Table of Contents

Tabel of Figures

Introduction to Assignment

In assignment I have todevelopan application is that manages the wardens and prisoners at a prison. Each prisoner is identified by a unique number. Details, such as name and number of days left to serve in prison, need to be stored for each prisoner as well as the unique identifier. Some prisoners are categorized as high security prisoners, in which case additional information must be stored, such as security level and whether or not the prisoner can share a cell with another prisoner. Wardens have a rank in the range 1 to 5, where a rank of 5 indicates the highest level of experience.

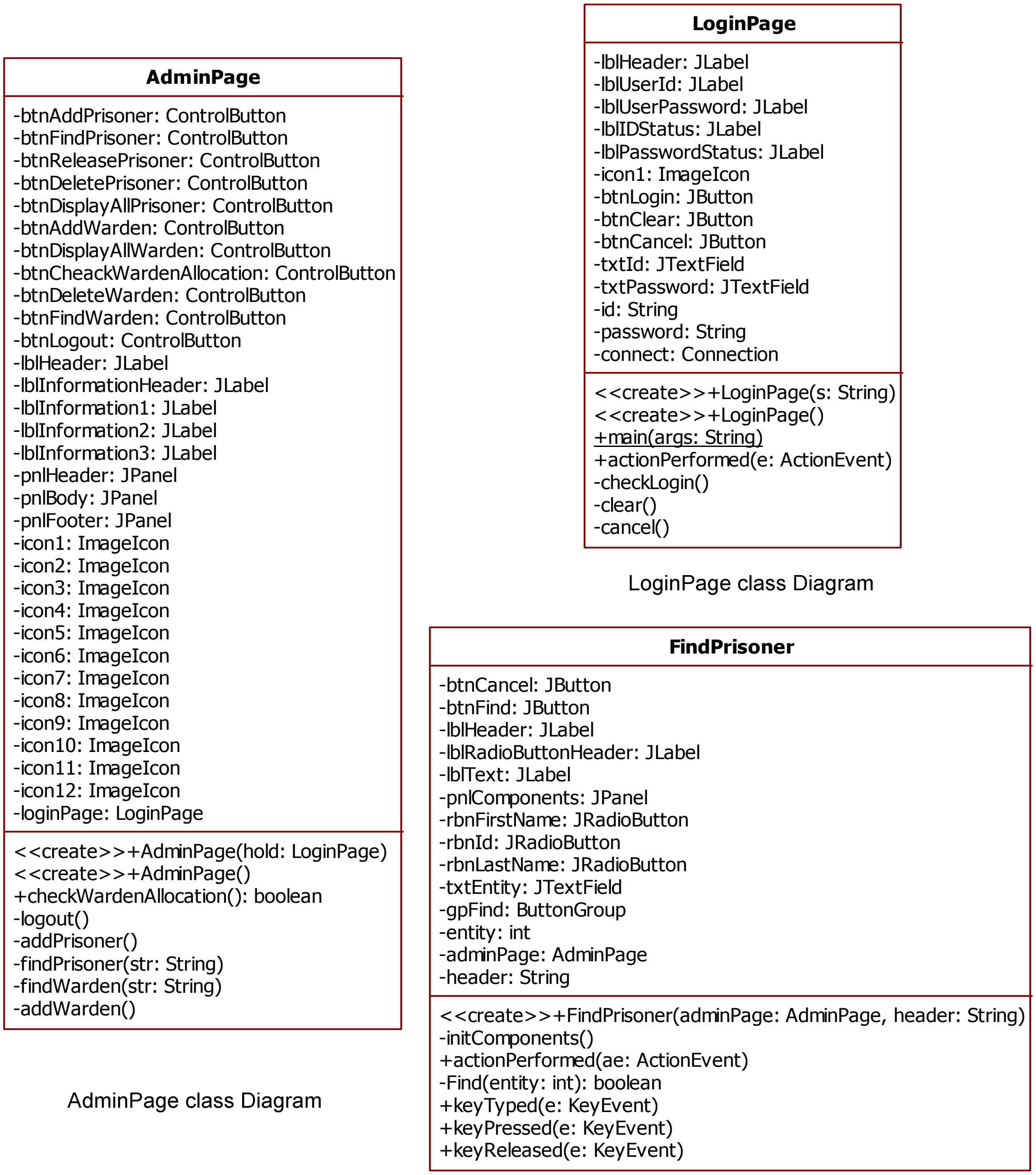
The application must provide at least the following functionalities:

- Add a new prisoner
- Find and display a prisoner, identified by number
- Release a prisoner, identified by number (a prisoner can only be released, at the discretion of the prison services, if they have less than 7 days left to serve)
- Delete a prisoner (this function will come into play in the event of death of a prisoner)
- Add a new prison warden
- Delete a prison warden
- Display all the wardens
- Display all the prisoners
- Check if the prison has an appropriate set of wardens for the current prisoners. The prison should have no less than 1 warden for every 5 prisoners. The prison must have 2 wardens of rank 3 or above for every 4 high security prisoners.

Along with these requirements I have consider some assumptions which are described in Assumptions part.

UML Diagrams

Class Diagrams



LoginPage

-lblHeader: JLabel
-lblUserId: JLabel
-lblUserPassword: JLabel
-lblIDStatus: JLabel
-lblPasswordStatus: JLabel
-icon1: ImageIcon
-btnLogin: JButton
-btnClear: JButton
-btnCancel: JButton
-txtId: JTextField
-txtPassword: JTextField
-id: String
-password: String
-connect: Connection

<<create>>+LoginPage(s: String)
<<create>>+LoginPage()
+main(args: String)
+actionPerformed(e: ActionEvent)
-checkLogin()
-clear()
-cancel()

LoginPage class Diagram

AdminPage

-btnAddPrisoner: ControlButton
-btnFindPrisoner: ControlButton
-btnReleasePrisoner: ControlButton
-btnDeletePrisoner: ControlButton
-btnDisplayAllPrisoner: ControlButton
-btnAddWarden: ControlButton
-btnDisplayAllWarden: ControlButton
-btnCheackWardenAllocation: ControlButton
-btnDeleteWarden: ControlButton
-btnFindWarden: ControlButton
-btnLogout: ControlButton
-lblHeader: JLabel
-lblInformationHeader: JLabel
-lblInformation1: JLabel
-lblInformation2: JLabel
-lblInformation3: JLabel
-pnlHeader: JPanel
-pnlBody: JPanel
-pnlFooter: JPanel
-icon1: ImageIcon
-icon2: ImageIcon
-icon3: ImageIcon
-icon4: ImageIcon
-icon5: ImageIcon
-icon6: ImageIcon
-icon7: ImageIcon
-icon8: ImageIcon
-icon9: ImageIcon
-icon10: ImageIcon
-icon11: ImageIcon
-icon12: ImageIcon
-loginPage: LoginPage

<<create>>+AdminPage(hold: LoginPage)
<<create>>+AdminPage()
+checkWardenAllocation(): boolean
-logout()
-addPrisoner()
-findPrisoner(str: String)
-findWarden(str: String)
-addWarden()

AdminPage class Diagram

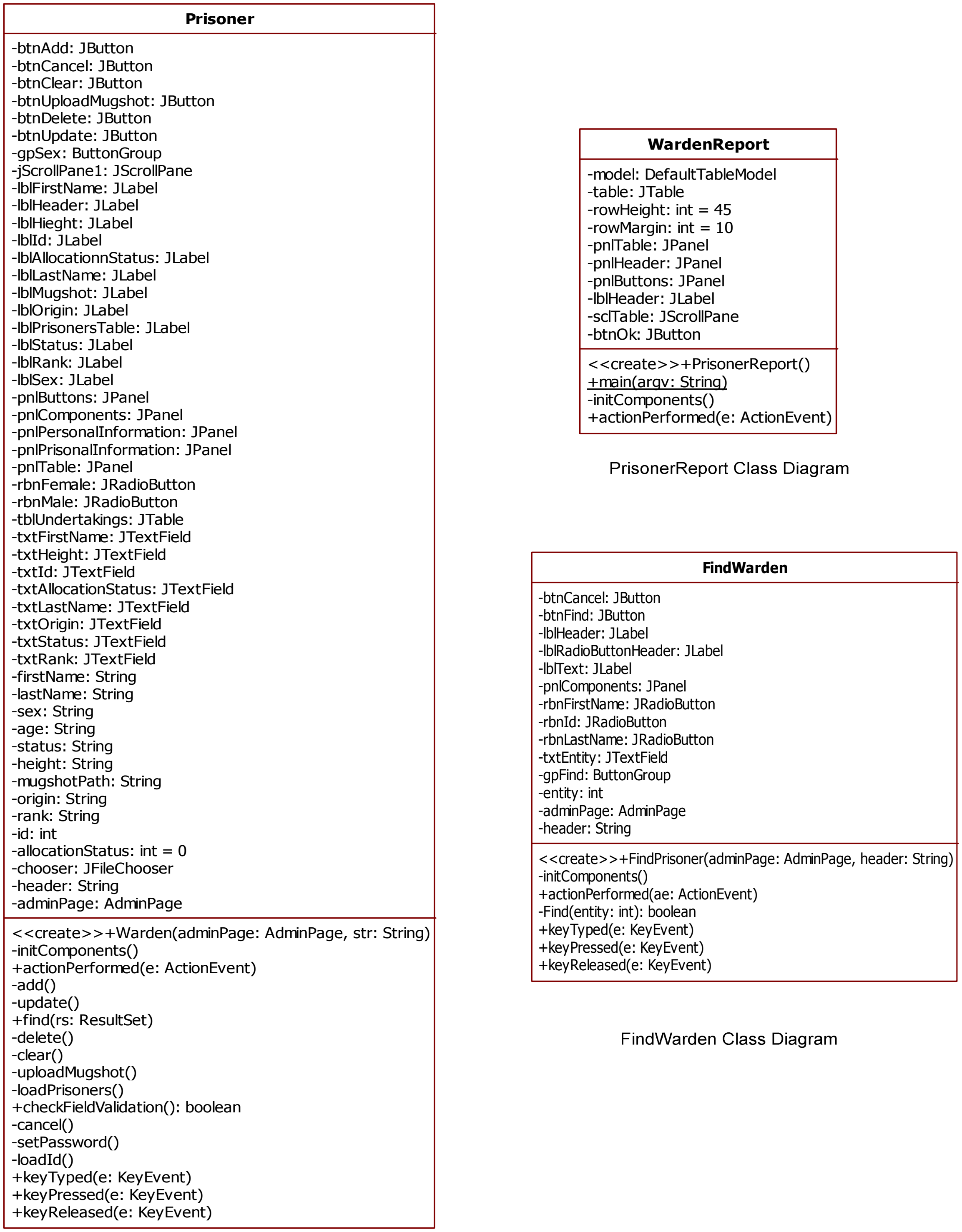
FindPrisoner

-btnCancel: JButton
-btnFind: JButton
-lblHeader: JLabel
-lblRadioButtonHeader: JLabel
-lblText: JLabel
-pnlComponents: JPanel
-rbnFirstName: JRadioButton
-rbnId: JRadioButton
-rbnLastName: JRadioButton
-txtEntity: JTextField
-gpFind: ButtonGroup
-entity: int
-adminPage: AdminPage
-header: String

<<create>>+FindPrisoner(adminPage: AdminPage, header: String)
-initComponents()
+actionPerformed(ae: ActionEvent)
-Find(entity: int): boolean
+keyTyped(e: KeyEvent)
+keyPressed(e: KeyEvent)
+keyReleased(e: KeyEvent)

FindPrisoner class Diagram

Figure 1 Class diagrams of adminPage, FindPrisoner, LginPage

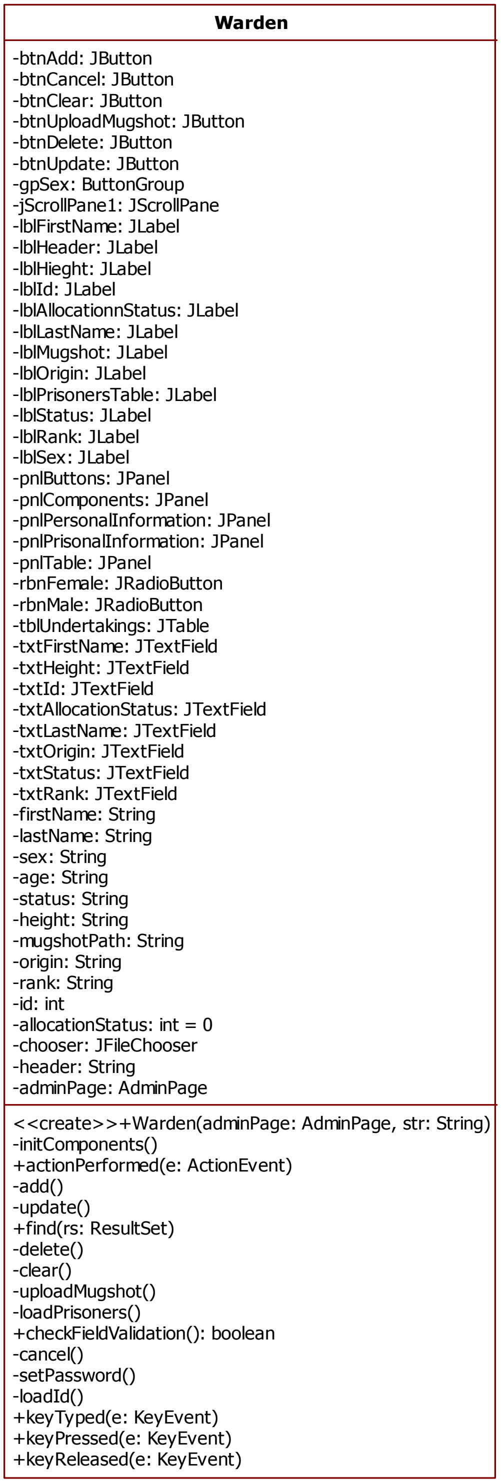


PrisonerReport Class Diagram

FindWarden Class Diagram

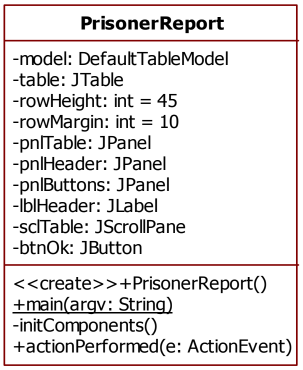
Prisoner Class Diagram

Figure 2 Class Diagrams of Prisoner, FindWarden, PrisonReport



```
Warden
-btnAdd: JButton
-btnCancel: JButton
-btnClear: JButton
-btnUploadMugshot: JButton
-btnDelete: JButton
-btnUpdate: JButton
-gpSex: ButtonGroup
-jScrollPane1: JScrollPane
-lblFirstName: JLabel
-lblHeader: JLabel
-lblHieght: JLabel
-lblId: JLabel
-lblAllocationnStatus: JLabel
-lblLastName: JLabel
-lblMugshot: JLabel
-lblOrigin: JLabel
-lblPrisonersTable: JLabel
-lblStatus: JLabel
-lblRank: JLabel
-lblSex: JLabel
-pnlButtons: JPanel
-pnlComponents: JPanel
-pnlPersonalInformation: JPanel
-pnlPrisonalInformation: JPanel
-pnlTable: JPanel
-rbnFemale: JRadioButton
-rbnMale: JRadioButton
-tblUndertakings: JTable
-txtFirstName: JTextField
-txtHeight: JTextField
-txtId: JTextField
-txtAllocationStatus: JTextField
-txtLastName: JTextField
-txtOrigin: JTextField
-txtStatus: JTextField
-txtRank: JTextField
-firstName: String
-lastName: String
-sex: String
-age: String
-status: String
-height: String
-mugshotPath: String
-origin: String
-rank: String
-id: int
-allocationStatus: int = 0
-chooser: JFileChooser
-header: String
-adminPage: AdminPage
<<create>>+Warden(adminPage: AdminPage, str: String)
-initComponents()
+actionPerformed(e: ActionEvent)
-add()
-update()
+find(rs: ResultSet)
-delete()
-clear()
-uploadMugshot()
-loadPrisoners()
+checkFieldValidation(): boolean
-cancel()
-setPassword()
-loadId()
+keyTyped(e: KeyEvent)
+keyPressed(e: KeyEvent)
+keyReleased(e: KeyEvent)
```

Class Diagram of Warden



```
PrisonerReport
-model: DefaultTableModel
-table: JTable
-rowHeight: int = 45
-rowMargin: int = 10
-pnlTable: JPanel
-pnlHeader: JPanel
-pnlButtons: JPanel
-lblHeader: JLabel
-sclTable: JScrollPane
-btnOk: JButton
<<create>>+PrisonerReport()
+main(argv: String)
-initComponents()
+actionPerformed(e: ActionEvent)
```

