**Project Technical Document**

**Grade 12 PAT**

**Jean De La Paix Kwizera Katabarwa**

**Student Register**

**Content**

**Cover page……………………………………………..1**

**Content page………………………………………….2**

**Login screen…………………………………………..3**

**New account screen………………………………7**

**Login/Register Help screen…………………9**

**One time Register screen……………………….10**

**Student attendance History interface…….17**

**Daily register interface…………………………19**

**Analysis interface…………………………………26**

**Late Students…………………………………….30**

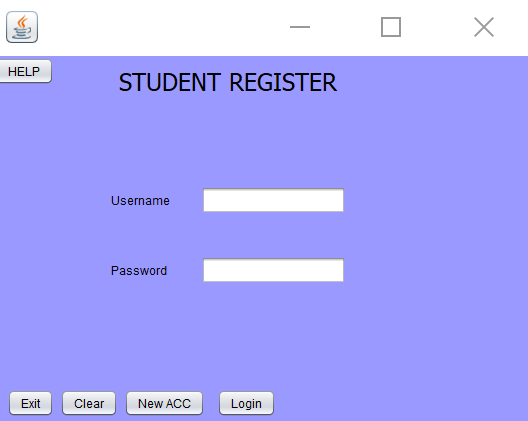
**Absentees……………………………………………33**

**Registering help…………………………………..37**

**Defensive Programming……………………….37**

**USER INTERFACE DESIGN**

**Login screen**

****

**\* Creates new form Login**

**\*/**

**ArrayList<Users> users;**

**public Login() {**

**initComponents();**

**users = new ArrayList<Users>();**

**getUsers();**

**}**

**// a method to get passwords and usernames in the users table**

**public ArrayList<Users> getUsers()**

**{**

**UserInterface g = new UserInterface();**

**Connection connection = g.getConnection();**

**String query = "Select \* FROM users";**

**Statement st;**

**ResultSet rs;**

**try**

**{**

**st = connection.createStatement();**

**rs = st.executeQuery(query);**

**Users user;**

**while (rs.next())**

**{**

**user = new Users(rs.getString("Name"), rs.getString("Password"));**

**users.add(user);**

**}**

**}catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**return users;**

**}**

**// a method to clear name and password fields**

**private void ClearForm(){**

**jTextFieldUsername.setText("");**

**jPasswordField1.setText("");**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**try{**

**getUsers();**

**// Presence check data validation for Username and Password**

**String password1 = new String(jPasswordField1.getText());**

**if (jTextFieldUsername.getText().isEmpty()){**

**JOptionPane.showMessageDialog(null,"Username required");**

**jTextFieldUsername.requestFocus();**

**}else if (password1.isEmpty()){**

**JOptionPane.showMessageDialog(null,"Password required");**

**jTextFieldUsername.requestFocus();**

**jPasswordField1.requestFocus();**

**}else{**

**boolean found = false;**

**int posInList = 0;**

**//if there are users in the users table**

**for (int i = 0; i< users.size(); i++){**

**// validation. if username matches the name in the table**

**if (users.get(i).getUsername().equals(jTextFieldUsername.getText().trim())){**

**found = true;**

**posInList = i;**

**}**

**}**

**// if it does not match display an error messahe**

**if ((found==false)){**

**JOptionPane.showMessageDialog(null,"Invalid Name");**

**ClearForm();**

**jTextFieldUsername.requestFocus();**

**}else**

**// if the name matches and the password matches the name in the user table proceed to the userInterface**

**if ((found==true) && (users.get(posInList).getPassword().equals(password1))){**

**new UserInterface().setVisible(true);//open the form MainUI**

**this.dispose();//Close the current form**

**}**

**// if the password is wrong display an error message**

**else{**

**JOptionPane.showMessageDialog(null, "Invalid Password");**

**ClearForm();**

**}**

**}**

**}//if there wwwere exceptions this message will be displayed**

**catch(Exception e){**

**JOptionPane.showMessageDialog(null,e);**

**}**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**ClearForm(); // TODO add your handling code here:**

**}**

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {**

**new NewAccount().setVisible(rootPaneCheckingEnabled);**

**this.dispose();// TODO add your handling code here:**

**}**

**private void HelpbtnActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpScreen help=new HelpScreen();**

**help.setVisible(true);**

**this.dispose();**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

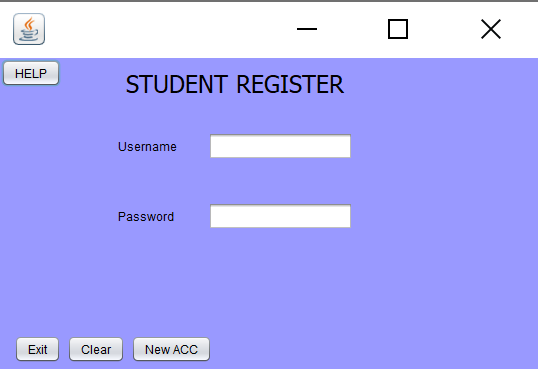
**new Login().setVisible(true);**

**}**

**});**

**}**

**New account screen**

****

**private void ExitbtnActionPerformed(java.awt.event.ActionEvent evt) {**

**System.exit(0);**

**}**

**private void CancelbtnActionPerformed(java.awt.event.ActionEvent evt) {**

**Usernametxt.setText("");**

**Passwordtxt.setText("");**

**}**

**private void HelpbtnActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpScreen help=new HelpScreen();**

**help.setVisible(true);**

**}**

**private void RegisterbtnActionPerformed(java.awt.event.ActionEvent evt) {**

**String password1 = new String(Passwordtxt.getText());**

**if (Usernametxt.getText().isEmpty()){**

**JOptionPane.showMessageDialog(null,"Username required");**

**Usernametxt.requestFocus();**

**}else if (password1.isEmpty()){**

**JOptionPane.showMessageDialog(null,"Password required");**

**Usernametxt.requestFocus();**

**Passwordtxt.requestFocus();**

**}else{**

**String query = "INSERT INTO users (Name,Password) values('" + Usernametxt.getText() + "','" +Passwordtxt.getText()+"')";**

