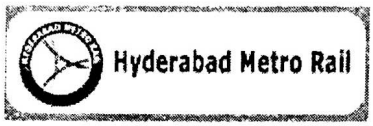
Winter Internship Report

An Organizational Study

At



Hyderabad

A project report on

Collection of data in real time .

Submitted by

Aryan Bampal (CSE17U003)

Under the guidance of Sohail Mathur, IT manager

CONTENTS

```
package gui;
import java.awt.Component;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.sql.*;
import java.text.ParseException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import javax.swing.BorderFactory;
import javax.swing.GroupLayout;
import javax.swing.JApplet;
import javax.swing.JButton;
import javax.swing.JFormattedTextField;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTextField;
import javax.swing.text.MaskFormatter;
public class classic extends JApplet {
// the only error shown is "no results shown by query" which does not affect or impede the program in anyway
//setting up a connection to database
private static   Connection connect() {
```

```
        String url="jdbc:postgresql://localhost:5432/mydb";
        String user="aryan";
        String password="eragon";
        Connection conn = null;
          try {
// registering the driver with drivermanager and loading it to get a connection object with getConnection()
            conn = DriverManager.getConnection(url, user, password);
            System.out.println("Connected to the PostgreSQL server successfully.");
        } catch (SQLException e) {
          System.out.println(e.getMessage());
        }
        return conn;
}
// function for turning seconds into (hh:mm:ss) format and returns a string
private static String conv(int timeInSeconds) {
        int hours = timeInSeconds / 3600;
        int secondsLeft = timeInSeconds - hours * 3600;
        int minutes = secondsLeft / 60;
        int seconds = secondsLeft - minutes * 60;
        String formattedTime = "";
        if (hours < 10)
          formattedTime += "0";
        formattedTime += hours + ":";

        if (minutes < 10)
          formattedTime += "0";
        formattedTime += minutes + ":";
```

```
            if (seconds < 10)
               formattedTime += "0";
            formattedTime += seconds ;

            return formattedTime;
}
//function for converting time(hh:mm:ss) into seconds and returns integer
private static int fstring(String x)
{
        int l,w,q;
        String[] parts = x.split(":");
        l=Integer.parseInt(parts[0]);
        w=Integer.parseInt(parts[1]);
        q=Integer.parseInt(parts[2]);
        l=l*3600+w*60+q;
        return l;
}
//function to calculate delay and return a string of (hh:mm:ss)
private static String funD(int i,int j) {
        String h;
        int f=0;
        f=j-i;
        if(f<=0)
        {

                f=0;
```

```
                h=conv(f);
                return h;
            }
            else
            {

                h=conv(f);
                return h;
    }
}
```

//function to calculate total time schedule and return string of (hh:mm:ss)

```
private static String ttschedule(int i,int k) {
        String h;
        int f=0;
        f=k-i;
        if(f<0)
        {
                f=f+86400;
                h=conv(f);
                return h;

        }
        else
        {
                h=conv(f);
                return h;
}
}
```

```
//function to calculate total time actual and return a string of (hh:mm:ss)
private static String ttactual(int j,int l) {
        String h;
        int f=0;
        f=l-j;
        if(f<0)
        {
                f=f+86400;
                h=conv(f);
                return h;
        }
        else
        {
                h=conv(f);
                return h;
        }
}
//function to calculate the count of delay and return 1 or 0
private static String funCD(int i,int j,int k,int l)
{
        int f,h;
        f=0;
        h=0;
        f=j-i;
        h=l-k;
        if(f<=60)
        {
                f=0;
```

```
            }
            else
            {
                    f=1;
            }
            if(h<=60)
            {
                    h=0;
            }
            else
            {
                    h=1;
            }
            if(f==1||h==1)
            {
              return Integer.toString(1);
    }
          else
          {
                  return Integer.toString(0);
          }
}
```

// these function make sure that the numbers displayed in GUI in editable fields are in the format of (##:##:##) and (##) and only accept numbers

```
private static MaskFormatter getSecondsMask()
        {
        MaskFormatter mask = null;
     try {
```

// setting up 4 propertychangelisteners for actual time arrival or departure to correct errors if 'get time' button is pressed at the wrong time