PrisonReport Class Diagram

Figure 3 Class Diagram of Prisoner Report, Warden

Class Relation Diagram

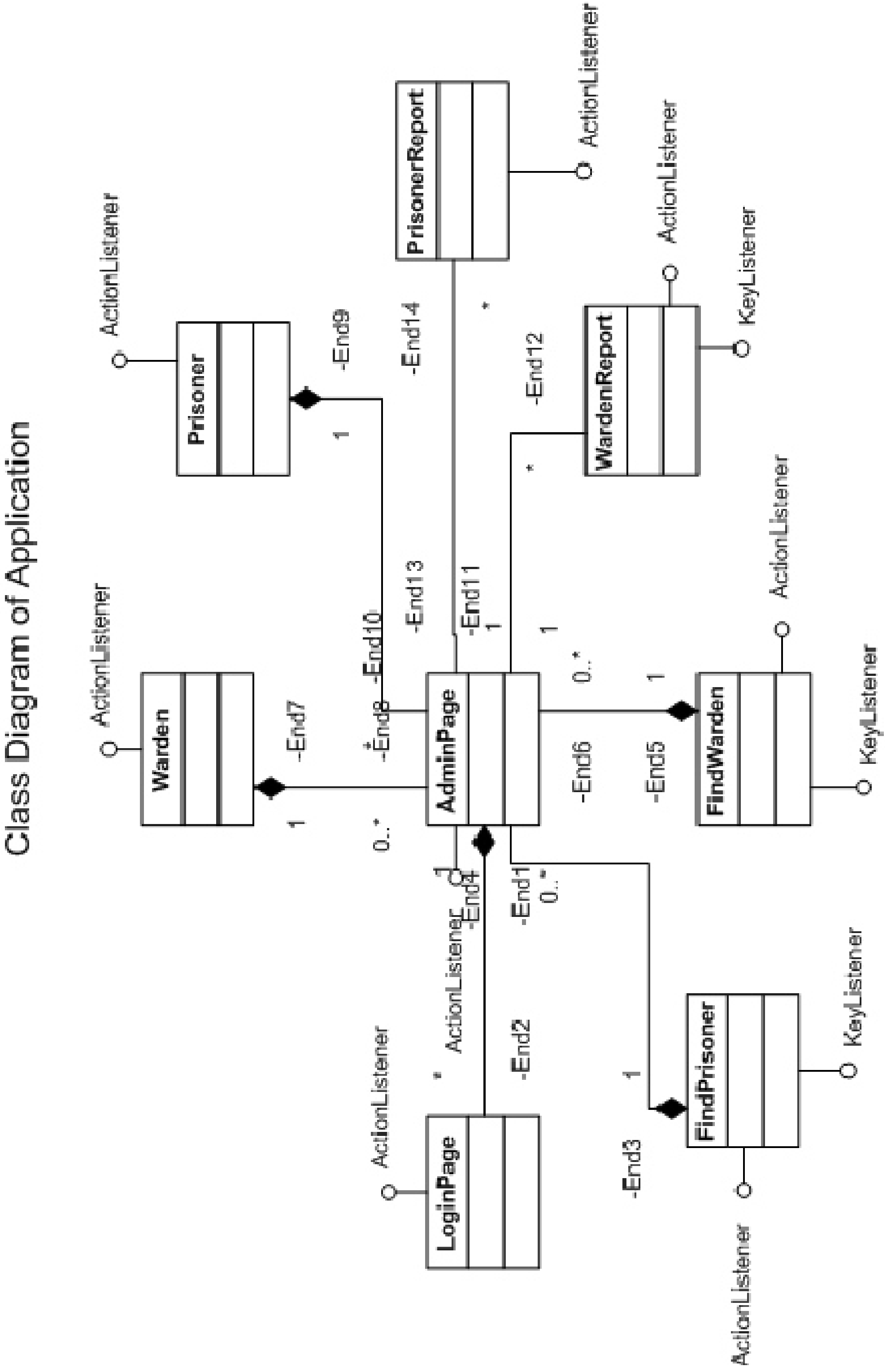


Figure 4 Class Diagram relation Diagram

Activity diagram for User/Admin/Warden

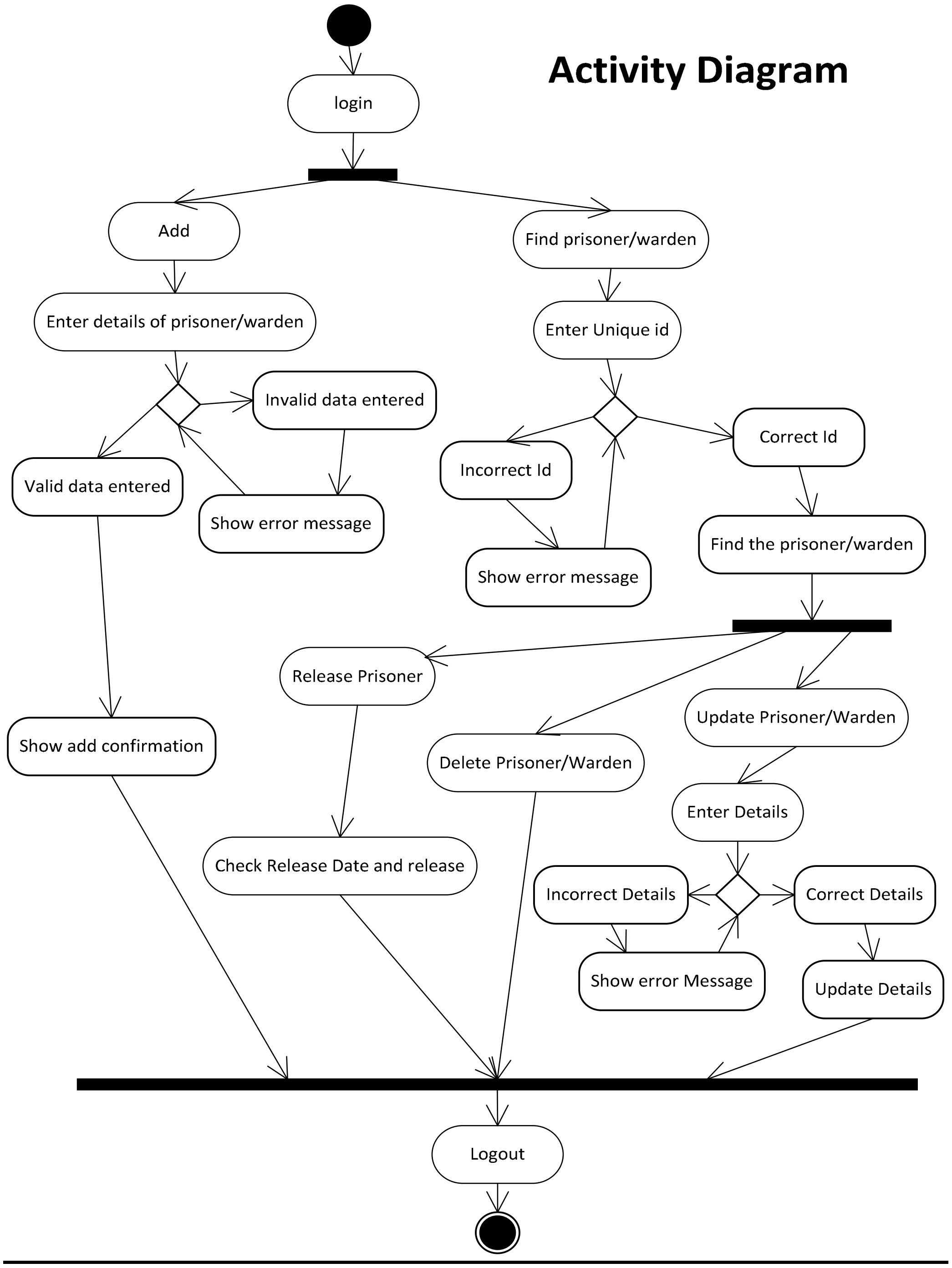


Figure 5 Activity Diagram

USE CASE DIAGRAMS

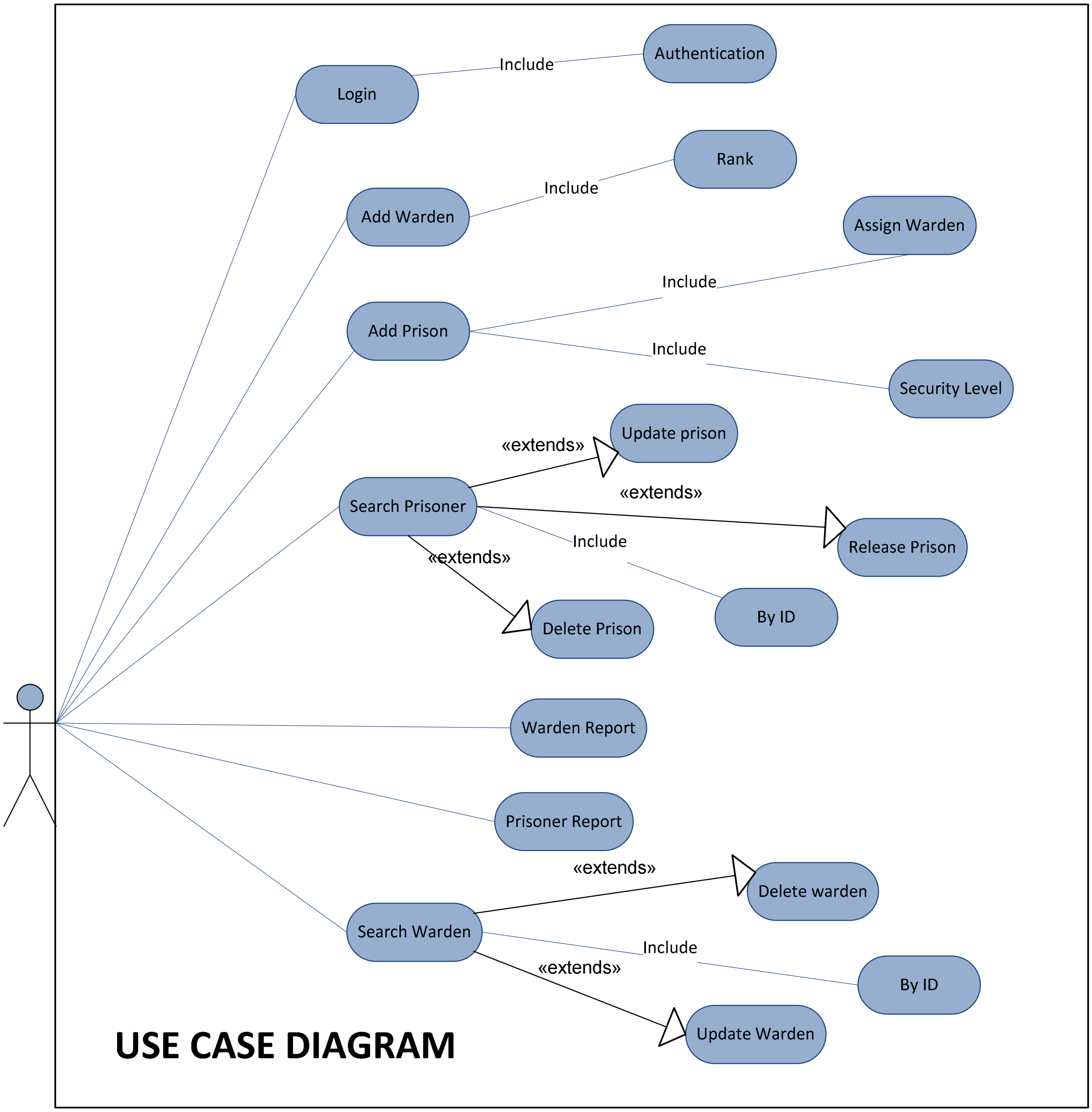


Figure 6 Use Case Diagram

Sequence Diagrams

Sequence Diagram for Add a Warden/Prisoner's Details

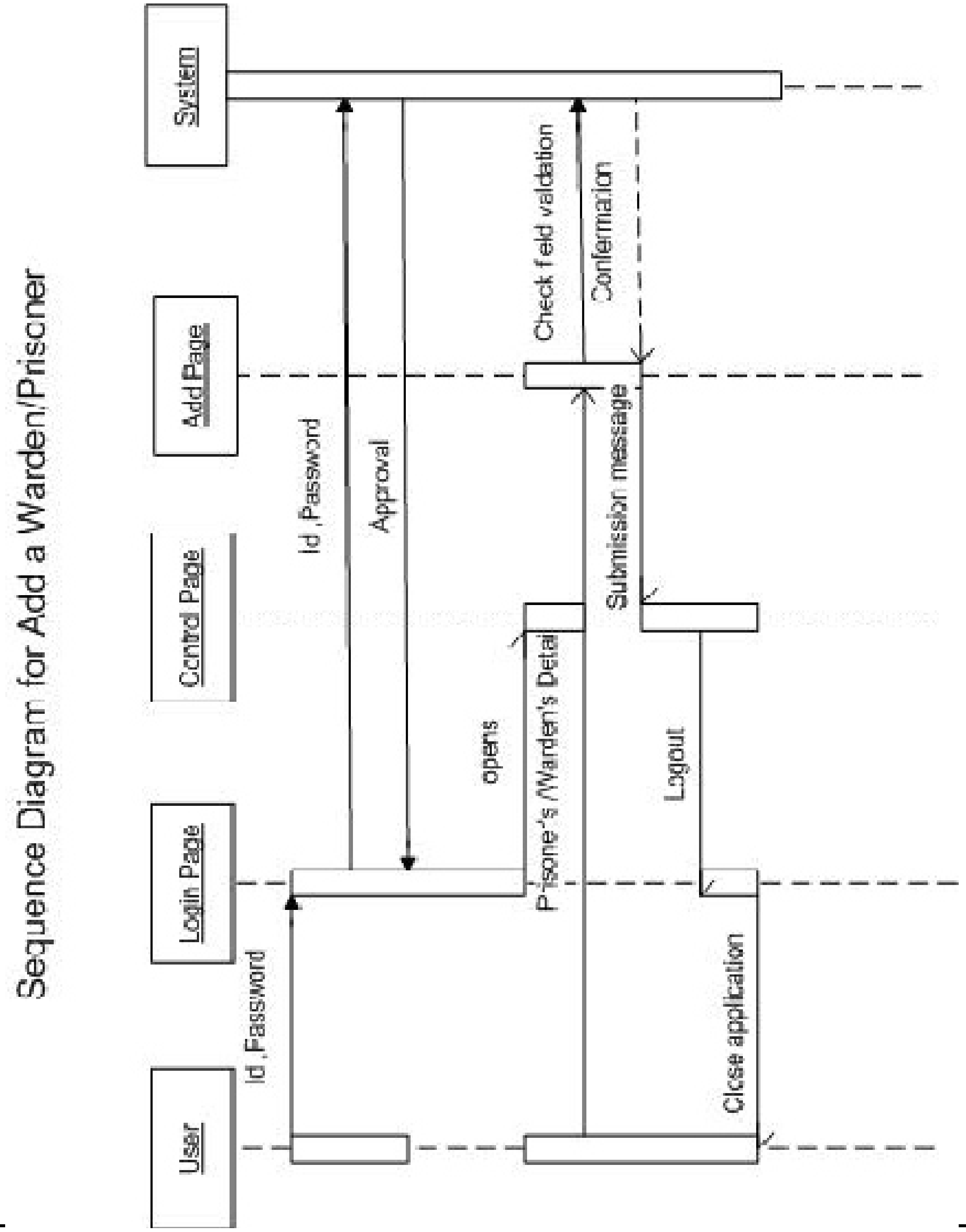


Figure 7Sequence Diagram for Add Records

Sequence Diagram for Update a Warden/Prisoner ‘s Details

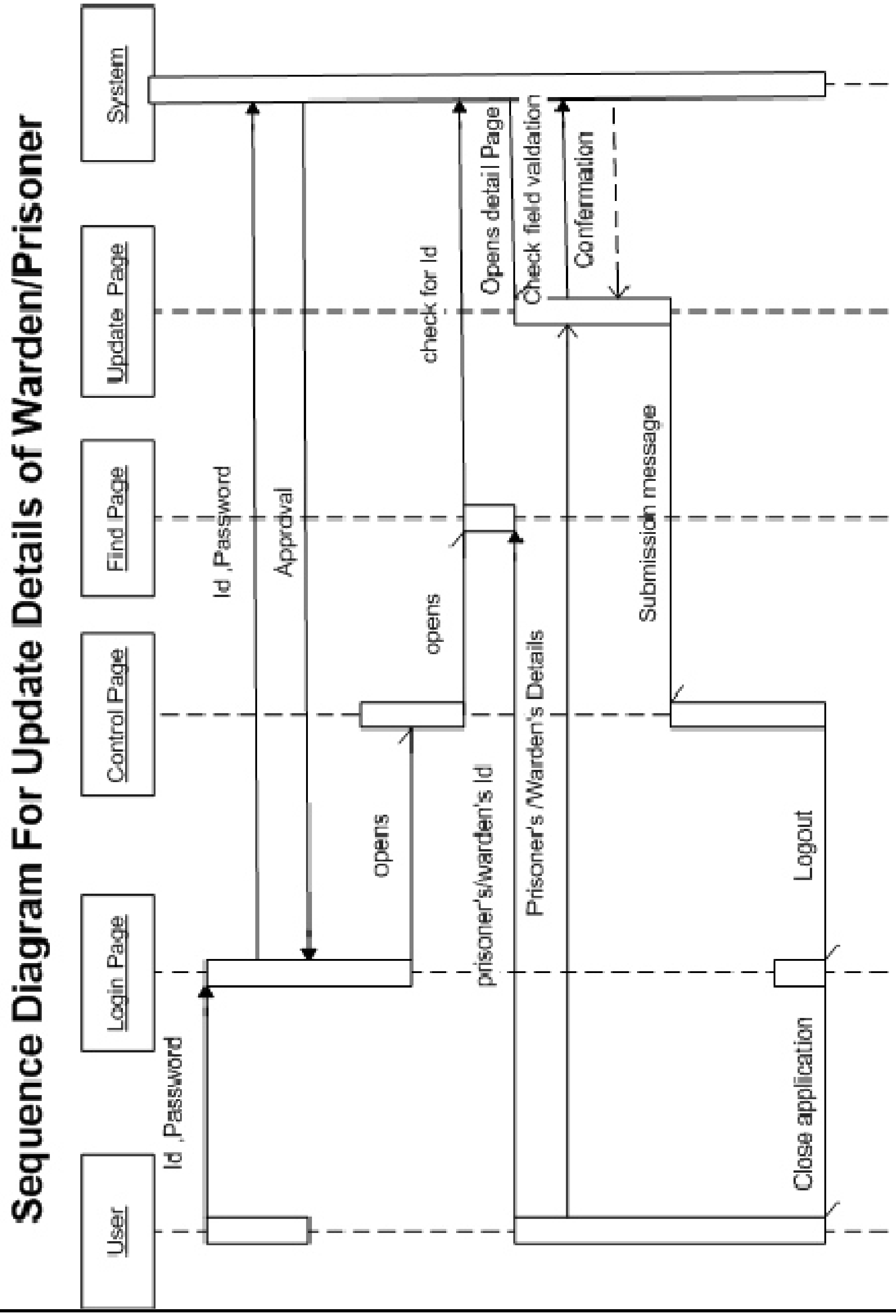


Figure 8 Sequence Diagram for Update Records

Sequence Diagram For Delete a Warden/Prisoner

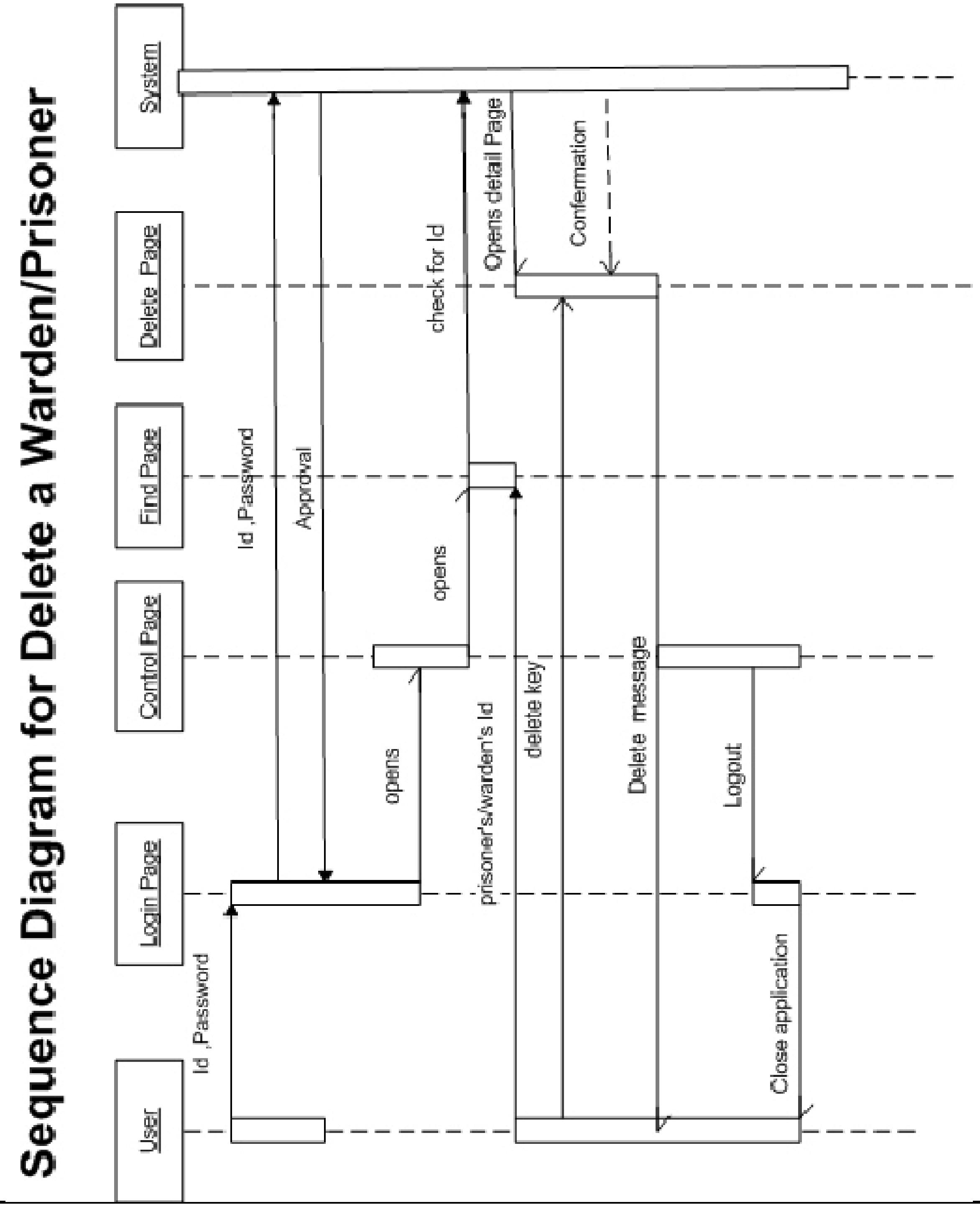


Figure 9 Sequence Diagram for Delete Record

Sequence Diagram for release a Prisoner

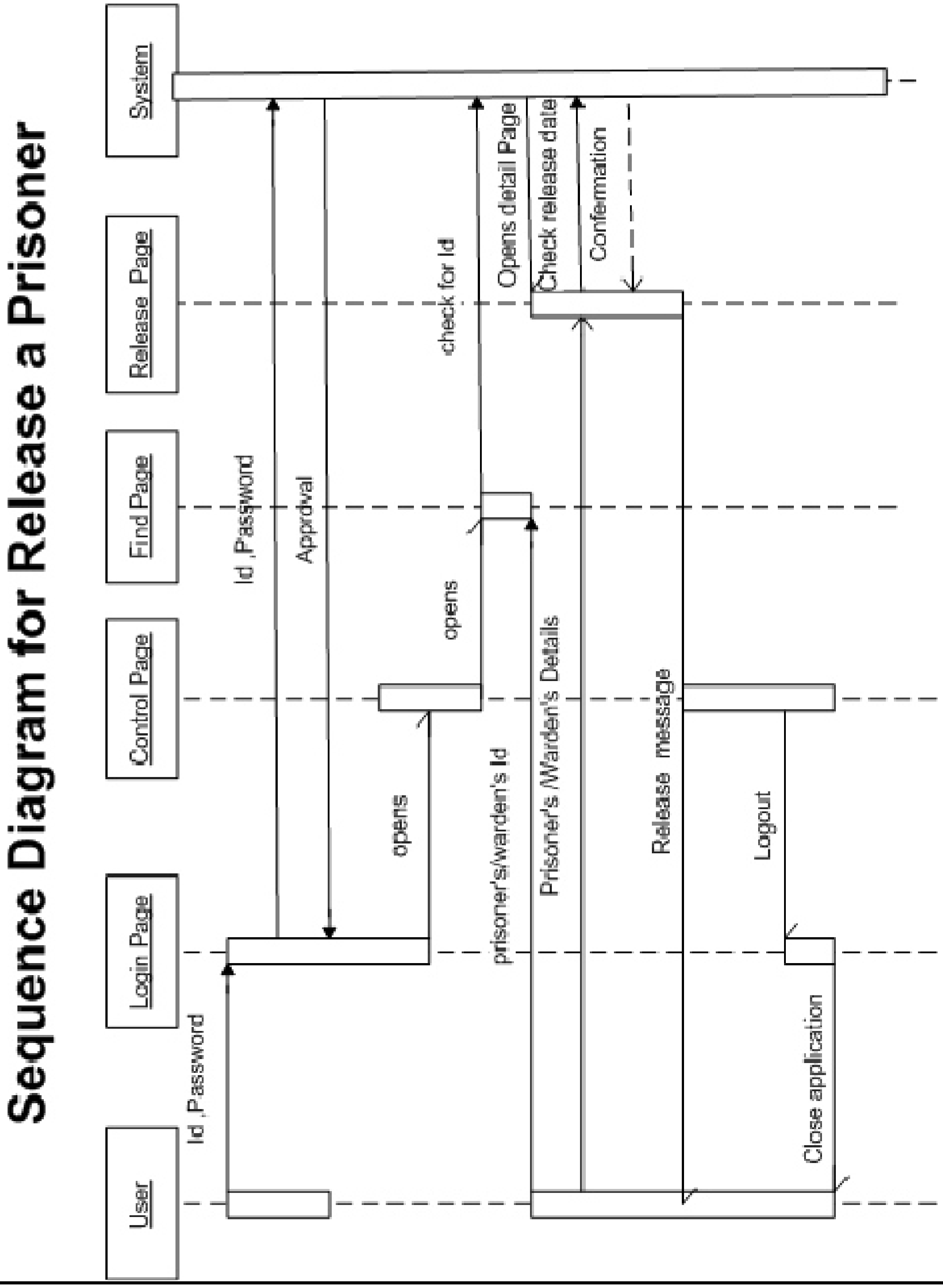


Figure 10Sequence Diagram to release a prisoner

Source Code of Application

LoginPage.java Source code

```
//package pms.login;
//import pms.admin.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
 import java.sql.*;
      /**
      * class {@ code Login Page } is the gate way of prison manage
ment system.
      */
public class LoginPage extends JFrame implements ActionListener {
      public LoginPage(String s){
            super(s);
      }
     JLabel
lblHeader,lblUserId,lblUserPassword,lblIDStatus,lblPasswordStatus;
      ImageIcon icon1;
      JButton btnLogin,btnClear,btnCancel;
      JTextField txtId,txtPassword;
      String id,password;
      Connection connect;
      /*
      * Constructor {@ case LoginPage()} is the only Constructerof
class. and it creates login page.
      */
      public LoginPage(){
         setPreferredSize(new Dimension(1366, 768));
            setLayout(null);
           icon1 = new ImageIcon("images/xyz.jpg");
            lblHeader=new JLabel(icon1);
            lblHeader.setBounds(300,50,790,72);
            add(lblHeader);
            lblUserId=new JLabel("Login ID");
            lblUserId.setBounds(500,300,100,30);
            lblUserId.setFont(new Font("Times New Roman",3,20));
            add(lblUserId);
            lblUserId.setForeground(new Color(255,255,255));

            lblUserPassword=new JLabel("Password");
            lblUserPassword.setBounds(500,350,100,30);
            lblUserPassword.setForeground(new Color(255,255,255));
            lblUserPassword.setFont(new Font("Times New Roman",3,20));
            add(lblUserPassword);

            txtId=new JTextField();
            txtId.setBounds(650,300,100,30);
            txtId.setForeground(new Color(72,72,74));
            add(txtId);
```

```
          lblIDStatus=new JLabel();
          lblIDStatus.setBounds(800,300,500,30);
          lblIDStatus.setForeground(new Color(255,0,0));
          add(lblIDStatus);

          txtPassword= new JTextField();
          txtPassword.setBounds(650,350,100,30);
          txtPassword.setForeground(new Color(72,72,74));
          add(txtPassword);

          lblPasswordStatus=new JLabel();
          lblPasswordStatus.setBounds(800,350,500,30);
          lblPasswordStatus.setForeground(new Color(255,0,0));
          add(lblPasswordStatus);

          btnLogin=new JButton("Login");
          btnLogin.setBounds(500,450,100,30);
          btnLogin.setForeground(new Color(255,255,255));
          btnLogin.setBackground(new Color(72,72,74));
          btnLogin.setOpaque(true);
          btnLogin.addActionListener(this);
          add(btnLogin);

          btnClear=new JButton("Clear");
          btnClear.setBounds(620,450,100,30);
          btnClear.setForeground(new Color(255,255,255));
          btnClear.setBackground(new Color(72,72,74));
          btnClear.addActionListener(this);
          btnClear.setOpaque(true);
          add(btnClear);

          btnCancel=new JButton("Cancel");
          btnCancel.setBounds(740,450,100,30);
          btnCancel.setForeground(new Color(255,255,255));
          btnCancel.setBackground(new Color(72,72,74));
          btnCancel.addActionListener(this);
          btnCancel.setOpaque(true);
          add(btnCancel);

          setLayout(null);
          getContentPane().setBackground(new Color(72,72,74));

          pack();

          try{Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
             System.out.println("Divers loaded successfully");
           connect = DriverManager.getConnection("jdbc:odbc:PMS");
             System.out.println("Connection established
successfully");

             }catch(Exception e){}
           setVisible(true);
           setResizable(false);
      }
      /*
      * Main Method is the Gate way of my application.
      */
      public static void main(String[] args)
      {
           LoginPage v=new LoginPage();
           System.out.println("Login Successfilly");
      }
      /**
      * method that works is called internally when any Button is
pressed
      * and then it call method asociated with button pressed.
      */
      public void actionPerformed(ActionEvent e){
           id=txtId.getText();
           password=txtPassword.getText();
           lblPasswordStatus.setText("");
           lblIDStatus.setText("");
           if(e.getSource()==btnLogin)
                 checkLogin();
           if(e.getSource()==btnClear)
                 clear();
           if(e.getSource()==btnCancel)
                 cancel();
      }
      private void checkLogin(){
           try{

                 Statement state = connect.createStatement();
                 String sql = "select Password from Login_Data where
ID= '"+ id +"'";
            ResultSet rs=state.executeQuery(sql);
                 if(rs.next()){
                      if(password.equals(rs.getString("Password"))){
                            clear();
                            new AdminPage(this).setVisible(true);
                            this.setVisible(false);

                      }else{
                            lblPasswordStatus.setText("!You have
entered a wrong password");
                            txtId.setText("");
                            txtPassword.setText("");
                      }
                 }else{
                      lblIDStatus.setText("!You have entered a Invalid
ID");
                      txtId.setText("");
```

```
                        txtPassword.setText("");
                    }
            }catch(Exception e){
                    System.out.println("Exception in check Login method at 
Login page"+e);
            }
    }
    private void clear(){
                        txtId.setText("");
                        txtPassword.setText("");
                    }
    private void cancel(){
            try{
                    connect.close();
            }catch(Exception e){}
            System.exit(0);
    }
}
```

AdminPage.java Source Code

```
//package pms.admin;
//import pms.prisoner.*;
//import pms.warden.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
 import java.sql.*;
/**
 * The {@code AdminPage} class represents Control page. It provides 
all the control
 * that admin or highest Rank warden can access.
 * <p>
 * Admin Page has following Controls For Prisoner :
 * <blockquote><pre>
 *      Add a Prisoner.
 *         Find a Prisoner.
 *         Release a Prisoner.
 *         Delete a Prisoner.
 *         Update Prisoner's Details.
 *         Genrate Reports of all Prisoner.
 *
 * </pre></blockquote>
 * <p>
 * Admin Page has following Controls For Warden :
 * <blockquote><pre>
 *      Add a Warden.
 *         Find a Warden.
 *         Delete a Warden.
 *         Update Warden's Details.
 *         Genrate Reports of all Warden.
 *
 * </pre></blockquote>
 *
 */


 public class AdminPage extends JFrame {

    ControlButton
btnAddPrisoner,btnFindPrisoner,btnReleasePrisoner,btnDeletePrisoner;
    ControlButton
btnDisplayAllPrisoner,btnAddWarden,btnDisplayAllWarden,btnCheackWarden
Allocation;
    ControlButton btnDeleteWarden,btnFindWarden,btnLogout;
    JLabel
lblHeader,lblInformationHeader,lblInformation1,lblInformation2,lblInfo
rmation3;
      JPanel pnlHeader,pnlBody,pnlFooter;
