IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY

Name: N.A.R. Dilshan

Student Reference Number: 10707181

Module Code: SOFT255SL	Module Name: Software Engineering for the Internet using Java
Coursework Title: SOFT255SL T1 C1	
Deadline Date: Tuesday, 1 December 2020	Member of staff responsible for coursework: Ms. Sulari Fernando
Programme: BSc (Hons) Software Engineering	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

N.A.R. Dilshan - 10707181
G.T.G.L.D. Abedeera - 10707120
P. A. H. N. Mihiranga - 10707281
G.T.U. Ariyathilake - 10707133
B.E.R.R. Jayathilaka - 10707229

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group: Rakshitha

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***
Signed :

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I have not used translation software.

If used, please state name of software……………………………………………………………………

Overall mark _____% Assessors Initials ______ Date_________

SOFT255SL

Software Engineering for the Internet using Java

Coursework - T1 C1

Security Logs System

Group No: 31

N.A.R. Dilshan
G.T.G.L.D. Abedeera
P. A. H. N. Mihiranga
G.T.U. Ariyathilake
B.E.R.R. Jayathilaka

Table of Content

Introduction

I would like to point out some situations that any of us as students of NSBM University may face in our day today life at NSBM. And also how we deal with those situations.

First one is forgetting to bring your Student ID card. The normal procedure is that the security personnel at the entrance checks your Student ID and lets you in. But if you forgot to bring it, they ask you to write your details in the logbook at the entrance. It could be difficult as well as time wasting task for everyone especially when it is a busy morning at the entrance.

Another situation is you lost something valuable inside the university premises. What we normally do is put a message about the lost item informing others in our WhatsApp chat groups. But the security personnel are the ones who could look out for the lost item and most probably the ones who may find it. But they are not members of your chat group. So they don't know to whom it belongs to when they found something.

Some of us travel to the university by our own vehicles. But we have no proper parking facility and we have to park our vehicles roadside beside the university without proper security.

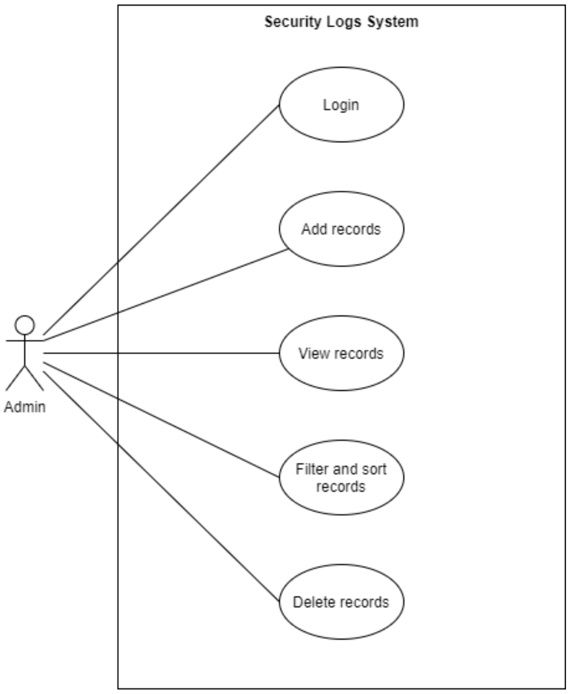
Considering situations like that we came up with a better solution to replace the current system. It is to have a security logs program to keep records of some situations like above and also to manage some processes in a better way. Both the security staff and the students could be benefited by this solution. As an example, if the university could arrange a proper space to park the vehicles and assign a security guard to the place, they can easily record the necessary details and manage the car park properly using this program. It is a great relief to the vehicle owners and also the university can charge a parking fee for their trouble since the program records all the necessary details of the vehicles in and out. So the main objective of our Security Logs Program is to manage some day today processes at university in a better way than they are right now and by that, to increase the productivity.

Analysis

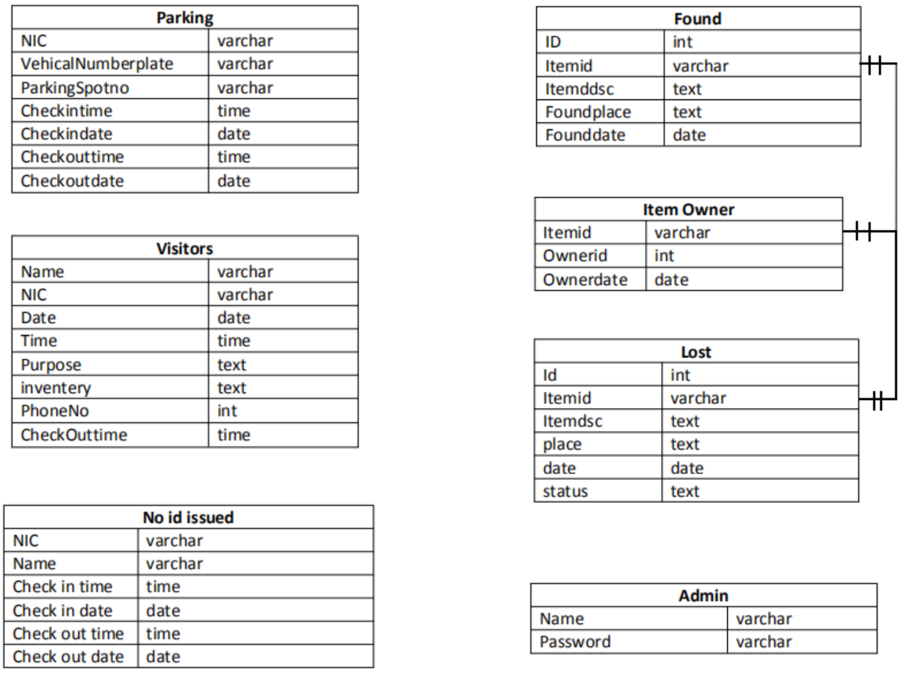
Requirements

- Functional Requirements
  1. Ability to add records of visitors, parking, lost and found items and people who doesn’t have their University ID.
  2. Ability to view all the records in a detailed form.
  3. Ability to sort and filter records.
  4. Ability to delete mistaken records.
  5. Ability to keep track of time of all entries.
  6. Only the security personal with credentials should be able to login to the system.
- Non-functional Requirements
  1. Security
  2. Simple and Functional GUI
  3. Stability of the software

Use Case Diagram



ER Diagram



Parking	
NIC	varchar
VehicalNumberplate	varchar
ParkingSpotno	varchar
Checkintime	time
Checkindate	date
Checkouttime	time
Checkoutdate	date

Visitors	
Name	varchar
NIC	varchar
Date	date
Time	time
Purpose	text
inventery	text
PhoneNo	int
CheckOuttime	time

No id issued	
NIC	varchar
Name	varchar
Check in time	time
Check in date	date
Check out time	time
Check out date	date

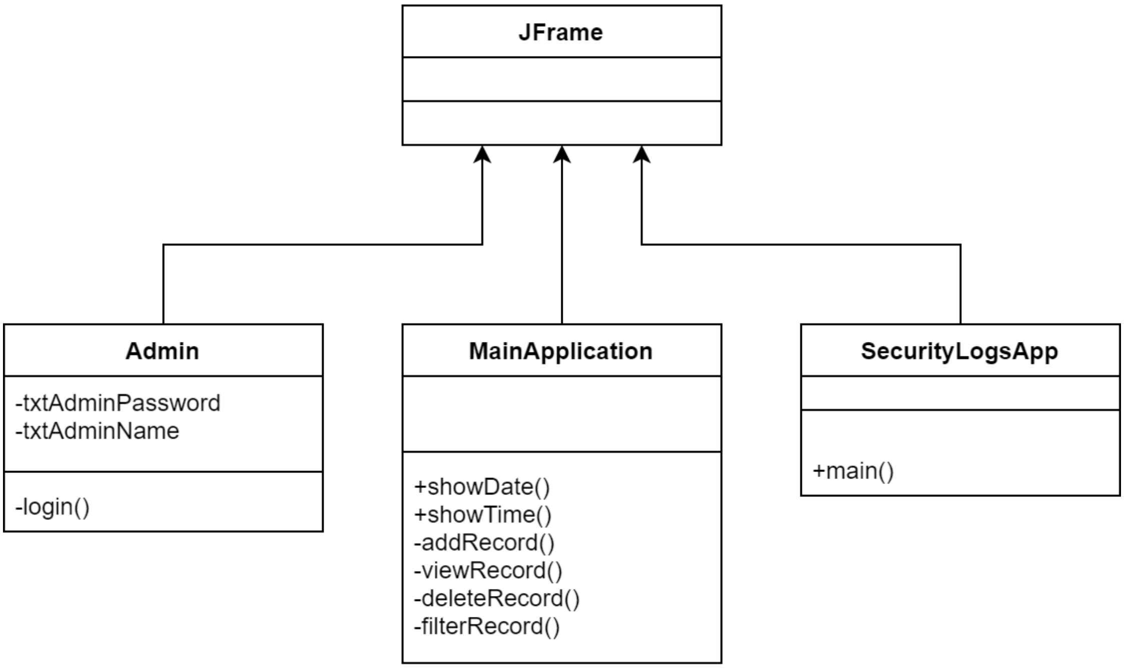
Found	
ID	int
Itemid	varchar
Itemddsc	text
Foundplace	text
Founddate	date

