Loan Management System using Java Swing
Part-I

Objective

Design and develop a Loan Management System using Java Swing for GUI and SQlite for data storage. The system will manage borrower data, loan records, interest calculation, and payment tracking with reporting and printing functionality.

Project Description

This system maintains loan-related information, including borrowers' details and payment records. The application will use two tables:

- **loan_table:** to store borrower and loan information
- **pay_record_table:** to store payment records related to each borrower

Main Functional Requirements

1. Borrower Management
2. Interest Calculation
3. Payment Record Management
4. Reporting and Printing
5. Data Persistence
6. Validations

Main Functional Requirements

1. **Borrower Management**
   - Add new borrower data (name, loan amount, loan date, interest rate, duration, etc.)
   - Edit borrower data
   - Delete borrower data
   - View list of all borrowers in a JTable (with sorting and searching features)

Main Functional Requirements

2. Interest Calculation

- Calculate interest(Monthly)

3. Payment Record Management

- Add payment record (amount paid, date of payment, etc.)
- View all payments made by a borrower
- Show remaining balance based on total loan and payments made

Main Functional Requirements

4. Reporting and Printing

- Generate and print Borrower Detail Report in PDF format
- Generate and print Payment History Report in PDF format
- Use a PDF library like **iText**

5. Data Persistence

- Use SQLite / MySQL to store borrower and payment records

6.Validations

- Input validation (e.g., loan amount > 0, name not empty, valid dates)
- Duplicate borrower check (based on name or unique ID)

Technology Requirements

- Language: Java
- GUI: Java Swing
- Database: SQLite
- PDF generation: iText library
- IDE: NetBeans

Database Design

```
CREATE TABLE loan (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL,
    nrc TEXT NOT NULL,
    loan_amount REAL NOT NULL,
    interest_rate REAL NOT NULL,
    duration INTEGER NOT NULL,
    start_date TEXT NOT NULL,
    total_pay_amount REAL
);
CREATE TABLE pay_record (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    loan_id INTEGER NOT NULL,
    payment_date TEXT NOT NULL,
    amount REAL NOT NULL,
    status TEXT NOT NULL,
    FOREIGN KEY (loan_id) REFERENCES loan(id)
);
```

Step(1) Do below steps

Step 1: create loandb database

>sqlite3 loandb.db

Step 2: create loan table

Step 3: create pay_record table

Step (2)

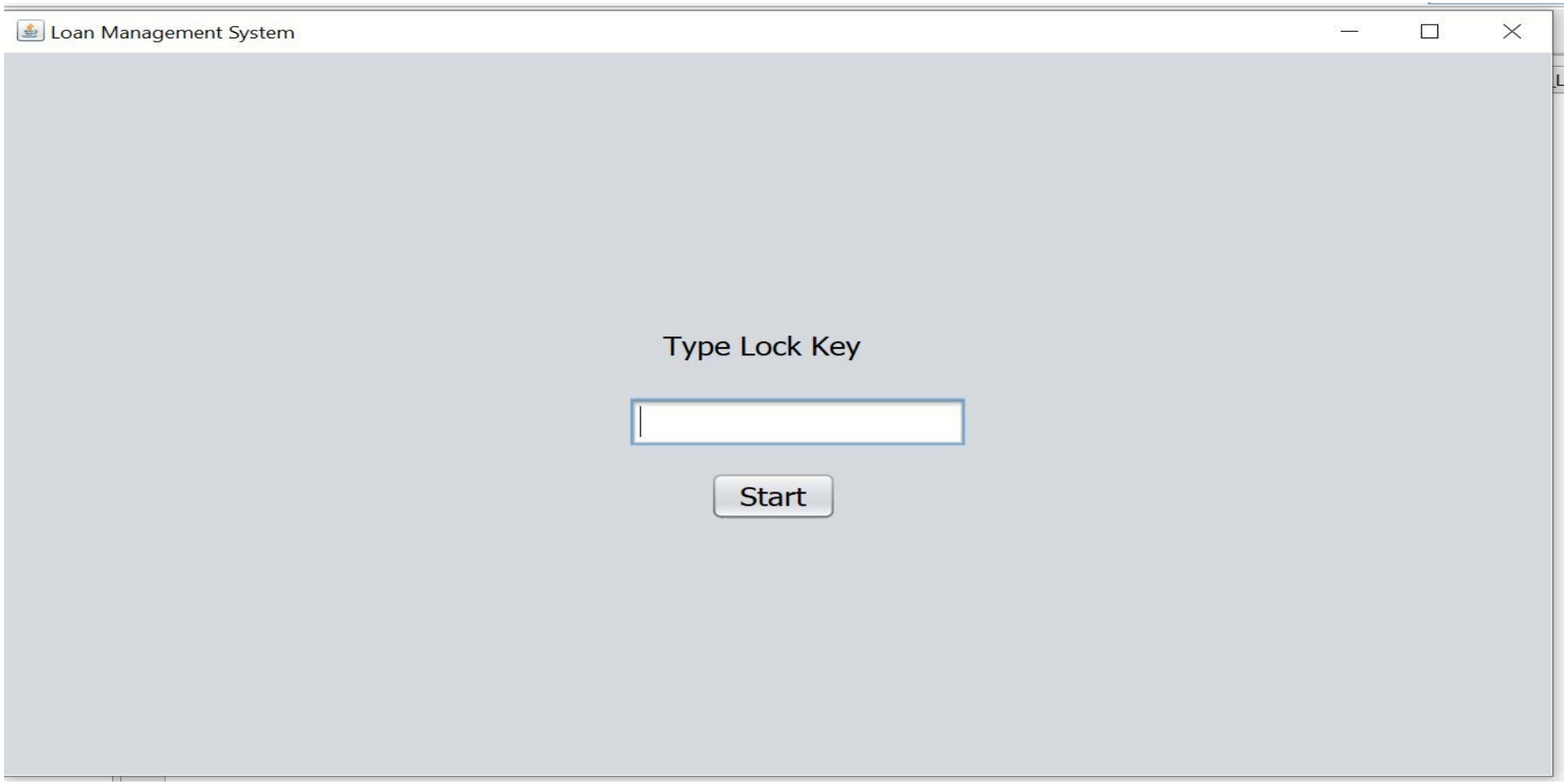
- Create two packages: dao, gui
- Define Dbconnection.java class under dao package
- After completing Step 1, run the `DbConnection` class. If the connection is successful, it will display the message: "Connection Success".

```
package dao;
import java.sql.*;

/**
 *
 * @author User
 */
public class DbConnection {
   public static Connection dbconnection() throws
SQLException{
      String url =
"jdbc:sqlite:C:\\Users\\User\\Desktop\\sqlite\\loandb.db";
     return  DriverManager.getConnection(url);
   }
   public static void main(String[] args) throws SQLException {
      Connection con=dbconnection();
      if(con!=null)
         System.out.println("Connection Success");



   }

}
```