      ImageIcon
icon1,icon2,icon3,icon4,icon5,icon6,icon7,icon8,icon9,icon10,icon11,ic
on12;
      LoginPage loginPage;
      /**
      * Constructer that calls another  admin page constructer .
      */
      public AdminPage(LoginPage hold){
            this();
            loginPage=hold;
      }
      /**
      * Constructer that creates admin page form and provides all the
controls.
      */
      public AdminPage(){

          super("Admin Controls");
          setSize(1366,768);
            setPreferredSize(new Dimension(1366, 768));
            setLayout(null);

        icon1 = new ImageIcon("images/xyz.jpg");
          icon2 = new ImageIcon("images/addprisoner.jpg");
          icon3 = new ImageIcon("images/findprisoner.jpg");
          icon4 = new ImageIcon("images/releaseprisoner.jpg");
          icon5 = new ImageIcon("images/deleteprisonerwarden.jpg");
            icon6 = new ImageIcon("images/displayprisoner.jpg");
            icon7 = new ImageIcon("images/addwarden.jpg");
            icon8 = new ImageIcon("images/displayAllWarden.jpg");
            icon9 = new ImageIcon("images/WardenAllocation.jpg");
            icon10 = new ImageIcon("images/deleteprisoner.jpg");
            icon11 =new ImageIcon("images/logout.jpg");
            icon12 = new ImageIcon("images/findWarden.jpg");
      lblHeader=new JLabel(icon1);
            lblHeader.setBounds(123,50,790,72);
            add(lblHeader);

            btnAddPrisoner=new ControlButton(icon2,123,150,this);
            btnFindPrisoner=new ControlButton(icon3,353,150,this);
            btnReleasePrisoner=new ControlButton(icon4,583,150,this);
            btnDeletePrisoner=new ControlButton(icon5,813,150,this);
            btnDisplayAllPrisoner=new
ControlButton(icon6,1043,150,this);
            btnAddWarden=new ControlButton(icon7,123,410,this);
            btnDisplayAllWarden=new ControlButton(icon8,353,410,this);
            btnCheackWardenAllocation=new
ControlButton(icon9,583,410,this);
      btnDeleteWarden=new ControlButton(icon10,813,410,this);
            btnFindWarden=new ControlButton(icon12,1043,410,this);
            btnLogout= new ControlButton(icon11,1220,20,100,50,this);

            //lblInformation
            lblInformationHeader=new JLabel();
            lblInformation1=new JLabel();
            lblInformation2=new JLabel();
            lblInformation3=new JLabel();
            lblInformationHeader.setBounds(50,650,400,25);
            lblInformationHeader.setFont(new Font("Tahoma", 1, 16));
            lblInformationHeader.setForeground(new Color(255,255,255));
            lblInformation1.setBounds(50,675,600,25);
            lblInformation1.setFont(new Font("Tahoma", 1, 16));
            lblInformation1.setForeground(new Color(255,255,255));
            lblInformation2.setBounds(600,675,700,25);
            lblInformation2.setFont(new Font("Tahoma", 1, 16));
            lblInformation2.setForeground(new Color(255,255,255));
            lblInformation3.setBounds(50,700,600,25);
            lblInformation3.setFont(new Font("Tahoma", 1, 16));
            lblInformation3.setForeground(new Color(255,0,0));
            add(lblInformationHeader);
            add(lblInformation1);
            add(lblInformation2);
            add(lblInformation3);
            setLayout(null);
            getContentPane().setBackground(new Color(72,72,74));

            pack();
            setResizable(false);

      }
      /**
      * method  that checks waden allocation and shows the status in
Information bar.
      */
      public boolean checkWardenAllocation(){
```

```
                int
wardenCount,prisonerCount,highLevelPrisoner=0,lowLevelPrisoner=0,highR
ankWarden=0,lowRankWarden=0;
                boolean flag=false;
            try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully in
checkwardenAllocation at adminPage");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in checkwardenAllocation at adminPage");
           Statement state = connect.createStatement();
           String sql1 = "select * from Prisoner_Details where Status=
'" + "Incarcerated" + "'";
           ResultSet rsWarden,rsPrisoner=state.executeQuery(sql1);
              while(rsPrisoner.next())
                {

     if(((String)rsPrisoner.getString("Security_Class")).equals("High
Risk"))
                                 highLevelPrisoner=highLevelPrisoner+1;
                           else
                                 lowLevelPrisoner=lowLevelPrisoner+1;
                }
              rsPrisoner.close();

               sql1 = "select * from Warden_Details where Status= '" +
"Working" +"'";
                rsWarden=state.executeQuery(sql1);
              while(rsWarden.next())
                {

                           if(rsWarden.getInt("Rank")>=3)
                                highRankWarden=highRankWarden+1;
                           else
                                 lowRankWarden=lowRankWarden+1;
                }

                lblInformationHeader.setText("Information Bar");
                      lblInformation1.setText("Number of High Rank
="+highRankWarden+" "+"Number of Low Rank Warden ="+lowRankWarden);
                      lblInformation2.setText("Number of Low Security
Prisoners ="+lowLevelPrisoner+" "+"Number of High Security Prisoners
="+highLevelPrisoner);
                      int hold=(highLevelPrisoner+1)/2-highRankWarden;
                      if((highLevelPrisoner+1)/2>highRankWarden)
                      lblInformation3.setText(hold+" "+"New High Rank
Warden are Required");
                      flag = true;
                rsWarden.close();
            state.close();
            connect.close();
            }catch (Exception e) {
            System.out.print(e);
         }
            return flag;
      }
      private void logout(){
            loginPage.setVisible(true);
            this.dispose();
      }
      private void addPrisoner(){
            new Prisoner(this,"Add a Prisoner").setVisible(true);
            setVisible(false);
      }
      private void findPrisoner(String str){
            new FindPrisoner(this,str);
      }
            private void findWarden(String str){
            new FindWarden(this,str);
      }
      private void addWarden(){
            new Warden(this,"Add a Warden").setVisible(true);
            setVisible(false);
      }

      class ControlButton extends JButton implements ActionListener{
      ControlButton(Icon c,int x,int y,Container cnt){
            super(c);
            this.setBounds(x,y,200,230);
            cnt.add(this);
            addActionListener(this);
      }
      ControlButton(Icon c,int w,int x,int y,int z,Container cnt){
            super(c);
            this.setBounds(w,x,y,z);
            cnt.add(this);
            addActionListener(this);
      }
      /**
      * method that works is called internally when any Button is
pressed
      * and then it call method asociated with button pressed.
      */
      public void actionPerformed(ActionEvent e){
            if(e.getSource()==btnFindPrisoner){
                  findPrisoner("Find the Prisoner");
            }
            if(e.getSource()==btnAddPrisoner){
                  addPrisoner();
            }
            if(e.getSource()==btnLogout){
                  logout();
            }
            if(e.getSource()==btnDeletePrisoner){
            findPrisoner("Delete a Prisoner");
            }
            if(e.getSource()==btnReleasePrisoner){
            findPrisoner("Release a Prisoner");
            }
            if(e.getSource()==btnDisplayAllPrisoner)
                  new PrisonerReport();
            if(e.getSource()==btnDisplayAllWarden)
                  new WardenReport();
            if(e.getSource()==btnFindWarden)
                  findWarden("Find the Warden");
            if(e.getSource()==btnDeleteWarden)
                  findWarden("Delete a Warden");
            if(e.getSource()==btnCheackWardenAllocation)
                  checkWardenAllocation();
            if(e.getSource()==btnAddWarden)
                  addWarden();
         }
      }
  }
```