Item Owner	
Itemid	varchar
Ownerid	int
Ownerdate	date

Lost	
Id	int
Itemid	varchar
Itemdsc	text
place	text
date	date
status	text

Admin	
Name	varchar
Password	varchar

Class Diagram

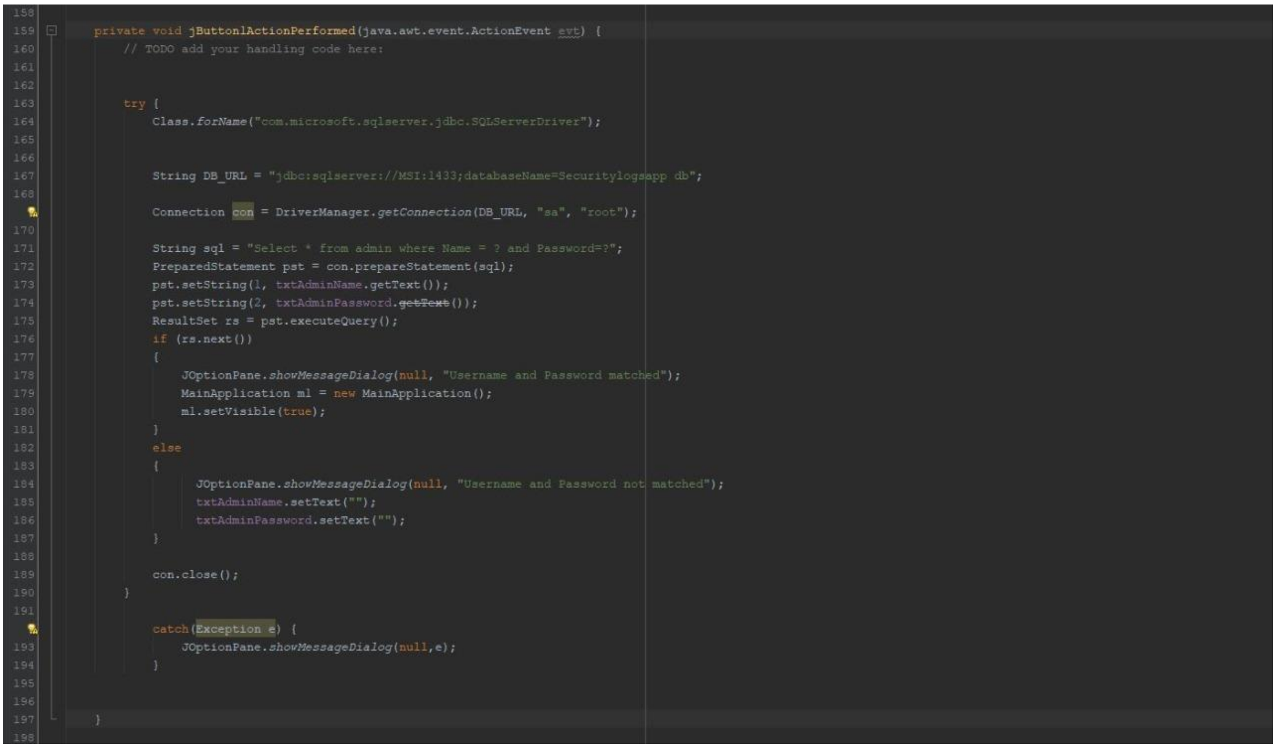


Implementation

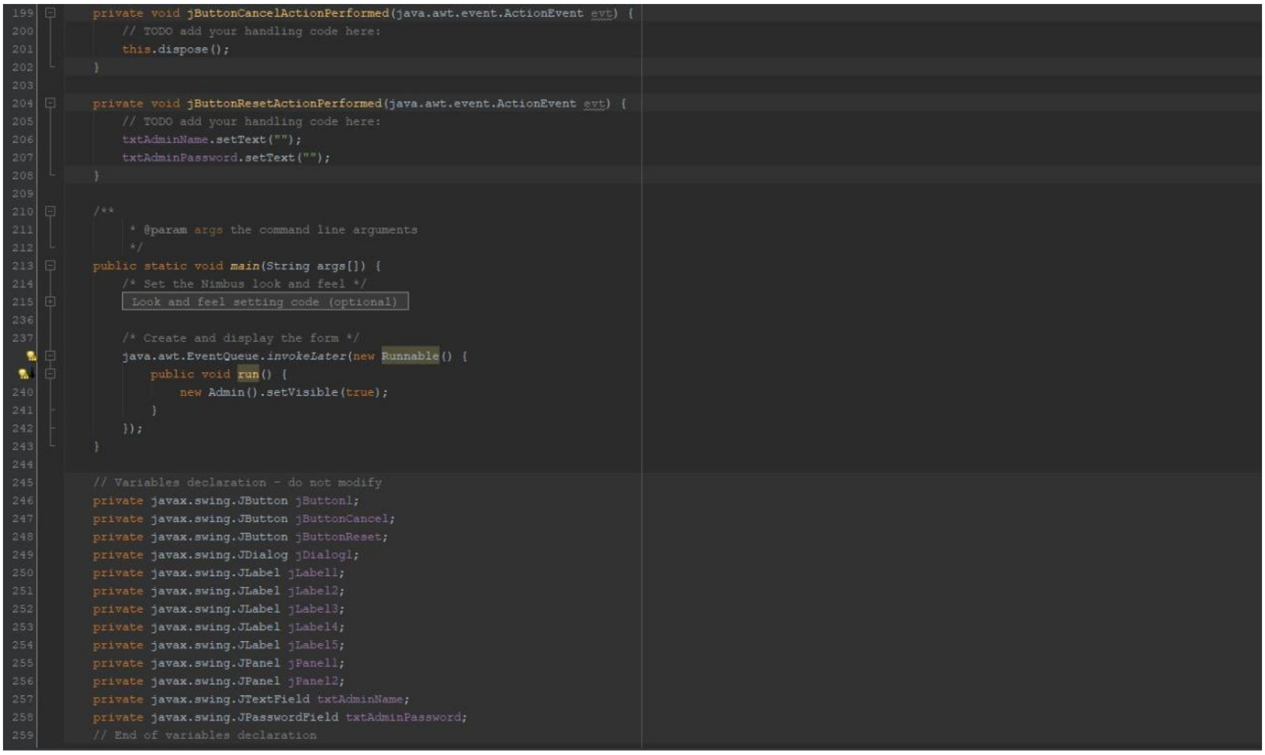
Code

In here we have used coding concepts like encapsulation, inheritance, exception handling to build our program. We have used a tabbed pane and buttons linked to tabs for easy transition between the sections of the software.

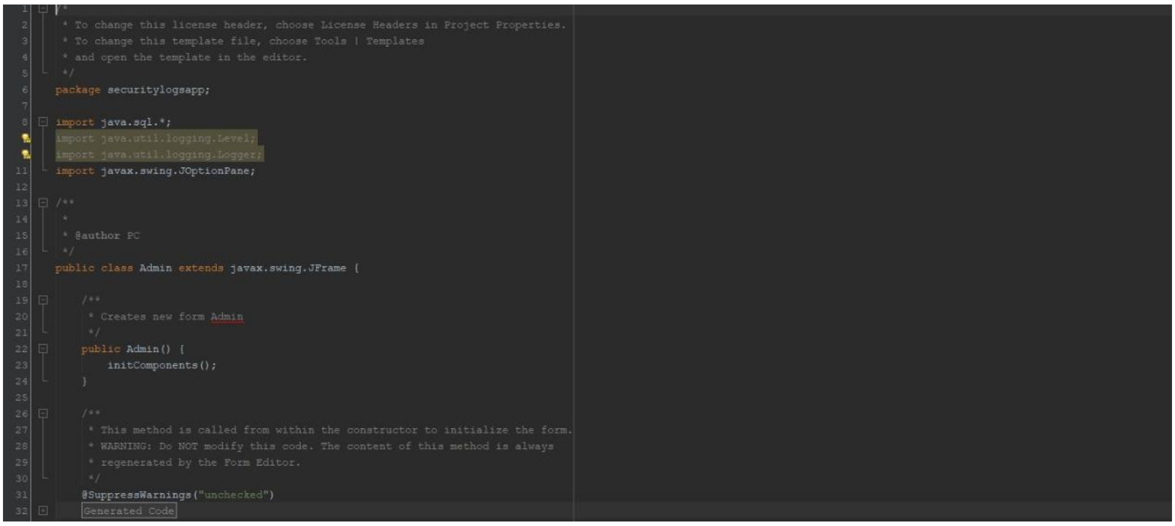
Admin



```java

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:


        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogsapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");

            String sql = "Select * from admin where Name = ? and Password=?";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setString(1, txtAdminName.getText());
            pst.setString(2, txtAdminPassword.getText());
            ResultSet rs = pst.executeQuery();
            if (rs.next())
            {
                JOptionPane.showMessageDialog(null, "Username and Password matched");
                MainApplication ml = new MainApplication();
                ml.setVisible(true);
            }
            else
            {
                 JOptionPane.showMessageDialog(null, "Username and Password not matched");
                 txtAdminName.setText("");
                 txtAdminPassword.setText("");
            }

            con.close();
        }

            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }


    }

```

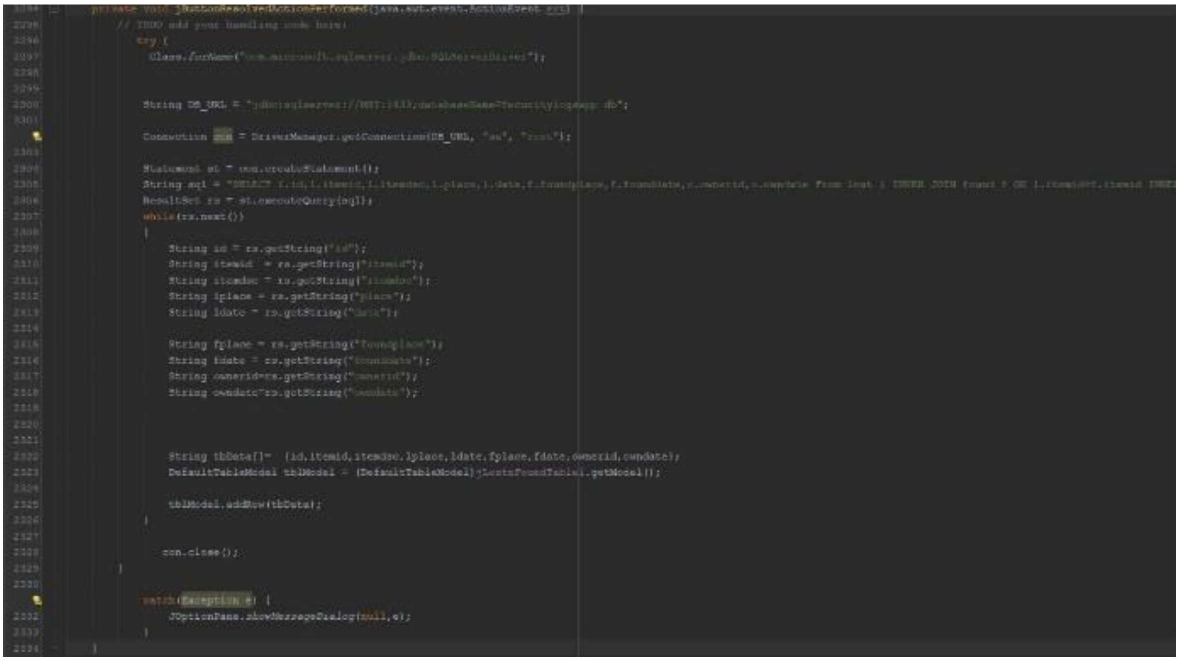


```java
    private void jButtonCancelActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        this.dispose();
    }

    private void jButtonResetActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        txtAdminName.setText("");
        txtAdminPassword.setText("");
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButtonCancel;
    private javax.swing.JButton jButtonReset;
    private javax.swing.JDialog jDialog1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JTextField txtAdminName;
    private javax.swing.JPasswordField txtAdminPassword;
    // End of variables declaration
```

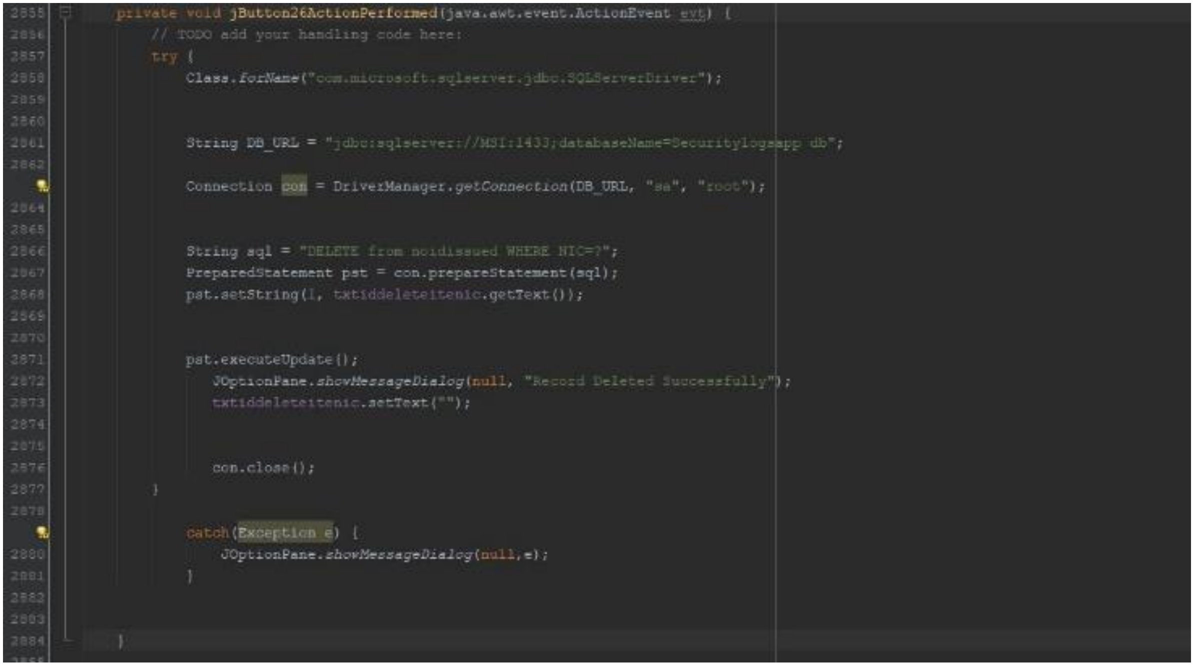


```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package securitylogsapp;

import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author PC
 */
public class Admin extends javax.swing.JFrame {

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

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code
```