Loan Management System using Java Swing
Part-II

Step(1) : Create Start Frame

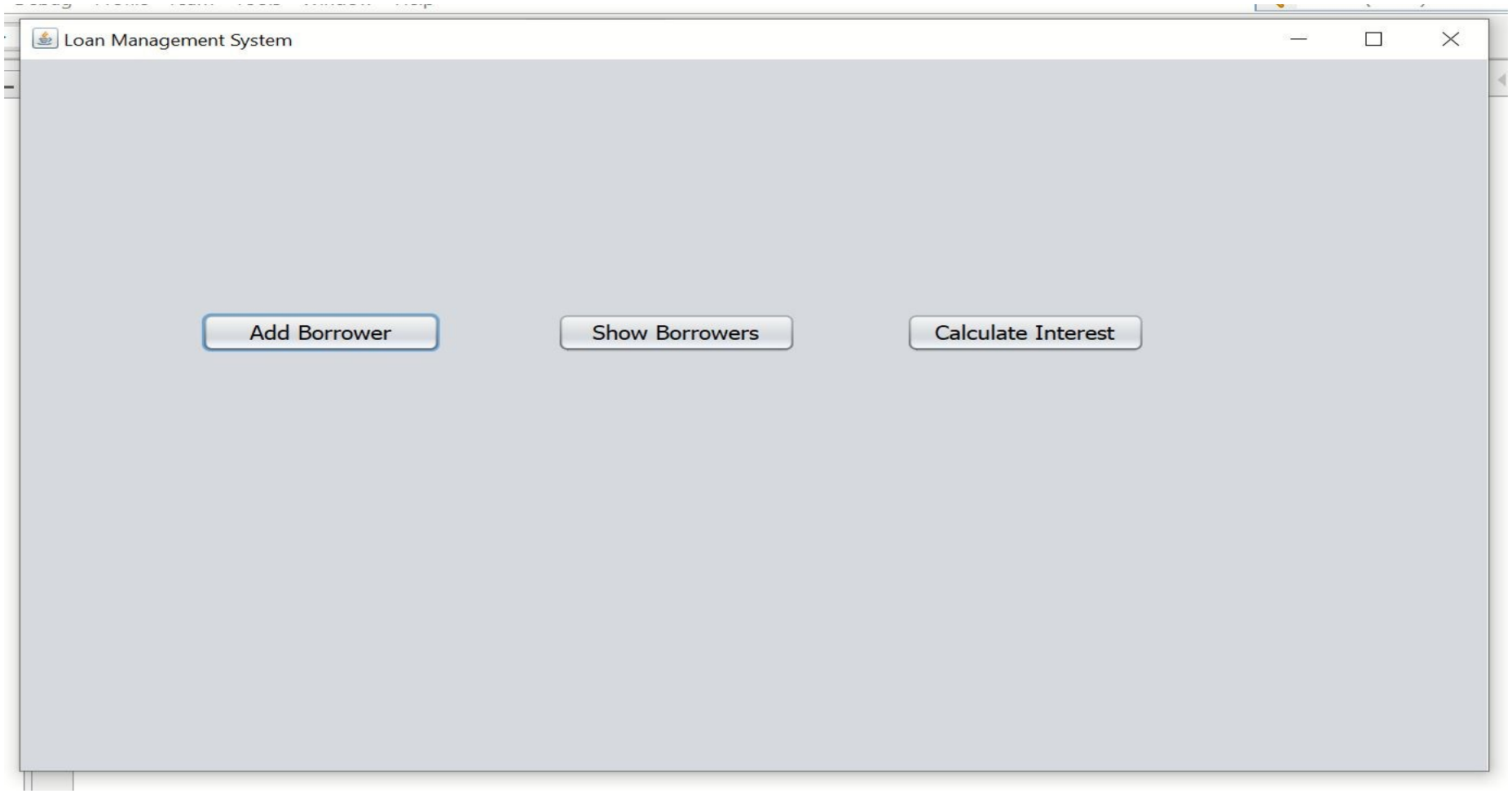
- Create Jframe: Start_Frame.java under gui package
- User types a password. If the password is valid, show the Dashboard_Frame. If the password is invalid, display a JOptionPane message dialog with the message: "Invalid Password".



```
private void btnStartActionPerformed(java.awt.event.ActionEvent evt) {
    char[] passChars = txtKey.getPassword();
    String pass = new String(passChars);
      //  String pass=txtKey.getText();
        if(pass.equals("12345")){
        this.setVisible(false);
      new  DashBoardFrame().setVisible(true);
       }
       else{
           JOptionPane.showMessageDialog(this, "Invalid Password, Try again!!");
      }
    }
```

Step(2) : Create Dashboard Frame

1. Create Jframe: Dashboard_Frame.java under gui package
2. You create three JFrames: Add_Borrower_Frame, Show_Borrowers_Frame, and Calculate_Interest_Frame. When the user clicks one of these buttons, the corresponding JFrame is displayed.



```
 private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
     this.setVisible(false);
     new Borrower_Frame().setVisible(true);
    }

 private void btnShowActionPerformed(java.awt.event.ActionEvent evt) {

          this.setVisible(false);
          new Show_Borrowers_Frame().setVisible(true);
}

    private void btnCalActionPerformed(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        new Calculate_Interest_Frame().setVisible(true);
    }
```

Loan Management System using Java Swing
Part-III

Step(1):

Define the following classes under the dao package:

1. DbConnection.java➔ Responsible for establishing and managing the database connection.
2. Loan.java➔ A model class used to create loan objects containing loan
3. details.PayRecord.java➔ A model class used to create payment record objects for each borrower's monthly payments.
4. LoanDao.java➔ An interface that declares all the abstract methods required for the project, such as adding, updating, retrieving, or deleting loan and payment data.
5. LoanDaoImpl.java➤ A class that implements the LoanDao interface by providing definitions for all its abstract methods. It interacts with the database to perform CRUD operations.