Prisoner.java Source Code

```
//package pms.prisoner;
// import pms.admin.*;
 import java.awt.*;
 import javax.swing.*;
 import javax.swing.border.*;
 import java.awt.event.*;
 import java.sql.*;
 import java.io.*;
 import java.util.Calendar;
 /**
 *The {@code Prisoner} class represents an actual prisoner and of its
attributes and properties for Prison Management sysytem.
 * Prisoner page provides following functionalities :
 * <blockquote><pre>
 *      Add a Prisoner.
 *         Find a Prisoner.
 *         Release a Prisoner.
 *         Delete a Prisoner.
 *         Update Prisoner's Details.
 *         Genrate Reports of all Prisoner.
 * </pre></blockquote>
 */
public class Prisoner extends JFrame implements ActionListener{
     /**
     *Button for {@code Submitting} the prisoner Form.
     */
```

```
    private JComboBox cbxSecurityClass;
      private JScrollPane scrollCrimeDescription;
      private int id;
      private int     month;
      private int     day;
      private int     year;
      private int     allocatedWardenId;
      private int     allocationStatus;
      private int     holdId;
      static private int count=0;
      private JFileChooser chooser;
      private Calendar calRelease,calToday;
      private ImageIcon mugshot;
      private Connection connect;
      private Statement state;
      private String header;
      private AdminPage adminPage;

    /**
     * Creates new  Prisoner form and add all the components to the
form
     */
      public Prisoner(AdminPage adminPage,String str){
            header=str;
            this.adminPage=adminPage;

          pnlComponents = new JPanel();
          pnlPersonalInformation = new JPanel();
          lblFirstName = new JLabel();
          txtFirstName = new JTextField();
          lblLastName = new JLabel();
          txtLastName = new JTextField();
          lblSex = new JLabel();
          rbnMale = new JRadioButton();
          rbnFemale = new JRadioButton();
          lblMugshot = new JLabel();
          lblRace = new JLabel();
          txtAge = new JTextField();
          lblOrigin = new JLabel();
          txtOrigin = new JTextField();
          lblHieght = new JLabel();
          txtHeight = new JTextField();
          btnUploadMugshot = new JButton();
          pnlPrisonalInformation = new JPanel();
          lblId = new JLabel();
          txtId = new JTextField();
          lblBookingDate = new JLabel();

          lblReleaseDate = new JLabel();
          lblDaysToIncarcerate = new JLabel();
          txtCountToIncarcerate = new JTextField();
          lblStatus = new JLabel();
```

```
        txtStatus = new JTextField();
        lblSecurityClass = new JLabel();
        cbxSecurityClass = new JComboBox();
        lblCrimeDescription = new JLabel();
           lblSelectWarden= new JLabel();

        scrollCrimeDescription = new JScrollPane();

        lblHeader = new JLabel();
        pnlButtons = new JPanel();
        btnAdd = new JButton();
        btnUpdate = new JButton();
           btnRelease= new JButton();
        btnDelete = new JButton();
        btnClear = new JButton();
        btnCancel = new JButton();
           gpSex=new ButtonGroup();
        textCrimeDescription = new JTextArea();
        cbxBookDay = new JComboBox();
        cbxBookMonth = new JComboBox();
        cbxBookYear = new JComboBox();
        cbxReleaseDay = new JComboBox();
        cbxReleaseMonth = new JComboBox();
        cbxReleaseYear = new JComboBox();
           cbxSelectWarden= new JComboBox();
           chooser = new JFileChooser();
           calRelease=Calendar.getInstance();
           calToday=Calendar.getInstance();
              try{Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully");

             connect = DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established
successfully");
             initComponents();
              loadWarden();
                 connect.close();
                 loadId();
              }catch(Exception e){}
     }

    private void initComponents() {
        GridBagConstraints gridBagConstraints;


        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(750, 660));

        pnlComponents.setLayout(new GridLayout(2, 1));
pnlPersonalInformation.setBorder(BorderFactory.createTitledBorder(null
, "Personal Information", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, new Font("Agency FB", 2, 24), new
Color(72, 72, 74))); // NOI18N
        pnlPersonalInformation.setLayout(new GridBagLayout());

        lblFirstName.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblFirstName.setText("First Name");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.ipadx = 7;
        gridBagConstraints.ipady = 4;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(lblFirstName, gridBagConstraints);

        txtFirstName.setText("");
        txtFirstName.setPreferredSize(new Dimension(90, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.ipadx = 7;
        gridBagConstraints.ipady = 4;
        gridBagConstraints.weightx = 0.01;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(txtFirstName, gridBagConstraints);

        lblLastName.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblLastName.setText("Last Name");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.ipadx = 7;
        gridBagConstraints.ipady = 4;
        gridBagConstraints.weightx = 0.01;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(lblLastName, gridBagConstraints);

        txtLastName.setText("");
        txtLastName.setPreferredSize(new Dimension(90, 20));

        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.ipadx = 2;
        gridBagConstraints.ipady = 4;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(txtLastName, gridBagConstraints);

        lblSex.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblSex.setText("Sex");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.ipadx = 7;
        gridBagConstraints.ipady = 4;
        gridBagConstraints.anchor = GridBagConstraints.EAST;
        gridBagConstraints.weightx = 0.01;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(lblSex, gridBagConstraints);

        rbnMale.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        rbnMale.setText("Male");
        rbnMale.setSelected(true);

           gpSex.add(rbnMale);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor = GridBagConstraints.WEST;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(rbnMale, gridBagConstraints);

        rbnFemale.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        rbnFemale.setText("Female");
           gpSex.add(rbnFemale);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 6;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor = GridBagConstraints.WEST;
        gridBagConstraints.weighty = 0.04;
        gridBagConstraints.insets = new Insets(0, 0, 0, 28);
        pnlPersonalInformation.add(rbnFemale, gridBagConstraints);


           lblMugshot.setForeground(new Color(72, 72, 74));
        lblMugshot.setHorizontalAlignment(SwingConstants.CENTER);
        lblMugshot.setText("Upload image here");
        lblMugshot.setBorder(BorderFactory.createMatteBorder(1, 1, 1,
1, new Color(72, 72, 74)));
        lblMugshot.setPreferredSize(new Dimension(150, 150));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 7;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.fill = GridBagConstraints.BOTH;
        gridBagConstraints.ipadx = 8;
        gridBagConstraints.anchor = GridBagConstraints.WEST;
        gridBagConstraints.weighty = 0.04;
        gridBagConstraints.insets = new Insets(0, 2, 0, 16);
        pnlPersonalInformation.add(lblMugshot, gridBagConstraints);

        lblRace.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblRace.setText("Age");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.weightx = 0.01;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(lblRace, gridBagConstraints);

        txtAge.setText(" ");
        txtAge.setPreferredSize(new Dimension(90, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.weightx = 0.01;
        pnlPersonalInformation.add(txtAge, gridBagConstraints);

        lblOrigin.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblOrigin.setText("Origin");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.weightx = 0.01;
        gridBagConstraints.weighty = 0.04;
        pnlPersonalInformation.add(lblOrigin, gridBagConstraints);

        txtOrigin.setPreferredSize(new Dimension(90, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.weightx = 0.01;
        pnlPersonalInformation.add(txtOrigin, gridBagConstraints);

        lblHieght.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblHieght.setHorizontalAlignment(SwingConstants.TRAILING);
        lblHieght.setText("Height");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 1;
        pnlPersonalInformation.add(lblHieght, gridBagConstraints);

        txtHeight.setPreferredSize(new Dimension(60, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 1;
        pnlPersonalInformation.add(txtHeight, gridBagConstraints);

        btnUploadMugshot.setFont(new Font("Tahoma", 3, 12)); // NOI18N
        btnUploadMugshot.setText("Upload Mugshot");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 7;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.anchor = GridBagConstraints.NORTH;
        gridBagConstraints.insets = new Insets(5, 0, 0, 23);
        pnlPersonalInformation.add(btnUploadMugshot,
gridBagConstraints);
           btnUploadMugshot.addActionListener(this);
           chooser.setCurrentDirectory(new File("."));


        pnlComponents.add(pnlPersonalInformation);


pnlPrisonalInformation.setBorder(javax.swing.BorderFactory.createTitle
dBorder(javax.swing.BorderFactory.createTitledBorder(null, "Prisonal
Information", javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, new
java.awt.Font("Agency FB", 2, 24), new java.awt.Color(72, 72, 74))));
// NOI18N
        pnlPrisonalInformation.setPreferredSize(new
java.awt.Dimension(60, 20));
        pnlPrisonalInformation.setLayout(new
java.awt.GridBagLayout());

        lblId.setFont(new java.awt.Font("Tahoma", 0, 12)); // NOI18N

lblId.setHorizontalAlignment(javax.swing.SwingConstants.RIGHT);
        lblId.setText("ID");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_END;
        pnlPrisonalInformation.add(lblId, gridBagConstraints);

        txtId.setMinimumSize(new java.awt.Dimension(90, 20));
        txtId.setPreferredSize(new java.awt.Dimension(60, 20));
           txtId.setEditable(false);
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_START;
        gridBagConstraints.weightx = 0.1;
        pnlPrisonalInformation.add(txtId, gridBagConstraints);

        lblBookingDate.setFont(new java.awt.Font("Tahoma", 0, 12)); //
NOI18N
        lblBookingDate.setText("Booking Date");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 1;
        pnlPrisonalInformation.add(lblBookingDate,
gridBagConstraints);

        lblReleaseDate.setFont(new java.awt.Font("Tahoma", 0, 12)); //
NOI18N
```

```
lblReleaseDate.setHorizontalAlignment(javax.swing.SwingConstants.TRAIL
ING);
        lblReleaseDate.setText("Release Date");
        lblReleaseDate.setPreferredSize(new java.awt.Dimension(80,
20));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 2;
        pnlPrisonalInformation.add(lblReleaseDate,
gridBagConstraints);

        lblDaysToIncarcerate.setFont(new java.awt.Font("Tahoma", 0,
12)); // NOI18N

lblDaysToIncarcerate.setHorizontalAlignment(javax.swing.SwingConstants
.TRAILING);
        lblDaysToIncarcerate.setText("Day to stay Incarcerate");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 6;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_END;
        pnlPrisonalInformation.add(lblDaysToIncarcerate,
gridBagConstraints);

        txtCountToIncarcerate.setPreferredSize(new
java.awt.Dimension(60, 20));
           txtCountToIncarcerate.setEditable(false);
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 7;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_START;
        gridBagConstraints.weightx = 0.1;
        pnlPrisonalInformation.add(txtCountToIncarcerate,
gridBagConstraints);

        lblStatus.setFont(new java.awt.Font("Tahoma", 0, 12)); //
NOI18N

lblStatus.setHorizontalAlignment(javax.swing.SwingConstants.TRAILING);
        lblStatus.setText("Status");

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_END;
        gridBagConstraints.weightx = 0.9;
        gridBagConstraints.weighty = 1.3;
        pnlPrisonalInformation.add(lblStatus, gridBagConstraints);
        txtStatus.setPreferredSize(new java.awt.Dimension(90, 20));
           txtStatus.setEditable(false);
           txtStatus.setText("Incarcerated");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_START;
        gridBagConstraints.weightx = 0.1;
        gridBagConstraints.weighty = 1.2;
        pnlPrisonalInformation.add(txtStatus, gridBagConstraints);

        lblSecurityClass.setFont(new java.awt.Font("Tahoma", 0, 12));
// NOI18N
        lblSecurityClass.setText("Security Class");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_END;
        gridBagConstraints.weightx = 0.2;
        gridBagConstraints.weighty = 1.2;
        pnlPrisonalInformation.add(lblSecurityClass,
gridBagConstraints);

           cbxSecurityClass.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Low Risk", "High
Risk" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LINE_START;
        gridBagConstraints.weightx = 0.1;
        gridBagConstraints.weighty = 1.2;
        pnlPrisonalInformation.add(cbxSecurityClass,
gridBagConstraints);

        lblCrimeDescription.setFont(new java.awt.Font("Tahoma", 0,
12)); // NOI18N
        lblCrimeDescription.setText("Crime Description");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.FIRST_LINE_END;
        gridBagConstraints.weighty = 1.2;
        pnlPrisonalInformation.add(lblCrimeDescription,
gridBagConstraints);
```

```
scrollCrimeDescription.setHorizontalScrollBarPolicy(javax.swing.Scroll
PaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        scrollCrimeDescription.setAutoscrolls(true);
        scrollCrimeDescription.setHorizontalScrollBar(null);
        scrollCrimeDescription.setPreferredSize(new
java.awt.Dimension(250, 80));

        textCrimeDescription.setColumns(20);
        textCrimeDescription.setRows(5);
        scrollCrimeDescription.setViewportView(textCrimeDescription);

        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.gridwidth = 4;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.PAGE_END;
        pnlPrisonalInformation.add(scrollCrimeDescription,
gridBagConstraints);

        cbxBookDay.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Day", "01", "02", "03", "04", "05", "06", "07",
"08", "09", "10", "11", "12", "13", "14", "15", "16", "17", "18",
"19", "20", "21", "22", "23", "24", "25", "26", "27", "28", "29",
"30", "31", "" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 1;
        pnlPrisonalInformation.add(cbxBookDay, gridBagConstraints);

        cbxBookMonth.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Month", "January", "Febuary", "March", "April",
"May", "June", "July", "August", "September", "October", "November",
"December", "" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 1;
        pnlPrisonalInformation.add(cbxBookMonth, gridBagConstraints);

        cbxBookYear.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "Select Year", "2011", "2012", "2013", "2014", "2015",
"2016", "2017", "2018", "2019", "2020", "2021", "2022", "2023",
"2024", "2025", "" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 1;
        pnlPrisonalInformation.add(cbxBookYear, gridBagConstraints);

        cbxReleaseDay.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Select Day", "01",
"02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12",
"13", "14", "15", "16", "17", "18", "19", "20", "21", "22", "23",
"24", "25", "26", "27", "28", "29", "30", "31", " " }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 2;
        pnlPrisonalInformation.add(cbxReleaseDay, gridBagConstraints);

        cbxReleaseMonth.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Select Month",
"January", "Febuary", "March", "April", "May", "June", "July",
"August", "September", "October", "November", "December", "" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 2;
        pnlPrisonalInformation.add(cbxReleaseMonth,
gridBagConstraints);

        cbxReleaseYear.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Select Year", "2011",
"2012", "2013", "2014", "2015", "2016", "2017", "2018", "2019",
"2020", "2021", "2022", "2023", "2024", "2025", "" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 2;
        pnlPrisonalInformation.add(cbxReleaseYear,
gridBagConstraints);

           lblSelectWarden.setText("Select Warden");
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 5;
        gridBagConstraints.gridy = 2;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LAST_LINE_END;
        pnlPrisonalInformation.add(lblSelectWarden,
gridBagConstraints);

        cbxSelectWarden.setModel(new
javax.swing.DefaultComboBoxModel(new String[] { "Warden ID" }));
        gridBagConstraints = new java.awt.GridBagConstraints();
        gridBagConstraints.gridx = 6;
        gridBagConstraints.gridy = 2;
        gridBagConstraints.anchor =
java.awt.GridBagConstraints.LAST_LINE_START;
        pnlPrisonalInformation.add(cbxSelectWarden,
gridBagConstraints);


        pnlComponents.add(pnlPrisonalInformation);

        getContentPane().add(pnlComponents,
java.awt.BorderLayout.CENTER);
        lblHeader.setFont(new Font("Tahoma", 3, 24)); // NOI18N
        lblHeader.setForeground(new Color(72, 72, 74));
        lblHeader.setHorizontalAlignment(SwingConstants.CENTER);
        lblHeader.setText(header);
        lblHeader.setAlignmentY(5.0F);
        lblHeader.setPreferredSize(new Dimension(500, 100));
        getContentPane().add(lblHeader, BorderLayout.NORTH);

        pnlButtons.setLayout(new FlowLayout(FlowLayout.CENTER, 5,
20));

        btnAdd.setText("Submit");
           btnAdd.addActionListener(this);
        pnlButtons.add(btnAdd);

        btnUpdate.setText("Update");
        pnlButtons.add(btnUpdate);
           btnUpdate.addActionListener(this);
           btnUpdate.setEnabled(false);

           btnRelease.setText("Release");
        pnlButtons.add(btnRelease);
           btnRelease.addActionListener(this);
           btnRelease.setEnabled(false);

        btnDelete.setText("Delete");
        pnlButtons.add(btnDelete);
           btnDelete.addActionListener(this);
           btnDelete.setEnabled(false);

        btnClear.setText("Clear");
           btnClear.addActionListener(this);
        pnlButtons.add(btnClear);

        btnCancel.setText("Cancel");
           btnCancel.addActionListener(this);
        pnlButtons.add(btnCancel);

        getContentPane().add(pnlButtons, BorderLayout.PAGE_END);

        pack();
           try {
            for (UIManager.LookAndFeelInfo info :
UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
```

```
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        }



    }
     /**
     * method that works is called internally when any Button is
pressed
     * and then it call method asociated with button pressed.
     */
       public void actionPerformed(ActionEvent e){
          firstName=txtFirstName.getText();
          lastName=txtLastName.getText();
             if (rbnMale.isSelected())
             {
                  sex=rbnMale.getText();
             }
             else
             sex=rbnFemale.getText();
             origin=txtOrigin.getText();
             age=txtAge.getText();
             height=txtHeight.getText();
             status=txtStatus.getText();
             securityClass=(String)cbxSecurityClass.getSelectedItem();
              crimeDescription=textCrimeDescription.getText();

bookDate=cbxBookDay.getSelectedItem()+"/"+cbxBookMonth.getSelectedItem
()+"/"+cbxBookYear.getSelectedItem();

releaseDate=(String)cbxReleaseDay.getSelectedItem()+"/"+(String)cbxRel
easeMonth.getSelectedItem()+"/"+(String)cbxReleaseYear.getSelectedItem
();
            if(cbxSelectWarden.getSelectedIndex()!=0)

allocatedWardenId=Integer.parseInt(((String)cbxSelectWarden.getSelecte
dItem()).substring(0,2));
                 boolean flag=true;
               if(e.getSource()==btnUploadMugshot){
                    uploadMugshot();
                 }
              if (e.getSource()==btnClear){
                   clear();
                   flag=false;
              }
              if(e.getSource()==btnCancel){
                   cancel();
                   flag=false;
              }
                      if(e.getSource()==btnAdd&&
checkFieldValidation())
                            add();
                      if (e.getSource()==btnUpdate&&
checkFieldValidation())
                            update();
                      if(e.getSource()==btnDelete)
                         delete();
                      if (e.getSource()==btnRelease)
                            release();
      }

       private void add(){
        try{
                status="Incarcerated";

     securityClass=(String)cbxSecurityClass.getSelectedItem();
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in add() method at Prisoner Page");
           Statement state = connect.createStatement();
           String sql = "insert into Prisoner_Details
(First_Name,Last_Name,Sex,Age,Origin,Height,Status,Security_Class,Book
_Date,Release_Date,Crime_Description,Mugshot,Allocated_Warden_Id)
values ('" + firstName + "','" + lastName + "','" + sex + "','" + age
+ "','" + origin + "','" + height + "','" + status + "','" +
securityClass + "','" + bookDate + "','" + releaseDate + "','" +
crimeDescription + "','" + mugshotPath + "','" + allocatedWardenId +
"')";
           state.execute(sql);
              JOptionPane.showMessageDialog(null, "Prisoner added
Successfully ");

              sql = "select * from Warden_Details where ID= "+
allocatedWardenId +"";
           ResultSet rs=state.executeQuery(sql);
              if(rs.next()){
              int allocationStatus=rs.getInt("Allocation_Status");
              allocationStatus++;
              sql = "UPDATE Warden_Details set Allocation_Status= '" +
allocationStatus + "' WHERE ID= "+allocatedWardenId+"";
              state.executeUpdate(sql);
              System.out.println("Statement executed successfully ");
```

```
                }
            state.close();
            connect.close();
            }
        catch (SQLException e) {
                System.out.println("Exception in add method at
prisoner page"+e);

        }
            clear();
            dispose();
            adminPage.setVisible(true);
      }
        private void update(){
                id=Integer.parseInt(txtId.getText());
        try{
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                System.out.println("Divers loaded successfully in
update() at prisoner");

            Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
                System.out.println("Connection established successfully
in update of prisoner");
            Statement state = connect.createStatement();
            String sql = "UPDATE Prisoner_Details set First_Name= '" +
firstName + "', Last_Name= '" + lastName + "', Sex= '" + sex + "',
Age= '" + age + "', Origin= '" + origin + "', Height= '" + height +
"', Status= '" + status + "', Security_Class= '" + securityClass +
"',Book_Date= '" + bookDate + "', Release_Date= '" + releaseDate + "',
Days_Count= '" + daysCount + "', Crime_Description= '" +
crimeDescription + "', Mugshot= '" + mugshotPath + "' WHERE ID=
"+id+"";
                state.executeUpdate(sql);
                JOptionPane.showMessageDialog(null, "Record Updated
Successfully");
                System.out.println("Statement executed successfully of
update at prisoner");
            state.close();
            connect.close();
            }
        catch (Exception e) {
                System.out.println("Exception in update method at
Prisoner page"+e);
        }
            clear();
            dispose();
            adminPage.setVisible(true);
      }
    public void find(ResultSet rs){
            adminPage.setVisible(false);
        btnAdd.setEnabled(false);
        btnDelete.setEnabled(true);
        btnUpdate.setEnabled(true);
        try{
            txtId.setText(rs.getString("ID"));
            txtFirstName.setText(rs.getString("First_Name"));
            txtHeight.setText(rs.getString("Height"));
            lblMugshot.setText(null);
            lblMugshot.setIcon(new
ImageIcon((String)rs.getObject("Mugshot")));

            txtLastName.setText(rs.getString("Last_Name"));
            txtOrigin.setText(rs.getString("Origin"));
            txtAge.setText(rs.getString("Age"));

       cbxSecurityClass.setSelectedItem(rs.getString("Security_Class"));

            txtStatus.setText(rs.getString("Status"));
            bookDate=rs.getString("Book_Date");
            releaseDate=rs.getString("Release_Date");
            if(bookDate.length()>6){
            bookDay=bookDate.substring(0,2);
            bookMonth=bookDate.substring(3,bookDate.length()-5);
            bookYear=bookDate.substring(bookDate.length()-
4,bookDate.length());
            releaseDay=releaseDate.substring(0,2);
            releaseMonth=releaseDate.substring(3,releaseDate.length()-
5);
            releaseYear=releaseDate.substring(releaseDate.length()-
4,releaseDate.length());
            }

       textCrimeDescription.setText(rs.getString("Crime_Description"));
            cbxBookDay.setSelectedItem(bookDay);
            cbxBookMonth.setSelectedItem(bookMonth);
            cbxBookYear.setSelectedItem(bookYear);
            cbxReleaseDay.setSelectedItem(releaseDay);

            cbxReleaseMonth.setSelectedItem(releaseMonth);

            cbxReleaseYear.setSelectedItem(releaseYear);

            int hold=(int)countIncarcerateDays();
             txtCountToIncarcerate.setText(""+hold+"");
              holdId=rs.getInt("Allocated_Warden_Id");
             rs.close();

             connect = DriverManager.getConnection("jdbc:odbc:PMS");
             state=connect.createStatement();
             String sql="select * from Warden_Details where ID= "+
holdId +"";
             ResultSet wd=state.executeQuery(sql);
```

```
            if(wd.next())
            cbxSelectWarden.setSelectedItem(""+holdId+" "+"Rank-
"+wd.getString("Rank"));
            if (hold<7)
                 btnRelease.setEnabled(true);
        }catch (Exception e) {
                 System.out.println("Exception in find method at
prisoner page"+e);
                 adminPage.setVisible(true);
           }
           setVisible(true);
     }

     private void delete(){
           status="Deleted";
          id=Integer.parseInt(txtId.getText());
        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully in delete
at prisoner");

           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfull
in delete at prisoner");
           Statement state = connect.createStatement();
           String sql = "UPDATE Prisoner_Details set Status= '"+
"Deleted" +"' WHERE ID= "+ id +"";

           state.execute(sql);
              JOptionPane.showMessageDialog(null, "Record deleted
Successfully");
              System.out.println("Statement executed successfully in
delete at Prisoner");

              sql = "select * from Warden_Details where ID= "+
allocatedWardenId +"";
           ResultSet rs=state.executeQuery(sql);
              if(rs.next()){
              int allocationStatus=rs.getInt("Allocation_Status");
              allocationStatus--;
              sql = "UPDATE Warden_Details set Allocation_Status= '" +
allocationStatus + "' WHERE ID= "+allocatedWardenId+"";
              state.executeUpdate(sql);
              System.out.println("Statement executed successfully ");
           state.close();
           connect.close();
              clear();
              }
           }
        catch (Exception e) {
                System.out.println("Exception in add method at 
prisoner page"+e);
                }
           dispose();
           adminPage.setVisible(true);
      }
     private void clear(){

           txtId.setText("");
           txtFirstName.setText("");
           txtHeight.setText("");

           txtLastName.setText("");
           txtOrigin.setText("");
           txtAge.setText("");
           txtCountToIncarcerate.setText("");
           txtStatus.setText("");

           textCrimeDescription.setText("");
           cbxBookDay.setSelectedIndex(0);
           cbxBookMonth.setSelectedIndex(0);
           cbxBookYear.setSelectedIndex(0);
           cbxReleaseDay.setSelectedIndex(0);
           cbxReleaseMonth.setSelectedIndex(0);
           cbxReleaseYear.setSelectedIndex(0);
           lblMugshot.setIcon(null);
           lblMugshot.setText("Upload Mugshot here");
           cbxSelectWarden.setSelectedIndex(0);

        }
     private void uploadMugshot(){
            try{
                           int result = chooser.showOpenDialog(null);
               if (result == JFileChooser.APPROVE_OPTION)
                mugshotPath = chooser.getSelectedFile().getPath();
                      lblMugshot.setText(null);
                      lblMugshot.setIcon(new ImageIcon(mugshotPath));
                }catch(Exception e){
                      System.out.println("Exception in uploadMugshot 
method at Admin page"+e);
                }
    }
     private void cancel(){
           adminPage.setVisible(true);
           this.dispose();
      }
     private void release(){
           status="Released";
         id=Integer.parseInt(txtId.getText());
        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in find() at prisoner");
           Statement state = connect.createStatement();
           String sql = "UPDATE Prisoner_Details set Status= '"+
"Relesed" +"' WHERE ID= "+ id +"";
           state.execute(sql);
              JOptionPane.showMessageDialog(null, "prisoner released
Successfully ");
              System.out.println("Statement executed successfully of
find() at prisoner");
           state.close();
           connect.close();
              clear();
              dispose();
           }
        catch (Exception e) {
                System.out.println("Exception in find() method at
prisoner page"+e);
                dispose();
        }
      }
       private void loadWarden(){
           int count=1;
           try{
             Statement  stmt = connect.createStatement();
             ResultSet rsWarden = stmt.executeQuery("Select * from
Warden_Details where Status= '" + "Working" + "'");
             System.out.println("statement executed in loadwarden at
prisoner");
             while(rsWarden.next()){
                 if(rsWarden.getInt("Allocation_Status")<5){

     cbxSelectWarden.insertItemAt(rsWarden.getString("ID")+" "+"Rank-
"+rsWarden.getString("Rank"),count);
                 count++;
                       }
                 }
                 System.out.println("statement finish execution in
loadwarden at prisoner");

           stmt.close();
        connect.close();
       }catch(Exception e){
                 System.out.println("Exception in loadWarden method at
prisoner page"+e);
           }
       }
       private boolean checkFieldValidation(){
             boolean flag=false;
             if(txtFirstName.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the First 
Name field");
             else if(txtLastName.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the Last 
Name Field");
             else if(txtAge.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the Age 
Field");
             else if(txtOrigin.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the Origin 
Field");
             else if(txtHeight.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the Height 
Field");
             else if(textCrimeDescription.getText().equals(""))
                 JOptionPane.showMessageDialog(null, "Fill the Crime 
Discription Field");
             else if(cbxBookDay.getSelectedIndex()==0 || 
cbxBookMonth.getSelectedIndex()==0 || 
cbxBookYear.getSelectedIndex()==0)
                 JOptionPane.showMessageDialog(null, "Please Selecte 
the All feilds of Booking Date Coulmn");
             else if(cbxReleaseDay.getSelectedIndex()==0 
||cbxReleaseMonth.getSelectedIndex()==0 
||cbxReleaseYear.getSelectedIndex()==0)
                 JOptionPane.showMessageDialog(null, "Please Selecte 
the All feilds of Release Date Coulmn");
             else if(cbxSelectWarden.getSelectedIndex()==0)
                 JOptionPane.showMessageDialog(null, "Please Selecte a 
warden from Select Warden Coulmn");
             else if (mugshotPath=="")
                 JOptionPane.showMessageDialog(null, "Please Selecte a 
mugshot from Upload Mugshot Coulmn");
             else if(checkDateValidation())
                   return true;

             return flag;
        }
        double countIncarcerateDays(){
            int day,month,year;
            day=cbxReleaseDay.getSelectedIndex();
            month=cbxReleaseMonth.getSelectedIndex();

      year=Integer.parseInt((String)cbxReleaseYear.getSelectedItem());
          Calendar dt1= Calendar.getInstance();
            Calendar dt2= Calendar.getInstance();
            dt2.set(year,month,day);
            double 
count1=dt2.getTimeInMillis(),count2=dt1.getTimeInMillis(),count;
            count1=count1/1000/60/60/24;
            count2=count2/1000/60/60/24;
```

```
			count=(int)count1-count2;
			return count;
			}
		boolean checkDateValidation(){
			boolean flag=false;
			if(
cbxBookDay.getSelectedIndex()==31&&(cbxBookMonth.getSelectedIndex()==4
||cbxBookMonth.getSelectedIndex()==6
||cbxBookMonth.getSelectedIndex()==9
||cbxBookMonth.getSelectedIndex()==11) )
			 JOptionPane.showMessageDialog(null, "Selected Booking Date
is an Invalid Date.Please Select day less then 31");
			else if(cbxBookMonth.getSelectedIndex()==2){
				int
year=Integer.parseInt((String)cbxBookYear.getSelectedItem());
				if(year%4==0&&cbxBookDay.getSelectedIndex()>29)
				JOptionPane.showMessageDialog(null, "Selected Booking
Date is an Invalid Date.Please Select day less then 29");
				else
if(!(year%4==0)&&cbxBookDay.getSelectedIndex()>28)
					JOptionPane.showMessageDialog(null, "Selected
Booking Date is an Invalid Date.Please Select day less then 28");
				else
					flag=true;
			}
			else if(
cbxReleaseDay.getSelectedIndex()==31&&(cbxReleaseMonth.getSelectedInde
x()==4 ||cbxReleaseMonth.getSelectedIndex()==6
||cbxReleaseMonth.getSelectedIndex()==9
||cbxReleaseMonth.getSelectedIndex()==11) )
			 JOptionPane.showMessageDialog(null, "Selected Release Date
is an Invalid Date.Please Select day less then 31");
			else if(cbxReleaseMonth.getSelectedIndex()==2){
				int
year=Integer.parseInt((String)cbxReleaseYear.getSelectedItem());
				if(year%4==0 && cbxReleaseDay.getSelectedIndex()>29)
				JOptionPane.showMessageDialog(null, "Selected Release
Date is an Invalid Date.Please Select day less then 29");
				else
if(!(year%4==0)&&cbxReleaseDay.getSelectedIndex()>28)
					JOptionPane.showMessageDialog(null, "Selected
Release Date is an Invalid Date.Please Select day less then 28");
				else
					flag=true;
			}
			else
				flag=true;
			return flag;
		}
		private void loadId(){
			try{
			Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
             System.out.println("Divers loaded successfully");
          Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
             System.out.println("Connection established successfully
in loadId() at Prisoner");
          Statement state =
connect.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE
,ResultSet.CONCUR_READ_ONLY);


               String sql = "select ID from Prisoner_Details ";
            ResultSet rs=state.executeQuery(sql);
                          int count=0;

                      if(rs.last()){
                           System.out.println("here check the last");
                      int hold=Integer.parseInt(rs.getString("ID"));
                      hold++;
                      txtId.setText(hold+"");
               }
             System.out.println("Statement executed successfully at
loadID() at prisoner");
          state.close();
          connect.close();
          }catch (Exception e) {
               System.out.println("Exception in loadID method at
prisoner page"+e);
          }
     }
}
```