**executeSQLQuery(query, "inserted");**

**new Login().setVisible(true);**

**this.dispose();**

**}**

**}**

**//A function that will create a connection to the database**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**JOptionPane.showMessageDialog(null, "Data " +message+ " sucessfully");**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null, "Data Not " +message+ " sucessfully");**

**}**

**}catch (Exception ex)**

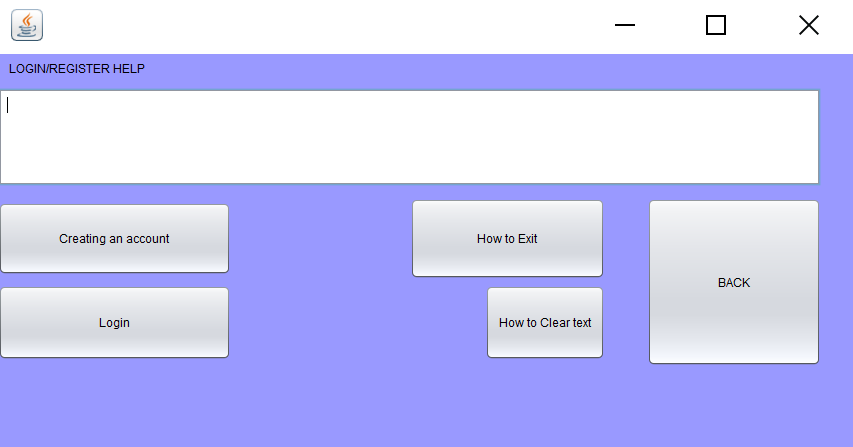
**{**

**ex.printStackTrace();**

**}**

**}**

**Login/Register Help screen**



**private void btnnewaccountActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("Write your name,password and the grade being registered then press NewACC to create a new account");**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("wrie your user name then password then press Login");**

**}**

**private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("Press the Exit button to exit the app"); // TODO add your handling code here:**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**new NewAccount().setVisible(true);**

**this.dispose();// TODO add your handling code here:**

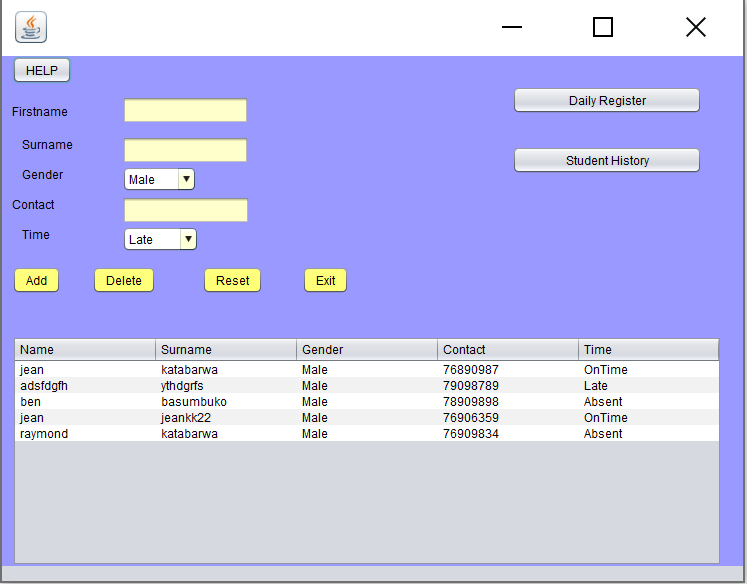
**}**

**private void btnClearTextActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("Press the Clear button to remove the written Name, Password and Grade"); // TODO add your handling code here:**

**}**

**One time Register screen**

****

**public ArrayList<StudentRegister> getStudents()**

**{**

**ArrayList<StudentRegister> StudentList = new ArrayList<StudentRegister>();**

**Connection connection = getConnection();**

**String query = "Select \* FROM students";**

**Statement st;**

**ResultSet rs;**

**try**

**{**

**st = connection.createStatement();**

**rs = st.executeQuery(query);**

**StudentRegister attendance;**

**while (rs.next())**

**{**

**attendance = new StudentRegister(rs.getString("Name"), rs.getString("Surname") ,rs.getString("Gender"),rs.getString("Contact"),rs.getString("Time"));**

**StudentList.add(attendance);**

**}**

**}catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**return StudentList;**

**}**

**public void show\_Members\_In\_jTable()**

**{**

**ArrayList<StudentRegister> list = getStudents();**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**Object[] row = new Object[5];**