// field 1:-

```
PropertyChangeListener l1=new PropertyChangeListener() {
        @Override
                public void propertyChange(PropertyChangeEvent event1) {
                        Object yo1=event1.getSource();
                  for(int m=0;m<Trips;m++)
                  {
                        if(yo1==ANAG1d[m])
                        {
                                String text2 = event1.getNewValue() != null ?
event1.getNewValue().toString() : "00:00:00";
                                String text1=PNAG1d[m].getText();
                                String text3=PAME2a[m].getText();
                        String text4=AAME2a[m].getText();
                        int i=fstring(text1);
                        int j=fstring(text2);
                        int k=fstring(text3);
                        int l=fstring(text4);
                        String d1=funD(i, j);
                        String d2=funCD(i,j,k,l);
                        String s1=ttschedule(i,k);
                        String s2=ttactual(j,l);
                        countDelay[m].setText(d2);
                        D1label[m].setText(d1);
                        TTS1[m].setText(s1);
                        TTA1[m].setText(s2);
```

```
                    break;
                    }
                }
            }
        };
for(i=0;i<Trips;i++)
{
        ANAG1d[i].addPropertyChangeListener("value",l1);
}
//field 2:-
PropertyChangeListener l2=new PropertyChangeListener() {
        @Override

            public void propertyChange(PropertyChangeEvent event2) {
                Object yo2=event2.getSource();
              for(int p=0;p<Trips;p++)
              {
                if(yo2==AAME2a[p])
                {
                        String text4 = event2.getNewValue() != null ?
event2.getNewValue().toString() : "00:00:00";
                        String text1=PNAG1d[p].getText();
                        String text2=ANAG1d[p].getText();
                        String text3=PAME2a[p].getText();
                        int i=fstring(text1);
                        int j=fstring(text2);
                        int k=fstring(text3);
                        int l=fstring(text4);
```

```
                    String text3=PNAG2a[m].getText();
                    int i=fstring(text1);
                    int j=fstring(text2);
                    int k=fstring(text3);
                    int l=fstring(text4);
                    String d1=funD(k, l);
                    String d2=funCD(i,j,k,l);
                    String s1=ttschedule(i,k);
                    String s2=ttactual(j,l);
                     countDelay2[m].setText(d2);
                    D4label[m].setText(d1);
                    TTS2[m].setText(s1);
                    TTA2[m].setText(s2);
                    break;
                        }
                    }
                }
        });
}
```

//setting up save button connected to database (PSQL) and updates the content of the table in the database

```
JButton[] save=new JButton[Trips];
for(i=0;i<Trips;i++)
{
        save[i]=new JButton("SAVE");
        save[i].addActionListener(new ActionListener(){
                @Override
```

```
public void actionPerformed (ActionEvent evt5) {
        Object source=evt5.getSource();
        for(int m=0;m<Trips;m++)
        {
            if(source==save[m])
            {
                try {
                PreparedStatement stmt=
connect().prepareStatement("UPDATE SCHEDULE5 SET train_set2=?, ANAG1d=?, DELAY=?,
"AAME2a=?, DELAY2=?, REMARK=?, COUNT_OF_DELAY=?,"+
"TTSUP=?, TTAUP=?, train_set=?, AAME2d=?, DELAY3=?, ANAG2a=? ,DELAY4=?, REMARK2=?, "+
"COUNT_OF_DELAY2=?, TTSDOWN=?, TTADOWN=?,"+
"SLOW_TRIPUP=?, SLOW_TRIPDOWN=? WHERE TRIP_NO=?; ");
                stmt.setString(1, trainSet2[m].getText());
                stmt.setString(2,ANAG1d[m].getText());
                stmt.setString(3,D1label[m].getText());
                stmt.setString(4, AAME2a[m].getText());
                stmt.setString(5,D2label[m].getText());
                stmt.setString(6, Rema1[m].getText());
                stmt.setString(7, countDelay[m].getText());
                stmt.setString(8, TTS1[m].getText());
                stmt.setString(9, TTA1[m].getText());
                stmt.setString(10, trainSet[m].getText());
                stmt.setString(11, AAME2d[m].getText());
                stmt.setString(12, D3label[m].getText());
                stmt.setString(13, ANAG2a[m].getText());
                stmt.setString(14, D4label[m].getText());
                stmt.setString(15, Rema2[m].getText());
                stmt.setString(16, countDelay2[m].getText());
```

```
                    stmt.setString(17, TTS2[m].getText());
                    stmt.setString(18, TTA2[m].getText());
                    stmt.setString(19, UPslo[m].getText() );
                    stmt.setString(20, DOWNslo[m].getText());
                    stmt.setString(21, tripN[m].getText());
                    stmt.executeQuery();
                    stmt.close();

                    }
                    catch (SQLException e1) {
                            e1.printStackTrace();

                    }
                    catch(Exception p1)
                    {
                            p1.printStackTrace();
                    }
                    break;
                }
            }
        }
    });
}
```

