



**Faculty of Computers &
Artificial Intelligence**



Benha University

Cars Inventory Management System

In

Information system

by

Student's Full Name (Student Section Number)

Hazem Refai Mohammed Refai (6)	Ibrahim Musa El-Syed Sweilem (1)
Ahmed Samer Eid Abd Elhamed (2)	Hazem Tarek Mahmoud Elsabbagh (6)
Eslam khaled Soliman Gad (4)	

Supervised by

Dr.Mona Arafa

Table of Contents

1. Introduction	1
2. Development tools.....	1
3. Home Section	1
4. Companies Section	2
5. Items Section	4
6. Cashing Section.....	8
7. Report Section	12
8. ERD	17
9. Mapping.....	18

1. Introduction

This project is used to manage car stores and manage the operations of auto stores, such as car company data. Supplier data such as name, address, email and phone number are saved and each supplier has its own id and also manages the types of cars in the stores and allows searching for them by the company name and add New car data, it also manages sales operations and customer data, and also allows inventory operations for finished sales and available cars.

2. Development tools

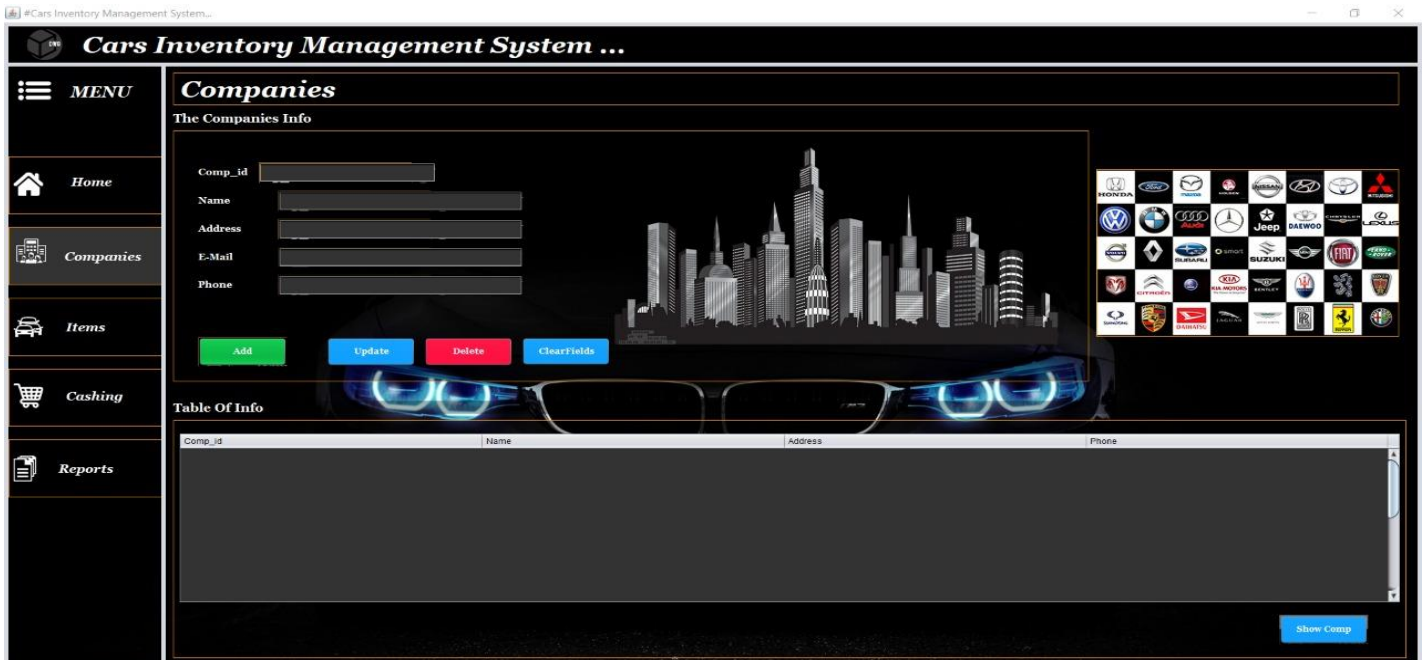
- 1-Java With netbeans IDE
- 2- SQL With my sql workbench
- 3- Connector between java and my sql

3. Home Section



It is through which he gives information about the program, gives information about the specific functions of each section, and gives notes specific to the program

4. Companies Section



It is through which all operations that are performed with companies' data Such as adding, updating , deleting and showing data

Add Button Code :

```
private void CAddBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(CIDText.getText().trim().isEmpty() || CNameText.getText().trim().isEmpty() ||
           CAddressText.getText().trim().isEmpty() || CEmailText.getText().trim().isEmpty() || CPhoneText.getText().trim().isEmpty())
        {
            JOptionPane.showMessageDialog(this, "Data Entry Error : "+"\\n"+"Maybe There Is An Empty TextField .");
        }
    }
    else {
        int Yas_No = JOptionPane.showConfirmDialog(this, "Are You Sure You Want To Add This Company", "Confirm", JOptionPane.YES_NO_OPTION);
        if(Yas_No == JOptionPane.YES_OPTION)
        {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
            PreparedStatement stmt = con.prepareStatement("insert into companies (Comp_ID,Comp_Name,Comp_Address,Comp_Email,Comp_Phone)Values(?,?,?,?,?)");
            String CID = CIDText.getText();
            String CName = CNameText.getText();
            String CAddress = CAddressText.getText();
            String CEmail = CEmailText.getText();
            String CPhone = CPhoneText.getText();
            stmt.setString(1,CID);
            stmt.setString(2,CName);
            stmt.setString(3,CAddress);
            stmt.setString(4,CEmail);
            stmt.setString(5,CPhone);
            stmt.executeUpdate();
            //-----
            DefaultComboBoxModel dm = (DefaultComboBoxModel) SBCNCBox.getModel();
            dm.addElement(CName);
            SBCNCBox.setModel(dm);
        }
    }
}
```

```
DefaultComboBoxModel dm2 = (DefaultComboBoxModel) R3ComboBox.getModel();
dm2.addElement(CName);
R3ComboBox.setModel(dm2);
//-----
JOptionPane.showMessageDialog(this, "Insert Success");
CIDText.setText("");
CNameText.setText("");
CAddressText.setText("");
CEmailText.setText("");
CPhoneText.setText("");
CShowBActionPerformed(evt);

}
} catch (SQLException ex) {
JOptionPane.showMessageDialog(this, "Sorry, one of these data [Comp_ID or Comp_Email or Comp_Phone] already exists"+
"\n"+",please change it and try again");
}
}
```

Update Button Code :

```
private void CUpdateBActionPerformed(java.awt.event.ActionEvent evt) {
try {
if (CTable.getSelectionModel().isSelectionEmpty())
{
JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Delete");
}
else
{
int Yas_No = JOptionPane.showConfirmDialog
(this, "Are You Sure That You Want To Update Info Of This Company ?", "Confirm", JOptionPane.YES_NO_CANCEL_OPTION);
if (Yas_No == JOptionPane.YES_OPTION)
{
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
PreparedStatement stmt = con.prepareStatement("update companies set Comp_Name = ? , Comp_Address= ?,Comp_Email=?,Comp_Phone=? where Comp_ID = ? ");
String CName = CNameText.getText();
String CAddress = CAddressText.getText();
String CEmail = CEmailText.getText();
String CPhone = CPhoneText.getText();
String Cid = CIDText.getText();
stmt.setString(1, CName);
stmt.setString(2, CAddress);
stmt.setString(3, CEmail);
stmt.setString(4, CPhone);
stmt.setString(5, Cid);
stmt.executeUpdate();
JOptionPane.showMessageDialog(this, "Update Success");
CNameText.setText("");
CAddressText.setText("");
CEmailText.setText("");
CPhoneText.setText("");
CIDText.setText("");
}
}
CShowBActionPerformed(evt);
}
} catch (SQLException ex) {
JOptionPane.showMessageDialog(this, "Company Info Entry Errorr , Maybe Each Of These Info is Existes ");
Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

Delete Button Code :