**for (int i = 0; i < list.size(); i ++)**

**{**

**row[0] = list.get(i).getName();**

**row[1] = list.get(i).getSurname();**

**row[2] = list.get(i).getGender();**

**row[3] = list.get(i).getContact();**

**row[4] = list.get(i).getTime();**

**model.addRow(row);**

**}**

**jTable1.setModel(model);**

**model.fireTableDataChanged();**

**}**

**//method to write each student data in a txt file**

**public void writeatxtfile(){**

**String name= txtFirstname.getText();**

**String Surname= txtSurname.getText() ;**

**String time=cmbArrivalTime.getSelectedItem().toString();**

**try**

**{**

**FileWriter Writer=new FileWriter("StudentData.txt",true);**

**Writer.write(""+name+" "+Surname+" "+time);**

**Writer.write(System.getProperty("line.separator"));**

**Writer.close();**

**System.out.println("TXT file written");**

**}**

**catch(Exception e)**

**{**

**System.out.println("Failed to write to txt file");**

**}**

**}**

**private void ClearForm()**

**{**

**txtFirstname.setText(" ");**

**txtSurname.setText(null);**

**txtContact.setText(null);**

**lblErrorFirstname.setText(null);**

**lblErrorSurname.setText(null);**

**lblErrorContact.setText(null);**

**}**

**private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {**

**// number Length check**

**int ValidNumber=8;**

**String phoneNumber=txtContact.getText();**

**int length=phoneNumber.length();**

**//Valid simcard**

**String MTNone="76";**

**String MTNtwo="78";**

**String SwaziMobile="79";**

**// //Presence Check for all fields**

**if(txtFirstname.equals("")){System.out.println("Student name is required");}**

**else if (txtSurname.getText().equals(null)){lblErrorSurname.setText("The Surname is required.");}**

**else if (txtContact.getText().equals(null)){lblErrorContact.setText("The Contact is required.");}**

**else if(length!=ValidNumber)**

**{**

**//Length check of the contact number**

**lblErrorContact.setText("The number you entered is not complete");**

**}**

**//MTN or SwaziMObile check**

**else if(txtContact.getText().substring(0,2)!=MTNone){lblErrorContact.setText("Invalid Number");}**

**else if(txtContact.getText().substring(0,2)!=MTNtwo){lblErrorContact.setText("Invalid Number");}**

**else if( txtContact.getText().substring(0,2)!=SwaziMobile){lblErrorContact.setText("Invalid Number");}**

**else{**

**String query = "INSERT INTO students (Name,Surname,Gender,Contact,Time) values('" + txtFirstname.getText() + "','" +txtSurname.getText()+"','" + cmbComboBox.getSelectedItem()+"','" +txtContact.getText()+"','" +cmbArrivalTime.getSelectedItem()+"')";**

**executeSQLQuery(query, "inserted");**

**txtFirstname.requestFocus();**

**DefaultTableModel model=(DefaultTableModel)jTable1.getModel();**

**model.setRowCount(0);**

**show\_Members\_In\_jTable();**

**writeatxtfile();**

**}**

**}**

**private void cmbComboBoxActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {**

**// Delete member usins a prepared statement**

**PreparedStatement pst = null;**

**Connection con = null;**

**try{**

**String sql = "DELETE from `students` WHERE Contact = ?";**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**pst = con.prepareStatement(sql);**

**pst.setString(1, txtFirstname.getText());**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Record removed");**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**int SelectedRowIndex = jTable1.getSelectedRow();**

**model.removeRow(SelectedRowIndex);**

**}catch(SQLException | HeadlessException e) {**

**JOptionPane.showMessageDialog(null, e);**

**}**

**jTable1.addNotify();**

**ClearForm();**

**txtFirstname.requestFocus();**

**}**

**private JFrame frame;**

**private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {**

**frame =new JFrame("Exit");**

**if(JOptionPane.showConfirmDialog(frame,"Are you sure you want to exit","Register",JOptionPane.YES\_NO\_OPTION)==JOptionPane.YES\_OPTION){**

**System.exit(0);**

**}**

**}**

**private void txtContactActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {**

**//clear written text**

**txtFirstname.setText("");**

**txtSurname.setText("");**

**txtContact.setText("");**

**//clear qrror messages**

**lblErrorFirstname.setText("");**

**lblErrorSurname.setText("");**

**lblErrorContact.setText("");**

**}**

**private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {**

**int i = jTable1.getSelectedRow();**

**TableModel model =jTable1.getModel();**

**txtFirstname.setText(model.getValueAt(i, 0).toString());**

**txtSurname.setText(model.getValueAt(i, 1).toString());**

**cmbComboBox.setSelectedItem(model.getValueAt(i, 2).toString());**

**txtContact.setText(model.getValueAt(i, 3).toString());**

**cmbArrivalTime.setSelectedItem(model.getValueAt(i, 4).toString());**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpScreen2 help=new HelpScreen2();**

**help.setVisible(true);**

**this.dispose();**

**// TODO add your handling code here:**

**}**

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {**

**StudentHistory SH=new StudentHistory();**

**SH.setVisible(true);**

**this.dispose();**

**// TODO add your handling code here:**

**}**

**private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {**

**DailyRegister AnalysisInterface=new DailyRegister();**

**AnalysisInterface.setVisible(true);**

**this.dispose(); // TODO add your handling code here:**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new UserInterface().setVisible(true);**

**}**

**});**

**}//A function that will create a connection to the database**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**JOptionPane.showMessageDialog(null, "Data " +message+ " sucessfully");**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null, "Data Not " +message+ " sucessfully");**

**}**

**}catch (Exception ex)**

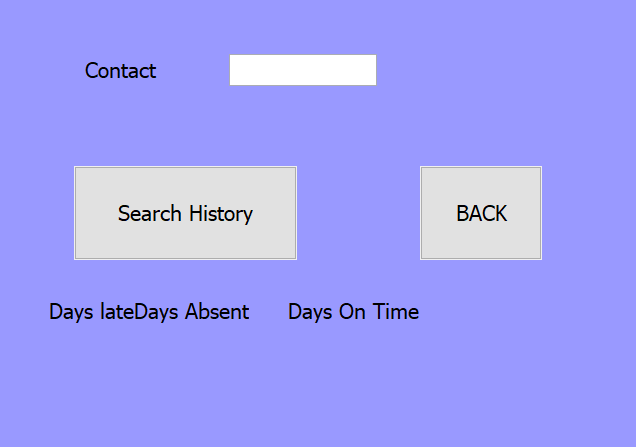
**{**

**ex.printStackTrace();**

**}**

**}**

**Student attendance History interface**

****

**public StudentHistory() {**

**initComponents();**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**String timesLate=txtContact.getText()+" "+"Late";**