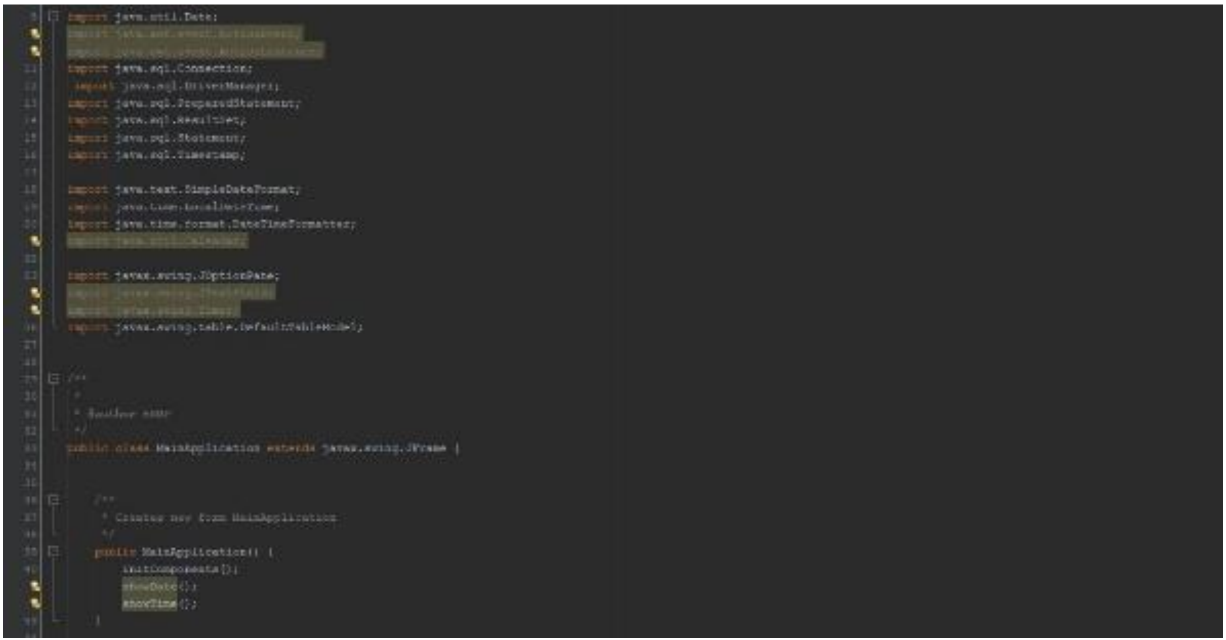
Main Application



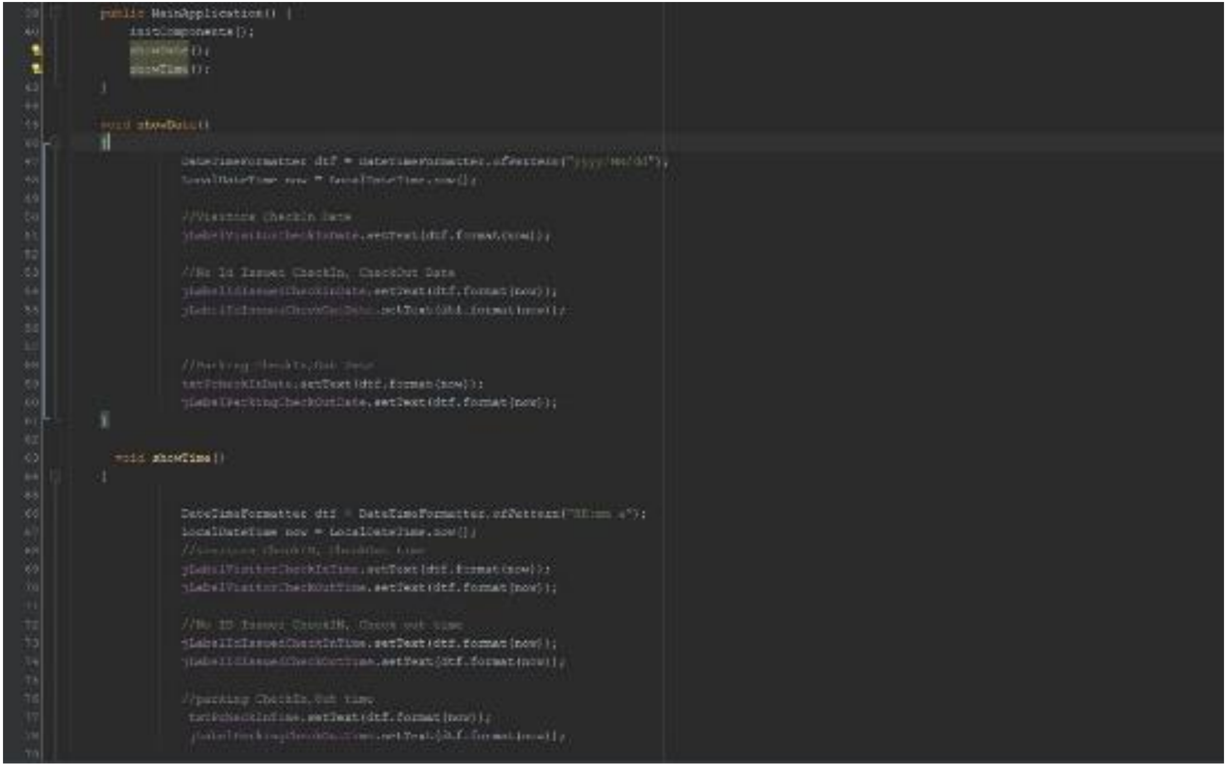
```
    private void jButtonResolvedActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
          try {
           Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogsapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");

            Statement st = con.createStatement();
            String sql = "SELECT l.id,l.itemid,l.itemdsc,l.place,l.date,f.foundplace,f.founddate,o.ownerid,o.owndate From lost l INNER JOIN found f ON l.itemid=f.itemid INNER
            ResultSet rs = st.executeQuery(sql);
            while(rs.next())
            {
                String id = rs.getString("id");
                String itemid  = rs.getString("itemid");
                String itemdsc = rs.getString("itemdsc");
                String lplace = rs.getString("place");
                String ldate = rs.getString("date");

                String fplace = rs.getString("foundplace");
                String fdate = rs.getString("founddate");
                String ownerid=rs.getString("ownerid");
                String owndate=rs.getString("owndate");



                String tbData[]=  {id,itemid,itemdsc,lplace,ldate,fplace,fdate,ownerid,owndate};
                DefaultTableModel tblModel = (DefaultTableModel)jLostnFoundTable1.getModel();

                tblModel.addRow(tbData);
            }

              con.close();
        }

            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }
    }
```



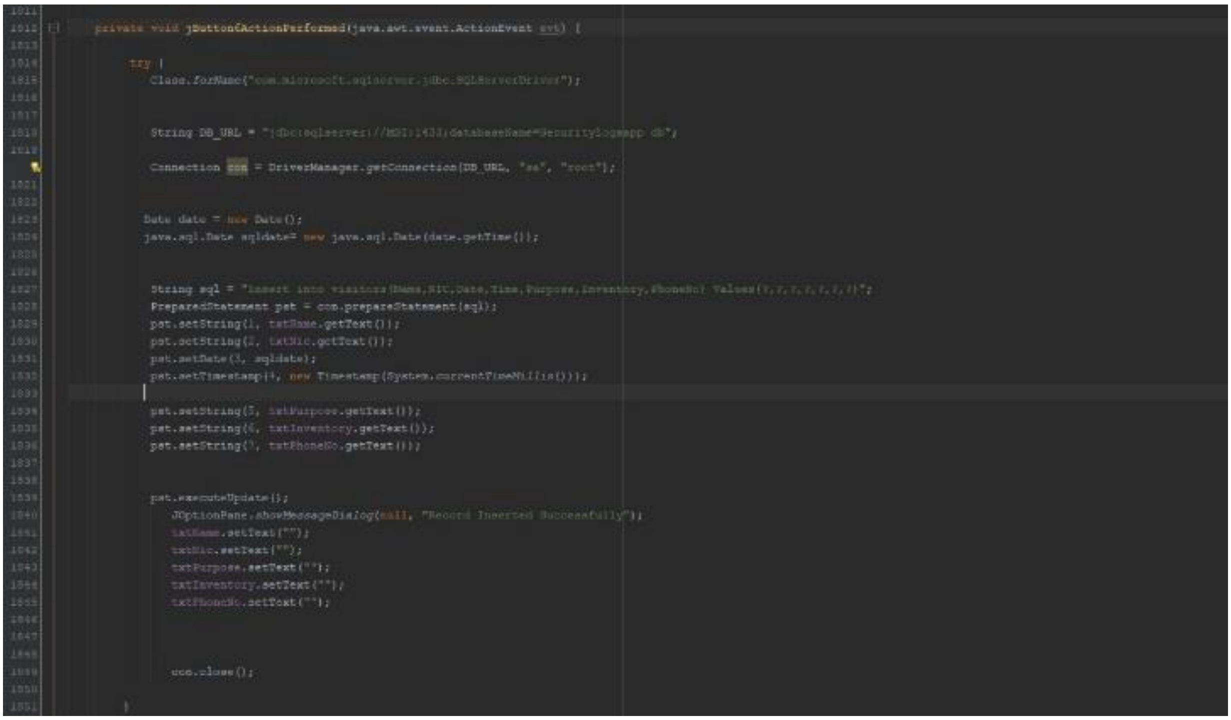
```
    private void jButton26ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogsapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");


            String sql = "DELETE from noidissued WHERE NIC=?";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setString(1, txtiddeleteitenic.getText());


            pst.executeUpdate();
               JOptionPane.showMessageDialog(null, "Record Deleted Successfully");
               txtiddeleteitenic.setText("");


               con.close();
        }

            catch(Exception e) {
             JOptionPane.showMessageDialog(null,e);
            }


    }
```



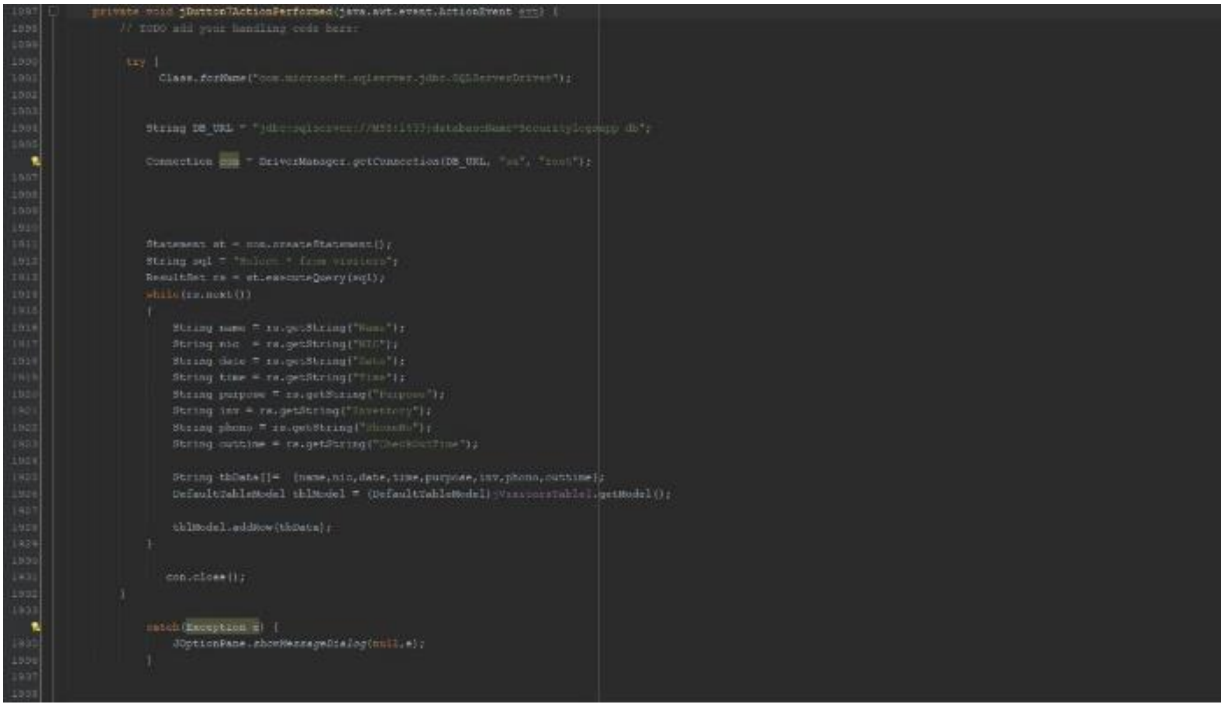
```java
import java.util.Date;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
 import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import java.sql.Timestamp;

import java.text.SimpleDateFormat;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.Calendar;

import javax.swing.JOptionPane;
import javax.swing.[illegible];
import javax.swing.Timer;
import javax.swing.table.DefaultTableModel;


/**
 *
 * @author user
 */
public class MainApplication extends javax.swing.JFrame {


    /**
     * Creates new form MainApplication
     */
    public MainApplication() {
        initComponents();
        showDate();
        showTime();
    }
```