Warden.java Source Code

```
//package pms.warden;
// import pms.admin.*;
 import java.awt.event.*;
 import javax.swing.*;
 import java.awt.*;
 import javax.swing.border.TitledBorder;
 import javax.swing.table.*;
 import java.sql.*;
 import java.io.*;
 /**
 * Class {@ cade Warden } represents a warden and all its attributes.
 *this page provides following functionalities :-
 * <blockquote><pre>
 *      Add a Warden.
 *         Find a Warden.
 *         Delete a Warden.
 *         Update Warden's Details.
 *         Genrate Reports of all Warden.
 * </pre></blockquote>
 */

public class Warden extends JFrame implements
ActionListener,KeyListener {
    private JButton btnAdd, btnCancel,btnClear,
btnUploadMugshot,btnDelete, btnUpdate;
    private ButtonGroup gpSex;
    private JScrollPane jScrollPane1;
    private JLabel
lblFirstName,lblHeader,lblHieght,lblId,lblAllocationnStatus,lblLastNam
e;
     private JLabel
lblMugshot,lblOrigin,lblPrisonersTable,lblStatus,lblRank,lblSex;
    private JPanel
pnlButtons,pnlComponents,pnlPersonalInformation,pnlPrisonalInformation
,pnlTable;
    private JRadioButton rbnFemale,rbnMale;
    private JTable tblUndertakings;
    private JTextField
txtFirstName,txtHeight,txtId,txtAllocationStatus,txtLastName,txtOrigin
,txtStatus,txtRank;
     private String
firstName,lastName,sex,age,status,height,mugshotPath,origin,rank;
     private int id,allocationStatus=0;
     private JFileChooser chooser;
     private String header;
     private AdminPage adminPage;

    /**
     * Creates new form Warden
     */
    public Warden(AdminPage adminPage,String str){
           header=str;
           this.adminPage=adminPage;

        gpSex = new ButtonGroup();
        pnlComponents = new JPanel();
        pnlPersonalInformation = new JPanel();
        lblFirstName = new JLabel();
        txtFirstName = new JTextField();
        lblLastName = new JLabel();
        txtLastName = new JTextField();
        lblSex = new JLabel();
        rbnMale = new JRadioButton();
        rbnFemale = new JRadioButton();
        lblMugshot = new JLabel();
        lblStatus = new JLabel();
        txtStatus = new JTextField();
        lblOrigin = new JLabel();
        txtOrigin = new JTextField();
        lblHieght = new JLabel();
        txtHeight = new JTextField();
        btnUploadMugshot = new JButton();
        pnlPrisonalInformation = new JPanel();
        lblId = new JLabel();
        txtId = new JTextField();
        lblAllocationnStatus = new JLabel();
        txtAllocationStatus = new JTextField();
        lblRank = new JLabel();
        txtRank = new JTextField();
        pnlTable = new JPanel();
        jScrollPane1 = new JScrollPane();
        tblUndertakings = new JTable();
        lblPrisonersTable = new JLabel();
        lblHeader = new JLabel();
        pnlButtons = new JPanel();
        btnAdd = new JButton();
        btnUpdate = new JButton();
        btnDelete = new JButton();
        btnClear = new JButton();
        btnCancel = new JButton();
           chooser = new JFileChooser();
        initComponents();
    }
    private void initComponents() {
       GridBagConstraints gridBagConstraints;

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setPreferredSize(new Dimension(800, 660));

        pnlComponents.setLayout(new GridLayout(2, 1));


pnlPersonalInformation.setBorder(BorderFactory.createTitledBorder(null
, "Personal Information", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, new Font("Agency FB", 3, 24), new
Color(72, 72, 74))); // NOI18N
           |
                            |
                Inb/n Section is Common to Prisoner Page
                            |
                            |
-----------------------------------------------------------------------

pnlPrisonalInformation.setBorder(BorderFactory.createTitledBorder(null
, "Prisonal Information", TitledBorder.DEFAULT_JUSTIFICATION,
TitledBorder.DEFAULT_POSITION, new Font("Agency FB", 3, 24), new
Color(72, 72, 74))); // NOI18N
        pnlPrisonalInformation.setPreferredSize(new Dimension(60,
20));
        pnlPrisonalInformation.setLayout(new GridBagLayout());

        lblId.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblId.setHorizontalAlignment(SwingConstants.RIGHT);
        lblId.setText(" ID");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        pnlPrisonalInformation.add(lblId, gridBagConstraints);

        txtId.setMinimumSize(new Dimension(60, 20));
        txtId.setPreferredSize(new Dimension(60, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.insets = new Insets(0, 13, 0, 13);
        pnlPrisonalInformation.add(txtId, gridBagConstraints);

        lblAllocationnStatus.setFont(new Font("Tahoma", 0, 12)); //
NOI18N
        lblAllocationnStatus.setText("Allocation Stattus");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 1;
        pnlPrisonalInformation.add(lblAllocationnStatus,
gridBagConstraints);

        txtAllocationStatus.setPreferredSize(new Dimension(60, 20));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 1;
        gridBagConstraints.insets = new Insets(0, 13, 0, 13);
        pnlPrisonalInformation.add(txtAllocationStatus,
gridBagConstraints);

        lblRank.setFont(new Font("Tahoma", 0, 12)); // NOI18N
        lblRank.setText("Rank");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 2;
        pnlPrisonalInformation.add(lblRank, gridBagConstraints);

        txtRank.setPreferredSize(new Dimension(60, 20));
           txtRank.addKeyListener(this);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 2;
        gridBagConstraints.insets = new Insets(0, 13, 0, 13);
        pnlPrisonalInformation.add(txtRank, gridBagConstraints);

        pnlTable.setBackground(new Color(72, 72, 74));
        pnlTable.setForeground(new Color(255, 255, 255));
        pnlTable.setMinimumSize(new Dimension(0, 0));
        pnlTable.setPreferredSize(new Dimension(500, 130));
        pnlTable.setLayout(new BorderLayout());
        tblUndertakings.setModel(new DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Serial No.", "Name", "Prisoner ID", "Security Class"
            }
        ));
        jScrollPane1.setViewportView(tblUndertakings);

        pnlTable.add(jScrollPane1, BorderLayout.CENTER);

        lblPrisonersTable.setFont(new Font("Tahoma", 0, 12)); //
NOI18N
        lblPrisonersTable.setForeground(new Color(255, 255, 255));
        lblPrisonersTable.setText("Undertaking Prisoners");
        lblPrisonersTable.setPreferredSize(new Dimension(105, 32));
        pnlTable.add(lblPrisonersTable, BorderLayout.PAGE_START);

        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.gridwidth = 4;
        gridBagConstraints.gridheight = 3;
        pnlPrisonalInformation.add(pnlTable, gridBagConstraints);

        pnlComponents.add(pnlPrisonalInformation);

        getContentPane().add(pnlComponents, BorderLayout.CENTER);

        lblHeader.setFont(new Font("Tahoma", 3, 24)); // NOI18N
        lblHeader.setForeground(new Color(72, 72, 74));
        lblHeader.setHorizontalAlignment(SwingConstants.CENTER);
        lblHeader.setText("Add a Warden");
        lblHeader.setAlignmentY(5.0F);
        lblHeader.setPreferredSize(new Dimension(500, 100));
        getContentPane().add(lblHeader, BorderLayout.NORTH);

        pnlButtons.setLayout(new FlowLayout(FlowLayout.CENTER, 5,
20));

        btnAdd.setText("Submit");
           btnAdd.addActionListener(this);
        pnlButtons.add(btnAdd);

        btnUpdate.setText("Update");
        pnlButtons.add(btnUpdate);
```

```
           btnUpdate.addActionListener(this);
        btnDelete.setText("Delete");
        pnlButtons.add(btnDelete);
           btnDelete.addActionListener(this);

        btnClear.setText("Clear");
           btnClear.addActionListener(this);
        pnlButtons.add(btnClear);

        btnCancel.setText("Cancel");
           btnCancel.addActionListener(this);
        pnlButtons.add(btnCancel);

        getContentPane().add(pnlButtons, BorderLayout.PAGE_END);

             txtStatus.setEditable(false);
             txtId.setEditable(false);
             txtAllocationStatus.setEditable(false);
             btnUpdate.setEnabled(false);
             btnDelete.setEnabled(false);
        pack();
           loadId();
    }
      /**
      * method that works is called internally when any Button is
pressed
      * and then it call method asociated with button pressed.
      */
      public void actionPerformed(ActionEvent e){
          firstName=txtFirstName.getText();
          lastName=txtLastName.getText();
             if(!(txtId.getText().equals("")))
             id=Integer.parseInt(txtId.getText());
             if (rbnMale.isSelected())
             {
                  sex=rbnMale.getText();
             }
             else
             sex=rbnFemale.getText();
             origin=txtOrigin.getText();
             height=txtHeight.getText();
             status="Working";
             rank=txtRank.getText();
                boolean flag=true;
              if(e.getSource()==btnUploadMugshot){
                   uploadMugshot();
                   flag=false;
                 }
              if (e.getSource()==btnClear){
                   clear();
                   flag=false;
              }
              if(e.getSource()==btnCancel){
                   cancel();
                   flag=false;
              }
              if(flag && checkFieldValidation()){
                     if(e.getSource()==btnAdd)
                          add();
                     if (e.getSource()==btnUpdate)
                          update();
                     if(e.getSource()==btnDelete)
                        delete();
              }
      }

       private void add(){

        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  //DSN
oriented Connection
              System.out.println("Divers loaded successfully in add()
method at warden");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in add at warden");
           Statement state = connect.createStatement();
           String sql = "insert into Warden_Details
(First_Name,Last_Name,Sex,Origin,Height,Rank,Status,Allocation_Status,
Mugshot) values ('" + firstName + "','" + lastName + "','" + sex +
"','" + origin + "','" + height + "','" + rank + "','" + status + "',"
+ allocationStatus + ",'" + mugshotPath + "')";
           state.execute(sql);
              JOptionPane.showMessageDialog(null, "Record added
Successfully ");
              System.out.println("Statement executed successfully in
add() at warden");
           state.close();
           connect.close();
              if(Integer.parseInt(rank)>4)
                   setPassword();
           }
        catch (Exception e) {
           System.out.print("error caught at add method inWarden"+e);
        }
           clear();
      }
       private void update(){
           id=Integer.parseInt(txtId.getText());
        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully in update
at warden");
```

```
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in update() at warden");
           Statement state = connect.createStatement();
           String sql = "UPDATE Warden_Details set First_Name= '" +
firstName + "', Last_Name= '" + lastName + "', Sex= '" + sex + "',
Origin= '" + origin + "', Height= '" + height + "', Rank= '" + rank +
"', Mugshot= '" + mugshotPath + "'  WHERE ID= "+id+"";
              state.executeUpdate(sql);
              JOptionPane.showMessageDialog(null, "Record Updated
Successfully");
              System.out.println("Statement executed successfully in
update at warden");
           state.close();
           connect.close();
           }
        catch (Exception e) {
           System.out.print("error caught at update method
inWarden"+e);
        }
           if(Integer.parseInt(rank)>4)
                setPassword();
           clear();
      }


    public void find(ResultSet rs){
         btnAdd.setEnabled(false);
           btnUpdate.setEnabled(true);
           btnDelete.setEnabled(true);
        try{
           txtId.setText(rs.getString("ID"));
           txtFirstName.setText(rs.getString("First_Name"));
           txtHeight.setText(rs.getString("Height"));

           txtLastName.setText(rs.getString("Last_Name"));
           txtOrigin.setText(rs.getString("Origin"));
           txtRank.setText(rs.getString("Rank"));

      txtAllocationStatus.setText(rs.getString("Allocation_Status"));
           txtStatus.setText(rs.getString("Status"));
           lblMugshot.setIcon(new ImageIcon(rs.getString("Mugshot")));
           loadPrisoners();
        }catch (Exception e) {
           System.out.print("error caught at find method inWarden"+e);
        }
           setVisible(true);
      }
      private void delete(){
           status="Deleted";
         id=Integer.parseInt(txtId.getText());
        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully in delete
at warden");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in delete at warden");
           Statement state = connect.createStatement();
           String sql = "UPDATE Warden_Details set Status= '"+
"Deleted" +"' WHERE ID= "+ id +"";
           state.execute(sql);
              JOptionPane.showMessageDialog(null, "Record deleted
Successfully indelete at warden");
              System.out.println("Statement executed successfully");
           state.close();
           connect.close();
              clear();
           }
        catch (Exception e) {
           System.out.print("error caught at delete() method
inWarden"+e);
        }
      }
     private void clear(){

           txtId.setText("");
           txtFirstName.setText("");
           txtHeight.setText("");

           txtLastName.setText("");
           txtOrigin.setText("");

           txtStatus.setText("");
           txtAllocationStatus.setText("");
           txtRank.setText("");
           rbnMale.setSelected(true);
           lblMugshot.setIcon(null);
           lblMugshot.setText("Upload Mugshot here");
           int count=0;
           while(tblUndertakings.getValueAt(count,0)!=null){
                tblUndertakings.setValueAt(null,count,0);
                tblUndertakings.setValueAt(null,count,1);
                tblUndertakings.setValueAt(null,count,2);
                tblUndertakings.setValueAt(null,count,3);
                count++;
           }
        }
     private void uploadMugshot(){
                int result = chooser.showOpenDialog(null);
                if (result == JFileChooser.APPROVE_OPTION)
                mugshotPath = chooser.getSelectedFile().getPath();
```

```
                      lblMugshot.setText(null);
                      lblMugshot.setIcon(new ImageIcon(mugshotPath));
                      }
    private void loadPrisoners(){
             try{
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
               System.out.println("Divers loaded successfully");
            Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
               System.out.println("Connection established successfully
in loadprisoners at warden");
            Statement state = connect.createStatement();
                      int
entity=(int)Integer.parseInt(txtId.getText());
                 String sql2 = "select * from Prisoner_Details where
Allocated_Warden_Id= "+ entity +" And Status= '" + "Incarcerated"
+"'";
             ResultSet prisoners=state.executeQuery(sql2);
                           int count=0;
             while(prisoners.next()){
                      tblUndertakings.setValueAt(count+1,count,0);

     tblUndertakings.setValueAt(prisoners.getString("First_Name"),coun
t,1);

     tblUndertakings.setValueAt(prisoners.getString("ID"),count,2);

     tblUndertakings.setValueAt(prisoners.getString("Security_Class"),
count,3);
                      count++;
               }
               System.out.println("Statement executed successfully in
laod prisoners at warden");
            state.close();
            connect.close();
            }catch (Exception e) {
            System.out.print("error caught at loadPrisoners method
inWarden"+e);
        }
     }
     public boolean checkFieldValidation(){
            boolean flag=false;
            if(txtFirstName.getText().equals(""))
               JOptionPane.showMessageDialog(null, "Fill the First
Name field");
            else if(txtLastName.getText().equals(""))
               JOptionPane.showMessageDialog(null, "Fill the Last
Name Field");
            else if(txtOrigin.getText().equals(""))
               JOptionPane.showMessageDialog(null, "Fill the Origin
Field");
            else if(txtHeight.getText().equals(""))
```

```
                JOptionPane.showMessageDialog(null, "Fill the Height
Field");
            else if(txtRank.getText().equals(""))
                JOptionPane.showMessageDialog(null, "Fill the Rank
Field properly");
            else if (mugshotPath=="")
                JOptionPane.showMessageDialog(null, "Please Selecte a
mugshot from Upload Mugshot Coulmn");

            else
                  flag=true;
            return flag;
        }
        private void cancel(){
            dispose();
            adminPage.setVisible(true);
        }
      private void setPassword(){

           String hold1 = JOptionPane.showInputDialog("Enter Your
Login Password");
           String hold2 = JOptionPane.showInputDialog("Confirm Login
Password");
           if(hold1.equals(hold2)){
                String password=hold1;
                     try{

                Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");  //DSN
oriented Connection
                   System.out.println("Divers loaded successfully in
setPassword at warden");

                  Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
                   System.out.println("Connection established
successfully  in setPassword at warden");

                  Statement state = connect.createStatement();
                  String sql = "insert into Login_Data (ID,Password)
values ('" + id + "','" + password + "')";
                  state.execute(sql);
                   JOptionPane.showMessageDialog(null, "Record added
Successfully");
                   System.out.println("Statement executed successfully
in setPassword at warden");
                  state.close();
                  connect.close();
                }catch (Exception e) {
           System.out.print("error caught at setPassword method
inWarden"+e);
               }
          }else{
               JOptionPane.showMessageDialog(null, "Password Check
Failed.");
                checkFieldValidation();
           }
      }
      private void loadId(){
                       try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
in loadId at warden");
           Statement state =
connect.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE
,ResultSet.CONCUR_READ_ONLY);
                String sql = "select ID from Warden_Details";
            ResultSet rs=state.executeQuery(sql);
                           int count=0;
            if(rs.next()){
                      rs.last();
                      int hold=rs.getInt("ID");
                      hold++;
                      txtId.setText(hold+"");
               }else
              JOptionPane.showMessageDialog(null, "Record Found
Successfully");
              System.out.println("Statement executed successfully   in
loadId at warden");
           state.close();
           connect.close();
           }catch (Exception e) {
           System.out.print("error caught at loadId method
inWarden"+e);
        }

      }
  public void keyTyped ( KeyEvent e ){ }

  public void keyPressed ( KeyEvent e){
      int hold=e.getKeyCode();
      if((hold>=49&& hold<=53)|| hold==8);
      else{
           JOptionPane.showMessageDialog(null, "Enter a valid Rank
Between 1-5.");
           txtRank.setText("");
      }
  }
  public void keyReleased ( KeyEvent e ){ }
}
```

