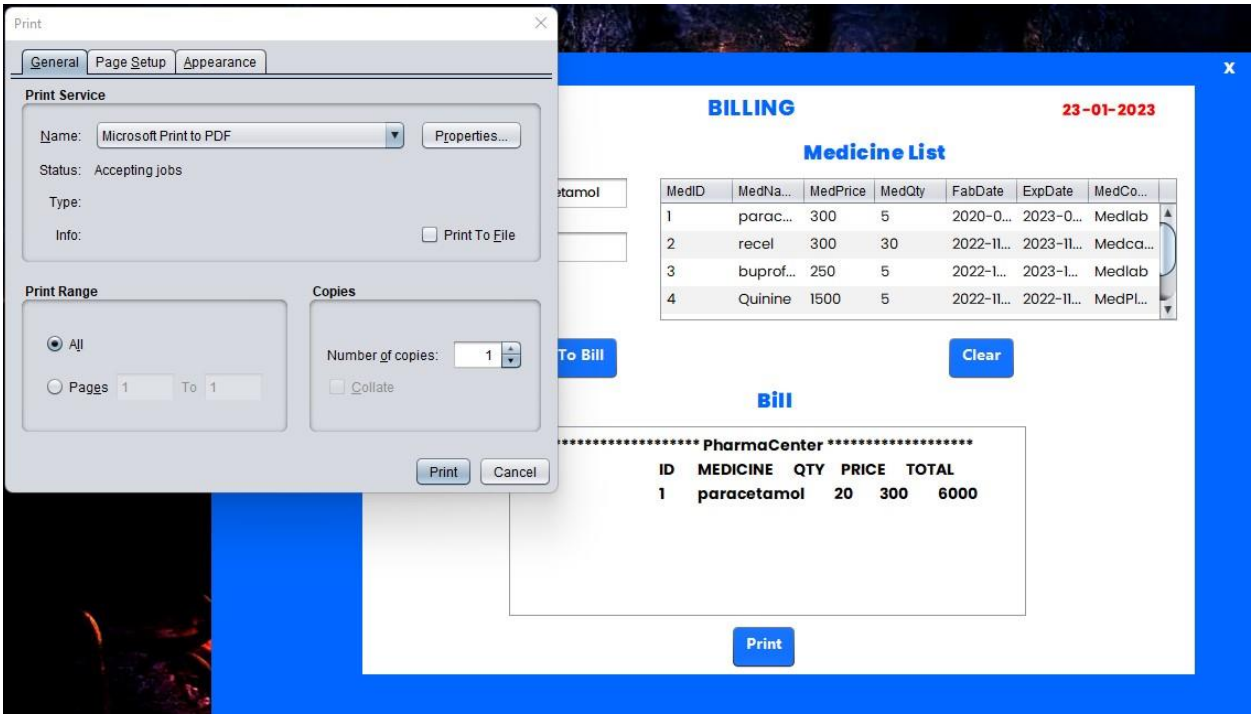
The company management page.

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.