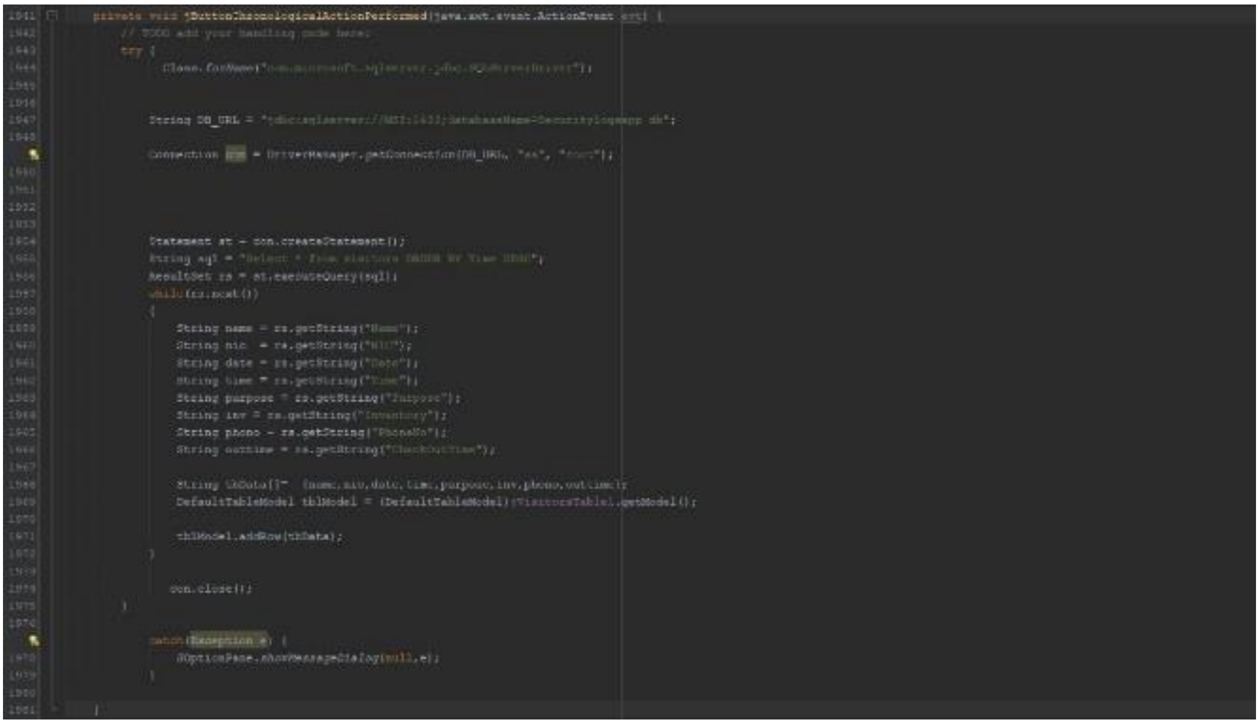
```java
    public MainApplication() {
        initComponents();
        showDate();
        showTime();
    }

    void showDate()
    {
                DateTimeFormatter dtf = DateTimeFormatter.ofPattern("yyyy/MM/dd");
                LocalDateTime now = LocalDateTime.now();

                //Visitors CheckIn Date
                jLabelVisitorCheckInDate.setText(dtf.format(now));

                //No Id Issued CheckIn, CheckOut Date
                jLabelIdIssuedCheckInDate.setText(dtf.format(now));
                jLabelIdIssuedCheckOutDate.setText(dtf.format(now));


                //Parking CheckIn,Out Date
                txtPcheckInDate.setText(dtf.format(now));
                jLabelParkingCheckOutDate.setText(dtf.format(now));
    }

     void showTime()
    {

                DateTimeFormatter dtf = DateTimeFormatter.ofPattern("hh:mm a");
                LocalDateTime now = LocalDateTime.now();
                //Visitors CheckIN, CheckOut time
                jLabelVisitorCheckInTime.setText(dtf.format(now));
                jLabelVisitorCheckOutTime.setText(dtf.format(now));

                //No ID Issued CheckIN, Check out time
                jLabelIdIssuedCheckInTime.setText(dtf.format(now));
                jLabelIdIssuedCheckOutTime.setText(dtf.format(now));

                //parking CheckIn,Out time
                 txtPcheckInTime.setText(dtf.format(now));
                 jLabelParkingCheckOutTime.setText(dtf.format(now));
```



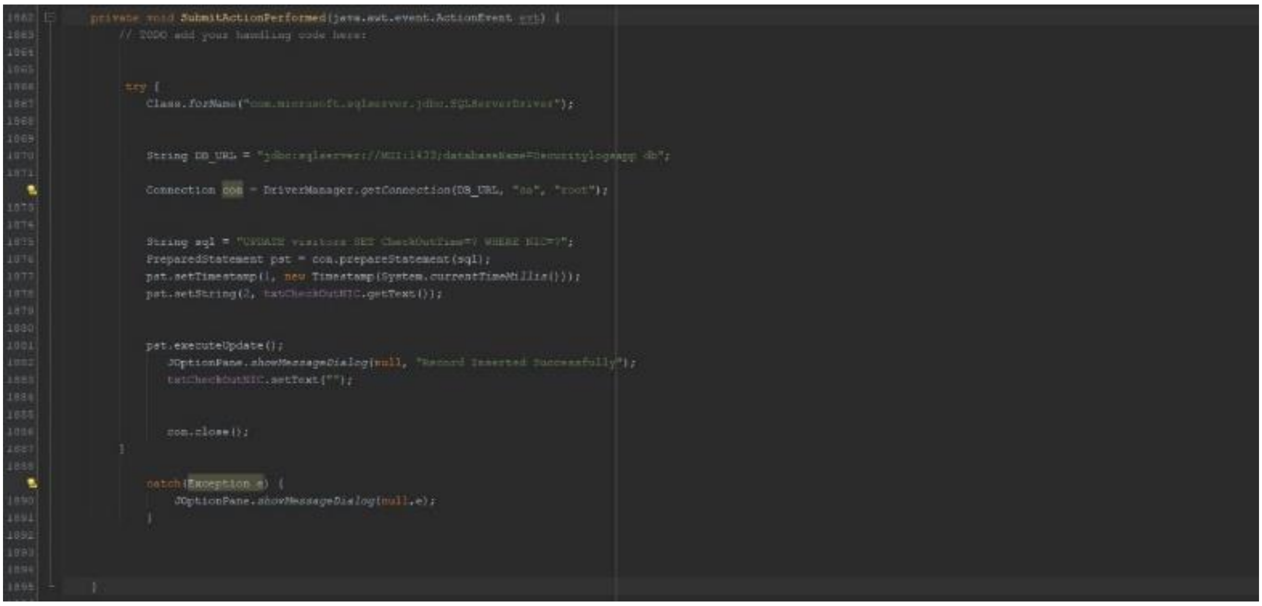
```
    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

        try {
          Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


          String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

          Connection con = DriverManager.getConnection(DB_URL, "sa", "root");


         Date date = new Date();
         java.sql.Date sqldate= new java.sql.Date(date.getTime());


          String sql = "Insert into visitors(Name,NIC,Date,Time,Purpose,Inventory,PhoneNo) Values(?,?,?,?,?,?,?)";
          PreparedStatement pst = con.prepareStatement(sql);
          pst.setString(1, txtName.getText());
          pst.setString(2, txtNic.getText());
          pst.setDate(3, sqldate);
          pst.setTimestamp(4, new Timestamp(System.currentTimeMillis()));

          pst.setString(5, txtPurpose.getText());
          pst.setString(6, txtInventory.getText());
          pst.setString(7, txtPhoneNo.getText());


          pst.executeUpdate();
             JOptionPane.showMessageDialog(null, "Record Inserted Successfully");
             txtName.setText("");
             txtNic.setText("");
             txtPurpose.setText("");
             txtInventory.setText("");
             txtPhoneNo.setText("");



             con.close();

        }
```



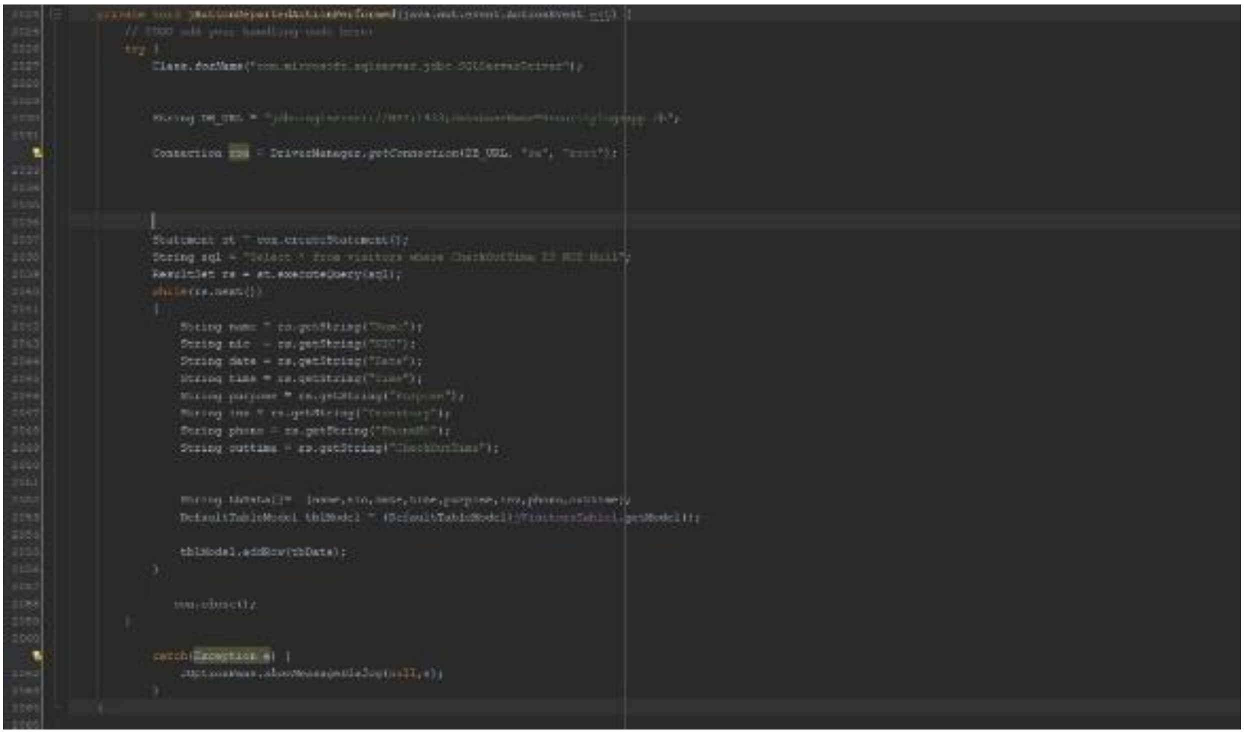
```
    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


          String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

          Connection con = DriverManager.getConnection(DB_URL, "sa", "root");




          Statement st = con.createStatement();
          String sql = "Select * from visitors";
          ResultSet rs = st.executeQuery(sql);
          while(rs.next())
          {
              String name = rs.getString("Name");
              String nic  = rs.getString("NIC");
              String date = rs.getString("Date");
              String time = rs.getString("Time");
              String purpose = rs.getString("Purpose");
              String inv = rs.getString("Inventory");
              String phone = rs.getString("PhoneNo");
              String outtime = rs.getString("CheckOutTime");

              String tbData[]=  {name,nic,date,time,purpose,inv,phone,outtime};
              DefaultTableModel tblModel = (DefaultTableModel)VisitorsTable1.getModel();

              tblModel.addRow(tbData);
          }

            con.close();
        }

          catch(Exception e) {
              JOptionPane.showMessageDialog(null,e);
          }