Find Prisoner Page

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
//package pms.prisoner;

//import pms.admin.*;
import java.util.logging.Logger;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.sql.*;

/**
 * The {@code FindPrisoner} class represents Find Prisoner Page.
 * It provides has all components to find a prisoner.
 */
public class FindPrisoner extends JFrame implements
ActionListener,KeyListener{

      private JButton btnCancel,btnFind;
    private JLabel lblHeader,lblRadioButtonHeader, lblText;
    private JPanel pnlComponents;
    private JRadioButton rbnFirstName,rbnId;
    private JRadioButton rbnLastName;
    private JTextField txtEntity;
      private ButtonGroup gpFind;
      private int entity;
      private AdminPage adminPage;
      private String header;

    /**
     * Creates new form FindPrisoner
     */
    public FindPrisoner(AdminPage adminPage,String header) {
           this.adminPage=adminPage;
           this.header=header;

        lblHeader = new JLabel();
        pnlComponents = new JPanel();
        lblText = new JLabel();
        btnFind = new JButton();
        btnCancel = new JButton();
        rbnId = new JRadioButton();
        rbnFirstName = new JRadioButton();
        rbnLastName = new JRadioButton();
        lblRadioButtonHeader = new JLabel();
        txtEntity = new JTextField();
           gpFind=new ButtonGroup();
        initComponents();
    }

    private void initComponents() {
        GridBagConstraints gridBagConstraints;

        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        setMinimumSize(new Dimension(500, 69));
        setPreferredSize(new Dimension(600, 250));
        setResizable(false);

        lblHeader.setBackground(new Color(102, 102, 102));
        lblHeader.setFont(new Font("Tahoma", 1, 24)); // NOI18N
        lblHeader.setForeground(new Color(72, 72, 74));
        lblHeader.setHorizontalAlignment(SwingConstants.CENTER);
        lblHeader.setText(header);
        lblHeader.setPreferredSize(new Dimension(600, 80));
        getContentPane().add(lblHeader, BorderLayout.PAGE_START);

lblHeader.getAccessibleContext().setAccessibleName("lblHeader\n");

        pnlComponents.setLayout(new GridBagLayout());

        lblText.setFont(new Font("Tahoma", 1, 14)); // NOI18N
        lblText.setForeground(new Color(72, 72, 74));
        lblText.setText("Enter Prisoner's ID");
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 0;
        pnlComponents.add(lblText, gridBagConstraints);

        btnFind.setFont(new Font("Tahoma", 2, 14)); // NOI18N
        btnFind.setForeground(new Color(72, 72, 74));
        btnFind.setText("Find");
           btnFind.addActionListener(this);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 0;
        pnlComponents.add(btnFind, gridBagConstraints);

        btnCancel.setFont(new Font("Tahoma", 2, 14)); // NOI18N
        btnCancel.setForeground(new Color(72, 72, 74));
        btnCancel.setText("Cancel");
           btnCancel.addActionListener(this);

        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 4;
        gridBagConstraints.gridy = 2;
        pnlComponents.add(btnCancel, gridBagConstraints);

        gpFind.add(rbnId);
        rbnId.setFont(new Font("Tahoma", 2, 11)); // NOI18N
        rbnId.setForeground(new Color(72, 72, 74));
        rbnId.setText("Prisoner  ID");
           rbnId.setSelected(true);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 4;
        gridBagConstraints.gridwidth = 2;
        pnlComponents.add(rbnId, gridBagConstraints);

        gpFind.add(rbnFirstName);
        rbnFirstName.setFont(new Font("Tahoma", 2, 11)); // NOI18N
        rbnFirstName.setForeground(new Color(72, 72, 74));
        rbnFirstName.setText("Prison's First Name");
           rbnFirstName.setEnabled(false);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 1;
        gridBagConstraints.gridy = 4;
        gridBagConstraints.gridwidth = 2;
        pnlComponents.add(rbnFirstName, gridBagConstraints);

        gpFind.add(rbnLastName);
        rbnLastName.setFont(new Font("Tahoma", 2, 11)); // NOI18N
        rbnLastName.setForeground(new Color(72, 72, 74));
        rbnLastName.setText("Prison's Last Name");
           rbnLastName.setEnabled(false);
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 3;
        gridBagConstraints.gridy = 4;
        pnlComponents.add(rbnLastName, gridBagConstraints);

        lblRadioButtonHeader.setFont(new Font("Tahoma", 1, 14)); //
NOI18N
        lblRadioButtonHeader.setForeground(new Color(72, 72, 74));
        lblRadioButtonHeader.setText("Select Entity Type");
        lblRadioButtonHeader.setPreferredSize(new Dimension(150, 25));
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 0;
        gridBagConstraints.gridy = 3;
        gridBagConstraints.gridwidth = 3;
        pnlComponents.add(lblRadioButtonHeader, gridBagConstraints);


        txtEntity.setPreferredSize(new Dimension(120, 25));
           txtEntity.addKeyListener (this) ;
        gridBagConstraints = new GridBagConstraints();
        gridBagConstraints.gridx = 2;
        gridBagConstraints.gridy = 0;
        gridBagConstraints.gridwidth = 2;
        pnlComponents.add(txtEntity, gridBagConstraints);

        getContentPane().add(pnlComponents, BorderLayout.CENTER);
        pack();
           setVisible(true);
    }
      /**
      * method that works is called internally when any Button is
pressed
      * and then it call method asociated with button pressed.
      */
        public void actionPerformed(ActionEvent ae){
           if(ae.getSource()==btnFind){
                 System.out.println(!(txtEntity.getText().equals("")));
                if(!(txtEntity.getText().equals(""))){

      entity=(int)Integer.parseInt(txtEntity.getText());
                      Find(entity);
                 }
           }else
                   dispose();

       }

        private boolean Find(int entity){
             boolean flag=false;
                 entity=(int)Integer.parseInt(txtEntity.getText());

        try{
           Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
              System.out.println("Divers loaded successfully  find()
in findPrisoner");
           Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
              System.out.println("Connection established successfully
find() in findPrisoner");
           Statement state = connect.createStatement();
           String sql = "select * from Prisoner_Details where ID= "+
entity +"";
           ResultSet rs=state.executeQuery(sql);
              if(rs.next()){
                          Prisoner p=new
Prisoner(adminPage,"Prisoner's Details");
                          p.find(rs);
                          dispose();
              JOptionPane.showMessageDialog(null, "Record Found
Successfully");
              System.out.println("Statement executed successfully in
find() in findPrisoner");
              flag=true;
                }else{
                          JOptionPane.showMessageDialog(null, "Record
is not Found ");
                           rs.close();
                      }
           state.close();
           connect.close();
           }catch (Exception e) {
           System.out.print("exception in  find() in findPrisoner"+e);
        }
           return flag;
      }
  public void keyTyped ( KeyEvent e ){ }

  public void keyPressed ( KeyEvent e){
      int hold=e.getKeyCode();
      if((hold>=48&& hold<=57)|| hold==8|| hold==10);
      else{
           JOptionPane.showMessageDialog(null, "Enter a valid Prisoner
ID.");
           txtEntity.setText("");
      }
  }
  public void keyReleased ( KeyEvent e ){ }
}
```

ReportPage.java Source Code

```
//package pms.admin;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.sql.*;
import java.awt.*;
import java.awt.event.*;

public class PrisonerReport extends JFrame implements ActionListener{
      DefaultTableModel model;
      JTable table;
      int rowHeight=45;
      int rowMargin=10;
      JPanel pnlTable,pnlHeader,pnlButtons;
      JLabel lblHeader;
      JScrollPane sclTable;
      JButton btnOk;

  public static void main(String[] argv) throws Exception {
       PrisonerReport m= new PrisonerReport();
  }
  public PrisonerReport(){
            model = new DefaultTableModel();
            table = new JTable(model);
          pnlTable=new JPanel();
            pnlHeader=new JPanel();
          lblHeader=new JLabel("Prisoner Reports");
            pnlButtons=new JPanel();
          btnOk=new JButton("OK");
```

```
            initComponents();
  }
    private void initComponents() {

      btnOk.addActionListener(this);

      pnlTable.setLayout(new FlowLayout());
      pnlTable.setSize(1366,700);

      lblHeader.setFont(new Font("Arial Black",1,25));
      pnlHeader.setLayout(new FlowLayout());
      pnlHeader.add(lblHeader);

      pnlButtons.add(btnOk);

      table.setRowHeight(30);
      table.setFont(new Font("Arial Black",0,12));
      table.setAlignmentY(25);
      table.setEnabled(false);
      table.setSize(1366,700);

      model.addColumn("Serial NO.");
      model.addColumn("PRISONER ID");
      model.addColumn("NAME");
      model.addColumn("SECURITY LEVEL");
      model.addColumn("BOOK DATE");
      model.addColumn("RELEASE DATE");
      Object content[]=new Object[6];
      int count=1;

      try{

        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         Connection connect =
DriverManager.getConnection("jdbc:odbc:PMS");
        Statement  stmt = connect.createStatement();
        ResultSet rs = stmt.executeQuery("Select * from
Prisoner_Details where Status= '" + "Incarcerated" +"'");
        while(rs.next()){
              content[0]=count;
              content[1]=rs.getString("ID");
              content[2]=rs.getString("First_Name")+"
"+rs.getString("Last_Name");
              content[3]=rs.getString("Security_Class");
              content[4]=rs.getString("Book_Date");
              content[5]=rs.getString("Release_Date");
               model.addRow(content);
               count++;
        }
            stmt.close();
         connect.close();
    }catch(Exception e){
    }

    setSize(1366, 700);
      sclTable=new JScrollPane(table);
      sclTable.setSize(1366,700);
    pnlTable.add(sclTable);
      setLayout(new BorderLayout());
      add(pnlHeader,BorderLayout.NORTH);
      add(pnlButtons,BorderLayout.SOUTH);
      add(sclTable);

           try {
            for (UIManager.LookAndFeelInfo info :
UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        } catch (UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Prisoner.class.getName()).log(java.
util.logging.Level.SEVERE, null, ex);
        }