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

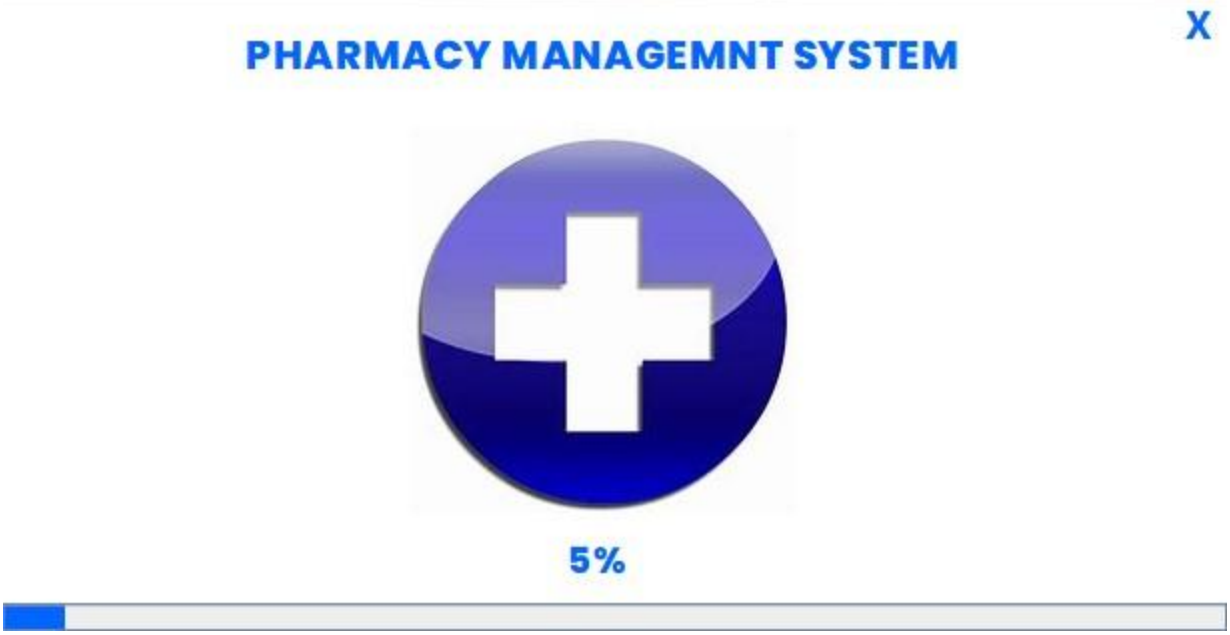
As mentioned previously in the strategy phase of the application development life cycle, our application is a java based web application. Both the front-end and backend were implemented with the help of Apache NetBeans IDE version 15 application.
The following libraries were used:

- **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

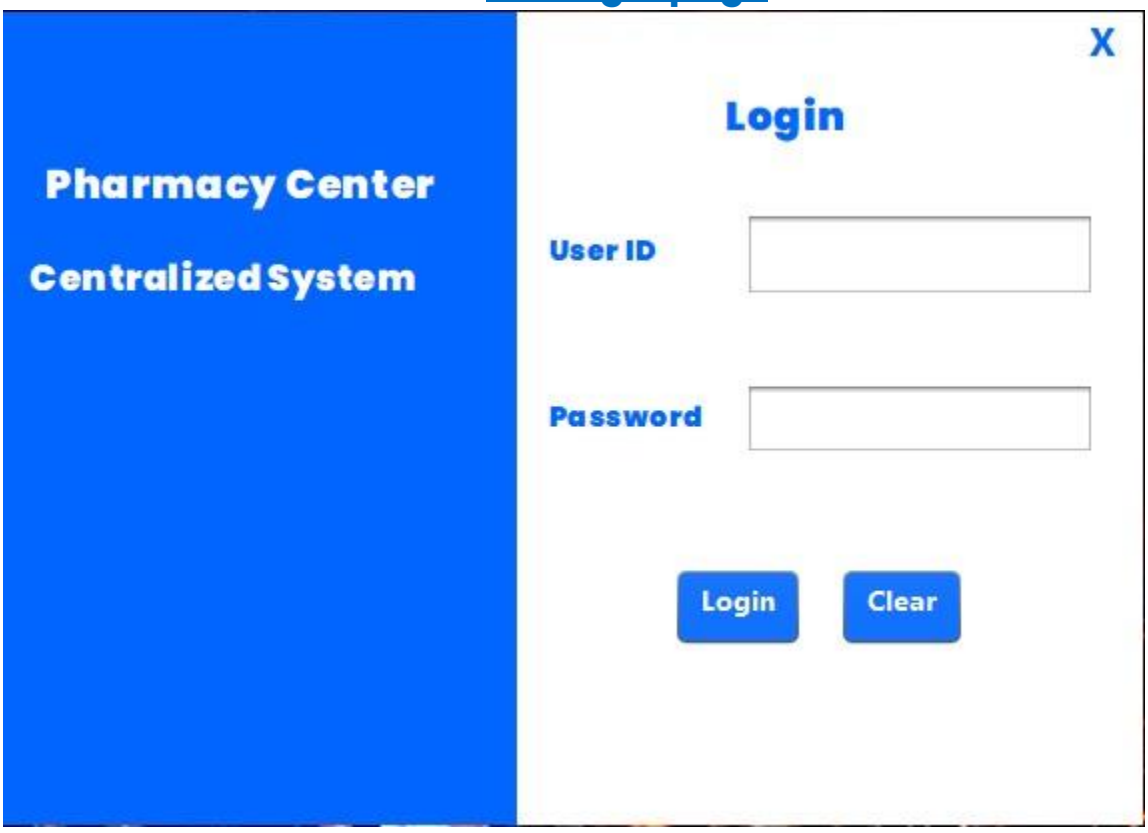


CODES

```
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

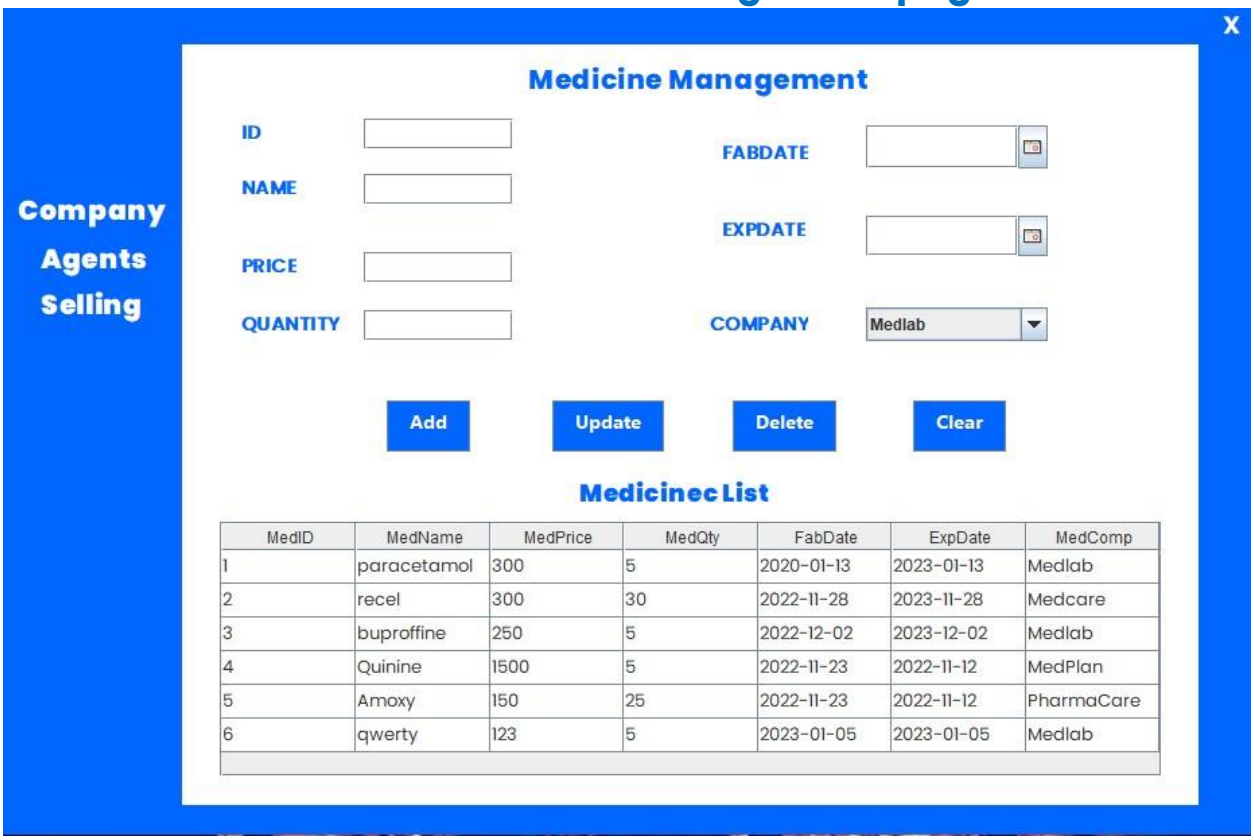