```



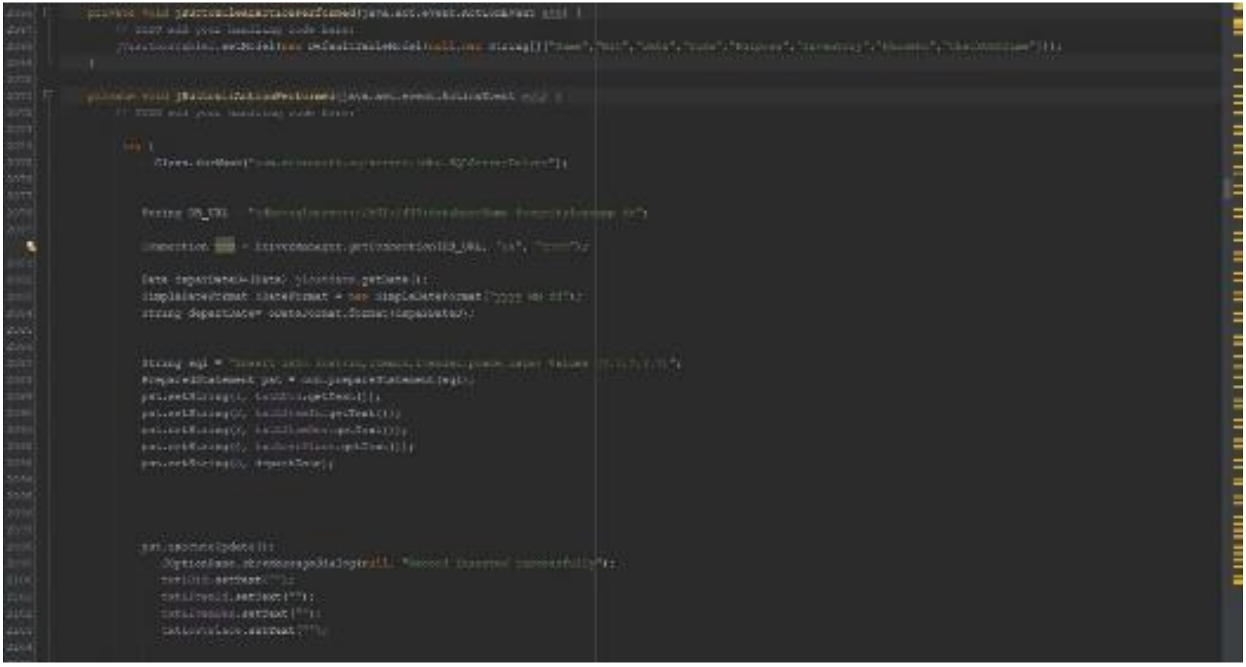
```java
    private void jButtonChronologicalActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
              Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogsapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");




            Statement st = con.createStatement();
            String sql = "Select * from visitors ORDER BY Time DESC";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next())
            {
                String name = rs.getString("Name");
                String nic  = rs.getString("NIC");
                String date = rs.getString("Date");
                String time = rs.getString("Time");
                String purpose = rs.getString("Purpose");
                String inv = rs.getString("Inventory");
                String phone = rs.getString("PhoneNo");
                String outtime = rs.getString("CheckOutTime");

                String tbData[]= {name,nic,date,time,purpose,inv,phone,outtime};
                DefaultTableModel tblModel = (DefaultTableModel)VisitorsTable1.getModel();

                tblModel.addRow(tbData);
            }

              con.close();
        }

            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }

    }
```



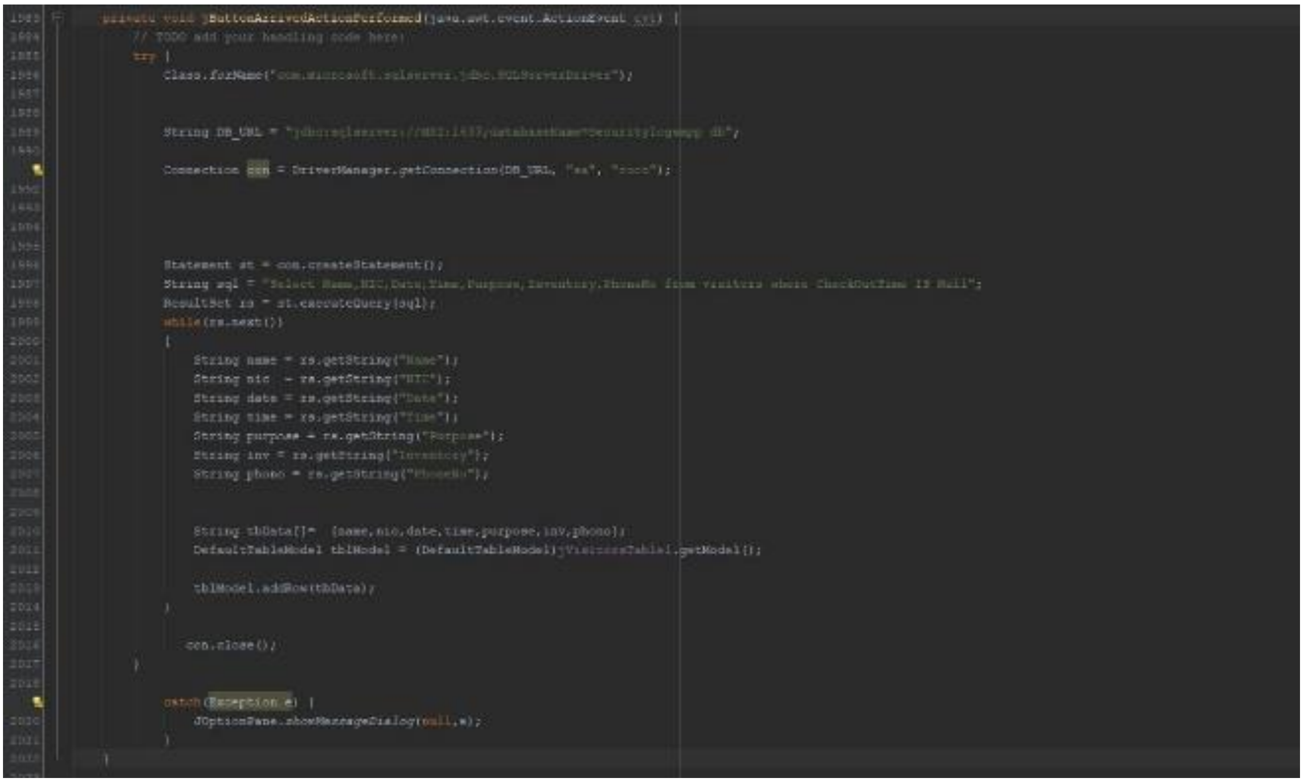
```java
    private void SubmitActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:


         try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogsapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");


            String sql = "UPDATE visitors SET CheckOutTime=? WHERE NIC=?";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setTimestamp(1, new Timestamp(System.currentTimeMillis()));
            pst.setString(2, txtCheckOutNIC.getText());


            pst.executeUpdate();
               JOptionPane.showMessageDialog(null, "Record Inserted Successfully");
               txtCheckOutNIC.setText("");


               con.close();
        }

            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }


    }
```



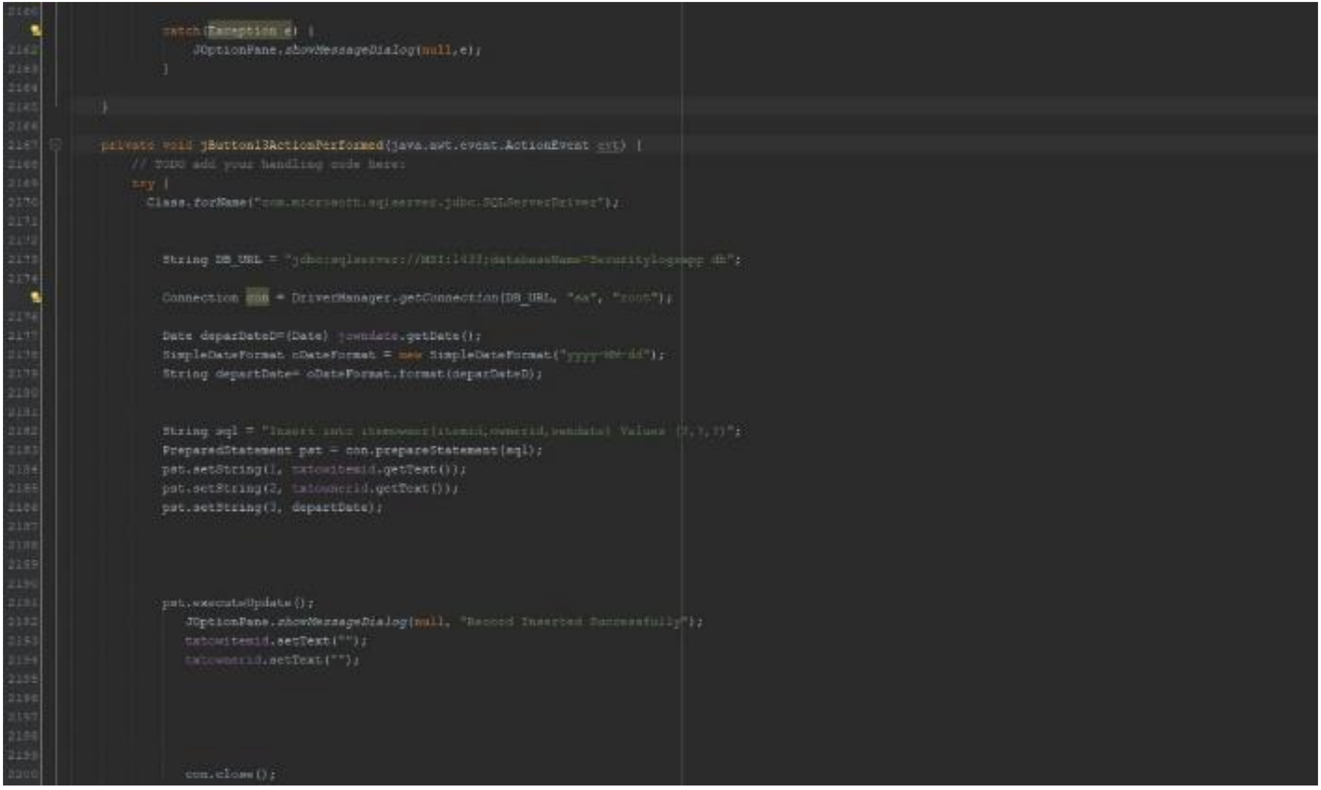
```
private void jButtonDepartedActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


        String DB_URL = "jdbc:sqlserver://[illegible]:1433;databaseName=[illegible]";

        Connection con = DriverManager.getConnection(DB_URL, "sa", "[illegible]");



        |
        Statement st = con.createStatement();
        String sql = "Select * from visitors where CheckOutTime IS NOT Null";
        ResultSet rs = st.executeQuery(sql);
        while(rs.next())
        {
            String name = rs.getString("Name");
            String nic = rs.getString("NIC");
            String date = rs.getString("Date");
            String time = rs.getString("time");
            String purpose = rs.getString("Purpose");
            String ins = rs.getString("Inventory");
            String phone = rs.getString("PhoneNo");
            String outtime = rs.getString("CheckOutTime");


            String tbData[]= {name,nic,date,time,purpose,ins,phone,outtime};
            DefaultTableModel tblModel = (DefaultTableModel)VisitorsTable1.getModel();

            tblModel.addRow(tbData);
        }

          con.close();
    }

        catch(Exception e) {
            JOptionPane.showMessageDialog(null,e);
        }
}
```