//creating 4 panels (containing the components) which is put on paneroot and is put on a loop to create duplicate paneroot(depending on number of trips) which is all put on panel11

//and the final panel11 is finally put on jscroll

//and finally jscroll is added to frame which can be displayed

```
JPanel panel11=new JPanel();
```

```
import java.sql.SQLException;

import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.xssf.usermodel.XSSFCell;
import org.apache.poi.xssf.usermodel.XSSFRow;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class testdatabase {
        private static Connection connect() {

                Connection c = null;

            try {
              Class.forName("org.postgresql.Driver");
              c = DriverManager
              .getConnection("jdbc:postgresql://localhost:5432/mydb","aryan", "eragon");
              System.out.println("Opened database successfully");
            }
              catch ( Exception e ) {
                      System.err.println( e.getClass().getName()+": "+ e.getMessage() );
                      System.exit(0);
              }
            return c;
        }
          public static void main( String args[] ) throws IOException, SQLException {

//using apache api to extract information from the the xlsm file format
                      File workbookFile = new
File("/home/aryan/Downloads/Datasheet2.xlsm");
                      FileInputStream file = new FileInputStream(workbookFile);
                      System.out.println("found file");

                      XSSFWorkbook workbook = new XSSFWorkbook(file);
                      System.out.println("in workbook");
                      XSSFSheet sheet = workbook.getSheet("Data");
                      System.out.println("got sheet");
        String sql=" INSERT INTO SCHEDULE5(trip_no,train_set,PNAG1d,PAME2a ,"
+
                      "train_set2,PAME2d ,PNAG2a) VALUES(?,?,?,?,?,?,?); ";
                      PreparedStatement statement = connect().prepareStatement(sql);
                      DataFormatter formatter = new DataFormatter();
```

```
                    XSSFRow row = null;

                    XSSFCell[] cell;
                    cell=new XSSFCell[7];
//trips here refer to the number trips you store in the database which is to be shown on
GUI
//starting_of_trips is index of trip no. from the datasheet where the starting trips can be
//  initialised from any point in the datasheet
                   int starting_of_trips=0;
                    int trips=0;
                   while(trips<36)
                   {
                      row = sheet.getRow(starting_of_trips+6);
                     cell[0] =row.getCell(1);
                     cell[1] =row.getCell(3);
                               cell[2] =row.getCell(4);
                               cell[3] =row.getCell(7);
                     //        cell[4] =row.getCell(15);   (NOT NEEDED ,AS CELL[4] IS
SAME AS CELL[1] or train_set=train_set2  (in the starting of planning stages)
                               cell[5] =row.getCell(16);
                               cell[6] =row.getCell(19);



                     statement.setString(1,formatter.formatCellValue(cell[0]));
                     statement.setString(2,formatter.formatCellValue(cell[1]));
                     statement.setString(3,formatter.formatCellValue(cell[2]));
                     statement.setString(4,formatter.formatCellValue(cell[3]));
                     statement.setString(5,formatter.formatCellValue(cell[1]));
                     statement.setString(6,formatter.formatCellValue(cell[5]));
                     statement.setString(7,formatter.formatCellValue(cell[6]));
                     statement.executeUpdate();
                     starting_of_trips++;
                     trips++;
                   }


                  statement.close();
                  workbook.close();
                  System.out.println("Table created successfully");
        }
```