```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import javax.swing.JOptionPane;
import java.sql.SQLException;
import java.sql.ResultSet;

/**
 *
 * @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



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);
```

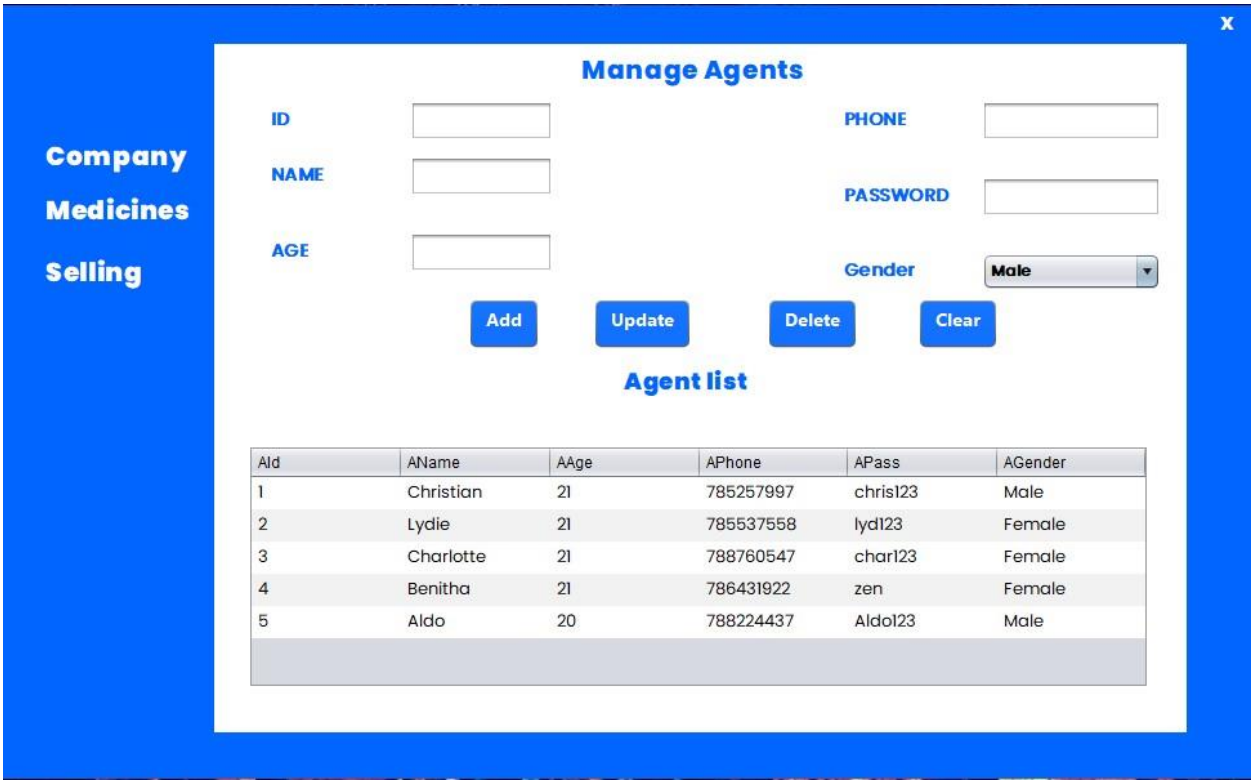
```
      }
   }
   catch(SQLException e)
   {
      e.printStackTrace();
   }
   }

   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.



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.