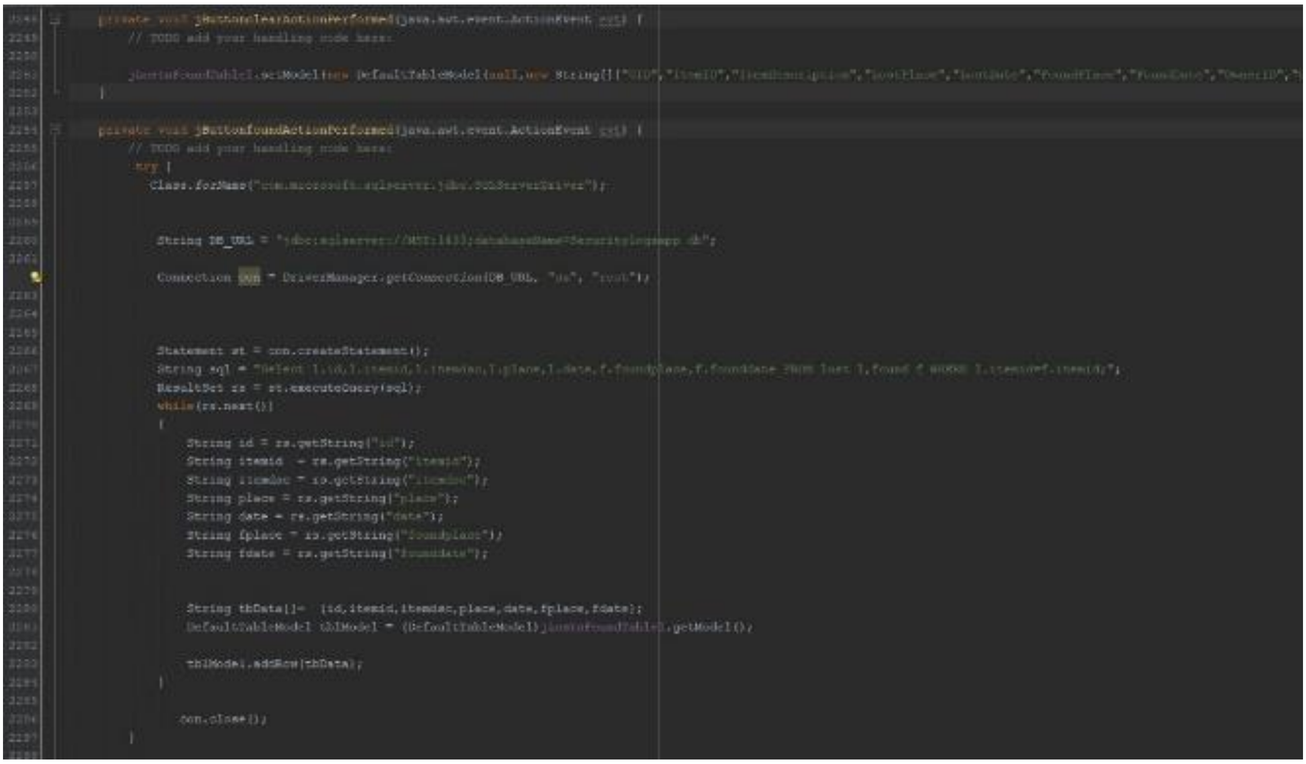
```
private void jButtonClearActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    //VisitorsTable1.setModel(new DefaultTableModel(null,new String[]{"Name","NIC","Date","Time","Purpose","Inventory","PhoneNo","CheckOutTime"}));
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

     try {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


      String DB_URL = "jdbc:sqlserver://[illegible]:1433;databaseName=[illegible]";

      Connection con = DriverManager.getConnection(DB_URL, "sa", "[illegible]");

      Date departDate=(Date) jDateChooser.getDate();
      SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy MM dd");
      String departDate= dateFormat.format(departDate);


      String sql = "insert into [illegible] values (?,?,?,?,?)";
      PreparedStatement pst = con.prepareStatement(sql);
      pst.setString(1, [illegible].getText());
      pst.setString(2, [illegible].getText());
      pst.setString(3, [illegible].getText());
      pst.setString(4, [illegible].getText());
      pst.setString(5, departDate);




      pst.executeUpdate();
        JOptionPane.showMessageDialog(null, "Record Inserted Successfully");
        [illegible].setText("");
        [illegible].setText("");
        [illegible].setText("");
        [illegible].setText("");
```



```
    private void jButtonArrivedActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");




            Statement st = con.createStatement();
            String sql = "Select Name,NIC,Date,Time,Purpose,Inventory,PhoneNo from visitors where CheckOutTime IS Null";
            ResultSet rs = st.executeQuery(sql);
            while(rs.next())
            {
                String name = rs.getString("Name");
                String nic  = rs.getString("NIC");
                String date = rs.getString("Date");
                String time = rs.getString("Time");
                String purpose = rs.getString("Purpose");
                String inv = rs.getString("Inventory");
                String phono = rs.getString("PhoneNo");


                String tbData[]=  {name,nic,date,time,purpose,inv,phono};
                DefaultTableModel tblModel = (DefaultTableModel)jVisitorsTable1.getModel();

                tblModel.addRow(tbData);
            }

              con.close();
        }

            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }
    }
```

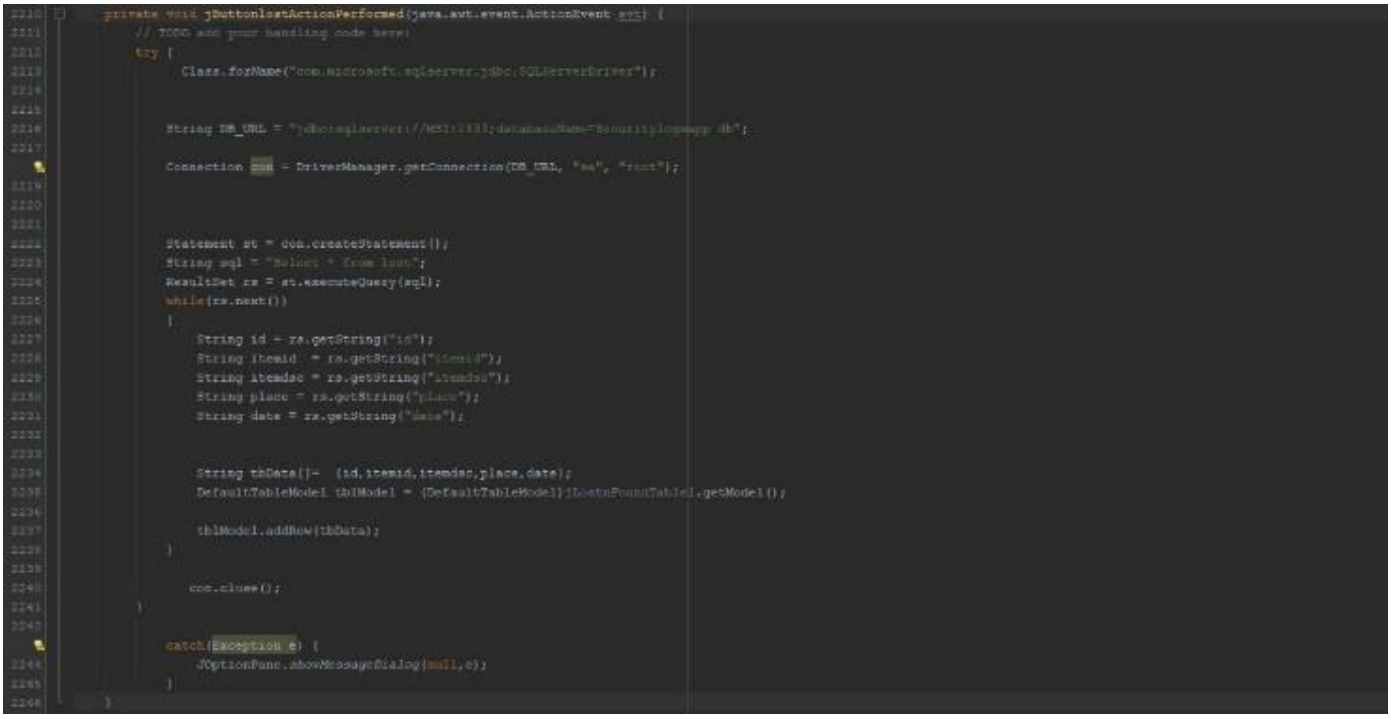


```
            catch(Exception e) {
                JOptionPane.showMessageDialog(null,e);
            }

    }

    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
          Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


            String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

            Connection con = DriverManager.getConnection(DB_URL, "sa", "root");

            Date deparDateD=(Date) jowndate.getDate();
            SimpleDateFormat oDateFormat = new SimpleDateFormat("yyyy-MM-dd");
            String departDate= oDateFormat.format(deparDateD);


            String sql = "Insert into itemowner(itemid,ownerid,owndate) Values (?,?,?)";
            PreparedStatement pst = con.prepareStatement(sql);
            pst.setString(1, txtowitemid.getText());
            pst.setString(2, txtownerid.getText());
            pst.setString(3, departDate);




            pst.executeUpdate();
               JOptionPane.showMessageDialog(null, "Record Inserted Successfully");
               txtowitemid.setText("");
               txtownerid.setText("");





               con.close();
```