```
private void CDeleteBActionPerformed(java.awt.event.ActionEvent evt) {
try {
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
if (CTable.getSelectionModel().isSelectionEmpty())
{
JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Delete");
}
else
{
int Yas_No = JOptionPane.showConfirmDialog
(this, "Are You Sure That You Want To Delete This Company ?", "Confirm", JOptionPane.YES_NO_CANCEL_OPTION);
if (Yas_No == JOptionPane.YES_OPTION)
{
String CID = CIDText.getText();
PreparedStatement stmt3 = con.prepareStatement("select inventory.Item_ID from inventory where inventory.Comp_ID = ? ");
stmt3.setString(1, CID);
ResultSet set = stmt3.executeQuery();
if (set.next())
{
JOptionPane.showMessageDialog(this, "Sorry!! This Company Cannot Be Deleted Because His Info Recorded In Item");
}
else {
PreparedStatement stmt = con.prepareStatement("delete from companies where Comp_ID = ?");
}
}
}
}
```

```
CShowBActionPerformed(evt);  
}  
}  
} catch (SQLException ex) {  
    System.out.println("Error");  
}  
}
```

Show Comp Button Code :

```
private void CShowBActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");  
        PreparedStatement stmt = con.prepareStatement("select * from companies");  
        ResultSet set = stmt.executeQuery();  
        DefaultTableModel dm = new DefaultTableModel();  
        dm.addColumn("Comp_ID");  
        dm.addColumn("Comp_Name");  
        dm.addColumn("Comp_Address");  
        dm.addColumn("Comp_Email");  
        dm.addColumn("Comp_Phone");  
        while (set.next())  
        {  
            String r[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5)};  
            dm.addRow(r);  
        }  
        CTable.setModel(dm);  
    } catch (SQLException ex) {  
        System.out.println("Error");  
    }  
}
```

5. Items Section

The screenshot displays the 'Cars Inventory Management System' application. The interface features a dark-themed sidebar menu on the left with options: MENU, Home, Companies, Items (selected), Cashing, and Reports. The main content area is titled 'Items' and includes a search bar with a 'Search:' label and a 'By:' dropdown menu set to 'All'. Below the search bar is a table with columns: Item_id, Brand, Model, Ability, Category, Color, and Amount. The table is currently empty. A large image of a car's headlights is displayed below the table. To the right of the headlights is a 'Show Item' button. Below the headlights is a form titled 'Item Details' with input fields for Item_id, Brand, Model, Color, Position, and Amount. To the right of this form is a 'Clear Fields' button. Further right is a form titled 'Comp Details' with input fields for Comp_id and Comp_Name, and a large text area for 'Item Description'. To the right of this form is a 'Show Item' button. At the bottom right of the interface are three buttons: 'Insert' (green), 'Update' (blue), and 'Remove' (red).

It is through which all the operations that are performed with the various cars in the inventory , Such as adding, updating , deleting and showing data

Show Item Button Code :

```
private void IShowBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = null;
        String selected_Text = SCBox.getSelectedItem().toString();
        if(selected_Text == "All")
        {
            stmt = con.prepareStatement
            ("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,inventory
            + ".Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID order by item.Item_ID");
        }
        else if(selected_Text == "Brand")
        {
            String Search_Text = IStext.getText();
            Search_Text=Search_Text+"%";
            stmt = con.prepareStatement(("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,"
            + "inventory.Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID and Brand_Car like ? order by item.Item_ID");
            stmt.setString(1,Search_Text);
        }
        else if(selected_Text == "Model")
        {
            String Search_Text = IStext.getText();
            Search_Text=Search_Text+"%";
            stmt = con.prepareStatement(("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,"
            + "inventory.Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID and Model_Car like ? order by item.Item_ID");
            stmt.setString(1,Search_Text);
        }
        else if(selected_Text == "Color")
        {
            String Search_Text = IStext.getText();
            Search_Text=Search_Text+"%";
            stmt = con.prepareStatement(("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,"
            + "inventory.Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID and Color_Car like ? order by item.Item_ID");
            stmt.setString(1,Search_Text);
        }
        else if(selected_Text == "Inventory Position")
        {
            String Search_Text = IStext.getText();
            Search_Text=Search_Text+"%";
            stmt = con.prepareStatement(("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,"
            + "inventory.Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID and inventory.Inventory_Position like ? order by item.Item_ID");
            stmt.setString(1,Search_Text);
        }
        else if(selected_Text == "Item Amount")
        {
            String Search_Text = IStext.getText();
            Search_Text=Search_Text+"%";
            stmt = con.prepareStatement(("select Item.Item_ID,Item.Brand_Car,Item.Model_Car,Item.Color_Car,Item.Item_Discription,inventory.Inventory_Position,"
            + "inventory.Item_Amount from item ,Inventory where Item.Item_ID = inventory.Item_ID and inventory.Item_Amount like ? order by item.Item_ID");
            stmt.setString(1,Search_Text);
        }
        ResultSet set = stmt.executeQuery();
        DefaultTableModel dm = new DefaultTableModel();
        while (set.next())
        {
            String z[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5),set.getString(6),set.getString(7));
            dm.addRow(z);
        }
        I_Table.setModel(dm);
    } catch (SQLException ex) {
        System.out.println("Error");
    }
}
```

Insert Button Code :

```
private void IInsertBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(IBrandText.getText().trim().isEmpty() || IModelText.getText().trim().isEmpty() || IColorText.getText().trim().isEmpty()
        || IPositionText.getText().trim().isEmpty() || IAmountText.getText().trim().isEmpty() || ICIDText.getText().trim().isEmpty()
        || IDescribText.getText().trim().isEmpty())
        {
            JOptionPane.showMessageDialog(this, "Data Entry Error : "+"\\n"+"Maybe There Is An Empty TextField .");
        }
        else
        {
            int Yas_No = JOptionPane.showConfirmDialog(this,"Are You Sure You Want To Insert This Item","Confirm",JOptionPane.YES_NO_OPTION);
            if(Yas_No == JOptionPane.YES_OPTION)
            {
                Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
                PreparedStatement stmt = con.prepareStatement("select companies.Comp_ID , companies.Comp_Name from companies where Comp_ID = ? ");
                String ICID = ICIDText.getText();
                stmt.setString(1,ICID);
                ResultSet set = stmt.executeQuery();
                boolean ID_Found=set.next();
                if(ID_Found)
                {
                    String IBrand = IBrandText.getText();
                    String IModel = IModelText.getText();
                    String IColor = IColorText.getText();
                    String IPosition = IPositionText.getText();
                    String IAmount = IAmountText.getText();
                    String IDescrib = IDescribText.getText();
                    stmt = con.prepareStatement("insert into item (Brand_Car,Model_Car,Color_Car,Item_Discription,Inventory_Position,Item_Amount)
                    values (?,?,?,?,?,?)");
                    stmt.setString(1,IBrand);
                    stmt.setString(2,IModel);
                    stmt.setString(3,IColor);
                    stmt.setString(4,IPosition);
                    stmt.setString(5,IAmount);
                    stmt.setString(6,IDescrib);
                    stmt.executeUpdate();
                }
            }
        }
    } catch (SQLException ex) {
        System.out.println("Error");
    }
}
```