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
 */
```

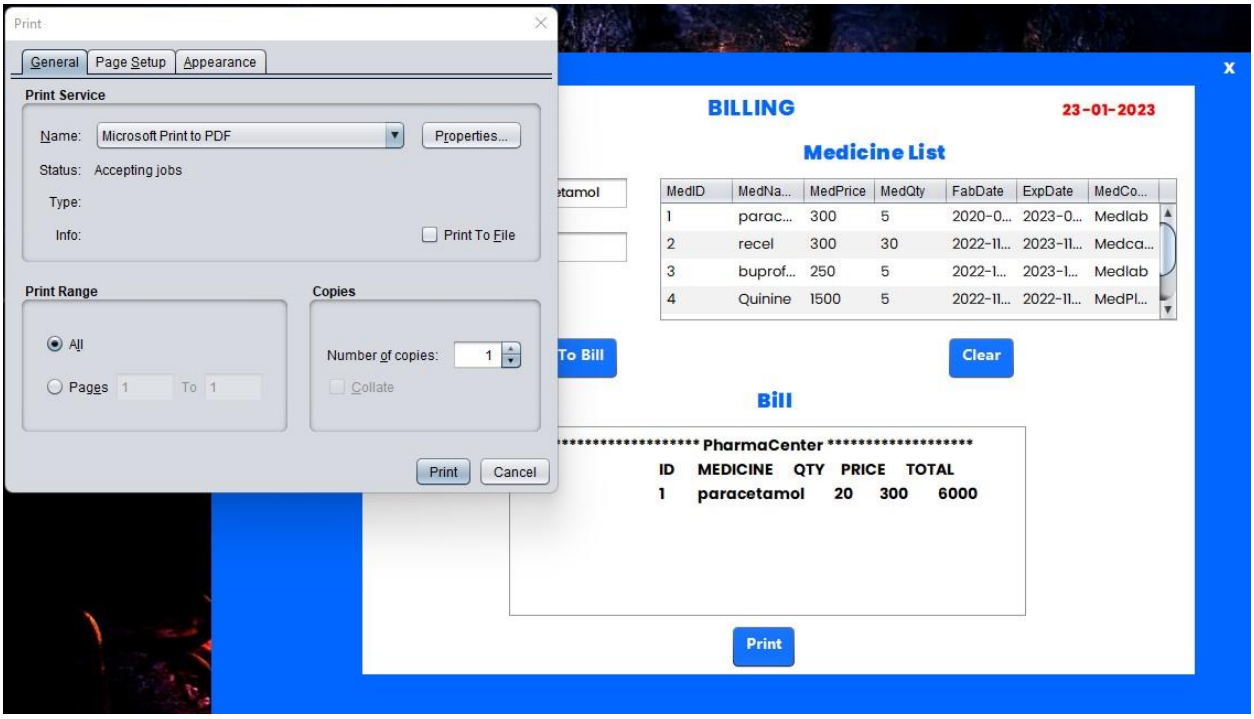
```
public class Company extends javax.swing.JFrame {

   /**
    * 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.



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

The concept of testing is based on insuring that every aspect of the application/project/system works as intended by the builder and as instructed by the client. As such, we insured that our application met both criteria's to ensure the best performance possible from the application. Each file of the project was run individually at the end of every build to insure everything was working as previously intended before joining everything up for a collective and overall testing of the entire project. It is therefore safe to assume that the system passed every one of our recommendations and can successfully run locally on any device/ computer presented.

However, we did encounter some difficulties while building the system some of the being the lack of some palettes that needed to be downloaded and imported from the internet and a malfunctioning of the XAMPP control panel that led to the re-installation of the whole system in order to resolve the problem.

We are confident that all issues were resolved successfully and if any should come up we anticipate that regular maintenance of the system we keep problems at bay.

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.