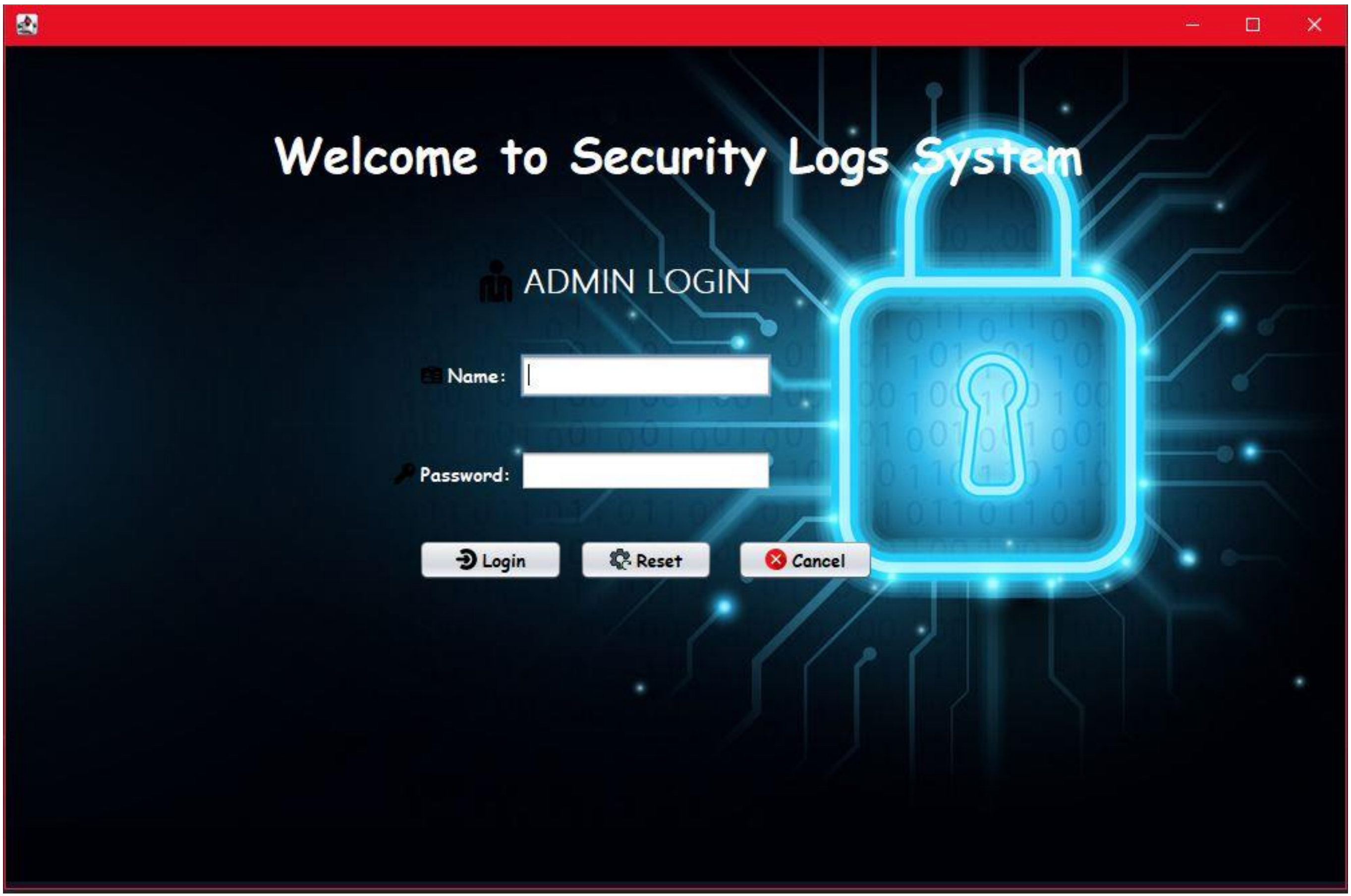
```
    private void jButtonclearActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        jLostnFoundTable1.setModel(new DefaultTableModel(null,new String[]{"UID","itemID","itemDescription","lostPlace","lostDate","FoundPlace","FoundDate","OwnerID","
    }

    private void jButtonfoundActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
         try {
          Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


           String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

           Connection con = DriverManager.getConnection(DB_URL, "sa", "root");



           Statement st = con.createStatement();
           String sql = "Select l.id,l.itemid,l.itemdsc,l.place,l.date,f.foundplace,f.founddate FROM lost l,found f WHERE l.itemid=f.itemid;";
           ResultSet rs = st.executeQuery(sql);
           while(rs.next())
           {
               String id = rs.getString("id");
               String itemid  = rs.getString("itemid");
               String itemdsc = rs.getString("itemdsc");
               String place = rs.getString("place");
               String date = rs.getString("date");
               String fplace = rs.getString("foundplace");
               String fdate = rs.getString("founddate");


               String tbData[]=  {id,itemid,itemdsc,place,date,fplace,fdate};
               DefaultTableModel tblModel = (DefaultTableModel)jLostnFoundTable1.getModel();

               tblModel.addRow(tbData);
           }

              con.close();
         }
```

Main Method



```
    private void jButtonlostActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");


          String DB_URL = "jdbc:sqlserver://MSI:1433;databaseName=Securitylogapp db";

          Connection con = DriverManager.getConnection(DB_URL, "sa", "root");



          Statement st = con.createStatement();
          String sql = "Select * from lost";
          ResultSet rs = st.executeQuery(sql);
          while(rs.next())
          {
              String id = rs.getString("id");
              String itemid  = rs.getString("itemid");
              String itemdsc = rs.getString("itemdsc");
              String place = rs.getString("place");
              String date = rs.getString("date");


              String tbData[]=  {id,itemid,itemdsc,place,date};
              DefaultTableModel tblModel = (DefaultTableModel)jLostnFoundTable1.getModel();

              tblModel.addRow(tbData);
          }

             con.close();
        }

          catch(Exception e) {
              JOptionPane.showMessageDialog(null,e);
          }
    }
```

GUI

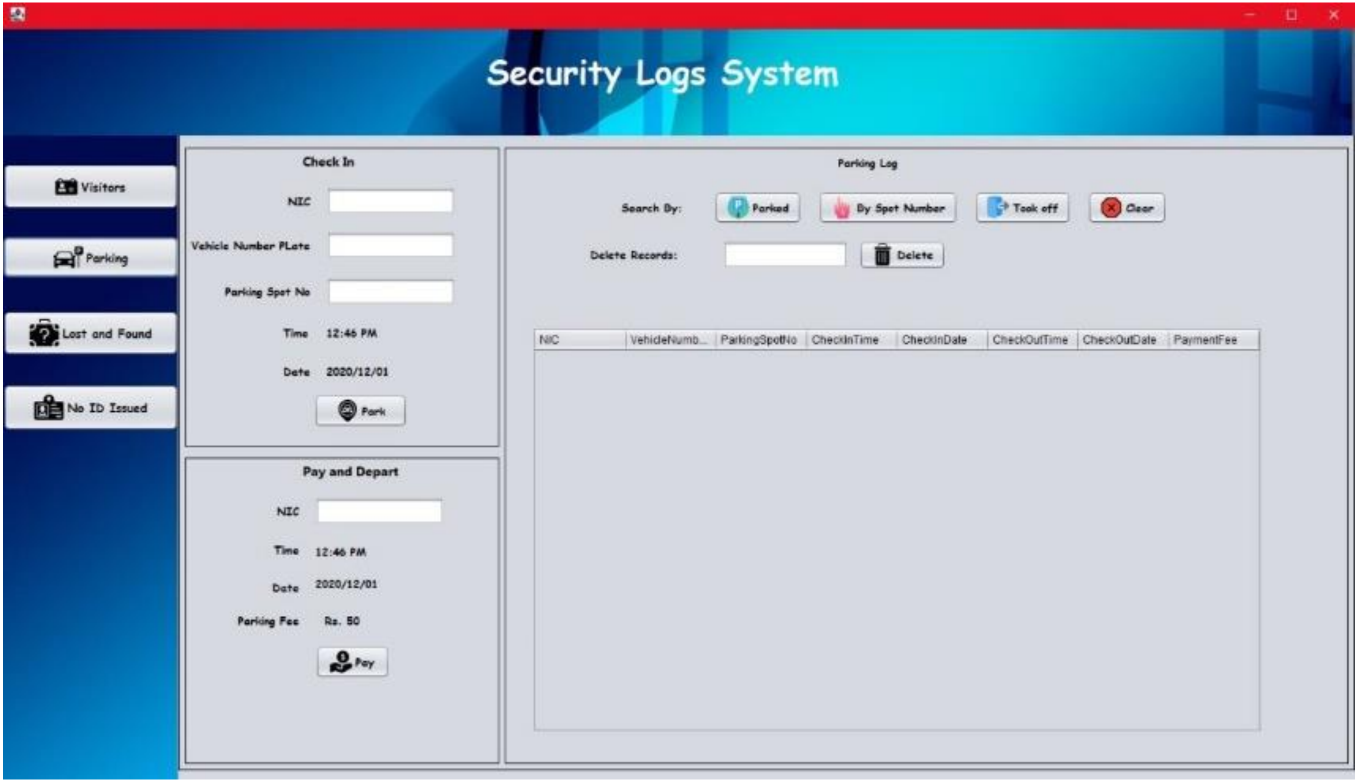


Welcome to Security Logs System

ADMIN LOGIN

Name:

Password:



Security Logs System

Check In

NIC

Vehicle Number PLate

Parking Spot No

Time 12:46 PM

Date 2020/12/01

Pay and Depart

NIC

Time 12:46 PM

Date 2020/12/01

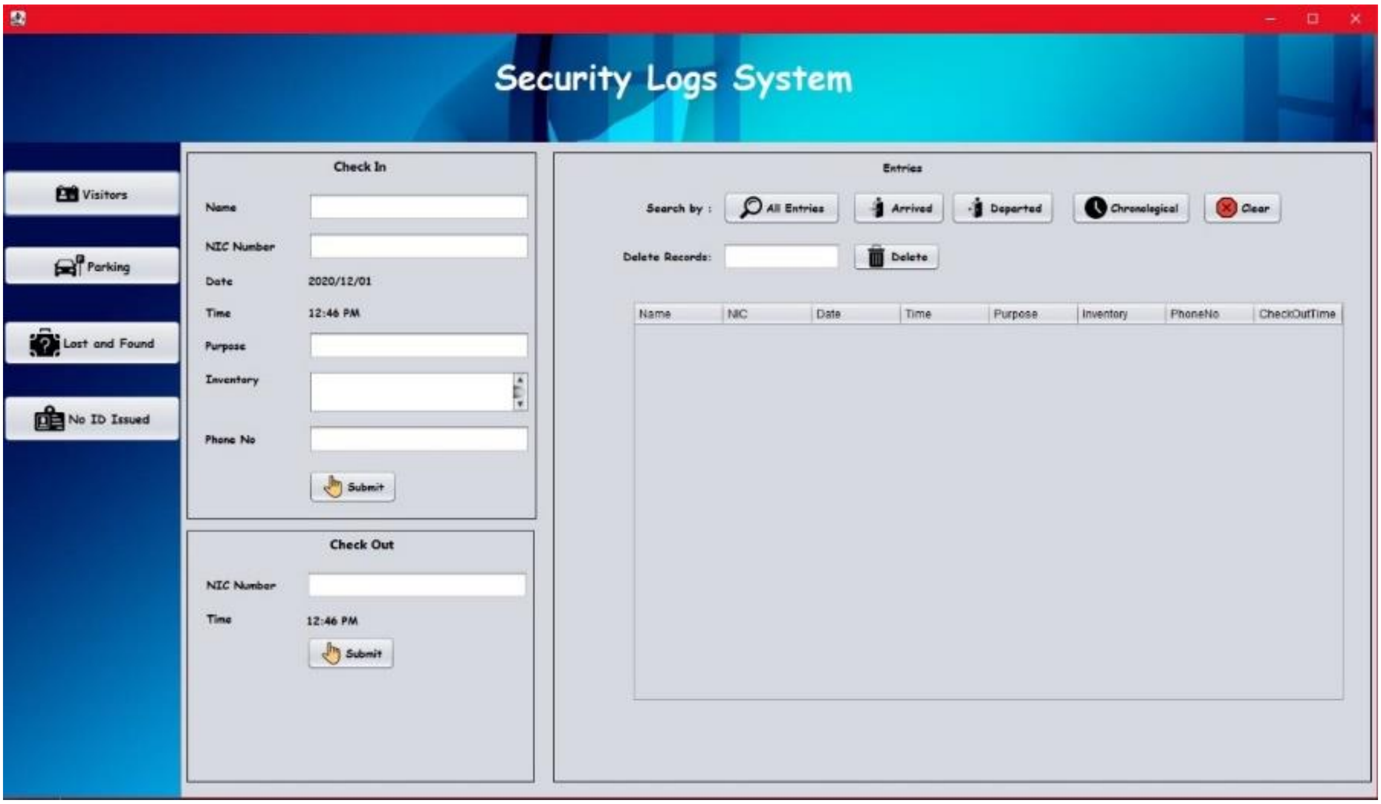
Parking Fee Rs. 50

Parking Log

Search By:

Delete Records:

NIC	VehicleNumb...	ParkingSpotNo	CheckInTime	CheckInDate	CheckOutTime	CheckOutDate	PaymentFee