**String timesEarly=txtContact.getText()+" "+"OnTime";**

**String timesAbsent=txtContact.getText()+" "+"Absent";**

**int Late=0;**

**int OnTime=0;**

**int Absent=0;**

**String s;**

**int none=0;**

**try {**

**Scanner scanner=new Scanner (new File("StudentData.txt"));**

**while(scanner.hasNext())**

**{**

**s=scanner.nextLine();**

**if (s.equals(timesLate)){**

**Late++;**

**}**

**if(s.equals(timesEarly)){**

**OnTime++;**

**}**

**if (s.equals(timesAbsent)){**

**Absent++;**

**}**

**}**

**if (Late>none||OnTime>none||Absent>none){**

**daysabsentlbl.setText(Integer.toString(Absent));**

**dayslatelbl.setText(Integer.toString(Late));**

**daysontimelbl.setText(Integer.toString(OnTime));**

**}**

**else{ JOptionPane.showMessageDialog(this, "This Student Does Not Exist");}**

**scanner.close();**

**} catch (FileNotFoundException ex) {**

**System.out.print("No student registered");**

**}**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**new DailyRegister().setVisible(true);**

**this.dispose(); // TODO add your handling code here:**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

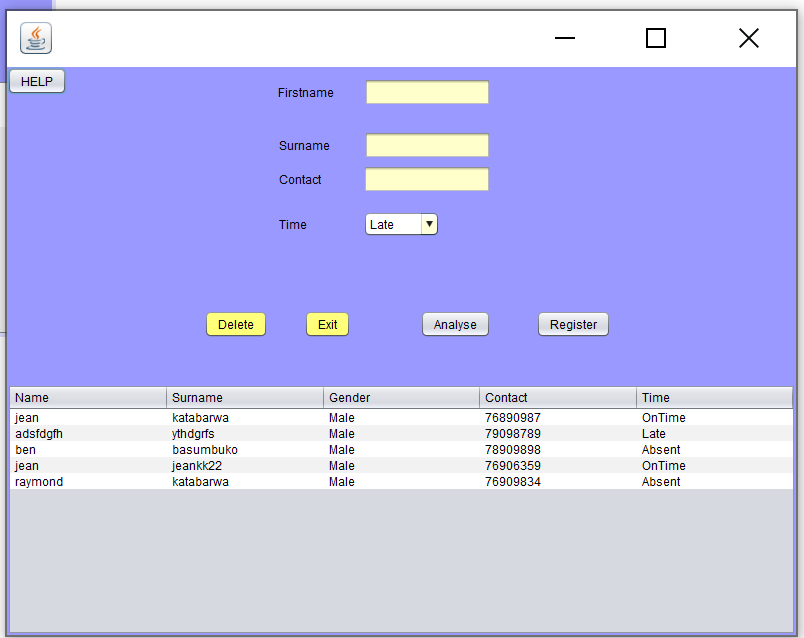
**new StudentHistory().setVisible(true);**

**}**

**});**

**}**

**Daily register interface**

****

**//creating an arrayList to hold data from the database in the class**

**public ArrayList<StudentRegister> getStudents()**

**{**

**ArrayList<StudentRegister> StudentList = new ArrayList<StudentRegister>();**

**Connection connection = getConnection();**

**String query = "Select \* FROM students";**

**Statement st;**

**ResultSet rs;**

**try**

**{**

**st = connection.createStatement();**

**rs = st.executeQuery(query);**

**StudentRegister attendance;**

**while (rs.next())**

**{**

**attendance = new StudentRegister(rs.getString("Name"), rs.getString("Surname") ,rs.getString("Gender"),rs.getString("Contact"),rs.getString("Time"));**

**StudentList.add(attendance);**

**}**

**}catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**return StudentList;**

**}**

**public void show\_Members\_In\_jTable()**

**{**

**ArrayList<StudentRegister> list = getStudents();**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**Object[] row = new Object[5];**

**for (int i = 0; i < list.size(); i ++)**

**{**

**row[0] = list.get(i).getName();**

**row[1] = list.get(i).getSurname();**

**row[2] = list.get(i).getGender();**

**row[3] = list.get(i).getContact();**

**row[4] = list.get(i).getTime();**

**model.addRow(row);**

**}**

**jTable1.setModel(model);**

**model.fireTableDataChanged();**

**}**

**private void ClearForm()**

**{**

**txtFirstname.setText(null);**

**txtContact.setText(null);**

**txtSurname.setText(null);**

**}**

**/\*method to write each student data in a txt file. this is important to keep**

**track of how many times each student was Late/Absent/On Time by**

**keeping track of their respective attendance daily\*/**

**public void writeatxtfile(){**

**String number=txtContact.getText();**

**String time=cmbArrivalTime.getSelectedItem().toString();**

**long millis=System.currentTimeMillis();**

**java.sql.Date date=new java.sql.Date(millis);**

**try**

**{**

**// to determine the date and the arrival time**

**FileWriter Writer=new FileWriter("ArrivalT.txt",true);**

**Writer.write(number+" "+time+" "+date);**

**Writer.write(System.getProperty("line.separator"));**

**Writer.close();**

**System.out.println("TXT file written");**

**// to determine the number of times students were registered**

**FileWriter AWriter=new FileWriter("StudentData.txt",true);**

**Writer.write(number+" "+time);**

**Writer.write(System.getProperty("line.separator"));**

**Writer.close();**

**System.out.println("TXT file written");**

**}**

**catch(Exception e)**

**{**

**System.out.println("Failed to write to txt file");**

**}**

**}**

**private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {**

**// Delete member usins a prepared statement**

**PreparedStatement pst = null;**

**Connection con = null;**

**try{**

**String sql = "DELETE from `students` WHERE Contact = ?";**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**pst = con.prepareStatement(sql);**

**pst.setString(1, txtFirstname.getText());**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Record removed");**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**int SelectedRowIndex = jTable1.getSelectedRow();**

**model.removeRow(SelectedRowIndex);**

**}catch(SQLException | HeadlessException e) {**

**JOptionPane.showMessageDialog(null, e);**

**}**

**jTable1.addNotify();**

**ClearForm();**

**txtFirstname.requestFocus();**

**}**

**private JFrame frame;**

**private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {**

**frame =new JFrame("Exit");**

**if(JOptionPane.showConfirmDialog(frame,"Are you sure you want to exit","Register",JOptionPane.YES\_NO\_OPTION)==JOptionPane.YES\_OPTION){**

**System.exit(0);**

**}**

**}**

**private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {**

**int i = jTable1.getSelectedRow();**

**TableModel model =jTable1.getModel();**

**txtFirstname.setText(model.getValueAt(i, 0).toString());**

**txtSurname.setText(model.getValueAt(i, 1).toString());**

**txtContact.setText(model.getValueAt(i, 3).toString());**

**cmbArrivalTime.setSelectedItem(model.getValueAt(i, 4).toString());**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**Analyse AnalysisInterface=new Analyse();**