```
ResultSet set2 = stmt2.executeQuery();
boolean Cars_Found = set2.next();
if(Cars_Found)
{
JOptionPane.showMessageDialog(this, "Sorry, The Main Info For This Car Is Already Exist In Another Car With Id = "
+set2.getString(2)+"\n If It Is The Same Car, You Can Update Its Amount . If it wasn't, Please Change This Info");
}
else
{
if (IAmount.matches("[0-9]+") && IAmount!="" && Integer.parseInt(IAmount)>0)
{
PreparedStatement stmt3 = con.prepareStatement
("insert into item (item.Brand_Car,item.Color_Car,item.Model_Car,item.Item_Discription)values (?,?,?,?)", Statement.RETURN_GENERATED_KEYS);
PreparedStatement stmt4 = con.prepareStatement
("insert into inventory (inventory.Comp_ID ,inventory.Item_ID, inventory.Inventory_Position , inventory.Item_Amount)Values (?,?,?,?)");
PreparedStatement stmt5 = con.prepareStatement
("insert into incoming (incoming.Comp_ID,incoming.Item_ID,incoming.Added_Amount,incoming.How_It_Add,incoming.Added_date)values (?,?,?,?,?)");
stmt3.setString(1, IBrand);
stmt3.setString(2, IColor);
stmt3.setString(3, IModel);
stmt3.setString(4, IDescrib);
stmt3.executeUpdate();
ResultSet rs = stmt3.getGeneratedKeys();
...
```

```
stmt4.executeUpdate();
//-----
Date date = new Date();
java.sql.Date sdate = new java.sql.Date(date.getTime());
stmt5.setString(1, ICID);
stmt5.setString(2, new_IID);
stmt5.setString(3, IAmount);
stmt5.setString(4, "By Insert It");
stmt5.setDate(5, sdate);
stmt5.executeUpdate();
//-----
JOptionPane.showMessageDialog(this, "Insert Success");
IBrandText.setText("");
IModelText.setText("");
IColorText.setText("");
IDescribText.setText("");
IPostionText.setText("");
IAmountText.setText("");
ICIDText.setText("");
ICNameText.setText("");
ItemIDText.setText("");
IShowBActionPerformed(evt);
}
else
{
```

```
{
JOptionPane.showMessageDialog(this, "Sorry, The Company Id You Entered Does Not Exist."+"\n"+" Please, Change It Or Add It In The Companies Sectionn")
}
}
catch (SQLException ex) {
Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

Update Button Code :

```
private void IUpdateBActionPerformed(java.awt.event.ActionEvent evt) {
try {
if (I_Table.getSelectionModel().isSelectionEmpty())
{
JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Update");
}
else
{
int Yas_No = JOptionPane.showConfirmDialog(this, "Are You Sure That You Want To Update Info Of This Item ?", "Confirm"
, JOptionPane.YES_NO_CANCEL_OPTION);
if (Yas_No == JOptionPane.YES_OPTION)
{
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
PreparedStatement stmt = con.prepareStatement("select companies.Comp_ID , companies.Comp_Name from companies where Comp_ID = ? ");
String ICID = ICIDText.getText();
stmt.setString(1, ICID);
ResultSet set = stmt.executeQuery();
boolean ID_Found=set.next();
if (ID_Found)
{
String Item_ID=ItemIDText.getText();
String IBrand = IBrandText.getText();
String IModel = IModelText.getText();
String IColor = IColorText.getText();
...
```



```
stmt2.setString(3, Item_ID);
ResultSet set2 = stmt2.executeQuery();
boolean Cars_Found = set2.next();
if (Cars_Found)
{
    JOptionPane.showMessageDialog(this, "Sorry, The Main Info For This Car Is Already Exist In Another Car With Id = "
        + set2.getString(2) + "\n If It Is The Same Car, You Can Update Its Amount . If it wasn't, Please Change This Info");
}
else
{
    if (IAmount.matches("[0-9]+") && IAmount != "" && Integer.parseInt(IAmount) > 0)
    {
        int row_se = I_Table.getSelectedRow();
        String BUAmount = I_Table.getValueAt(row_se, 6).toString();
        if (Integer.parseInt(IAmount) > Integer.parseInt(BUAmount))
        {
            int N_Amount = Integer.parseInt(IAmount) - Integer.parseInt(BUAmount);
            PreparedStatement stmt4 = con.prepareStatement("insert into incoming (incoming.Comp_ID, incoming.Item_ID, incoming.Added_Amount, incoming.How_It_Add, "
                + "incoming.Added_date) values (?, ?, ?, ?, ?)");
            Date date = new Date();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            stmt4.setString(1, ICID);
            stmt4.setString(2, Item_ID);
            stmt4.setString(3, String.valueOf(N_Amount));
        }
    }
}
```

```
stmt3.setString(3, IModel);
stmt3.setString(4, IDescrib);
stmt3.setString(5, ICID);
stmt3.setString(6, IPosition);
stmt3.setString(7, IAmount);
stmt3.setString(8, Item_ID);
stmt3.executeUpdate();
JOptionPane.showMessageDialog(this, "Update Success");
IBrandText.setText("");
IModelText.setText("");
IColorText.setText("");
IDescribText.setText("");
IPositionText.setText("");
IAmountText.setText("");
ICIDText.setText("");
ICNameText.setText("");
ItemIDText.setText("");
IShowBActionPerformed(evt);
}
else
{
    JOptionPane.showMessageDialog(this, "Plese, In Amount Enter Positive Num > 0");
}
```

Delete Button Code :

```
private void IRemoveBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        if (I_Table.getSelectionModel().isSelectionEmpty())
        {
            JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Delete");
        }
        else
        {
            int Yes_No = JOptionPane.showConfirmDialog(this, "Are You Sure That You Want To Remove This Items ?", "Confirm",
                JOptionPane.YES_NO_CANCEL_OPTION);
            if (Yes_No == JOptionPane.YES_OPTION)
            {
                String IID = ItemIDText.getText();
                PreparedStatement stmt3 = con.prepareStatement("select cashing.item_ID from cashing where cashing.item_ID= ?");
                stmt3.setString(1, IID);
                ResultSet set = stmt3.executeQuery();
                PreparedStatement stmt4 = con.prepareStatement("select incoming.Item_ID from incoming where incoming.Item_ID= ?");
                stmt4.setString(1, IID);
                ResultSet set2 = stmt4.executeQuery();
                if (set.next() || set2.next())
                {
                    JOptionPane.showMessageDialog(this, "Sorry!! This Item Cannot Be Deleted Because His Info Recorded In Cashing Or Incoming List");
                }
            }
        }
        IBrandText.setText("");
        IModelText.setText("");
        IColorText.setText("");
        IDescribText.setText("");
        IPositionText.setText("");
        IAmountText.setText("");
        ICIDText.setText("");
        ICNameText.setText("");
        ItemIDText.setText("");
        IShowBActionPerformed(evt);
    }
    catch (SQLException ex) {
        Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

6. Cashing Section

It is through it that all the operations that perform the clints that the cars dispose to them, Such as adding, updating , deleting and showing data . And also the cashing and completion operations

Make Item Cashing Button Code :