Loan.java

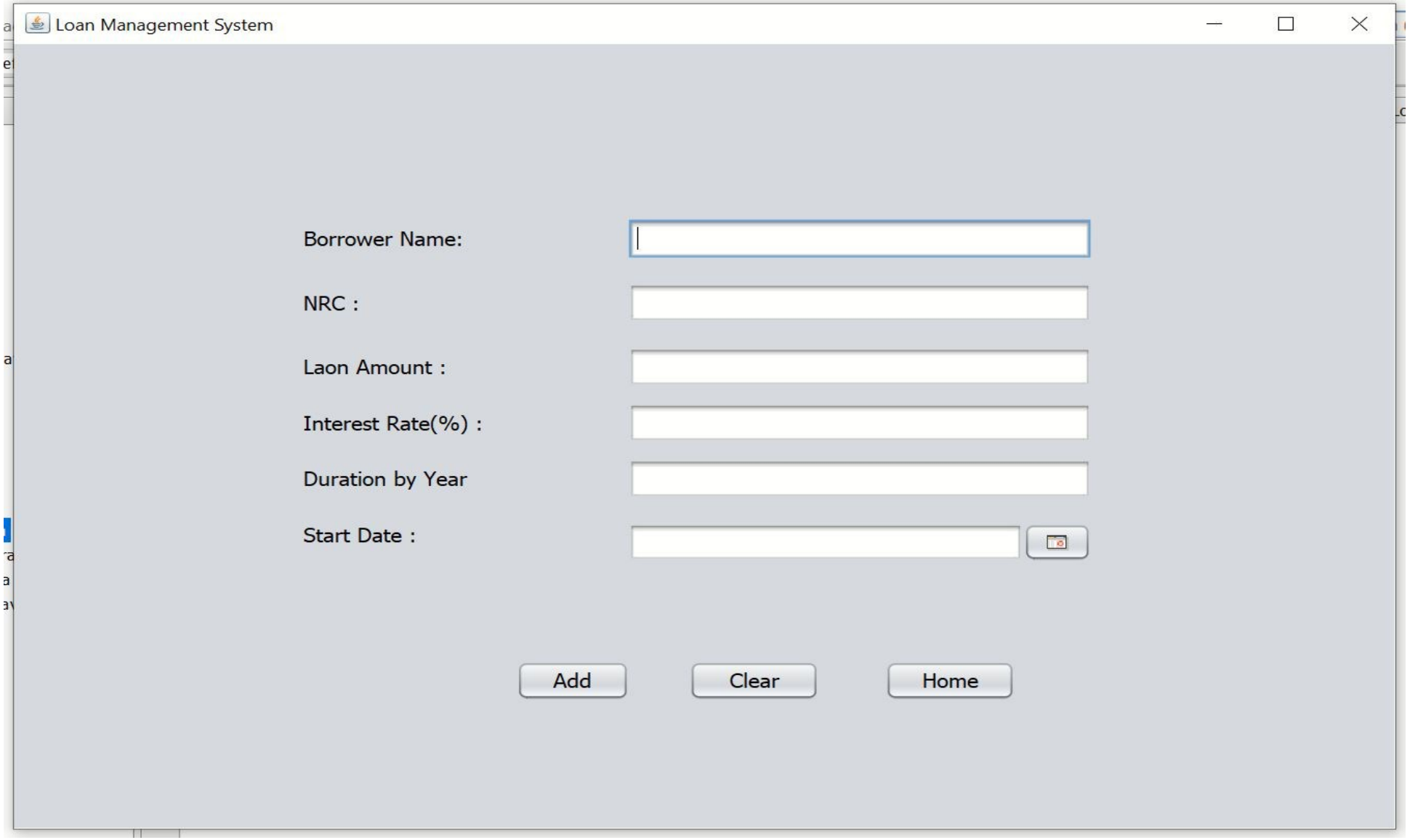
```
package dao;

/**
 *
 * @plmyan
 * to generate getter/setter methods=> press alt+insert
 *
 */
public class Loan {

private  int id, duration;
private  String bname,nrc,startDate;
private double amount,interest, total_pay_amount;

//Generate getter/setter methods


}
```

PayRecord.java

```
package dao;

/**
 *
 * @plmyan
 * to generate getter/setter methods=> press alt+insert
 *
 */
public class PayRecord {
    private int pid, lid;
    private String pdate,status;
    double amount;

//Generate getter/setter methods


}
```

Step(2) Create Add_Borrower_Frame

- Open Add_Borrower_Frame.java and add components according to given UI design



Step1.1 open LoanDao.java and insert below three abstract methods:

```
/*
 *Dao( Data Access Object)
 *
 */

package dao;
import java.util.*;
import dao.Loan;

public interface LaonDao {
    public List<Loan> showAllLoans();
    public void delete(String nrc);
    public void update(Loan loan);
}
```

Step1.2 open LoanDaoImpl.java and add below method

```
public List<Loan> showAllLoans() {
    List<Loan> laonlist = new ArrayList<>();
    try {
        PreparedStatement pstat;
        ResultSet rs;
        String sql;
        sql = "select name,nrc,loan_amount,interest_rate,duration,start_date,total_pay_amount from loan";
        pstat = con.prepareStatement(sql);
        rs = pstat.executeQuery();
        while (rs.next()) {
            Loan laon = new Loan(); // it inside loop
            laon.setBname(rs.getString("name"));
            laon.setNrc(rs.getString("nrc"));
            laon.setAmount(rs.getDouble("loan_amount"));
            laon.setInterest(rs.getDouble("interest_rate"));
            laon.setDuration(rs.getInt("duration"));
            laon.setStartDate(rs.getString("start_date"));
            laon.setTotal_pay_amount(rs.getDouble("total_pay_amount"));
            laonlist.add(laon);
        }
        pstat.close();
        rs.close();
    } catch (SQLException ex) {
        Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
    }
    return laonlist;
  }
```

```
@Override
  public void delete(String nrc) {
     int lid = 0;
     try {
        String sql1;
        PreparedStatement pstat;
        ResultSet rs;
        sql1 = "select id from loan where nrc=?";
        pstat = con.prepareStatement(sql1);
        pstat.setString(1, nrc);
        rs = pstat.executeQuery();
        if (rs.next()) {
           lid = rs.getInt("id");
        }
        deletePayRecord(lid);
        deleteLoanId(lid);
     } catch (SQLException ex) {
        System.out.println(ex.getMessage());

     }

  }
```

```
private void deletePayRecord(int lid) {
      try {
         String sql1;
         PreparedStatement pstat;
         sql1 = "DELETE FROM pay_record WHERE loan_id =?";
         pstat = con.prepareStatement(sql1);
         pstat.setInt(1, lid);
         pstat.executeUpdate();
      } catch (SQLException ex) {
Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
```

```
private void deleteLoanId(int lid) {
      try {
         String sql1;
         PreparedStatement pstat;
         sql1 = "DELETE FROM loan WHERE id =?";
         pstat = con.prepareStatement(sql1);
         pstat.setInt(1, lid);
         pstat.executeUpdate();
      } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
```