**AnalysisInterface.setVisible(true);**

**this.dispose();**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpScreen2 help=new HelpScreen2();**

**help.setVisible(true);**

**this.dispose();**

**// TODO add your handling code here:**

**}**

**private void txtContactActionPerformed(java.awt.event.ActionEvent evt) {**

**// TODO add your handling code here:**

**}**

**private void updatebtnActionPerformed(java.awt.event.ActionEvent evt) {**

**writeatxtfile();**

**DefaultTableModel model=(DefaultTableModel)jTable1.getModel();**

**model.setRowCount(0);**

**show\_Members\_In\_jTable();**

**// TODO add your handling code here:**

**String query = "UPDATE students SET `Time` = '" + cmbArrivalTime.getSelectedItem() + "' WHERE contact = '" + txtContact.getText() + "'" ;**

**executeSQLQuery(query, "Updated");**

**ClearForm();**

**txtFirstname.requestFocus();**

**jTable1.repaint();**

**// TODO add your handling code here:**

**}**

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

**// Delete member usins a prepared statement**

**PreparedStatement pst = null;**

**Connection con = null;**

**try{**

**String sql = "DELETE from `students` WHERE Contact = ?";**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**pst = con.prepareStatement(sql);**

**pst.setString(1, txtFirstname.getText());**

**pst.executeUpdate();**

**JOptionPane.showMessageDialog(null, "Record removed");**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**int SelectedRowIndex = jTable1.getSelectedRow();**

**model.removeRow(SelectedRowIndex);**

**}catch(SQLException | HeadlessException e) {**

**JOptionPane.showMessageDialog(null, e);**

**}**

**jTable1.addNotify();**

**ClearForm();**

**txtFirstname.requestFocus();**

**}**

**/\*\***

**\* @param args the command line arguments**

**/\* Set the Nimbus look and feel \*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new DailyRegister().setVisible(true);**

**}**

**});**

**}**

**//A function that will create a connection to the database**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**JOptionPane.showMessageDialog(null, "Data " +message+ " sucessfully");**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null, "Data Not " +message+ " sucessfully");**

**}**

**}catch (Exception ex)**

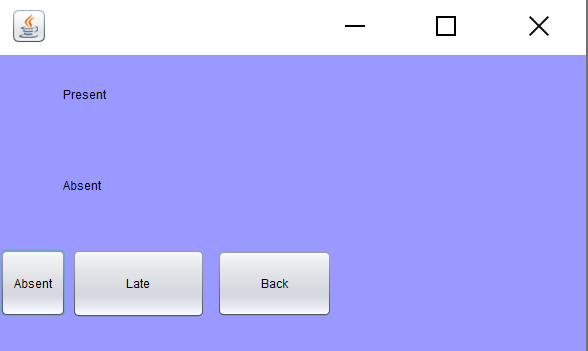
**{**

**ex.printStackTrace();**

**}**

**}**

**Analysis interface**

****

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**Absentee AbsentStudents=new Absentee();**

**AbsentStudents.setVisible(true);**

**this.dispose();// TODO add your handling code here:**

**}**

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**Late LateAttendance=new Late();**

**LateAttendance.setVisible(true);**

**this.dispose();**

**// TODO add your handling code here:**

**}**

**private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {**

**DailyRegister AnalysisInterface=new DailyRegister();**

**AnalysisInterface.setVisible(true);**

**this.dispose();// TODO add your handling code here:**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Analyse().setVisible(true);**

**}**

**});**

**}**

**public void present(){**

**String query = "SELECT COUNT(\*) AS count FROM students WHERE Time='Late' or Time='OnTime'";**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**PreparedStatement pst = con.prepareStatement(query);**

**ResultSet rs=pst.executeQuery(query);**

**if (rs.next())**

**{**

**presentlbl.setText(""+rs.getInt("count"));**

**}**

**}**

**catch (Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex,"error",JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**public void absent(){**

**String query = "SELECT COUNT(\*) AS count FROM students WHERE Time='Absent'";**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**PreparedStatement pst = con.prepareStatement(query);**

**ResultSet rs=pst.executeQuery(query);**

**if (rs.next())**

**{**

**absentlbl.setText(""+rs.getInt("count"));**

**}**

**}**

**catch (Exception ex)**

**{**

**JOptionPane.showMessageDialog(null,ex,"error",JOptionPane.ERROR\_MESSAGE);**

**}**

**}**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**System.out.println(message);**

**}**

**else**

**{**

**System.out.println(message+" are not being displayed");**

**}**

**}catch (Exception ex)**

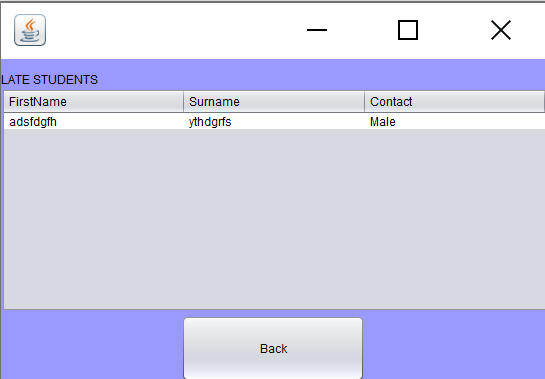
**{**

**ex.printStackTrace();**

**}**

**}**

**Late Students**

****

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**Analyse AnalysisInterface=new Analyse();**

**AnalysisInterface.setVisible(true);**

**this.dispose();// TODO add your handling code here:**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Late().setVisible(true);**