```
private void CMICBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        int Yas_No = JOptionPane.NO_OPTION;
        if (ClientIDText.getText().trim().isEmpty() || ClientEText.getText().trim().isEmpty() || ClientFText.getText().trim().isEmpty()) {
            JOptionPane.showMessageDialog(this, "Data Entry Error : "+"\\n"+"Maybe There Is An Empty TextField .");
        }
        else {
            Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
            String Next_ID;
            if (ClientIDText.getText().trim().isEmpty()) {
                Yas_No = JOptionPane.showConfirmDialog(this, "Is He New Client ?", "Confirm", JOptionPane.YES_NO_OPTION);
                if (Yas_No == JOptionPane.YES_OPTION) {
                    PreparedStatement stmt1 = con.prepareStatement("set @@SESSION.information_schema_stats_expiry = 0;");
                    stmt1.executeQuery();
                    PreparedStatement stmt = con.prepareStatement("select Auto_increment from information_schema.tables where table_schema = 'project' and table_name = 'client' ;");
                    ResultSet set = stmt.executeQuery();
                    set.next();
                    Next_ID = set.getString(1);
                    ClientIDText.setText(Next_ID);
                    CClient_IDText.setText(Next_ID);
                }
                else {
                    JOptionPane.showMessageDialog(this, "Sorry!! The Client info didnot Exist ");
                }
            }
            else {
                String ClientID = ClientIDText.getText().toString();
                CClient_IDText.setText(ClientID);
                if (Yas_No == JOptionPane.YES_OPTION || !ClientIDText.getText().trim().isEmpty()) {
                    SBCNBox.setEnabled(true);
                    WIEFCSAll.setEnabled(true);
                    CClearB.setEnabled(true);
                    CClearAB.setEnabled(true);
                    CDoneB.setEnabled(true);
                    Date date = new Date();
                    DateText.setText(date.toString());
                    PreparedStatement stmt = con.prepareStatement("select * from Cashing ");
                    ResultSet set = stmt.executeQuery();
                    int LO_NO=1;
                }
            }
        }
    }
}
```

```

        while (set.next())
        {
            LO_NO = Integer.parseInt(set.getString(1));
            LO_NO++;
        }
        OrderNOText.setText(String.valueOf(LO_NO));
    }

} catch (SQLException ex) {
    Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}

```

Update Button Code :

```

private void CUpdateBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        if(CInfoT.getSelectionModel().isSelectedEmpty())
        {
            JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Update");
        }
        else
        {
            int Yas_No = JOptionPane.showConfirmDialog(
                this, "Are You Sure That You Want To Update Info Of This Clint ?", "Confirm", JOptionPane.YES_NO_CANCEL_OPTION);
            if (Yas_No == JOptionPane.YES_OPTION)
            {
                Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
                PreparedStatement stmt = con.prepareStatement(
                    "Update Client set Client_Name = ? , Client_Email = ? , Client_Phone= ? where Client_ID = ? ");
                String ClintName = ClintNText.getText();
                String ClintEmail = ClintEText.getText();
                String ClintPhone = ClintPText.getText();
                String ClintId = ClintIDText.getText();
                stmt.setString(1, ClintName);
                stmt.setString(2, ClintEmail);
                stmt.setString(3, ClintPhone);
                stmt.setString(4, ClintId);
                stmt.executeUpdate();
                JOptionPane.showMessageDialog(this, "Update Success");
                ClintNText.setText("");
                ClintEText.setText("");
                ClintPText.setText("");
                ClintIDText.setText("");
                CShowBActionPerformed(evt);
            }
        }
    }
    catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Sorry, one of these data [Email or Phone] already exists+"\n+", please change it and try again");
    }
}

```

Remove Button Code :

```

private void CRemoveBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        if(CInfoT.getSelectionModel().isSelectedEmpty())
        {
            JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Delete");
        }
        else
        {
            int Yas_No = JOptionPane.showConfirmDialog(this, "Are You Sure That You Want To Remove This Clint ?", "Confirm",
                JOptionPane.YES_NO_CANCEL_OPTION);
            if (Yas_No == JOptionPane.YES_OPTION)
            {
                String Clint_ID = ClintIDText.getText();
                PreparedStatement stmt = con.prepareStatement("select cashing.Client_ID from cashing where cashing.Client_ID= ?;");
                stmt.setString(1, Clint_ID);
                ResultSet set = stmt.executeQuery();
                if (set.next())
                {
                    JOptionPane.showMessageDialog(this, "Sorry!! This Client Cannot Be Deleted Because His Info Recorded In Cashing List");
                }
                else {
                    PreparedStatement stmt2 = con.prepareStatement("delete from client where Client_ID = ?");
                    stmt2.setString(1, Clint_ID);
                    stmt2.executeUpdate();
                    JOptionPane.showMessageDialog(this, "The Clint Delete Success");
                    ClintNText.setText("");
                    ClintEText.setText("");
                    ClintPText.setText("");
                    ClintIDText.setText("");
                    ClintShowBActionPerformed(evt);
                }
            }
        }
    }
    catch (SQLException ex) {
        Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

Show Client Data Button Code :

```
private void ClintShowBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = con.prepareStatement("select * from Client");
        ResultSet set = stmt.executeQuery();
        DefaultTableModel dm = new DefaultTableModel();
        dm.addColumn("ClintID");
        dm.addColumn("ClintName");
        dm.addColumn("ClintE-mail");
        dm.addColumn("ClintPhone");

        while (set.next()) {
            String r[] = {set.getString(1), set.getString(2), set.getString(3), set.getString(4)};
            dm.addRow(r);
        }
        CInfoT.setModel(dm);
    } catch (SQLException ex) {
        System.out.println("Errorr");
    }
}
```

Add Button Code:

```
private void ClintAddBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        if(WIEFCTable.getSelectionModel().isSelectionEmpty()) {
            JOptionPane.showMessageDialog(this, "There Are No Selected Row . Please, Select The Row You Want To Update");
        }
        else {
            if(AmontText.getText().matches("[0-9]+") && AmontText.getText().length() != 0 && Integer.parseInt(AmontText.getText()) > 0) {
                int IAmount = Integer.parseInt(AmontText.getText());
                System.out.println(IAmount);
                int row_se = WIEFCTable.getSelectedRow();
                int Avi_amount = Integer.parseInt(WIEFCTable.getValueAt(row_se, 4).toString());
                System.out.println(Avi_amount);
                if (IAmount <= Avi_amount) {
                    String Item_ID = WIEFCTable.getValueAt(row_se, 0).toString();
                    String Brand_Car = WIEFCTable.getValueAt(row_se, 1).toString();
                    String Model_Car = WIEFCTable.getValueAt(row_se, 2).toString();
                    String Color_Car = WIEFCTable.getValueAt(row_se, 3).toString();
                    if (ItemAddBCheck == 0) {
                        DefaultTableModel dm = new DefaultTableModel();
                        dm.addColumn("Item_ID");
                        dm.addColumn("Brand_Car");
                        dm.addColumn("Model_Car");
                        dm.addColumn("Color_Car");
                        dm.addColumn("Cashing_Amount");
                        ItemAddBCheck = 1;
                        String r[] = {Item_ID, Brand_Car, Model_Car, Color_Car, AmontText.getText()};
                        dm.addRow(r);
                        WWBCFCTable.setModel(dm);
                    }
                    else {
                        boolean Ch_Duplicate = true;
                        DefaultTableModel dm = (DefaultTableModel) WWBCFCTable.getModel();
                        int RowCount = WWBCFCTable.getRowCount();
                        for (int i = RowCount - 1; i >= 0; i--) {
                            if (WWBCFCTable.getValueAt(i, 0).toString() == Item_ID) {
                                Ch_Duplicate = false;
                                break;
                            }
                        }
                        if (Ch_Duplicate) {
                            String r[] = {Item_ID, Brand_Car, Model_Car, Color_Car, AmontText.getText()};
                            dm.addRow(r);
                            WWBCFCTable.setModel(dm);
                        }
                        else {
                            JOptionPane.showMessageDialog(this, "Sorry! You Add This Item Befor. ");
                        }
                    }
                }
            }
            else {
                JOptionPane.showMessageDialog(this, "not Available");
            }
        }
        AmontText.setText("");
    }
    else {
        JOptionPane.showMessageDialog(this, "Plese, In Amount Enter Positive Num > 0");
    }
} catch (SQLException ex) {
    Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

Show Button Code :