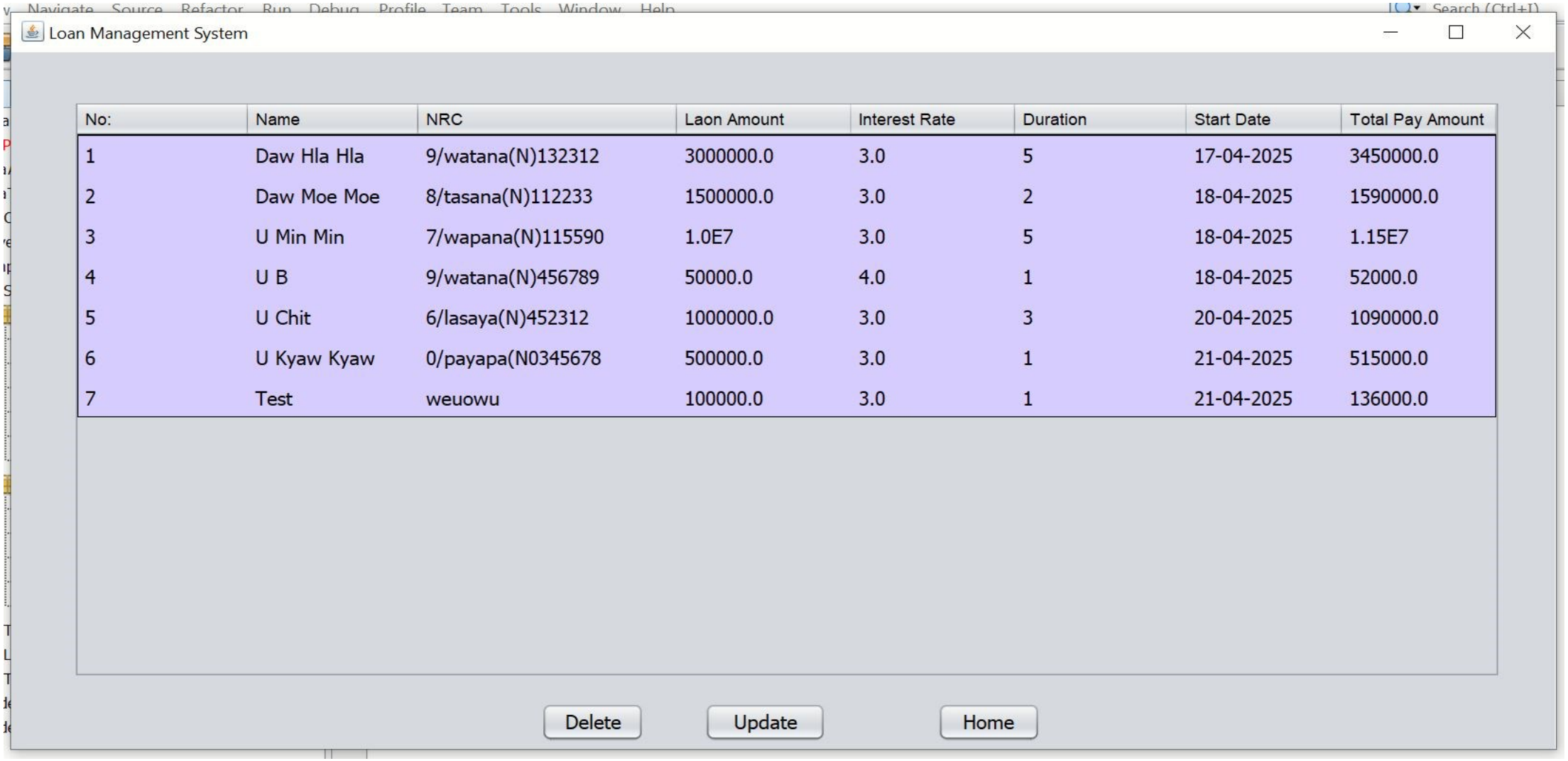
```
private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        double amount, total_pay_amount,interest_rate;
        int duration;
        amount=Integer.parseInt(txtLamount.getText());
        interest_rate=Double.parseDouble(txtInterest.getText());
        duration=Integer.parseInt(txtDuration.getText());
        Loan loan1=new Loan();
        loan1.setBname(txtName.getText());
        loan1.setNrc(txtNRC.getText());
        loan1.setAmount(amount);
        loan1.setInterest(interest_rate);
        loan1.setDuration(duration);
        total_pay_amount=amount+(amount*interest_rate/100*duration*12);
         loan1.setTotal_pay_amount(total_pay_amount);
        SimpleDateFormat df = new SimpleDateFormat("d-MM-y");
        loan1.setStartDate(df.format(sDate.getDate()));
       int status= new LaonDaoImple().insert(loan1);
       if(status!=0)
          JOptionPane.showMessageDialog(this, "Successfully Inserted!!!");
    } catch (SQLException ex) {
        Logger.getLogger(Borrower_Frame.class.getName()).log(Level.SEVERE, null, ex);
    }
  }
```

Step2.3 open Borrower_Frame.java and insert below code in the “Add” button ActionListener area:

Loan Management System using Java Swing
Part-IV

Step(1) Create Show_Borrowers_Frame

- Open Show_Borrowers_Frame.java and add components according to the provided UI design.
- This JTable is displayed when the user clicks the "Show Borrowers" button in Dashboard_Frame.



Step 1.1: Open Show_Borrowers_Frame.java and insert the following method into the constructor, as the JTable should be displayed when this frame is shown.

```
private void showAllLoan() throws SQLException {
      List<Loan> laonlist = new LaonDaoImple().showAllLoans();
      Object data[] = new Object[8];
      DefaultTableModel tmodel = (DefaultTableModel) showLoanTable.getModel();
      tmodel.setRowCount(0);
      int i = 0;
      for (Loan l : laonlist) {
         data[0] = ++i;
         data[1] = l.getBname();
         data[2] = l.getNrc();
         data[3] = l.getAmount();
         data[4] = l.getInterest();
         data[5] = l.getDuration();

         data[6] = l.getStartDate();
         data[7] = l.getTotal_pay_amount();
         tmodel.addRow(data);
      }
   }}
```

Step1.2 open LoanDao.java and insert below method:

public int insert(Loan laon);

```
/*
 *Dao( Data Access Object)
 *
 */

package dao;
import java.util.*;
import dao.Loan;

public interface LaonDao {
 public void delete(String nrc);
  public void update(Loan loan);
}
```

Step 1.3 open LoanDaoImpl.java and insert below method:

```
public void delete(String nrc) {
      int lid = 0;
      try {
         String sql1;
         PreparedStatement pstat;
         ResultSet rs;
         sql1 = "select id from loan where nrc=?";
         pstat = con.prepareStatement(sql1);
         pstat.setString(1, nrc);
         rs = pstat.executeQuery();
         if (rs.next()) {
            lid = rs.getInt("id");
         }
         deletePayRecord(lid);  // private method
         deleteLoanId(lid);  //private method
      } catch (SQLException ex) {
         System.out.println(ex.getMessage());

      }
  }
```

```
private void deletePayRecord(int lid) {
      try {
         String sql1;
         PreparedStatement pstat;
         sql1 = "DELETE FROM pay_record WHERE loan_id =?";
         pstat = con.prepareStatement(sql1);
         pstat.setInt(1, lid);
         pstat.executeUpdate();
      } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
  }
  private void deleteLoanId(int lid) {
      try {
         String sql1;
         PreparedStatement pstat;
         sql1 = "DELETE FROM loan WHERE id =?";
         pstat = con.prepareStatement(sql1);
         pstat.setInt(1, lid);
         pstat.executeUpdate();
      } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }    }
```

```
 @Override
   public void update(Loan loan) {
      try {
         String sql;
         PreparedStatement pstat;
         int id = getLoanId(loan.getNrc()); // private method
         System.out.println("Testing 1 " + id);
         System.out.println("Testing 1Name " + loan.getBname());
         sql = "update loan set  name=?,nrc=?,loan_amount=?,interest_rate=?,duration=?,start_date=?,total_pay_amount=?
where id=?";
         pstat = con.prepareStatement(sql);
         pstat.setString(1, loan.getBname());
         pstat.setString(2, loan.getNrc());
         pstat.setDouble(3, loan.getAmount());
         pstat.setDouble(4, loan.getInterest());
         pstat.setInt(5, loan.getDuration());
         pstat.setString(6, loan.getStartDate());
         pstat.setDouble(7, loan.getTotal_pay_amount());
         pstat.setInt(8, id);
         pstat.executeUpdate();
      } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
   }
```

```
private int getLoanId(String nrc) {
      String sql1;
      int lid = 0;
      try {
         PreparedStatement pstat;
         ResultSet rs;
         sql1 = "select id from loan where nrc=?";
         pstat = con.prepareStatement(sql1);
         pstat.setString(1, nrc);
         rs = pstat.executeQuery();
         if (rs.next())
            lid = rs.getInt("id");
        } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
      return lid;
  }
```