      setVisible(true);
  }
  public void actionPerformed(ActionEvent e){
  dispose();}
}
```

Screen Shots of Application

Components of Login Page

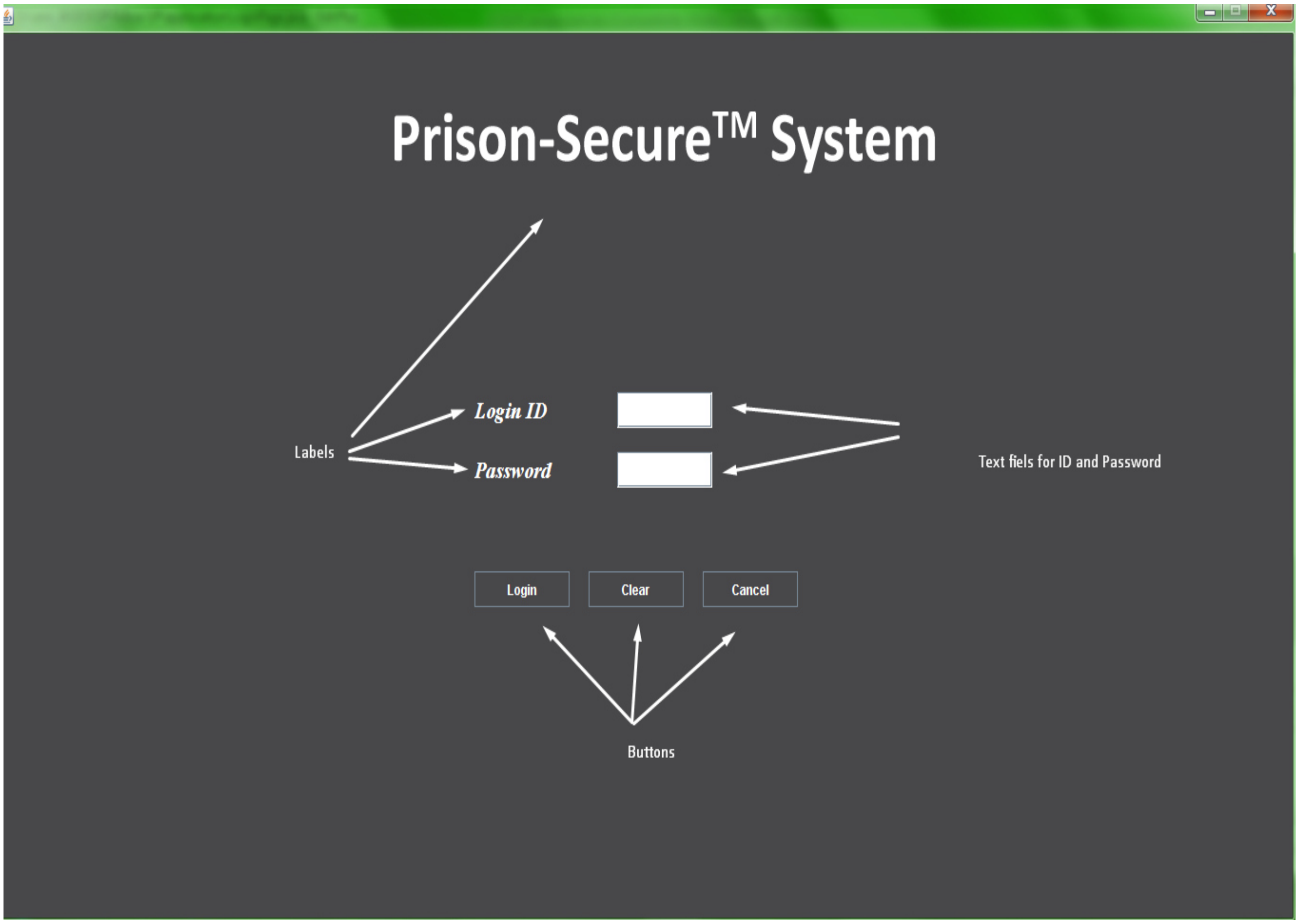


Figure 11Component of login Page

Description

Login Page is provided with two text field where user has to put his id and login password having respective labels.

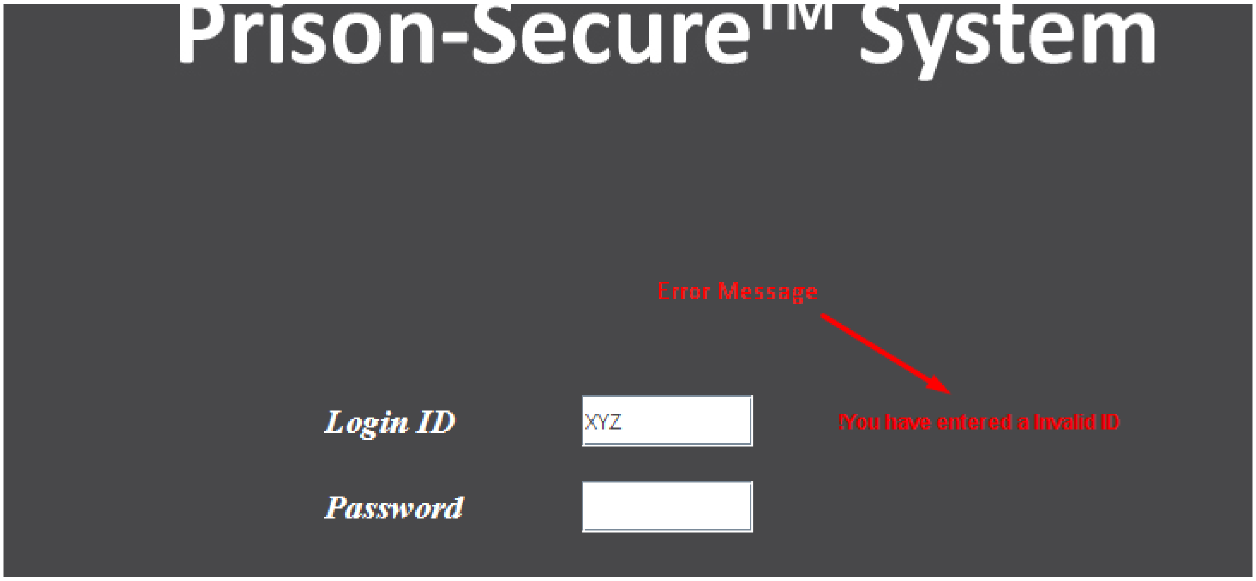
Three labels are there to provide one for header of the page and other two are for two text fields. Three buttons are there which works as per their labels and these tasks are :-

Login into the System

Clear the text fields.

Exit the login page.

Error Messages on Login Page

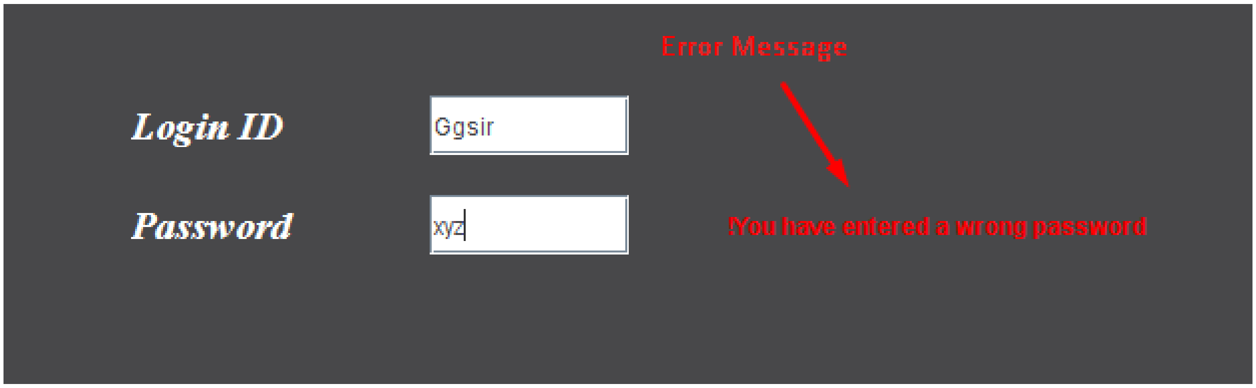


Login ID XYZ

Password

!You have entered a Invalid ID

Figure 12 Error on Login Page



Login ID Ggsir

Password xyz

!You have entered a wrong password

Figure 13Error on Login Page

Components of Admin Page

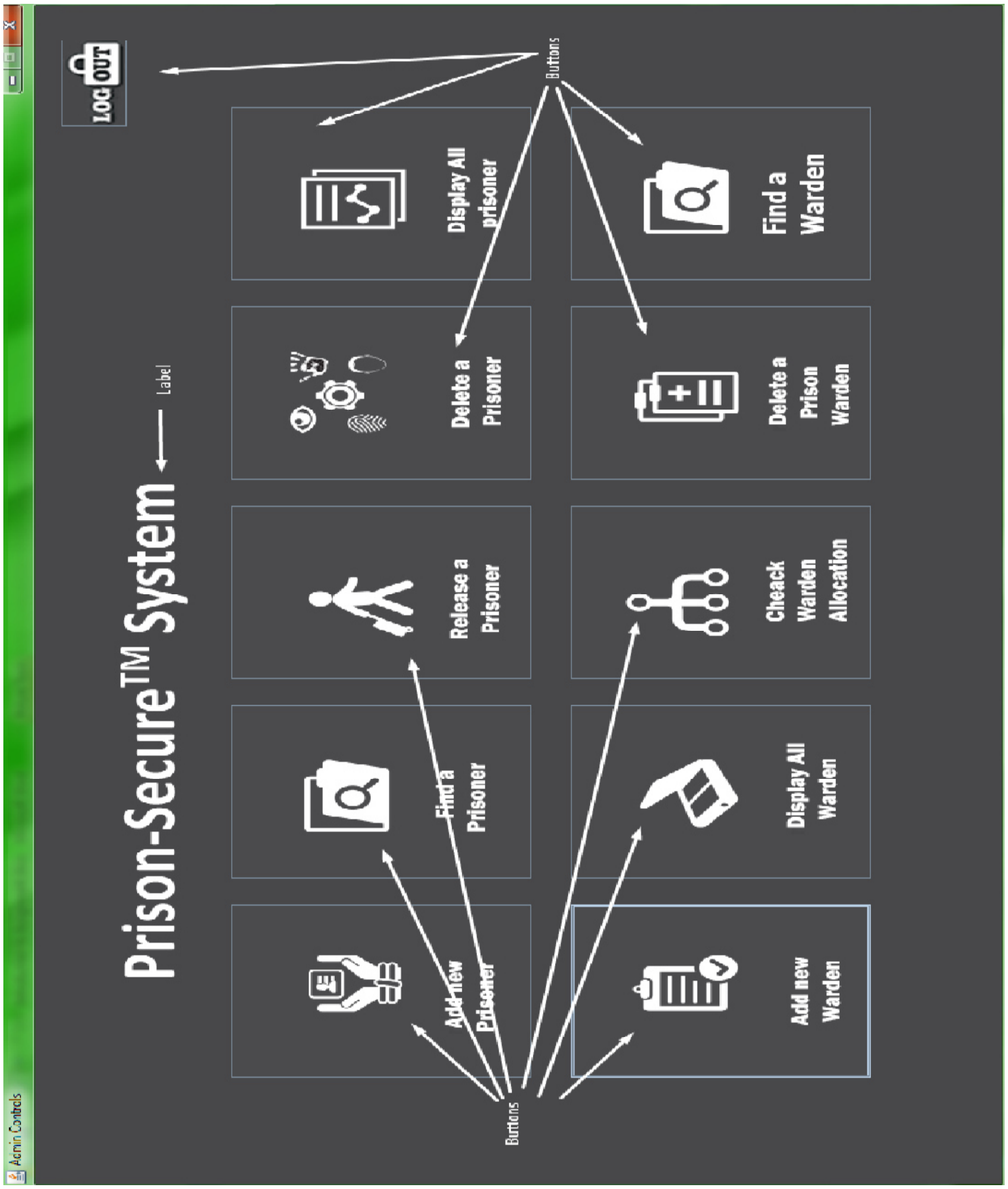


Figure 14 Components of Admin page

Description

Admin page is provided with eleven buttons and one label. Single label defines the header of the page. All buttons perform following tasks :-

- Add, find, and delete a Prisoner/Warden
- Release a prisoner.
- Generate reports of Prisoners/Wardens
- Check Warden Allocation.
- Logout from system.

Add Prisoner Page

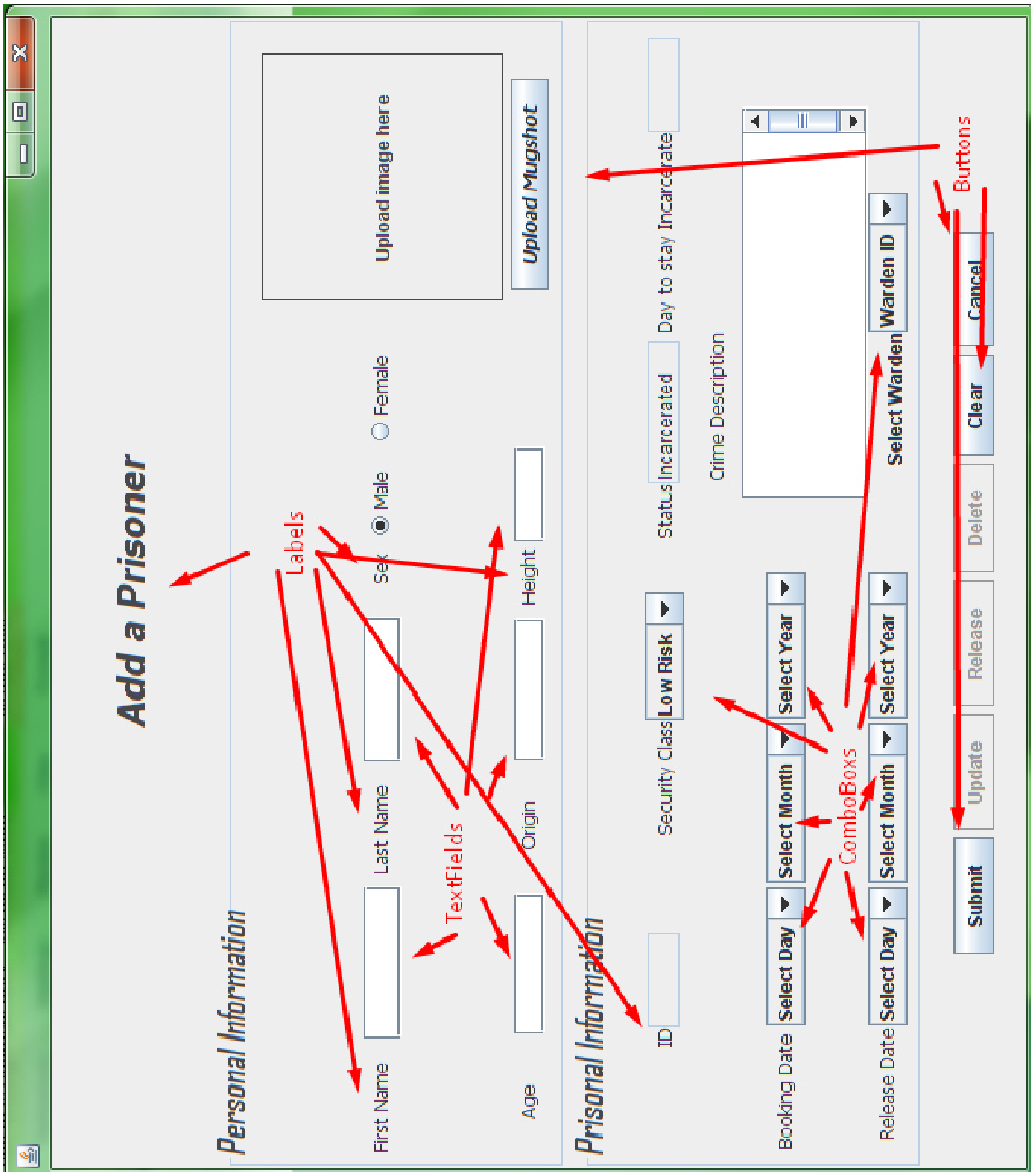


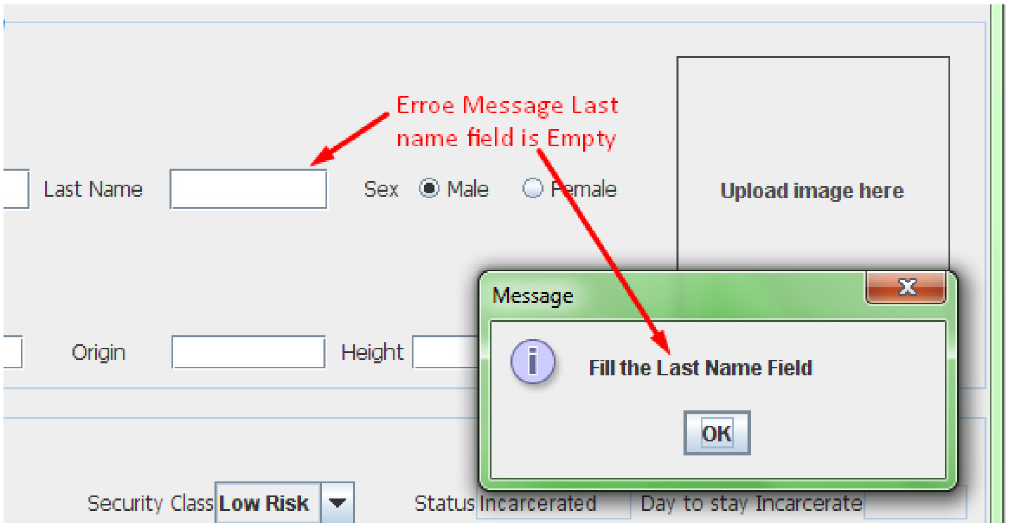
Figure 15 Components of Add Prisoner Page

Description

Add Prisoner page has fourteen labels which defines purpose of their adjacent components. Nine text fields which take input as per their labels. Three active buttons which provides the following functionalities:-

- Submit record
- Cancel form
- Clear fields

Error messages on Add Prisoner Page



Erroe Message Last name field is Empty

Message

Fill the Last Name Field

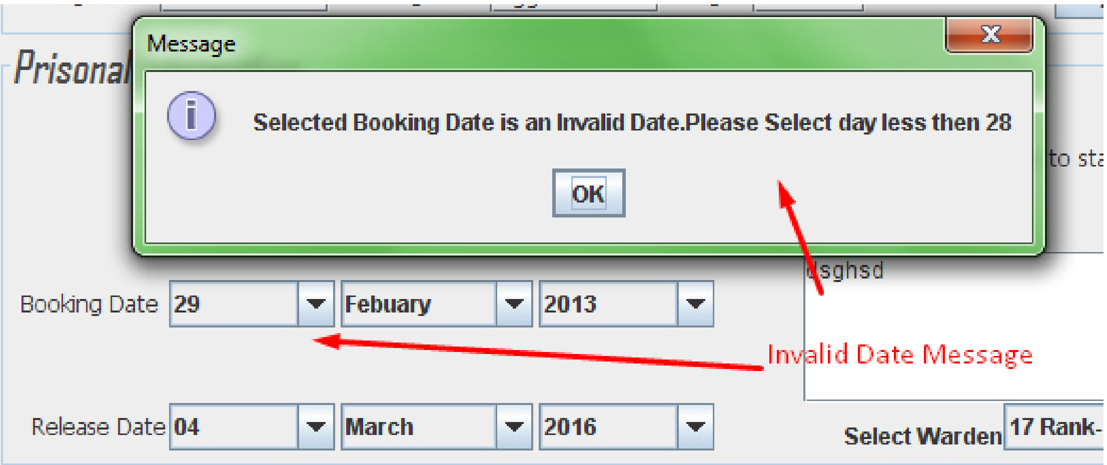
Last Name

Sex ● Male ○ Female

Security Class: Low Risk

Status: Incarcerated

Figure 16 Error on Add Prisoner Page



Message

Selected Booking Date is an Invalid Date.Please Select day less then 28

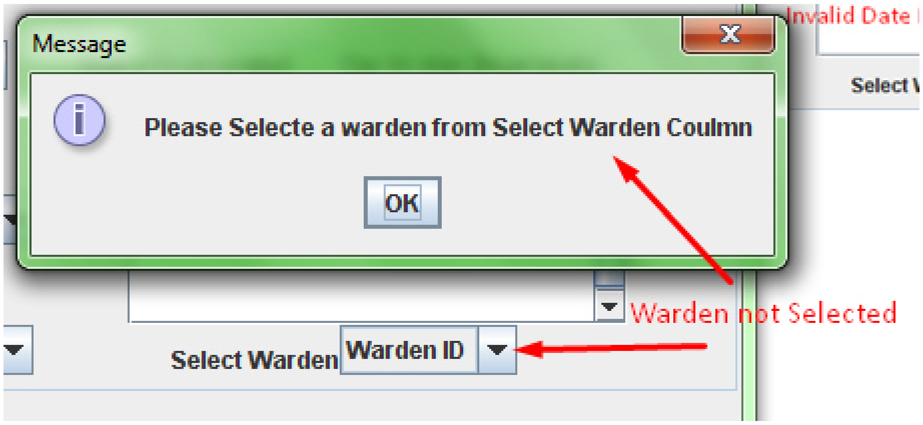
Invalid Date Message

Booking Date: 29 | Febuary | 2013

Release Date: 04 | March | 2016

Select Warden: 17 Rank-

Figure 17Error onAdd Prisoner Page



Message

Please Selecte a warden from Select Warden Coulmn

Warden not Selected

Select Warden: Warden ID

Figure 18Error onAdd Prisoner Page

Components of Add warden Page

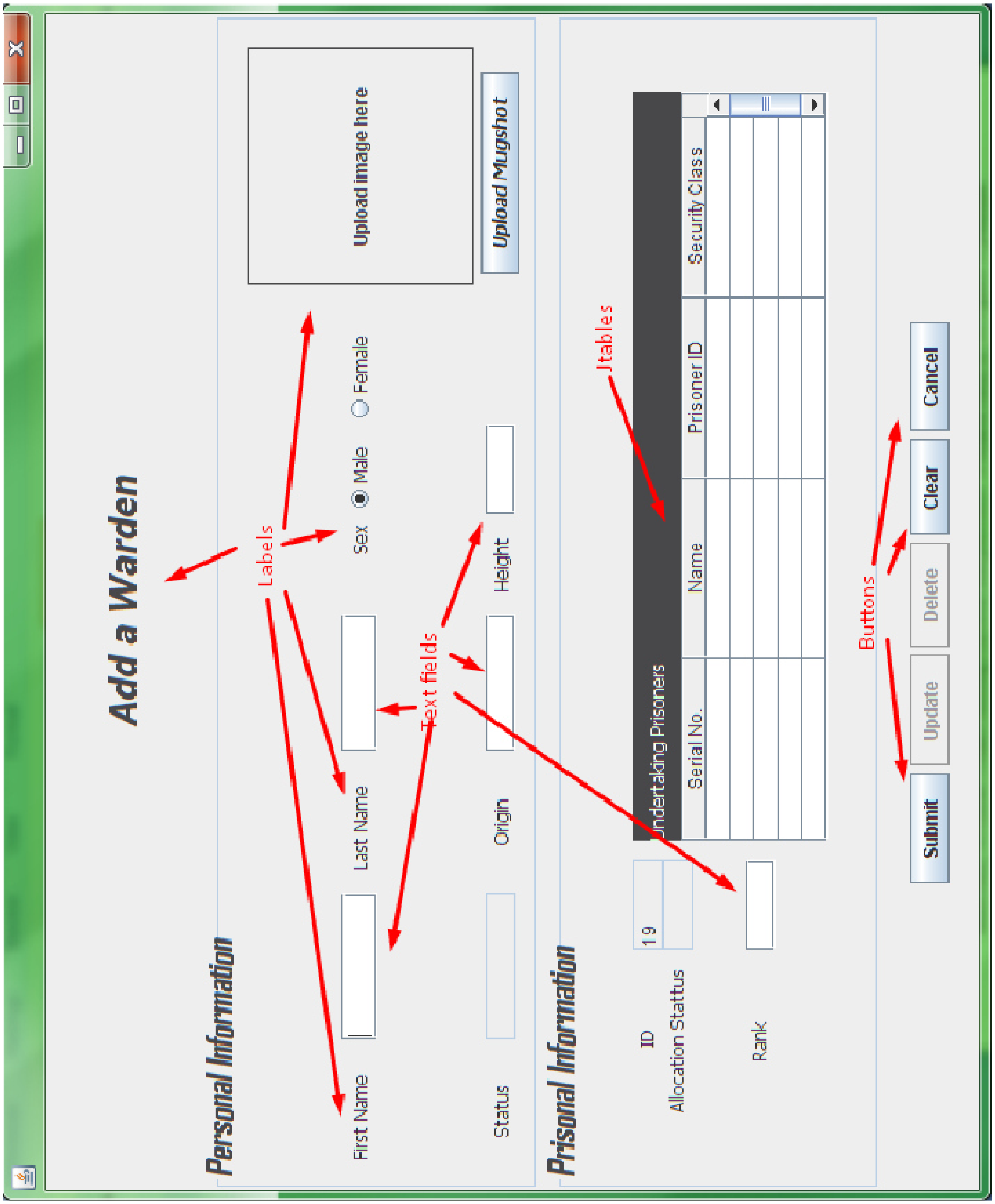


Figure 19 Components of Add WardenPage

Components of Find Prisoner Page

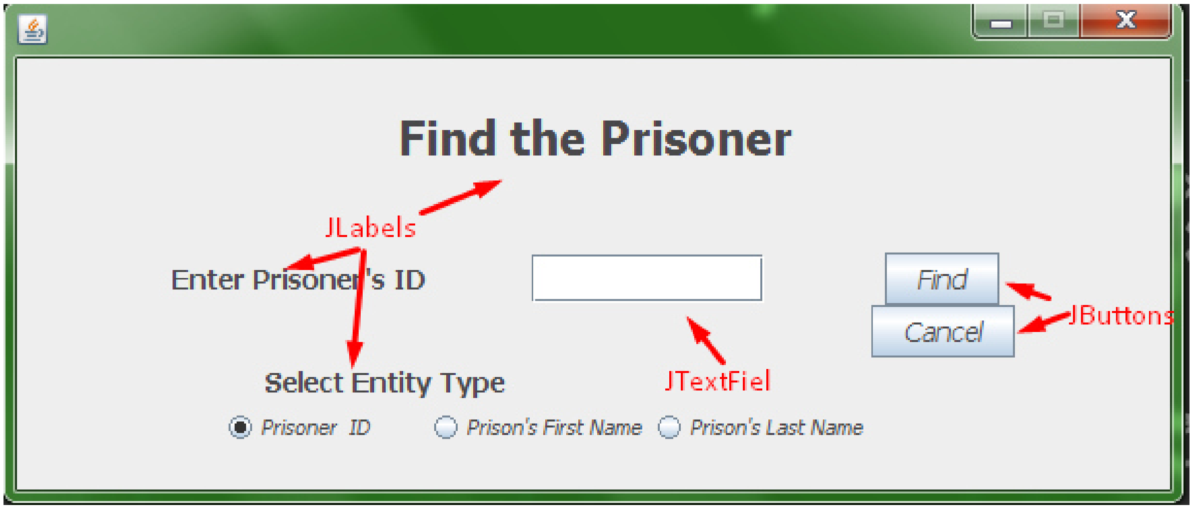
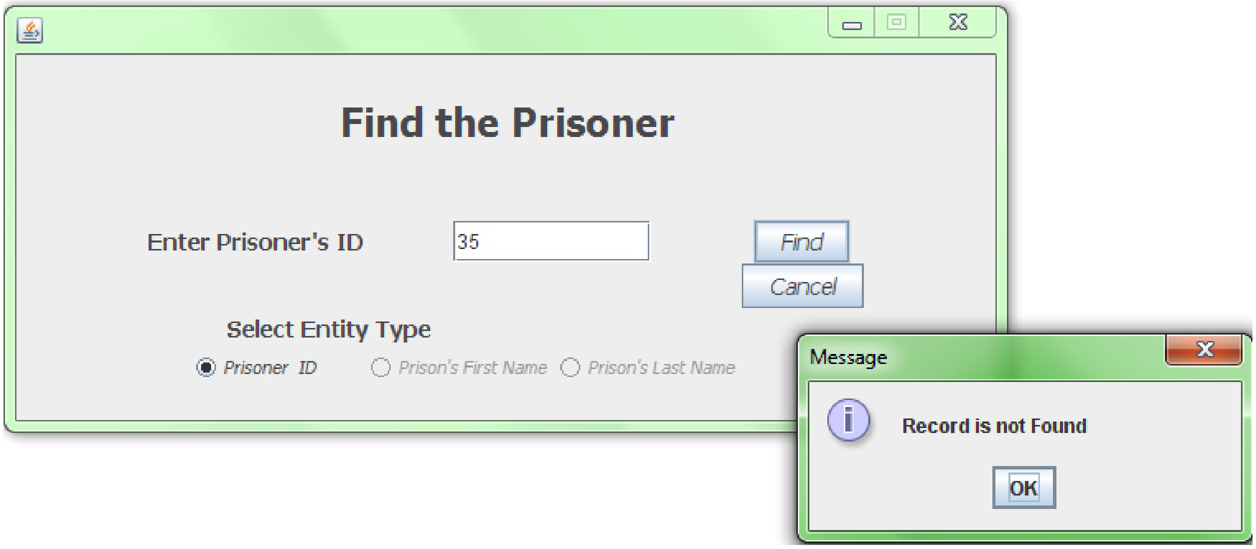


Figure 20 Components of Find Prisoner Page

Error Message on Find Prisoner Page

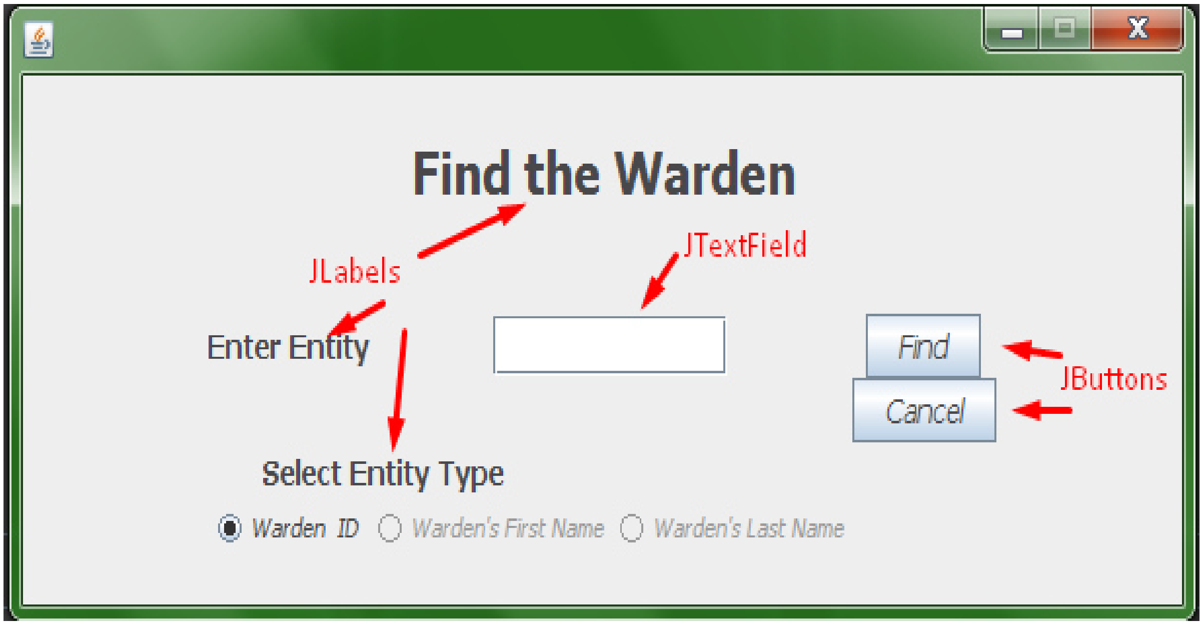


Enter Prisoner's ID 35

Record is not Found

Figure 21 Error on Find Components Page

Components of Warden Page



Find the Warden

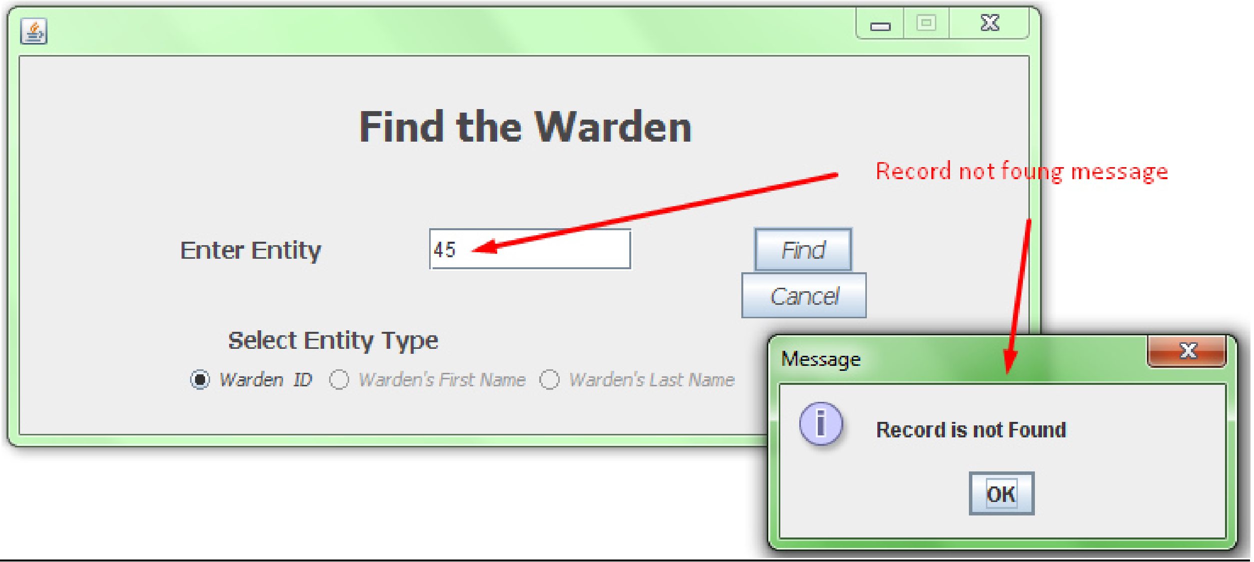
Enter Entity

Select Entity Type

● Warden ID ○ Warden's First Name ○ Warden's Last Name

Figure 22 Components of Find Warden Page

Error Message on Find Warden age



Find the Warden

Enter Entity: 45

Select Entity Type

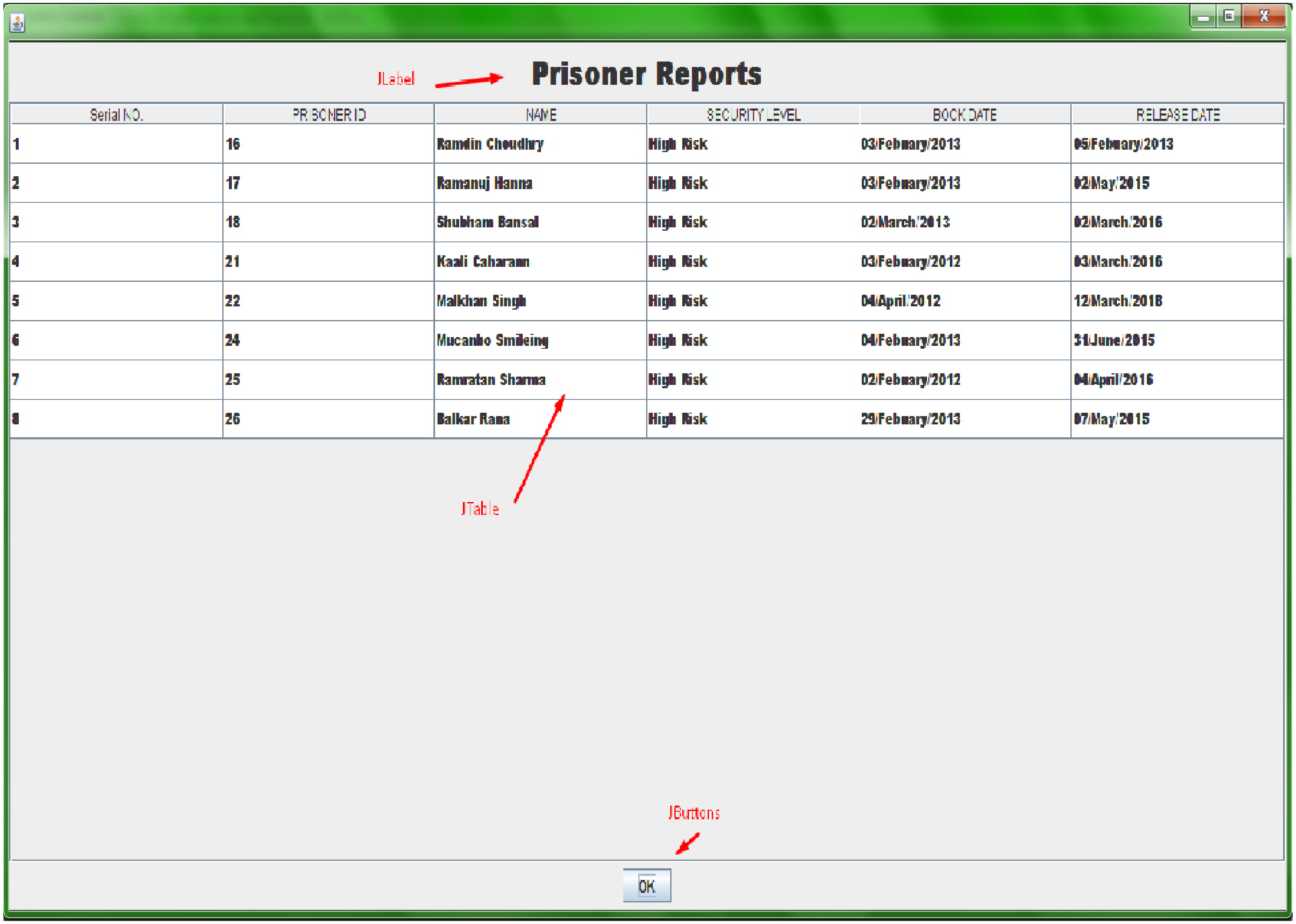
● Warden ID ○ Warden's First Name ○ Warden's Last Name

Message

Record is not Found

Figure 23 Error On Find Warden Page

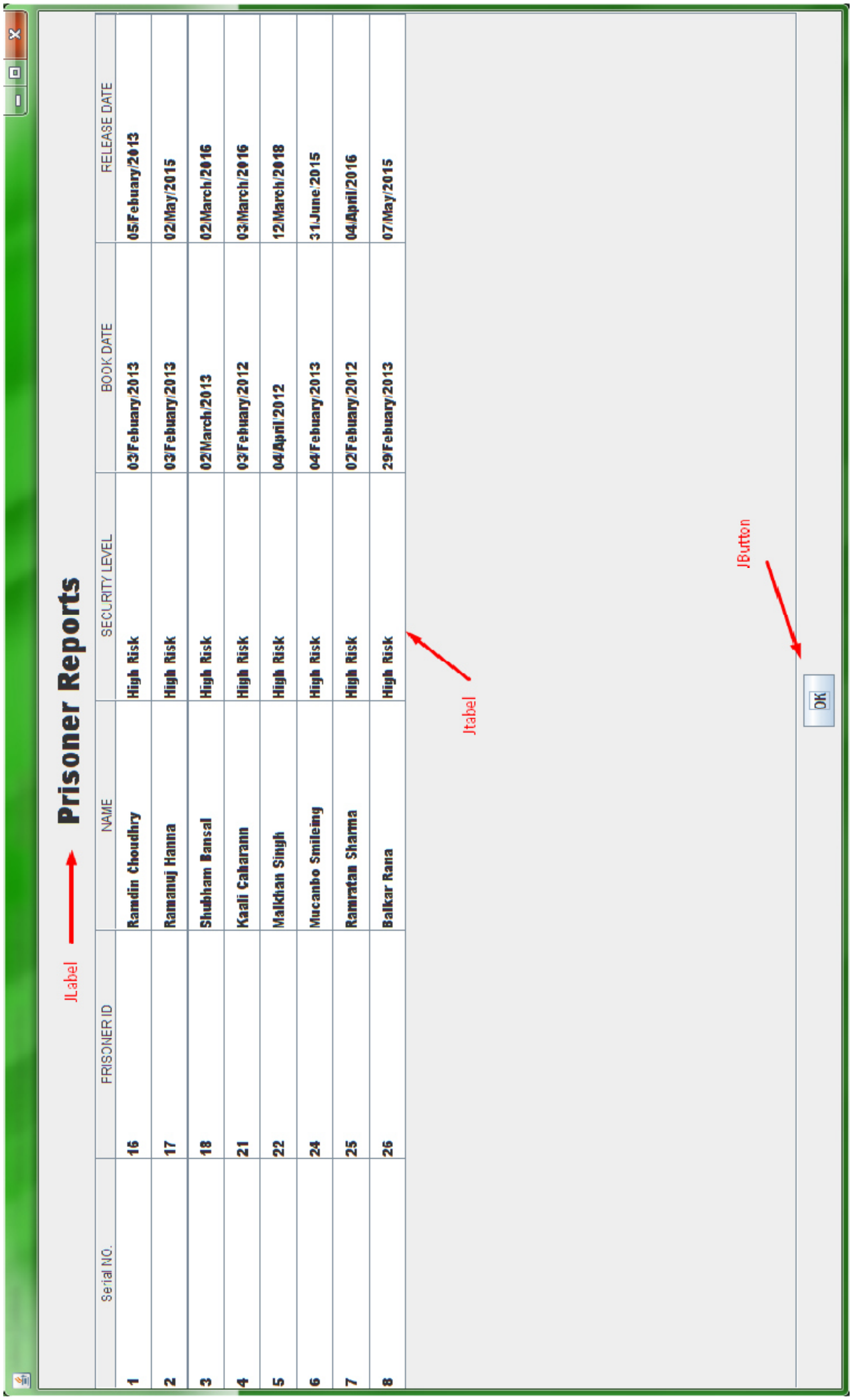
Components of Prisoner Report Page



Serial NO.	PRISONER ID	NAME	SECURITY LEVEL	BOOK DATE	RELEASE DATE
1	16	Ramdin Choudhry	High Risk	03/Febuary/2013	05/Febuary/2013
2	17	Ramanuj Hanna	High Risk	03/Febuary/2013	02/May/2015
3	18	Shubham Bansal	High Risk	02/March/2013	02/March/2016
4	21	Kaali Caharam	High Risk	03/Febuary/2012	03/March/2016
5	22	Malkhan Singh	High Risk	04/April/2012	12/March/2018
6	24	Mucanbo Smileing	High Risk	04/Febuary/2013	31/June/2015
7	25	Ramratan Sharma	High Risk	02/Febuary/2012	04/April/2016
8	26	Balkar Rana	High Risk	29/Febuary/2013	07/May/2015

Figure 24 Components of Prisoner Report Page

Components of Warden Report Page



Prisoner Reports

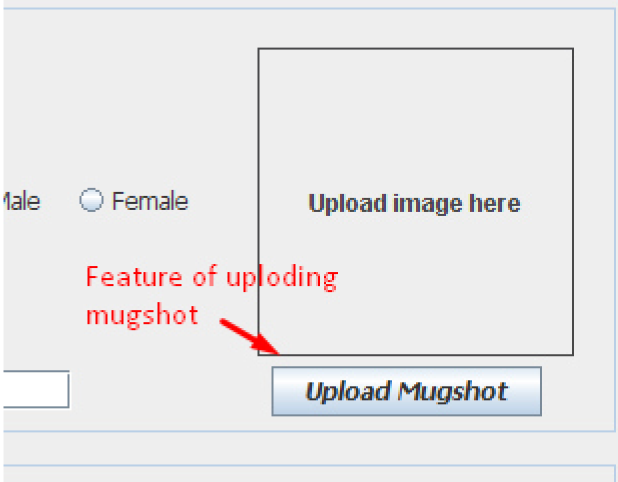
Serial NO.	FRISONER ID	NAME	SECURITY LEVEL	BOOK DATE	RELEASE DATE
1	16	Ramdin Choudhry	High Risk	03/Febuary/2013	05/Febuary/2013
2	17	Ramanuj Hanna	High Risk	03/Febuary/2013	02/May/2015
3	18	Shubham Bansal	High Risk	02/March/2013	02/March/2016
4	21	Kaali Caharann	High Risk	03/Febuary/2012	03/March/2016
5	22	Malkhan Singh	High Risk	04/April/2012	12/March/2018
6	24	Mucanbo Smileing	High Risk	04/Febuary/2013	31/June/2015
7	25	Ramratan Sharma	High Risk	02/Febuary/2012	04/April/2016
8	26	Balkar Rana	High Risk	29/Febuary/2013	07/May/2015

Figure 25Components of Warden Report Page

Additional Features

Store mug shots of warden/prisoner

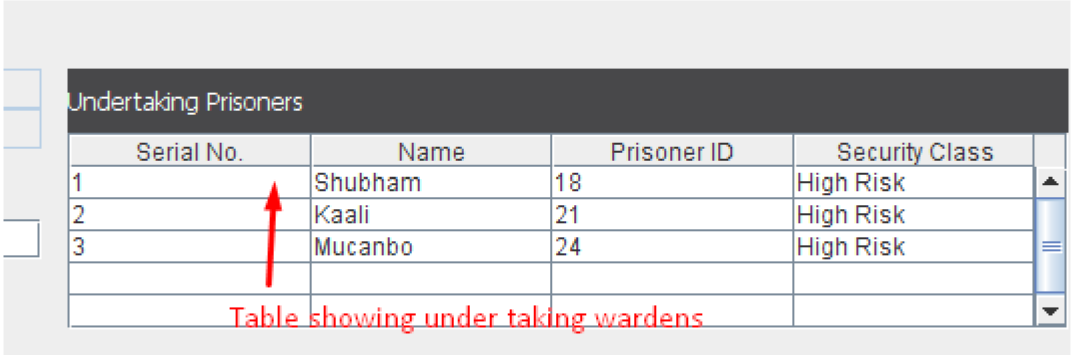
Professional Prison management system stores mug shot of all its wardens and Prisoners. I have provided the same functionality to my system. It makes my application to look more professional and enhance the efficiency of my system.