```
private void WIEFCSAllActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = null;
        String selected_Text = SBCNCBox.getSelectedItem().toString();
        if(selected_Text=="All")
        {
            stmt = con.prepareStatement("select item.item_ID, item.Brand_Car , item.Model_Car ,item.Color_Car ,inventory.Item_Amount from item ,"
            + " inventory where item.Item_ID = inventory.Item_ID and inventory.Item_Amount > 0 order by item.Item_ID");
        }
        else
        {
            stmt = con.prepareStatement("select item.item_ID, item.Brand_Car , item.Model_Car ,item.Color_Car ,inventory.Item_Amount from item ,"
            + " inventory , companies where companies.Comp_ID = inventory.Comp_ID and item.Item_ID = inventory.Item_ID and companies."
            + "Comp_Name = ? and inventory.Item_Amount > 0 order by item.Item_ID;");
            stmt.setString(1,selected_Text);
        }
        ResultSet set = stmt.executeQuery();
        DefaultTableModel dm = new DefaultTableModel();
        dm.addColumn("Item_ID");
        dm.addColumn("Brand_Car");
        dm.addColumn("Model_Car");
        dm.addColumn("Color_Car");
        dm.addColumn("Avi_Amount");
        while (set.next())
        {
            String r[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5)};
            dm.addRow(r);
        }
        WIEFCTable.setModel(dm);
    } catch (SQLException ex) {
        Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Done Button Code :

```
private void CDoneBActionPerformed(java.awt.event.ActionEvent evt) {
    String ItemID = String.valueOf(WWBCFCTable.getValueAt(0, 0));
    System.out.println(ItemID);
    if(!ItemID.isEmpty() && ItemID != "null")
    {
        int Yas_No = JOptionPane.showConfirmDialog(this,"Are You Sure That You Want To Cashing This Items ?", "Confirm",
        JOptionPane.YES_NO_CANCEL_OPTION);
        if(Yas_No == JOptionPane.YES_OPTION)
        {
            try {
                Connection con =DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
                String Clint_ID = CClint_IDText.getText();
                PreparedStatement stmt = con.prepareStatement("select Client_ID from client where client_ID = ? ");
                stmt.setString(1, Clint_ID);
                ResultSet set = stmt.executeQuery();
                boolean if_Exist = set.next();
                if(!if_Exist)
                {
                    String CName = ClintNText.getText();
                    String CEmail = ClintEText.getText();
                    String CPhone = ClintPText.getText();
                    PreparedStatement stmt2 = con.prepareStatement("insert into client (client.client_Name , client.client_EMail , "
                    + " client.client_Phone)values(?, ?, ?)");
                    stmt2.setString(1, CName);
                    stmt2.setString(2, CEmail);
                    stmt2.setString(3, CPhone);
                    stmt2.executeUpdate();
                    JOptionPane.showMessageDialog(this, "A New Client Has Been Added Successfully");
                    ClintShowBActionPerformed(evt);
                }
                PreparedStatement stmt3 = con.prepareStatement("update inventory . item set inventory.Item Amount = ? where item.Item ID = ? "
                + "and item.Item_ID = inventory.Item_ID;");
                PreparedStatement stmt4 = con.prepareStatement("insert into cashing (cashing.Order_No , cashing.Client_ID , cashing.Item_ID , "
                + "cashing.Cashing_Amount, cashing.Cashing_Date)values (?,?,?,?) ");
                PreparedStatement stmt5 = con.prepareStatement("select inventory.Item_Amount from inventory where inventory.Item_ID = ? ;");
                int RowCount = WWBCFCTable.getRowCount();
                String Item_ID ;
                String Brand_Car ;
                String Color_Car ;
                String Model_Car ;
                String Clint_Id = CClint_IDText.getText();
                Date date = new Date();
                java.sql.Date sdate = new java.sql.Date(date.getTime());
                System.out.println(sdate);
                String O_NO = OrderNOText.getText();
                int Avi_Amount;
                int IAmount;

                for(int i=RowCount-1;i>=0;i--)
                {
                    Item_ID = WWBCFCTable.getValueAt(i, 0).toString();
                    Brand_Car = WWBCFCTable.getValueAt(i, 1).toString();
                    Color_Car = WWBCFCTable.getValueAt(i, 2).toString();
                    Model_Car = WWBCFCTable.getValueAt(i, 3).toString();
                    Avi_Amount = Integer.parseInt(WWBCFCTable.getValueAt(i, 4).toString());
                    System.out.println(Avi_Amount);
                    stmt5.setString(1, Item_ID);
                    ResultSet set5 = stmt5.executeQuery();
                    set5.next();
                    IAmount = Integer.parseInt(set5.getString(1));
                    System.out.println(IAmount);
                    //-----
                    stmt3.setString(1, String.valueOf((IAmount-Avi_Amount)));
                }
            }
        }
    }
}
```

```

stmt3.setString(2, Item_ID);
stmt3.executeUpdate();
//-----
stmt4.setString(1, O_NO);
stmt4.setString(2, Clint_Id);
stmt4.setString(3, Item_ID);
stmt4.setString(4, String.valueOf(Avi_Amount));
stmt4.setDate(5, sdate);
stmt4.executeUpdate();
}
JOptionPane.showMessageDialog(this, "Done");
ClintNText.setText("");
ClintEText.setText("");
ClintPText.setText("");
ClintIDText.setText("");
AmontText.setText("");
OrderNOText.setText("");
CClint_IDText.setText("");
DateText.setText("");
//-----
SBCNCBox.setEnabled(false);
WIEFCSAll.setEnabled(false);
CClearB.setEnabled(false);
CClearAB.setEnabled(false);
CDoneB.setEnabled(false);
AmontText.setEnabled(false);
ClintAddB.setEnabled(false);
//-----
CClearABActionPerformed(evt);
WIEFCSAllActionPerformed(evt);
} catch (SQLException ex) {

}

JOptionPane.showMessageDialog(this, "Clint Info Entry Error , Maybe Each Of These Info is Existes ");
Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}

else
{
JOptionPane.showMessageDialog(this, "There Are No Item Selected for Cashing");
}
}
}