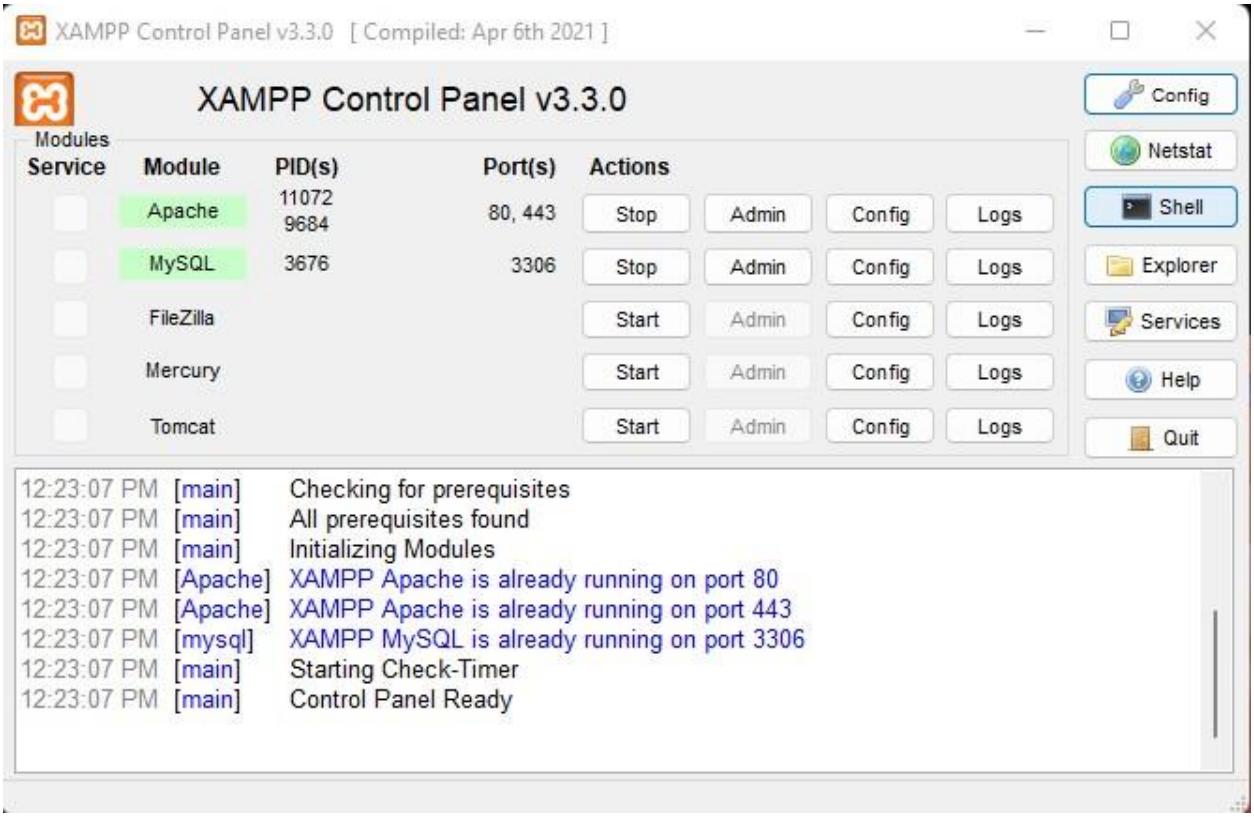
Stage 6: MAINTAINACE

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.

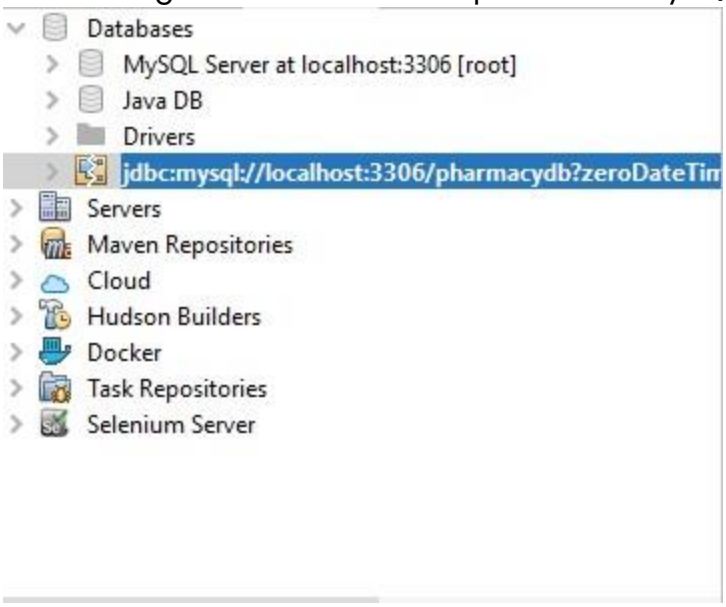
We begin with the installation of all the necessary applications i.e. **MySQL database server, XAMPP control panel,JDK-19 & Apache NetBeans IDE 15** necessary to build the application. Next, we had to install other additional files to ensure the application run smoothly. This are: **Mysql-connector-j-8.0.31.jar,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