Step 1.4 open Show_Borrower_Frame.java and insert below code in the “Delete” button ActionListener area:

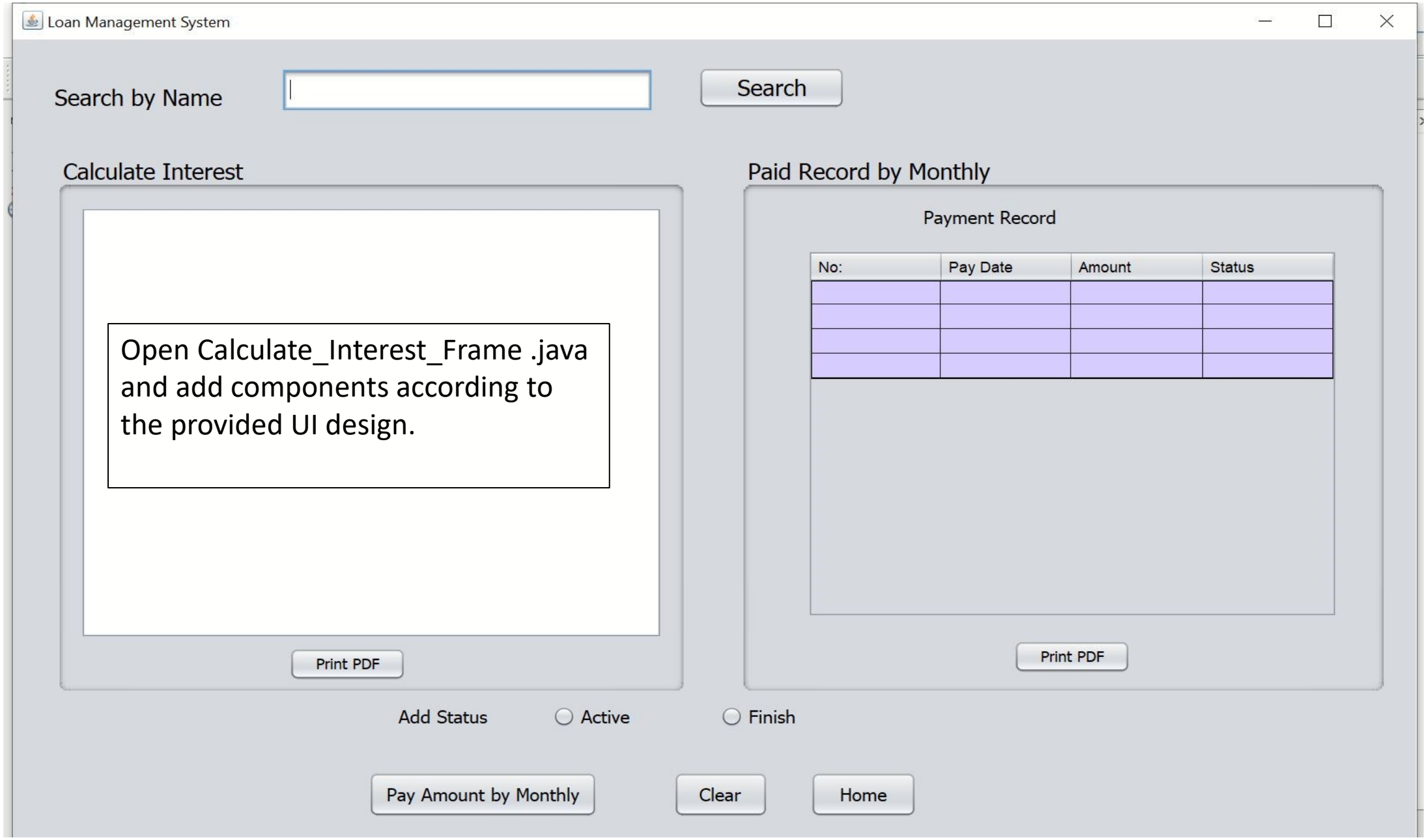
```
private void btnDelete1ActionPerformed(java.awt.event.ActionEvent evt) {

      try {
         DefaultTableModel tmodel = (DefaultTableModel) showLoanTable.getModel();
         int row = showLoanTable.getSelectedRow();
         System.out.println("Test row" + row);
         String nrc = (String) showLoanTable.getModel().getValueAt(row, 2);
         System.out.println("Test nrc" + nrc);
         new LaonDaoImple().delete(nrc);
         tmodel.removeRow(row);
         JOptionPane.showMessageDialog(this, " Success Delete Row");
      } catch (SQLException ex) {
         Logger.getLogger(Show_Laons_frame.class.getName()).log(Level.SEVERE, null, ex);
      }


   }
```