```

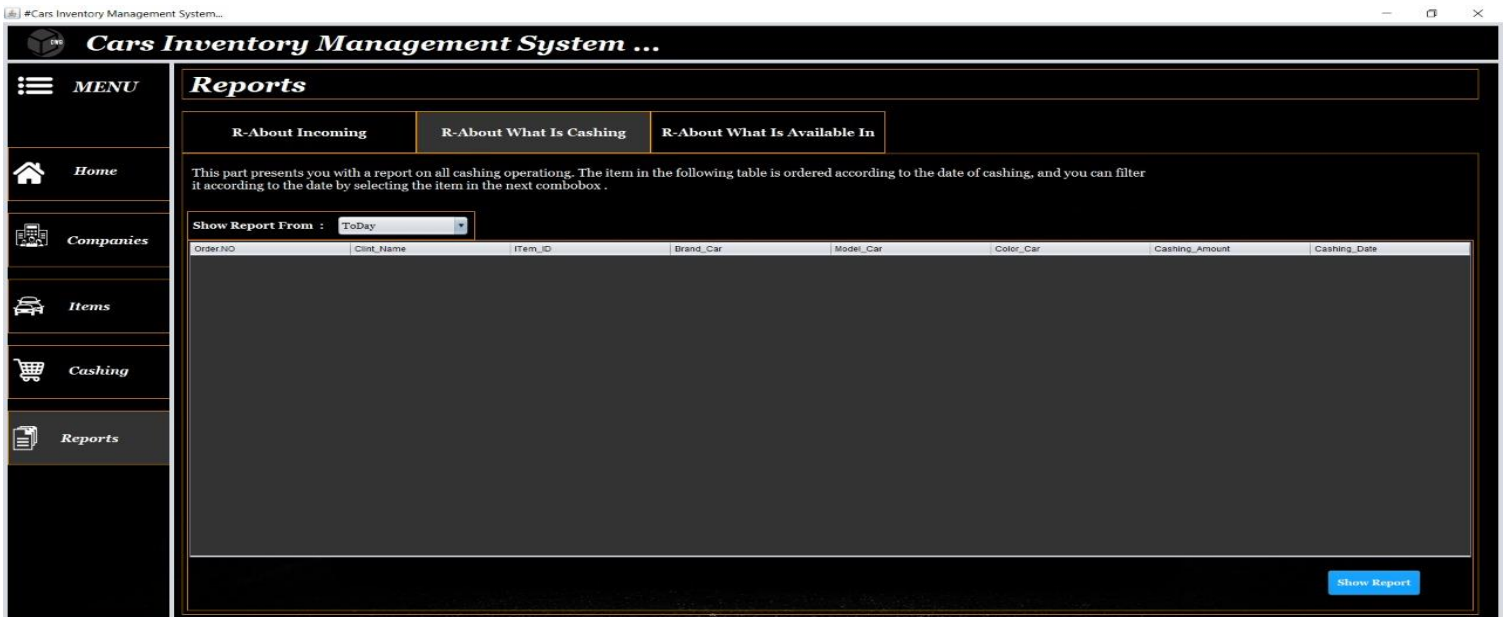
7. Report Section



It is through show all reports for the project, such as displaying the item that was incoming to the inventory, the item available in the inventory, and what was cashign for client

Show Report Button Code :

```
private void RiShowActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = null;
        String selected_Text = RiComboBox.getSelectedItem().toString();
        if(selected_Text=="ToDay")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-1);
            Date date =new Date();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select incoming.Supply_Process_No , companies.Comp_Name , incoming.Item_ID , item.Brand_Car ,
            + " item.Model_Car , item.Color_Car , incoming.Added_Amount , incoming.How_It_Add , incoming.Added_date from incoming ,
            + " item , companies where item.Item_ID = incoming.Item_ID and companies.Comp_ID = incoming.Comp_ID and incoming.Added_date >= ?;");
            System.out.println(date.toString());
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="Yesterday")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select incoming.Supply_Process_No , companies.Comp_Name , incoming.Item_ID , item.Brand_Car ,
            + " item.Model_Car , item.Color_Car , incoming.Added_Amount , incoming.How_It_Add , incoming.Added_date from incoming ,
            + " item , companies where item.Item_ID = incoming.Item_ID and companies.Comp_ID = incoming.Comp_ID and incoming.Added_date >= ?;");
            System.out.println(date.toString());
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastWeek")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-7);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select incoming.Supply_Process_No , companies.Comp_Name , incoming.Item_ID , item.Brand_Car ,
            + " item.Model_Car , item.Color_Car , incoming.Added_Amount , incoming.How_It_Add , incoming.Added_date from incoming ,
            + " item , companies where item.Item_ID = incoming.Item_ID and companies.Comp_ID = incoming.Comp_ID and incoming.Added_date >= ?;");
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastMonth")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.MONTH,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select incoming.Supply_Process_No , companies.Comp_Name , incoming.Item_ID , item.Brand_Car ,
            + " item.Model_Car , item.Color_Car , incoming.Added_Amount , incoming.How_It_Add , incoming.Added_date from incoming ,
            + " item , companies where item.Item_ID = incoming.Item_ID and companies.Comp_ID = incoming.Comp_ID and incoming.Added_date >= ?;");
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastYear")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.YEAR,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select incoming.Supply_Process_No , companies.Comp_Name , incoming.Item_ID , item.Brand_Car , item.Model_Car
            stmt.setDate(1,sdate);
        }
    }
    ResultSet set = stmt.executeQuery();
    DefaultTableModel dm = new DefaultTableModel();
    dm.addColumn("Supply_Process_No");
    dm.addColumn("Comp_Name");
    dm.addColumn("Item_ID");
    dm.addColumn("Brand_Car");
    dm.addColumn("Model_Car");
    dm.addColumn("Color_Car");
    dm.addColumn("Added_Amount");
    dm.addColumn("How_It_Added");
    dm.addColumn("Added_Date");
    while (set.next())
    {
        String r[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5),set.getString(6),set.getString(7),
        set.getString(8),set.getString(9)};
        dm.addRow(r);
    }
    RiTable.setModel(dm);
} catch (SQLException ex) {
    Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
}
}
```