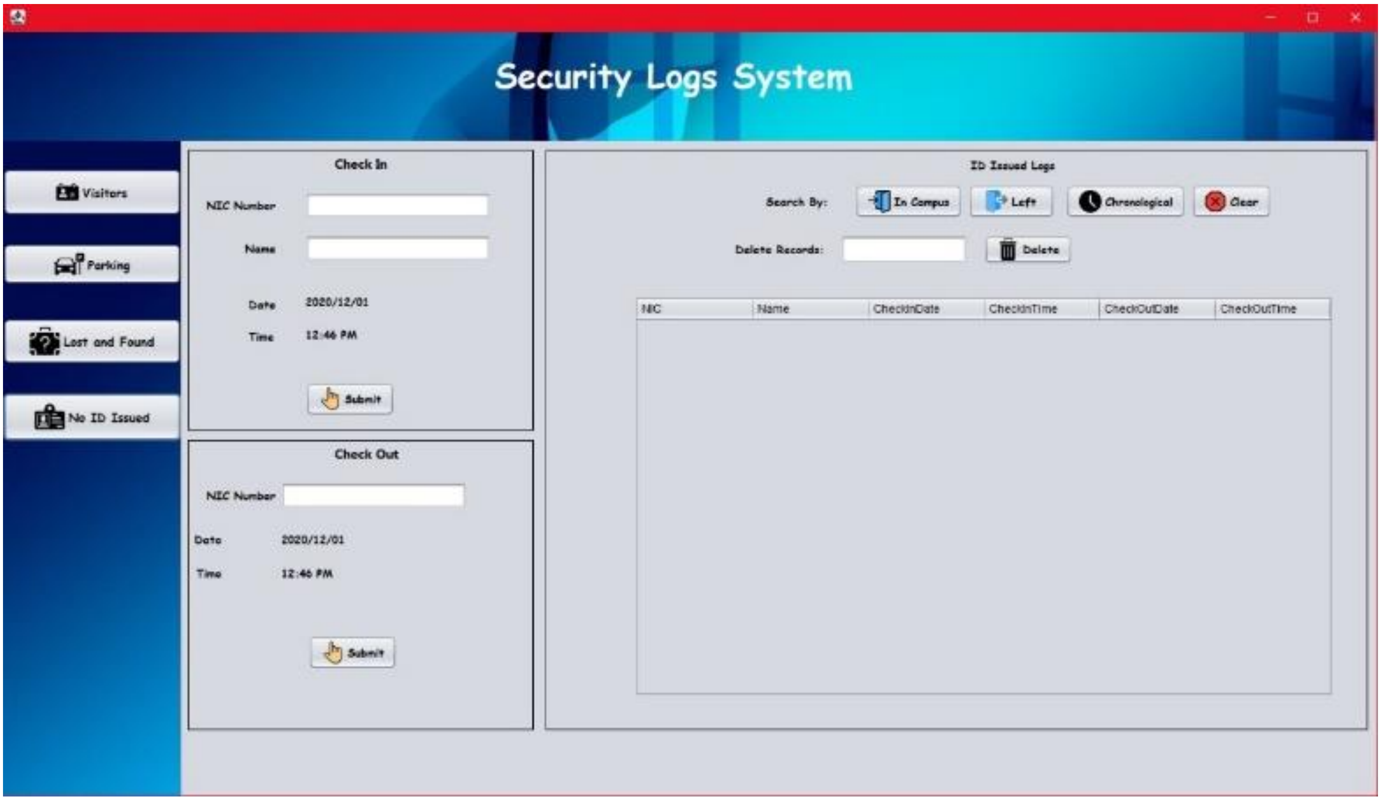
Date 2020/12/01

Time 12:46 PM

NIC Number

Time 12:46 PM

Name	NIC	Date	Time	Purpose	Inventory	PhoneNo	CheckOutTime



Date 2020/12/01

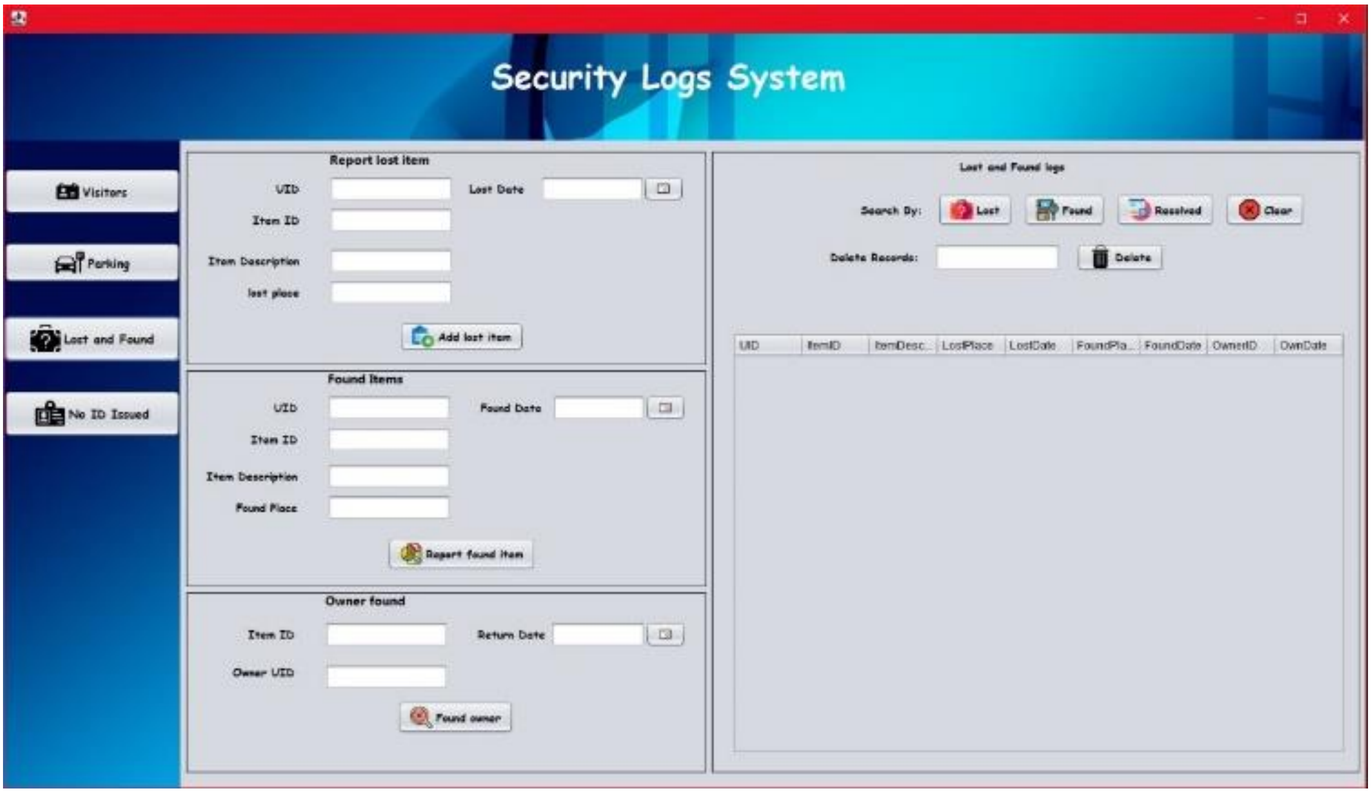
Time 12:46 PM

NIC Number

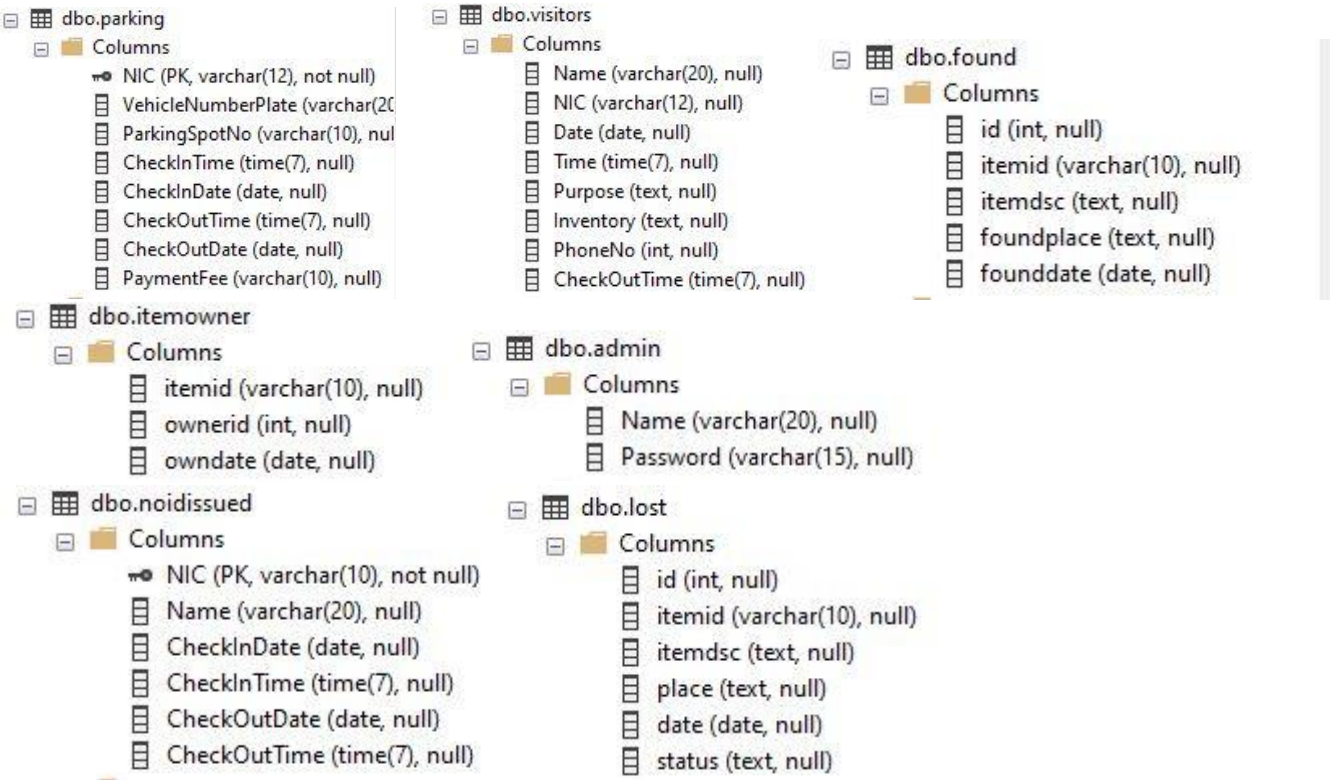
Date 2020/12/01

Time 12:46 PM

NIC	Name	CheckInDate	CheckInTime	CheckOutDate	CheckOutTime



Database



dbo.parking
Columns
NIC (PK, varchar(12), not null)
VehicleNumberPlate (varchar(2
ParkingSpotNo (varchar(10), nul
CheckInTime (time(7), null)
CheckInDate (date, null)
CheckOutTime (time(7), null)
CheckOutDate (date, null)
PaymentFee (varchar(10), null)

dbo.itemowner
Columns
itemid (varchar(10), null)
ownerid (int, null)
owndate (date, null)

dbo.noidissued
Columns
NIC (PK, varchar(10), not null)
Name (varchar(20), null)
CheckInDate (date, null)
CheckInTime (time(7), null)
CheckOutDate (date, null)
CheckOutTime (time(7), null)

dbo.visitors
Columns
Name (varchar(20), null)
NIC (varchar(12), null)
Date (date, null)
Time (time(7), null)
Purpose (text, null)
Inventory (text, null)
PhoneNo (int, null)
CheckOutTime (time(7), null)

dbo.admin
Columns
Name (varchar(20), null)
Password (varchar(15), null)

dbo.lost
Columns
id (int, null)
itemid (varchar(10), null)
itemdsc (text, null)
place (text, null)
date (date, null)
status (text, null)

dbo.found
Columns
id (int, null)
itemid (varchar(10), null)
itemdsc (text, null)
foundplace (text, null)
founddate (date, null)

Conclusion

The users are able to easily keep track of the records of the visitors, parking details, lost and found and people who visits the university without a ID card. Unlike the traditional system, they can filter records and sort records when they need. Users must be given credentials to login the system by a higher position.

Future Improvements,

- The software can be connected to the NSBM main database so that the relevant student and staff information can be viewed through our system.
- Include and alert system for lost and found section to notify people about lost and found items.
- Include a card payment integration for the parking section.