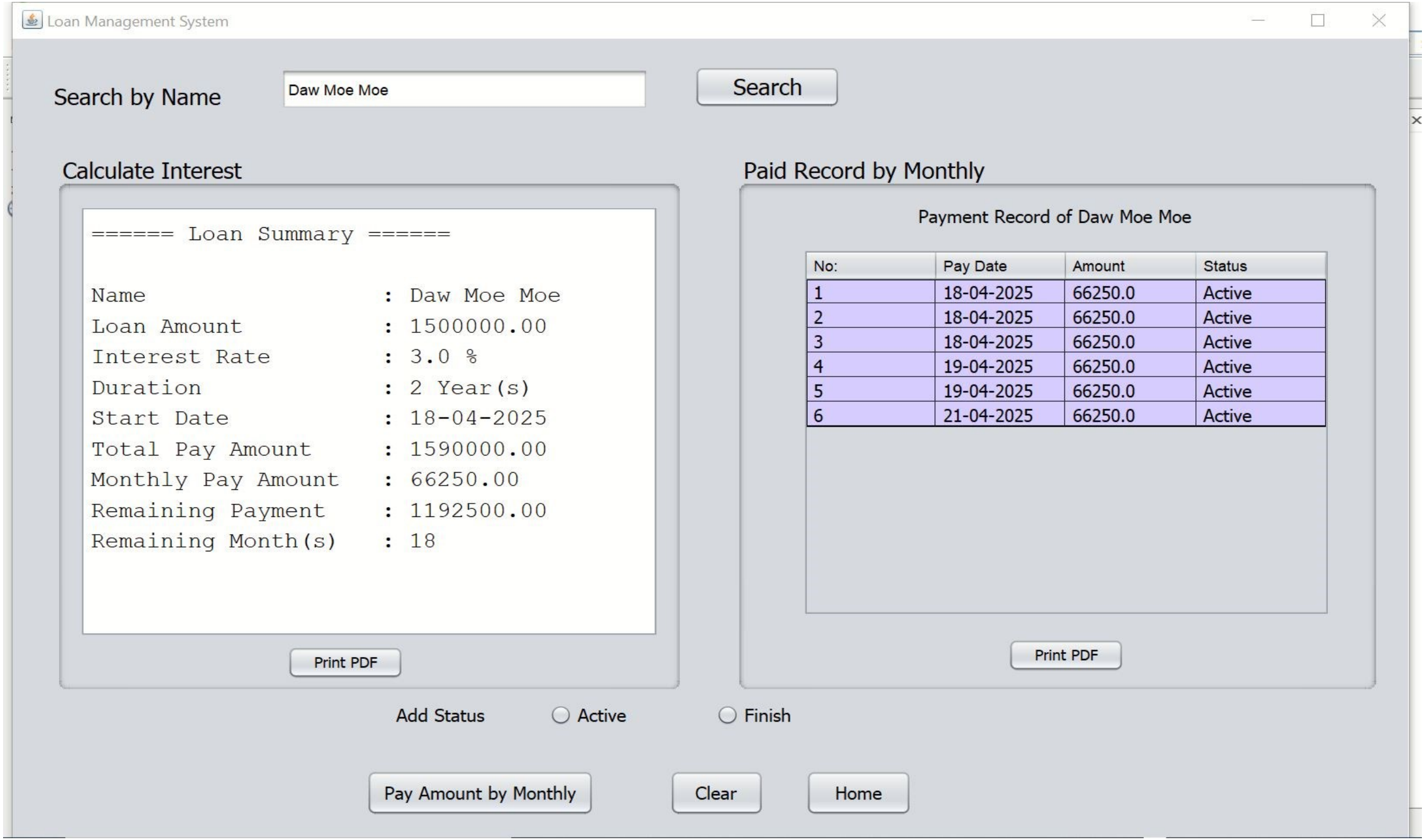
Insert below code in the “Update” button ActionListener area:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
        try {
        int row = showLoanTable.getSelectedRow();
        if (row == -1) {
            JOptionPane.showMessageDialog(this, "Please select a row to update.");
            return;
        }
        Loan ll = new Loan();
        DefaultTableModel tmodel = (DefaultTableModel) showLoanTable.getModel();
        ll.setBname((String) tmodel.getValueAt(row, 1));
        ll.setNrc((String) tmodel.getValueAt(row, 2));
        ll.setAmount(Double.parseDouble(tmodel.getValueAt(row, 3).toString()));
        ll.setInterest(Double.parseDouble(tmodel.getValueAt(row, 4).toString()));
        ll.setDuration(Integer.parseInt(tmodel.getValueAt(row, 5).toString()));
        ll.setStartDate((String) tmodel.getValueAt(row, 6));
        ll.setTotal_pay_amount(Double.parseDouble(tmodel.getValueAt(row, 7).toString()));
        new LaonDaoImple().update(ll);
        JOptionPane.showMessageDialog(this, "Successfully updated row.");
    } catch (SQLException ex) {
        Logger.getLogger(Show_Laons_frame.class.getName()).log(Level.SEVERE, null, ex);
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage());
    }    }
```

Loan Management System using Java Swing
Part-V

Step(1) Create Calculate_Interest_Frame




Loan Management System
Search by Name
Daw Moe Moe
Search
Calculate Interest
====== Loan Summary ======
Name : Daw Moe Moe
Loan Amount : 1500000.00
Interest Rate : 3.0 %
Duration : 2 Year(s)
Start Date : 18-04-2025
Total Pay Amount : 1590000.00
Monthly Pay Amount : 66250.00
Remaining Payment : 1192500.00
Remaining Month(s) : 18
Print PDF
Paid Record by Monthly
Payment Record of Daw Moe Moe
No: Pay Date Amount Status
1 18-04-2025 66250.0 Active
2 18-04-2025 66250.0 Active
3 18-04-2025 66250.0 Active
4 19-04-2025 66250.0 Active
5 19-04-2025 66250.0 Active
6 21-04-2025 66250.0 Active
Print PDF
Add Status
Active
Finish
Pay Amount by Monthly
Clear
Home

Step1.2 open LoanDao.java and insert below method:

public int insert(Loan laon);

```
/*
 *Dao( Data Access Object)
 *
 */
package dao;
import java.util.*;
import dao.Loan;

public interface LaonDao {
 public Loan searchBorrowerByName(String sname);
    public double getTotalPaymentByid(int loanId);
    public int getTotalMonthByid(int loadId);
    public void savePayRecord(PayRecord record);
    public List<PayRecord> showPayRecordbyId(int id);
}
```

Step 1.3 open LoanDaoImpl.java and insert below methods included in next slide :

```
@Override
   public Loan searchBorrowerByName(String sname) {
      Loan laon = null;
      try {
         PreparedStatement pstat;
         ResultSet rs;
         String sql;
         sql = "select id, name,loan_amount,interest_rate,duration,start_date,total_pay_amount from loan where name=? ";
         pstat = con.prepareStatement(sql);
         pstat.setString(1, sname);
         rs = pstat.executeQuery();
         while (rs.next()) {
            laon = new Loan();
            laon.setId(rs.getInt("id"));
            laon.setBname(rs.getString("name"));
            laon.setAmount(rs.getDouble("loan_amount"));
            laon.setInterest(rs.getDouble("interest_rate"));
            laon.setDuration(rs.getInt("duration"));
            laon.setStartDate(rs.getString("start_date"));
            laon.setTotal_pay_amount(rs.getDouble("total_pay_amount"));
         }
         pstat.close();
         rs.close();
      } catch (SQLException ex) {
         Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
      }
      return laon;    }
```

To Calculate remaining payment of each borrower

```
@Override
  public double getTotalPaymentByid(int loanId) {
     double totalPayment = 0.0;
     try {
        PreparedStatement pstat;
        ResultSet rs;
        String sql;
        sql = "select SUM(amount) As total_paid from pay_record where loan_id=? GROUP BY loan_id ";
        pstat = con.prepareStatement(sql);
        pstat.setInt(1, loanId);
        rs = pstat.executeQuery();
        if (rs.next()) {
           totalPayment = rs.getDouble("total_paid");
        }
        pstat.close();
        rs.close();

     } catch (SQLException ex) {
        Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
     }
     return totalPayment;
  }
```

To Calculate remaining month of each borrower

```
@Override
  public int getTotalMonthByid(int loadId) {
     int totalMonth = 0;
     try {
        PreparedStatement pstat;
        ResultSet rs;
        String sql;
        sql = "select COUNT(*) As total_month from pay_record where loan_id=? GROUP BY loan_id ";
        pstat = con.prepareStatement(sql);
        pstat.setInt(1, loadId);
        rs = pstat.executeQuery();
        if (rs.next()) {
           totalMonth = rs.getInt("total_month");
        }
        pstat.close();
        rs.close();