Feature of uploding mugshot

Warden Screen displaying under taking prisoners

My system shows All under taking prisoner of a warden on his details page.



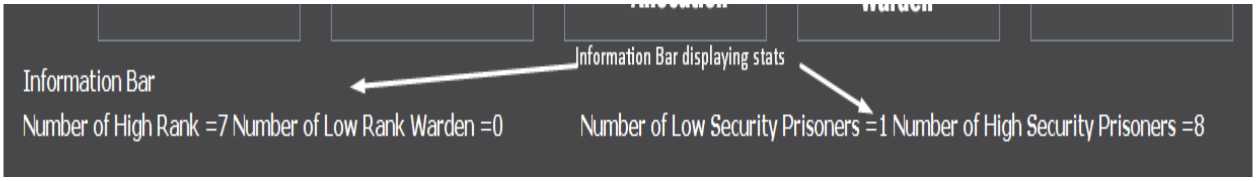
Undertaking Prisoners

Serial No.	Name	Prisoner ID	Security Class
1	Shubham	18	High Risk
2	Kaali	21	High Risk
3	Mucanbo	24	High Risk

Table showing under taking wardens

Check Warden Allocation

My system also covers the functionality of warden allocation. System displays the stats regarding to all prisoners and warden and do prison require more wardens.



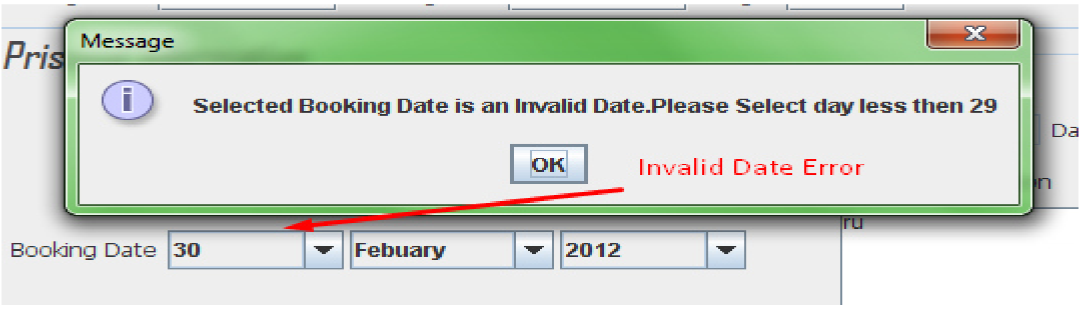
Information Bar displaying stats

Information Bar

Number of High Rank =7 Number of Low Rank Warden =0 Number of Low Security Prisoners =1 Number of High Security Prisoners =8

Invalid date Error

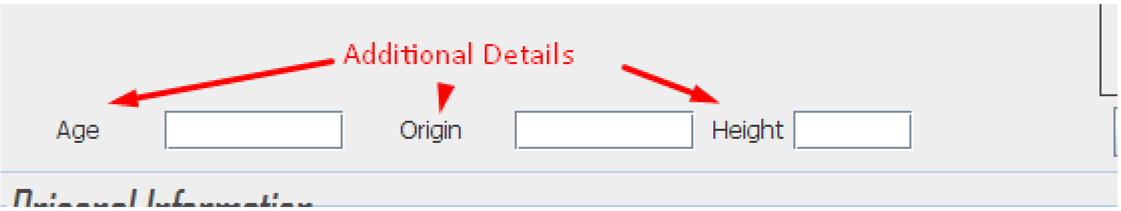
My system bounds user to enter some to enter valid date.



Pris
Message
Selected Booking Date is an Invalid Date.Please Select day less then 29
OK
Invalid Date Error
Booking Date
30
Febuary
2012

Stores additional Details

My System stores some additional details that are not mentioned in assignment. These details are usually stored by any prison management system so bring my system more close or comparable to real world management system I have also provided the same fields.



Assumptions

Assumption helps developer to increase flexibility and reliability of the system. In Assignment we must have to implement all the functionalities specified in Introduction but real world prison management application has many other features. I have assumed that my system is a complete prison management system with these all functionalities only.

Only Highest Rank warden have Login Permission

In my System we have to maintain 1:5 ratio of warden to the prisoner which is quite greater than actual prison management system .so I have assume it to be impractical to provide login functionalities to all the wardens in my system. Reasons for this assumptions are :

- Greater ratio of wardens to prisoners than practical one.
- Our system is providing features which should not be provided to all rank wardens.
- Provided features are to be used by Administrative office only
- Highest rank warden (5) being the most top officials can login the system and use all the features of system .

Unique ID for warden and prisoner

In my system I assumed that warden and prisoner will also have a unique ID because in general there may be name conflict among wardens and prisoners. It makes easy to recognize a prisoner and warden uniquely and avoid any type of confusion. Using primary key also makes it easier to update or delete the desired warden or prisoner.

User friendly

Instead of providing menu item controls to user (Admin) I have implemented my system via button options where I have provided all the control directly from the admin form. Admin don't need to search for a control in different menus. Reasons for button style menu :-

- Very easy to operate.
- No need to train the users.
- Very less number of provided functionalities.

References

- International Leads. 2015. *International Leads*. [ONLINE] Available at:http://www.elearners.com/international/ForeignLanding.aspx?. [Accessed 26 April 2015].
- Java Tutorial . 2015. *Java Tutorial* . [ONLINE] Available at:http://www.java2s.com/Tutorial/Java/CatalogJava.htm. [Accessed 26 April 2015].
- Java Desktop application: SWT vs. Swing - Stack Overflow. 2015. *Java Desktop application: SWT vs. Swing - Stack Overflow*. [ONLINE] Available at: http://stackoverflow.com/questions/2306190/java-desktop-application-swt-vs-swing. [Accessed 26 April 2015].
- Establishing a Connection (The Java™ Tutorials > JDBC(TM) Database Access > JDBC Basics) . 2015. *Establishing a Connection (The Java™ Tutorials > JDBC(TM) Database Access > JDBC Basics)* . [ONLINE] Available at:http://docs.oracle.com/javase/tutorial/jdbc/basics/connecting.html. [Accessed 26 April 2015].
- Using JDBC with GUI API (The Java™ Tutorials > JDBC(TM) Database Access > JDBC Basics) . 2015. *Using JDBC with GUI API (The Java™ Tutorials > JDBC(TM) Database Access > JDBC Basics)* . [ONLINE] Available at:http://docs.oracle.com/javase/tutorial/jdbc/basics/jdbcswing.html. [Accessed 26 April 2015].
- How to Use Panels (The Java™ Tutorials > Creating a GUI With JFC/Swing > Using Swing Components) . 2015. *How to Use Panels (The Java™ Tutorials > Creating a GUI With JFC/Swing > Using Swing Components)* . [ONLINE] Available at:http://docs.oracle.com/javase/tutorial/uiswing/components/panel.html. [Accessed 26 April 2015].
- How to Use Password Fields (The Java™ Tutorials > Creating a GUI With JFC/Swing > Using Swing Components) . 2015. *How to Use Password Fields (The Java™ Tutorials > Creating a GUI With JFC/Swing > Using Swing Components)* . [ONLINE] Available at:http://docs.oracle.com/javase/tutorial/uiswing/components/passwordfield.html. [Accessed 26 April 2015].