**}**

**});**

**}**

**public ArrayList<StudentRegister> getStudents()**

**{**

**ArrayList<StudentRegister> StudentList = new ArrayList<StudentRegister>();**

**Connection connection = getConnection();**

**String query = "Select \* FROM students Where Time='Late'";**

**Statement st;**

**ResultSet rs;**

**try**

**{**

**st = connection.createStatement();**

**rs = st.executeQuery(query);**

**StudentRegister attendance;**

**while (rs.next())**

**{**

**attendance = new StudentRegister(rs.getString("Name"),**

**rs.getString("Surname") ,rs.getString("Gender"),**

**rs.getString("Contact"),rs.getString("Time"));**

**StudentList.add(attendance);**

**}**

**}catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**return StudentList;**

**}**

**public void show\_Members\_In\_jTable()**

**{**

**ArrayList<StudentRegister> list = getStudents();**

**DefaultTableModel model = (DefaultTableModel)jTable1.getModel();**

**Object[] row = new Object[5];**

**for (int i = 0; i < list.size(); i ++)**

**{**

**row[0] = list.get(i).getName();**

**row[1] = list.get(i).getSurname();**

**row[2] = list.get(i).getGender();**

**row[3] = list.get(i).getContact();**

**row[4] = list.get(i).getTime();**

**model.addRow(row);**

**}**

**jTable1.setModel(model);**

**model.fireTableDataChanged();**

**}**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**JOptionPane.showMessageDialog(null, "Data " +message+ " sucessfully");**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null, "Data Not " +message+ " sucessfully");**

**}**

**}catch (Exception ex)**

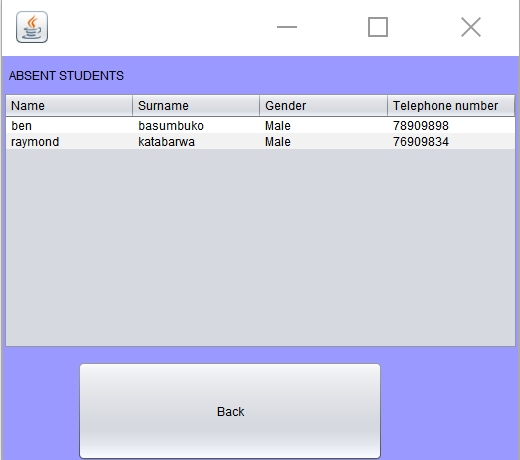
**{**

**ex.printStackTrace();**

**}**

**}**

**Absentees**

****

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**Analyse AnalysisInterface=new Analyse();**

**AnalysisInterface.setVisible(true);**

**this.dispose();// TODO add your handling code here:**

**}**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String args[]) {**

**/\* Set the Nimbus look and feel \*/**

**//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">**

**/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.**

**\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html**

**\*/**

**try {**

**for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {**

**if ("Nimbus".equals(info.getName())) {**

**javax.swing.UIManager.setLookAndFeel(info.getClassName());**

**break;**

**}**

**}**

**} catch (ClassNotFoundException ex) {**

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

**} catch (InstantiationException ex) {**

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

**} catch (IllegalAccessException ex) {**

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

**} catch (javax.swing.UnsupportedLookAndFeelException ex) {**

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

**}**

**//</editor-fold>**

**/\* Create and display the form \*/**

**java.awt.EventQueue.invokeLater(new Runnable() {**

**public void run() {**

**new Absentee().setVisible(true);**

**}**

**});**

**}**

**public ArrayList<StudentRegister> getStudents()**

**{**

**ArrayList<StudentRegister> StudentList = new ArrayList<StudentRegister>();**

**Connection connection = getConnection();**

**String query = "Select \* FROM students Where Time='Absent'";**

**Statement st;**

**ResultSet rs;**

**try**

**{**

**st = connection.createStatement();**

**rs = st.executeQuery(query);**

**StudentRegister attendance;**

**while (rs.next())**

**{**

**attendance = new StudentRegister(rs.getString("Name"),**

**rs.getString("Surname") ,rs.getString("Gender"),**

**rs.getString("Contact"),rs.getString("Time"));**

**StudentList.add(attendance);**

**}**

**}catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**return StudentList;**

**}**

**public void show\_Members\_In\_jTable()**

**{**

**ArrayList<StudentRegister> list = getStudents();**

**DefaultTableModel model = (DefaultTableModel)jTable2.getModel();**

**Object[] row = new Object[5];**

**for (int i = 0; i < list.size(); i ++)**

**{**

**row[0] = list.get(i).getName();**

**row[1] = list.get(i).getSurname();**

**row[2] = list.get(i).getGender();**

**row[3] = list.get(i).getContact();**

**row[4] = list.get(i).getTime();**

**model.addRow(row);**

**}**

**jTable2.setModel(model);**

**model.fireTableDataChanged();**

**}**

**public Connection getConnection()**

**{**

**Connection con;**

**try**

**{**

**con = DriverManager.getConnection("jdbc:mysql://localhost:3306/Studentregister", "root","");**

**return con;**

**}catch (Exception ex) {**

**ex.printStackTrace();**

**return null;**

**}**

**}**

**public void executeSQLQuery(String query, String message)**

**{**

**//Initialising the connection**

**Connection con = getConnection();**

**//declaring a variable to hold the SQL statement**

**Statement st;**

**try**

**{**

**//Creating the SQL Statement**

**st = con.createStatement();**

**//Testing to see if the query was executed**

**if (st.executeUpdate(query)==1)**

**{**

**JOptionPane.showMessageDialog(null, "Data " +message+ " sucessfully");**

**}**

**else**

**{**

**JOptionPane.showMessageDialog(null, "Data Not " +message+ " sucessfully");**

**}**

**}catch (Exception ex)**

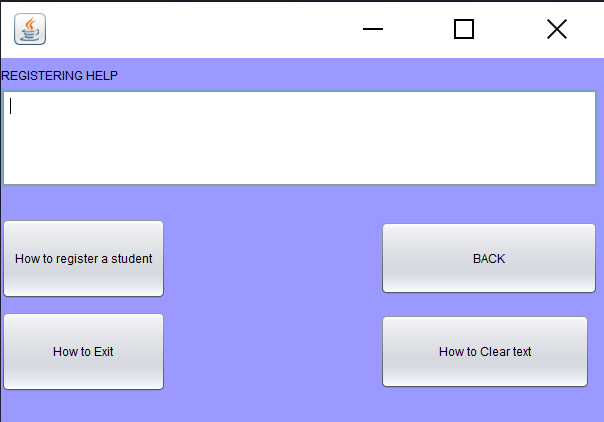
**{**

**ex.printStackTrace();**

**}**

**}**

**Registering help**

****

**private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("Write the First Name, Surname,Gender,Contact number and Arrival time then press add"); // TODO add your handling code here:**