     } catch (SQLException ex) {
        Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
     }
     return totalMonth;
  }
```

To save monthly payment of each borrower

```
@Override
  public void savePayRecord(PayRecord record) {
     // System.out.println("In savePayRecord#########"+record.getLid());
     try {
        PreparedStatement pstat;
        String sql;
        sql = "insert into pay_record(loan_id, payment_date,amount,status) values(?,?,?,?)";
        pstat = con.prepareStatement(sql);
        pstat.setInt(1, record.getLid());
        pstat.setString(2, record.getPdate());
        pstat.setDouble(3, record.getAmount());
        pstat.setString(4, record.getStatus());
        status = pstat.executeUpdate();
        pstat.close();
     } catch (SQLException ex) {
        Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
     }

  }
```

```
@Override
  public List<PayRecord> showPayRecordbyId(int id) {
     List<PayRecord> paylist = new ArrayList<>();
     try {
        PreparedStatement pstat;
        ResultSet rs;
        String sql;
        sql = "select payment_date,amount,status from pay_record where loan_id=?";
        pstat = con.prepareStatement(sql);
        pstat.setInt(1, id);
        rs = pstat.executeQuery();
        while (rs.next()) {
           PayRecord pay = new PayRecord(); // it inside loop
           pay.setPdate(rs.getString("payment_date"));
           pay.setAmount(rs.getDouble("amount"));
           pay.setStatus(rs.getString("status"));
           paylist.add(pay);
        }
        pstat.close();
        rs.close();
     } catch (SQLException ex) {
        Logger.getLogger(LaonDaoImple.class.getName()).log(Level.SEVERE, null, ex);
     }
     return paylist;
  }
```

To show all payment record of each borrower

Step 1.4 open Calculate_Interest_Frame.java and insert below code in the “Search” button ActionListener area and call private method: showTextArea()

```
 private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {
        lblPay.setText("");
        btnPayByMonth.setEnabled(true);
        showTextArea();  // private method
}
```

```
private void showTextArea() {
    double remaining_payment;
    boolean checkAorF = false;
    int remaining_month;
    myArea.setText("");
    myArea.append("====== Loan Summary ======\n\n");
    try {
       Loan loan = new LaonDaoImple().searchBorrowerByName(txtSname.getText());
       if (loan == null) {
          JOptionPane.showMessageDialog(this, "Not found borrower ,! " + txtSname.getText());
          return;
       } else {

          id = loan.getId();
          pay_amount_monthly = (loan.getTotal_pay_amount() / (loan.getDuration() * 12));
          lblPay.setText("Payment Record of " + loan.getBname());
          myArea.append(String.format("%-20s : %s\n", "Name", loan.getBname()));
          myArea.append(String.format("%-20s : %.2f\n", "Loan Amount", loan.getAmount()));
          myArea.append(String.format("%-20s : %.1f %%\n", "Interest Rate", loan.getInterest()));
          myArea.append(String.format("%-20s : %d Year(s)\n", "Duration", loan.getDuration()));
          myArea.append(String.format("%-20s : %s\n", "Start Date", loan.getStartDate()));
          myArea.append(String.format("%-20s : %.2f\n", "Total Pay Amount", loan.getTotal_pay_amount()));
          myArea.append(String.format("%-20s : %.2f\n", "Monthly Pay Amount", pay_amount_monthly));
          remaining_payment = new LaonDaoImple().getTotalPaymentByid(id);
```

```
if (remaining_payment == 0.0) {
                remaining_payment = loan.getTotal_pay_amount();
            } else {
                remaining_payment = loan.getTotal_pay_amount() - remaining_payment;
            }
            remaining_month = new LaonDaoImple().getTotalMonthByid(id);
            if (remaining_month == (loan.getDuration() * 12)) {
                checkAorF = true;
            } else if (remaining_month == 0) {
                remaining_month = loan.getDuration() * 12;
            } else {
                remaining_month = (loan.getDuration() * 12) - remaining_month;
            }
            myArea.append(String.format("%-20s : %.2f\n", "Remaining Payment", remaining_payment));
            myArea.append(String.format("%-20s : %d\n", "Remaining Month(s)", remaining_month));

            if (checkAorF) {
                myArea.setText("Done!  Congratulation!!!!!!!!!!");
                btnPayByMonth.setEnabled(false);
            }
            showPayRecord(id); // private method
        }//end outer else
    } catch (SQLException ex) {
        Logger.getLogger(Calculate_Interest_Frame.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

```
private void showPayRecord(int id) throws SQLException {

    List<PayRecord> paylist = new LaonDaoImple().showPayRecordbyId(id);
    Object data[] = new Object[4];
    tmodel2 = (DefaultTableModel) pay_record_table.getModel();
    tmodel2.setRowCount(0);
    int i = 0;
    for (PayRecord p : paylist) {
       data[0] = ++i;
       data[1] = p.getPdate();
       data[2] = p.getAmount();
       data[3] = p.getStatus();
       tmodel2.addRow(data);
    }
  }
```

```
private void btnPayByMonthActionPerformed(java.awt.event.ActionEvent evt) {
        btnG1.add(rdA);
        btnG1.add(rdF);
        try {
            PayRecord record = new PayRecord();
            record.setLid(id);
            SimpleDateFormat df = new SimpleDateFormat("d-MM-y");
            Date dd = new Date();
            record.setPdate(df.format(dd));
            record.setAmount(pay_amount_monthly);
            //  new LaonDaoImple().getTotalMonthByid(id)
            if (rdA.isSelected()) {
                record.setStatus("Active");
            } else {
                record.setStatus("Finish");
            }
            new LaonDaoImple().savePayRecord(record);
            showTextArea();
            showPayRecord(id);
        } catch (SQLException ex) {
            Logger.getLogger(Calculate_Interest_Frame.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
```

Insert below code in the “Pay Amount by Monthly” button ActionListener area:

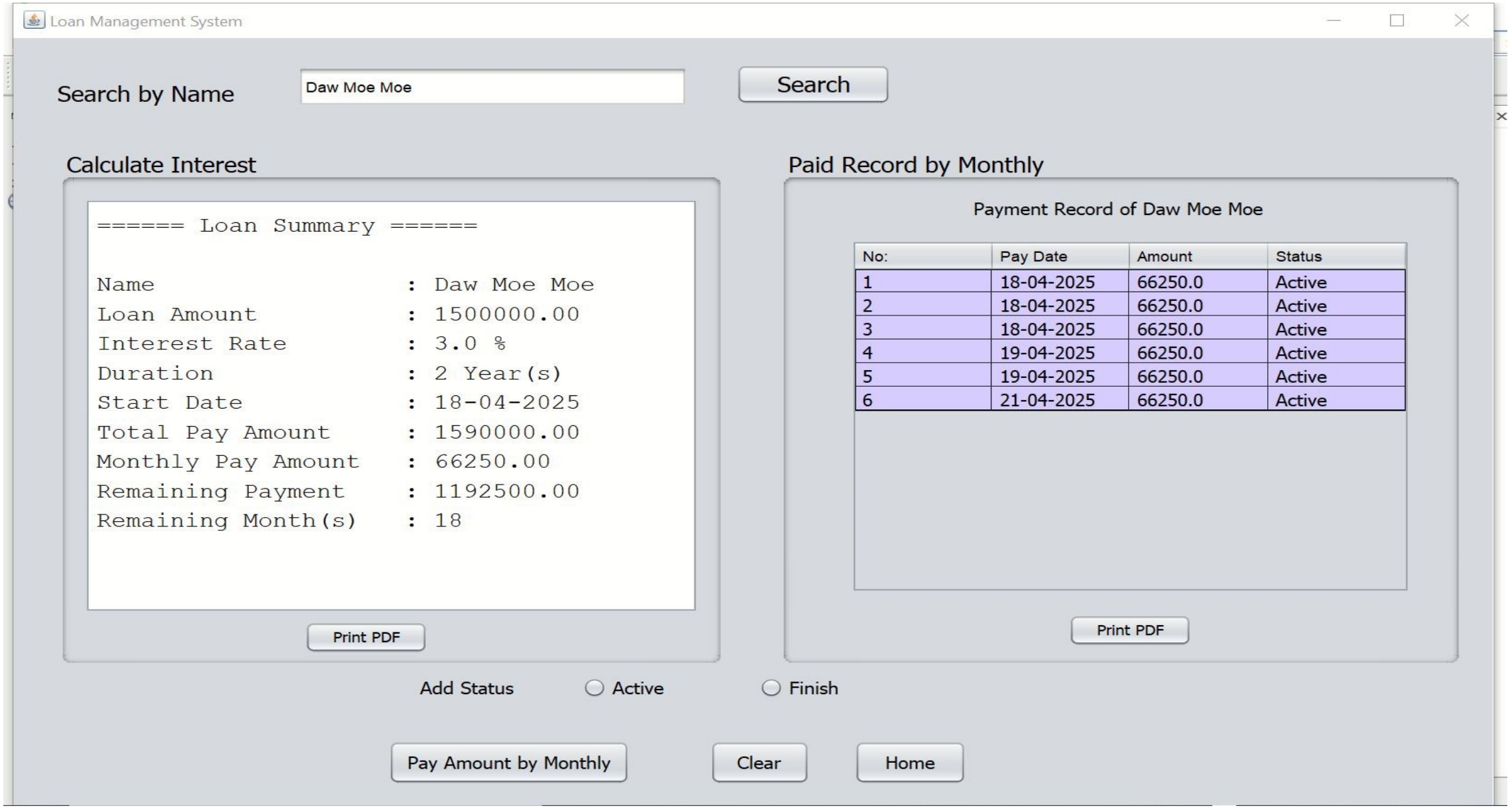
Insert below code in the “Cancel” button ActionListener area:

```
private void btnClearActionPerformed(java.awt.event.ActionEvent evt) {
      myArea.setText("");
     txtSname.setText("");
     tmodel2.setRowCount(0);
     lblPay.setText("");
  }
```

Loan Management System using Java Swing
Part-VI

To Print Loan data with PDF Format using iText library

Step(1) add itextpdf-5.5.13.2 jar into your project under Libraries Section



Step 1.2 open Calculate_Interest_Frame.java and insert below code in the "To Print" button ActionListener area and call private method: exportTextAreaToPDF(fileToSave.getAbsolutePath())

```
private void btnPrint1ActionPerformed(java.awt.event.ActionEvent evt) {

  JFileChooser fileChooser = new JFileChooser();
  fileChooser.setDialogTitle("Save PDF");
  fileChooser.setSelectedFile(new File("LoanSummary.pdf")); // default name

  int userSelection = fileChooser.showSaveDialog(this); // 'this' = your JFrame
  if (userSelection == JFileChooser.APPROVE_OPTION) {
     try {
        File fileToSave = fileChooser.getSelectedFile();
        exportTextAreaToPDF(fileToSave.getAbsolutePath()); //  Call private
     } catch (FileNotFoundException ex) {
        Logger.getLogger(Calculate_Interest_Frame.class.getName()).log(Level.SEVERE, null, ex);
     }
  }

  }
```

```
 private void exportTextAreaToPDF(String filePath) throws FileNotFoundException {
      Document document = new Document();
    try {
        PdfWriter.getInstance(document, new FileOutputStream(filePath));
        document.open();

        // Use Monospaced font to preserve formatting
        Font font = new Font(Font.FontFamily.COURIER, 12);
        String content = myArea.getText();
        Paragraph para = new Paragraph(content, font);
        document.add(para);

        JOptionPane.showMessageDialog(this, "PDF exported successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
    } catch (DocumentException | IOException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(this, "Error exporting PDF: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    } finally {
        document.close();
    }
}
```

Step 1.3 Insert below code in the “To Print” button ActionListener area and call private methodexportJTableToPDF(pay_record_table, fileToSave.getAbsolutePath());

```
private void btnPrint2ActionPerformed(java.awt.event.ActionEvent evt) {

    JFileChooser fileChooser = new JFileChooser();
  fileChooser.setDialogTitle("Save Table as PDF");
  fileChooser.setSelectedFile(new File("PayRecordTable.pdf"));

  int userSelection = fileChooser.showSaveDialog(this);
  if (userSelection == JFileChooser.APPROVE_OPTION) {
     File fileToSave = fileChooser.getSelectedFile();
     exportJTableToPDF(pay_record_table, fileToSave.getAbsolutePath());
  }
  }
```

```
 private void exportJTableToPDF(JTable table, String filePath) {
   Document document = new Document();
   try {
      PdfWriter.getInstance(document, new FileOutputStream(filePath));
      document.open();
      document.add(new Paragraph("          ====== Payment Record Table ======\n\n"));
       document.add(new Paragraph("          ======"+ txtSname.getText()+" ======\n\n"));
      PdfPTable pdfTable = new PdfPTable(table.getColumnCount());
      pdfTable.setWidthPercentage(100);

      // Add table headers
      for (int i = 0; i < table.getColumnCount(); i++) {
         pdfTable.addCell(new PdfPCell(new Phrase(table.getColumnName(i))));
      }
      // Add rows
      for (int rows = 0; rows < table.getRowCount(); rows++) {
         for (int cols = 0; cols < table.getColumnCount(); cols++) {
            Object value = table.getValueAt(rows, cols);
            pdfTable.addCell(value != null ? value.toString() : "");
         }
      }
      document.add(pdfTable);
      JOptionPane.showMessageDialog(null, "PDF exported successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);
   } catch (DocumentException | IOException e) {
      e.printStackTrace();
      JOptionPane.showMessageDialog(null, "Error exporting PDF: " + e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
   } finally {
      document.close();
   }
}
   }
```

Finished JavaSE Course.

Thank you for everything, my students!