Show Report Button Code :

```
private void R2ShowBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = null;
        String selected_Text = R2ComboBox.getSelectedItem().toString();
        if(selected_Text=="ToDay")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-1);
            Date date =new Date();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select cashing.Order_No ,client.Client_Name, cashing.Item_ID, item.Brand_Car, item.Model_Car , "
                + " item.Color_Car , cashing.Cashing_Amount , Cashing.Cashing_Date from cashing ,item , "
                + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;");
            System.out.println(date.toString());
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="Yesterday")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select cashing.Order_No ,client.Client_Name, cashing.Item_ID, item.Brand_Car, item.Model_Car , "
                + " item.Color_Car , cashing.Cashing_Amount , Cashing.Cashing_Date from cashing ,item , "
                + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;");
            System.out.println(date.toString());
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastWeek")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.DATE,-7);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select cashing.Order_No ,client.Client_Name, cashing.Item_ID, item.Brand_Car, item.Model_Car , "
                + " item.Color_Car , cashing.Cashing_Amount , Cashing.Cashing_Date from cashing ,item , "
                + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;");
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastMonth")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.MONTH,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select cashing.Order_No ,client.Client_Name, cashing.Item_ID, item.Brand_Car, item.Model_Car , "
                + " item.Color_Car , cashing.Cashing_Amount , Cashing.Cashing_Date from cashing ,item , "
                + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;");
            stmt.setDate(1,sdate);
        }
        else if(selected_Text=="LastYear")
        {
            Calendar cal = Calendar.getInstance();
            cal.add(Calendar.YEAR,-1);
            Date date =cal.getTime();
            java.sql.Date sdate = new java.sql.Date(date.getTime());
            System.out.println(sdate);
            stmt = con.prepareStatement("select cashing.Order_No ,client.Client_Name, cashing.Item_ID, item.Brand_Car, "
                + " item.Model_Car , item.Color_Car , cashing.Cashing_Amount , Cashing.Cashing_Date from cashing ,item , "
                + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;");
            stmt.setDate(1,sdate);
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, ex.getMessage());
    }
}
```



```

        + " client where item.Item_ID = cashing.Item_ID and client.Client_ID = cashing.Client_ID and cashing.Cashing_Date >= ?;";
        stmt.setDate(1,sdate);
    }
    ResultSet set = stmt.executeQuery();
    DefaultTableModel dm = new DefaultTableModel();
    dm.addColumn("Order.NO");
    dm.addColumn("Client_Name");
    dm.addColumn("Item_ID");
    dm.addColumn("Brand_Car");
    dm.addColumn("Model_Car");
    dm.addColumn("Color_Car");
    dm.addColumn("Cashing_Amount");
    dm.addColumn("Cashing_Date");
    int dupl_Check = 0 ;
    int i = 0 ;
    while (set.next())
    {
        if(dupl_Check == Integer.parseInt(set.getString(1)) )
        {
            String r[]={"-----", "-----", set.getString(3), set.getString(4), set.getString(5), set.getString(6), set.getString(7), "-----"};
            dm.addRow(r);
        }
        else
        {
            //System.out.println(set.getString(1));
            if(i!=0)
            {
                String s[]={"-----", "-----", "-----", "-----", "-----", "-----", "-----", "-----"};
                dm.addRow(s);
            }
            String r[]={set.getString(1), set.getString(2), set.getString(3), set.getString(4), set.getString(5), set.getString(6), set.
                getString(7), set.getString(8)};
            dm.addRow(r);
            dupl_Check = Integer.parseInt(set.getString(1));
            i++;
        }
        R2Table.setModel(dm);
    }
    catch (SQLException ex) {
        Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

#Cars Inventory Management System...

Cars Inventory Management System ...

MENU

Home

Companies

Items

Cashing

Reports

Reports

R-About Incoming

R-About What Is Cashing

R-About What Is Available In

This part presents you with a report on all the available item in inventory, ordered according to the company name, and you can filter it according to the company name by selecting the item in the combobox.

Show Report About Item For : Company/ies

Title 1	Title 2	Title 3	Title 4

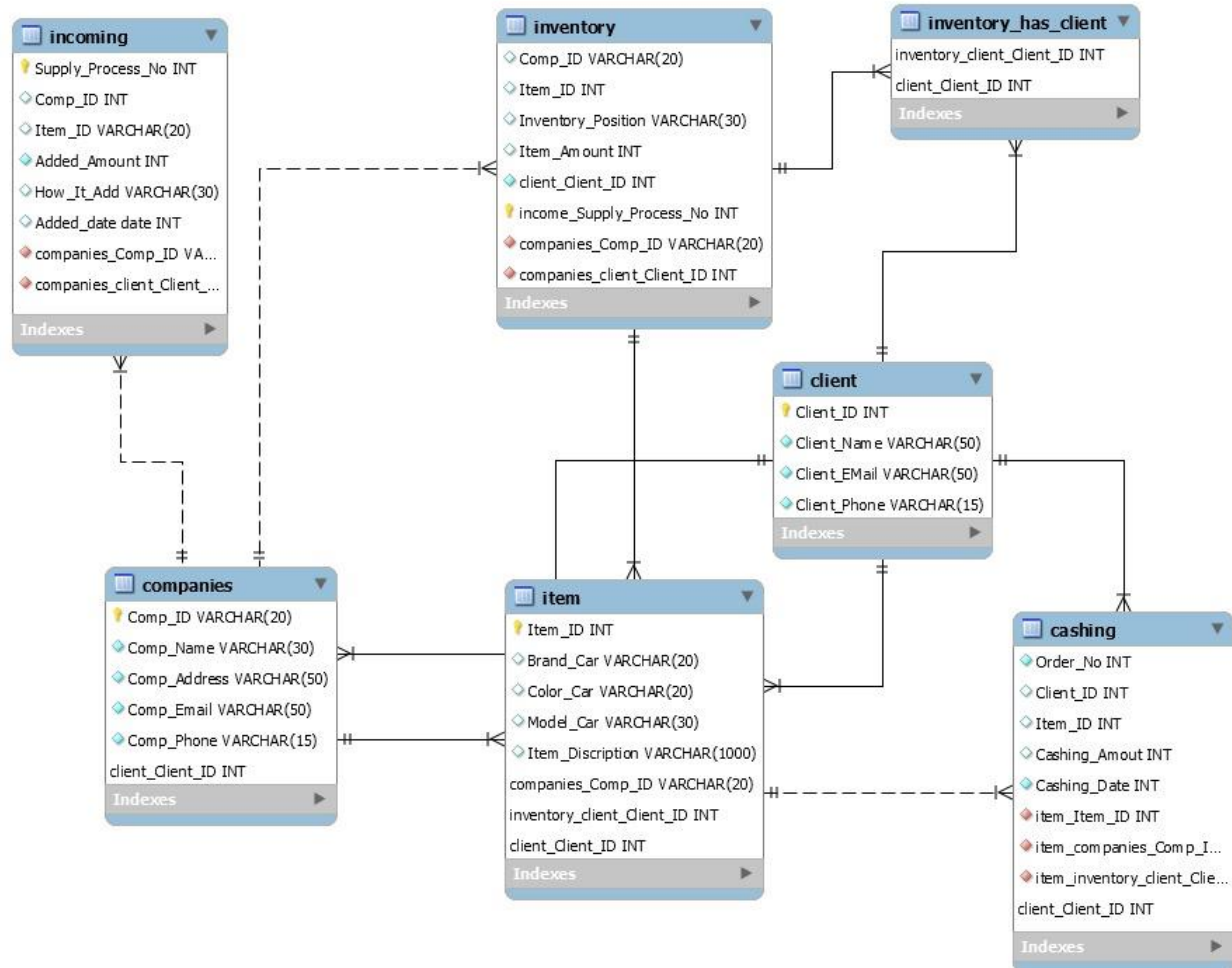
Show Report

Show Report Button Code :

```
private void R3ShowBActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/project", "root", "Root");
        PreparedStatement stmt = null;
        String selected_Text = R3ComboBox.getSelectedItem().toString();
        if(selected_Text=="ALL")
        {
            stmt = con.prepareStatement("select companies.Comp_Name, item.item_ID, item.Brand_Car , item.Model_Car ,item.Color_Car ,
            + " item.Item_Discription ,inventory.Item_Amount from companies, item , inventory where item.Item_ID = inventory.Item_ID and"
            + " inventory.Comp_ID = companies.Comp_ID and inventory.Item_Amount > 0 order by companies.Comp_Name;");
        }
        else
        {
            stmt = con.prepareStatement("select companies.Comp_Name, item.item_ID, item.Brand_Car , item.Model_Car ,item.Color_Car ,
            + " item.Item_Discription ,inventory.Item_Amount from companies, item , inventory where item.Item_ID = inventory.Item_ID and"
            + " inventory.Comp_ID = companies.Comp_ID and companies.Comp_Name = ? and inventory.Item_Amount > 0 order by companies.Comp_Name;");
            stmt.setString(1,selected_Text);
        }
        ResultSet set = stmt.executeQuery();
        DefaultTableModel dm = new DefaultTableModel();
        dm.addColumn("Company_Name");
        dm.addColumn("Item_ID");
        dm.addColumn("Brand_Car");
        dm.addColumn("Model_Car");
        dm.addColumn("Color_Car");
        dm.addColumn("Item_Discription");
        dm.addColumn("AviIn_Amount");
        String chForSeparate = "";
        int i =0;
        while (set.next())
        {
            if(chForSeparate.equals(set.getString(1)))
            {
                String z[]={set.getString(2),set.getString(3),set.getString(4),set.getString(5),set.getString(6),set.getString(7)};
                dm.addRow(z);
            }
            else
            {
                //System.out.println(set.getString(1));
                if(i!=0)
                {
                    String s[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5),set.getString(6),set.getString(7)};
                    dm.addRow(s);
                }
                String z[]={set.getString(1),set.getString(2),set.getString(3),set.getString(4),set.getString(5),set.getString(6),set.getString(7)};
                dm.addRow(z);
                chForSeparate = set.getString(1);
                System.out.println(chForSeparate);
            }
            i++;
        }
        R3Table.setModel(dm);
    } catch (SQLException ex) {
        Logger.getLogger(system.class.getName()).log(Level.SEVERE, null, ex);
    }
}