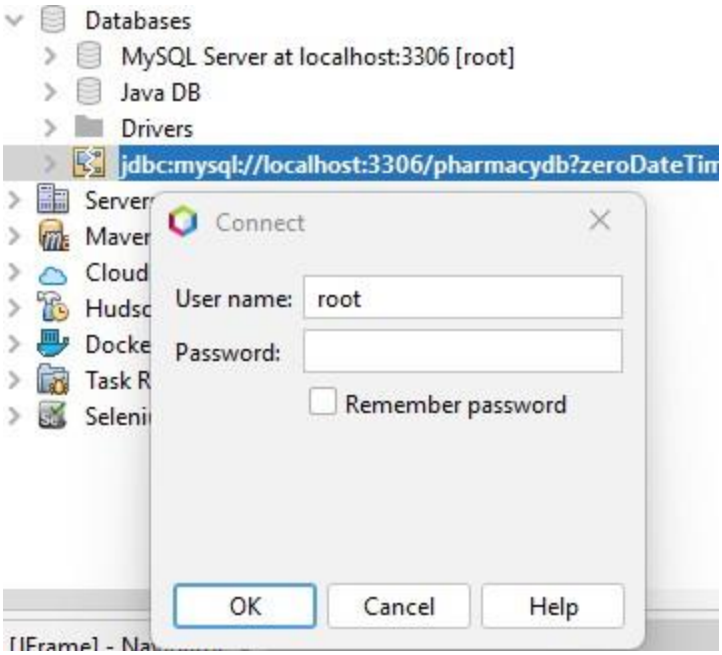
Next we have the testing phase which is meant to ensure that the applications works and performs tasks as intended. During this phase we had to ensure that Mysql server was up and running (we had to start the server and Apache) on the XAMPP Control Panel and make sure that the project was connected to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



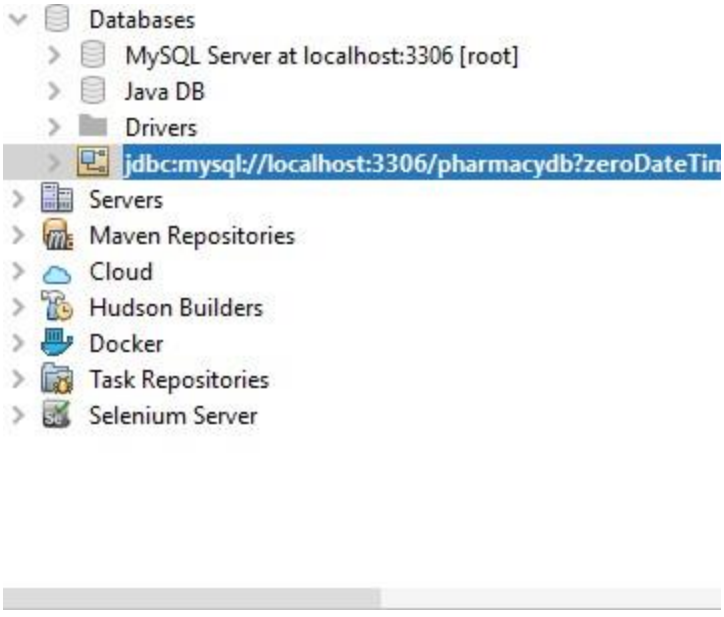
In this image we see that both Apache and MySQL are started and running.



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



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.