**}**

**private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {**

**UserInterface back=new UserInterface();**

**back.setVisible(true);**

**this.dispose(); // TODO add your handling code here:**

**}**

**private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {**

**HelpText.setText("Press the Exit button to exit the app"); // TODO add your handling code here:**

**}**

**private void btnClearTextActionPerformed(java.awt.event.ActionEvent evt) {**

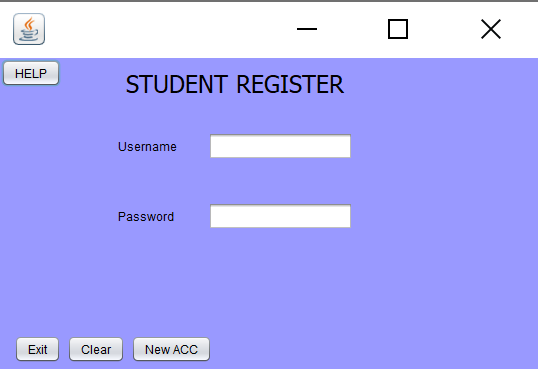
**HelpText.setText("Press the Clear button to remove the written Name, Password and Grade"); // TODO add your handling code here:**

**}**

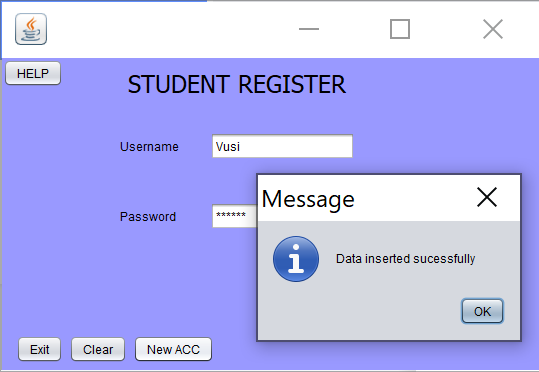
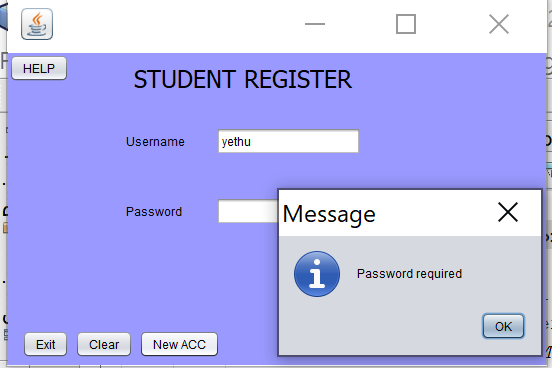
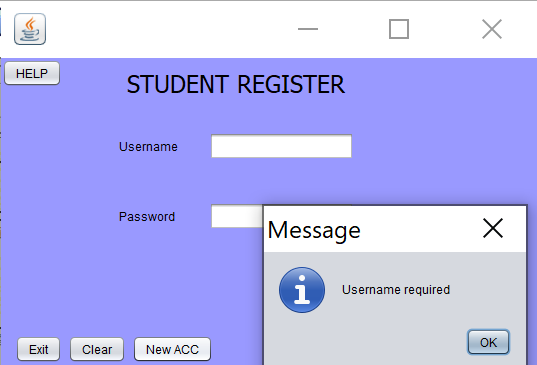
**Defective programming**

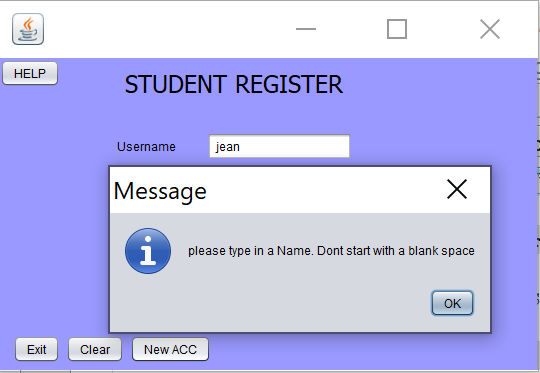
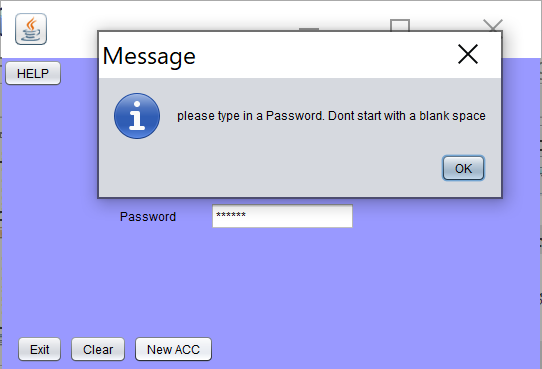
**The user is throughout limited, to allow no correct input to be taken in and reduce any possible errors that might make the program crash. The user is presented with buttons, text fields, combo boxes and tables in order to get the required data from the user. The user will receive error messages either through a message dialog or labels on the sides of the incorrect fields. The help button will be in interfaces that require some understanding to the program to operate or that simply require input from the user.**

**Test Plan**

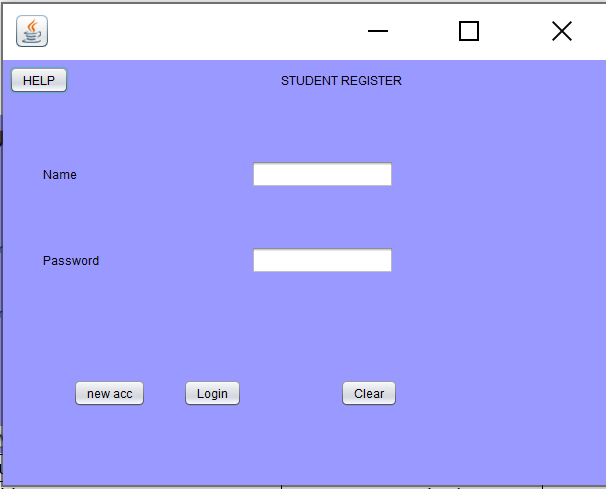
****

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Data | Input | Expected Result | Result |
| Presence check | nothing | Username required  Password required | Username required  Password required |
| standard | New name | Data inserted successfully | Data inserted successfully |
| Abnormal | Name/password start with a blank space | Error message stating “please type in a Name/Password (respectively) . Dont start with a blank space” | Error message displayed |