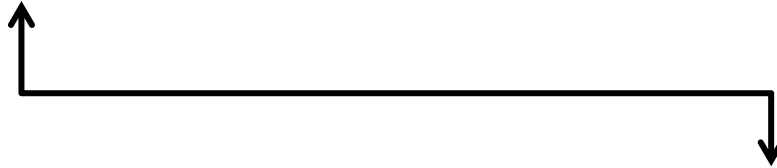
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    if(CIDText.getText().trim().isEmpty() && CNameText.getText().trim().isEmpty() && CAddressText.getText().trim().isEmpty()
    && CEmailText.getText().trim().isEmpty() && CPhoneText.getText().trim().isEmpty())
    {
        JOptionPane.showMessageDialog(this, "Ooh!! There Are No data To Clear It");
    }
    else
    {
        CIDText.setText("");
        CNameText.setText("");
        CAddressText.setText("");
        CEmailText.setText("");
        CPhoneText.setText("");
    }
}
```

8. ERD



9. Mapping

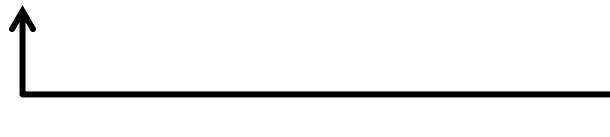
<u>Comp-ID</u>	Name	Address	Phone	E-mail
-----------------------	------	---------	-------	--------



<u>Item-ID</u>	Brand -Car	Color - Car	Model - Car	Item Description	<u>Comp-ID</u>
-----------------------	---------------	----------------	----------------	---------------------	-----------------------

1. Companies 1 to M Item.

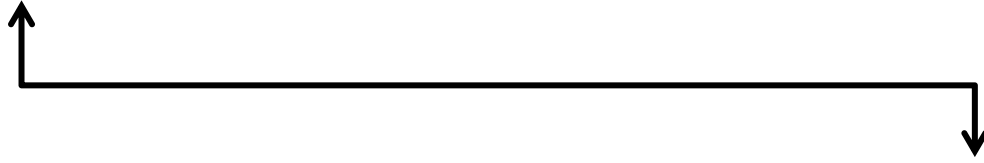
<u>Comp-ID</u>	Name	Address	Phone	E-mail
-----------------------	------	---------	-------	--------



<u>Client-ID</u>	Inventory- Position	Item- Amount	<u>Comp-ID</u>	<u>Item-ID</u>
-------------------------	------------------------	-----------------	-----------------------	-----------------------

2. Companies 1 to M Inventory.

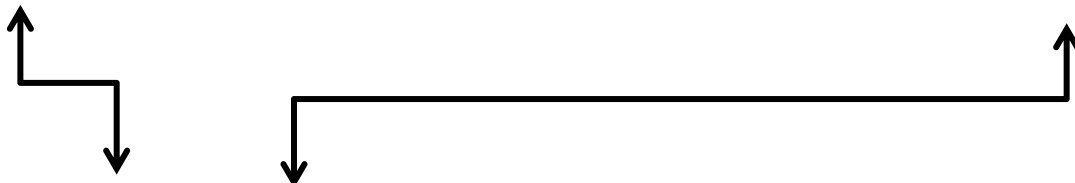
<u>Client-ID</u>	Name-Client	Phone	E-Mail
-------------------------	-------------	-------	--------



<u>Comp-ID</u>	Name	Address	Phone	E-mail	Client-ID
-----------------------	------	---------	-------	--------	-----------

3. Client 1 to M Companies.

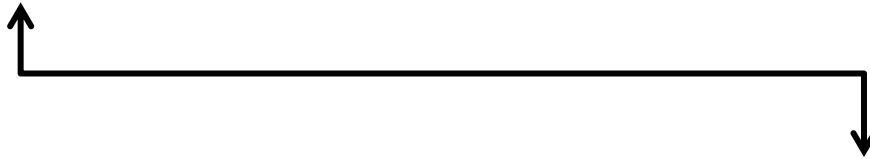
<u>Comp-ID</u>	Name	Address	Phone	E-mail	<u>Client-ID</u>
-----------------------	------	---------	-------	--------	-------------------------



<u>Comp-ID</u>	<u>Client-ID</u>	<u>Supply-Process-No</u>	Comp-Name	<u>Item-ID</u>	Added-Amount	Added-date.
-----------------------	-------------------------	---------------------------------	-----------	-----------------------	--------------	-------------

4. Companies 1 to M incoming.

<u>Item-ID</u>	Brand - Car	Color - Car	Model - Car	Item Description
-----------------------	----------------	----------------	----------------	---------------------



<u>Client-ID</u>	Inventory- Position	Item- Amount	<u>Item-ID</u>
-------------------------	------------------------	-----------------	-----------------------

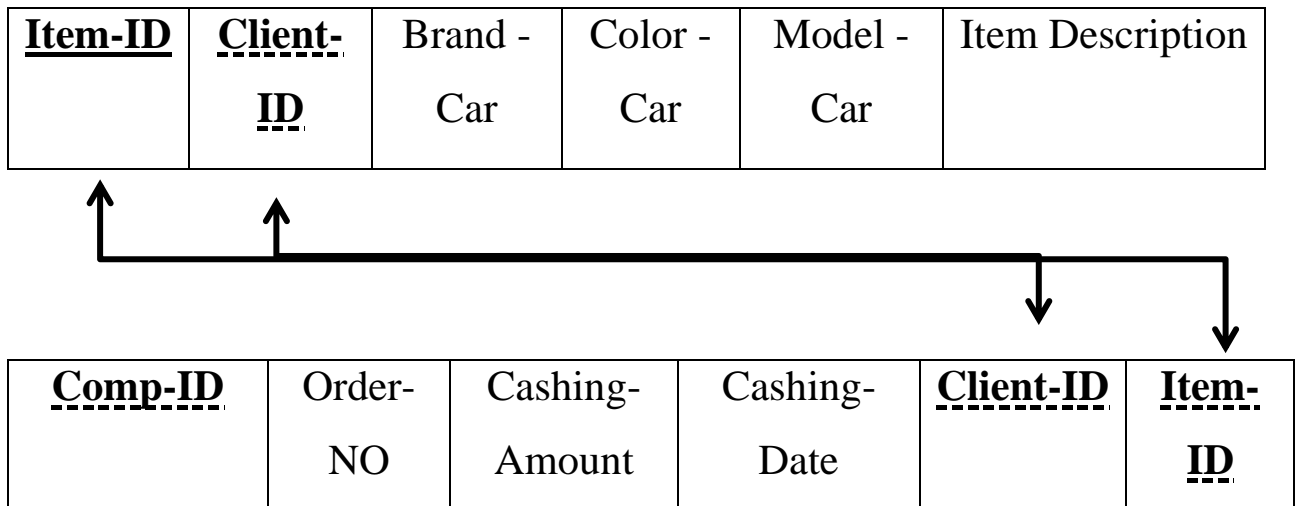
5. Item 1 to M Inventory.

<u>Client-ID</u>	Name-Client	Phone	E-Mail
-------------------------	-------------	-------	--------



<u>Item-ID</u>	Brand - Car	Color - Car	Model - Car	Item Description	<u>Client-ID</u>
-----------------------	----------------	----------------	----------------	---------------------	-------------------------

6. Client 1 to M Item.



7. Item I to M Cashing.

<u>Client-ID</u>	Name-Client	Phone	E-Mail
-------------------------	-------------	-------	--------

<u>Comp-ID</u>	Name	Address	Phone	E-mail
-----------------------	------	---------	-------	--------

<u>Comp-ID</u>	<u>Client-ID</u>
-----------------------	-------------------------

8. Inventory M to M Client.