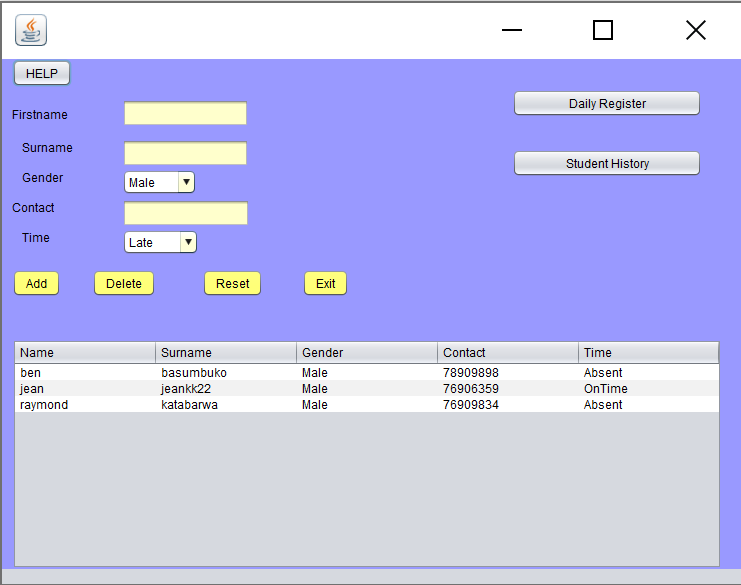
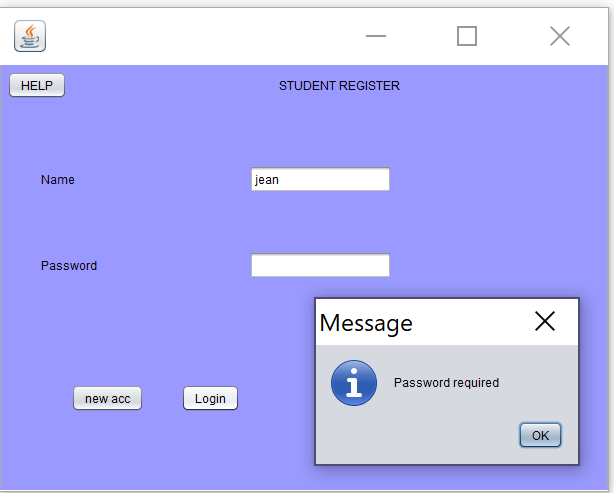
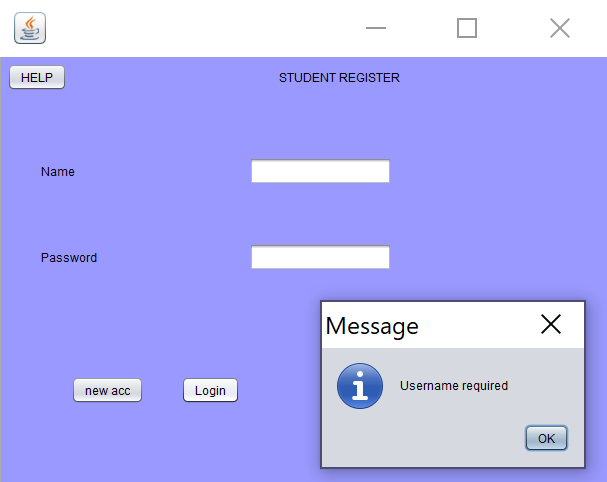
****

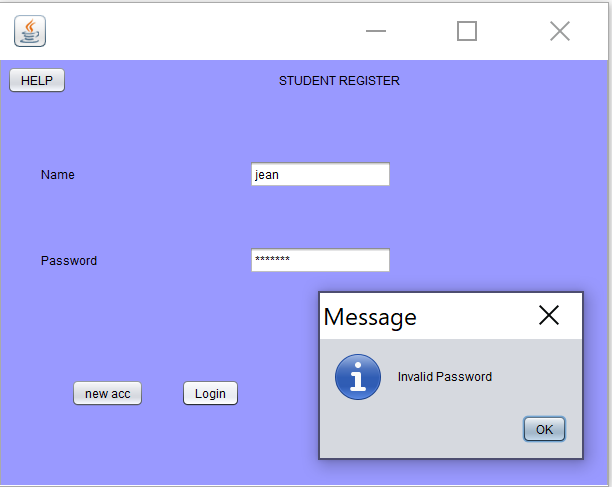
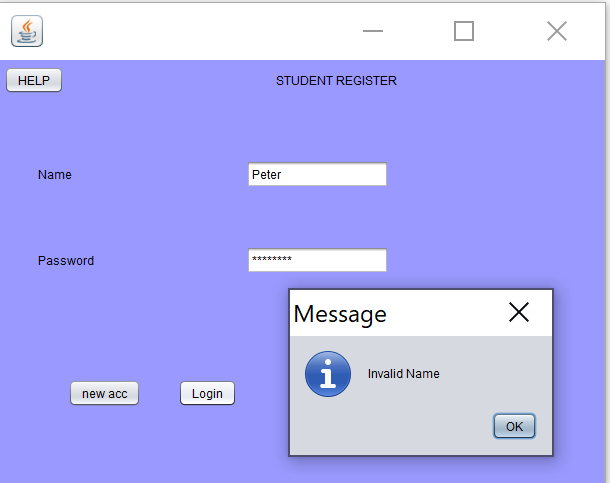
****

**Login**

****

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Data | Input | Expected Result | Result |
| Presence check | nothing | Username required  Password required | Username required  Password required |
| standard | Name exists/Password exists | Login screen dis appears and One time register User interface screen | Login screen dis appears and One time register User interface screen |
| Invalid | Unknown name/ password | Incorrect name/password errors are displayed | Error messages displayed |

****

****