





**Enjabulweni Independent School  
Grade 12  
Banzile Z. Nhlebela  
IT PAT  
Project Coding Document**

Contents

[**Persistent Storage** 4](#_Toc83844503)

[**Logging in as a User** 4](#_Toc83844504)

[**Creating a new Account** 6](#_Toc83844505)

[**Resetting The Password as the User** 8](#_Toc83844506)

[**Entering Current Username** 8](#_Toc83844507)

[**Entering Answer to security question** 9](#_Toc83844508)

[**Resetting the password** 10](#_Toc83844509)

[**Accessing the different forms from main menu** 11](#_Toc83844510)

[**Student Log** 11](#_Toc83844511)

[**Student Offence Register** 11](#_Toc83844512)

[**Edit Profile** 11](#_Toc83844513)

[**Parent Log** 11](#_Toc83844514)

[**Offences Screen** 11](#_Toc83844515)

[**Help Screen** 11](#_Toc83844516)

[**Updating and Deleting user data** 12](#_Toc83844517)

[**Update** 12](#_Toc83844518)

[**Delete** 14](#_Toc83844519)

[**Inserting, Updating and Deleting Student Data** 15](#_Toc83844520)

[**Insert** 15](#_Toc83844521)

[**Update** 15](#_Toc83844522)

[**Delete** 16](#_Toc83844523)

[**Inserting, Updating and Deleting Student Offence Registration Data** 19](#_Toc83844524)

[**Insert** 19](#_Toc83844525)

[**Update** 19](#_Toc83844526)

[**Delete** 20](#_Toc83844527)

[**Inserting, Updating and Deleting Parent Data** 24](#_Toc83844528)

[**Insert** 24](#_Toc83844529)

[**Update** 24](#_Toc83844530)

[**Delete** 26](#_Toc83844531)

[**Inserting, Updating and Deleting Offence Data** 28](#_Toc83844532)

[**Insert** 28](#_Toc83844533)

[**Update** 28](#_Toc83844534)

[**Delete** 29](#_Toc83844535)

[**Accessing Help Topics** 31](#_Toc83844536)

# **Persistent Storage**

## **Logging in as a User**

Logging in is handled by the Login Page form. The code below searches the accountdetails database and checks the username and password entered in the form against those in the database. Should the information match, the user is granted access to the program and should there be a mismatch, an error message is displayed.

String query = "SELECT \* FROM `accountdetails` WHERE `Username` = '" + txtUN.getText() + "' AND `Password` = '" + pfPwd.getText() + "'";

//Begin try-catch

try

{

//get connection to database

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

pst = con.prepareStatement(query);

//get all data from database

ResultSet rs = pst.executeQuery(query);

MainMenu menu = new MainMenu();

//If data is found in database/matches data in database

if(rs.next())

{

//Begin try-catch

try

{

//Write Username, Password, Position to new text file

try (PrintWriter px = new PrintWriter(new FileWriter(new File("temp.txt"))))

{

String s = txtUN.getText() + "#" + pfPwd.getText() + "#" + rs.getString("Position");

px.print(s);

}

}

catch(SQLException | IOException ex)

{

//Display if Exception appears

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

}

//end of try-catch

//Set fields to green colour to show successful login

lblUNerr.setForeground(g);

lblPwdErr.setForeground(g);

pfPwd.setBorder(a);

txtUN.setBorder(a);

pfPwd.setForeground(g);

txtUN.setForeground(g);

//if user is male

if(rs.getString("Gender").equals("Male"))

{

//Welcome message to be displayed when login is successful

JOptionPane.showMessageDialog(null, "Login Successful"+ "\nWelcome Mr. " + rs.getString("Lastname"), rs.getString("Position"), JOptionPane.INFORMATION\_MESSAGE);

}

//if user is female

else

{

//Welcome message to be displayed when login is successful

JOptionPane.showMessageDialog(null, "Login Successful"+ "\nWelcome Ms. " + rs.getString("Lastname"), rs.getString("Position"), JOptionPane.INFORMATION\_MESSAGE);

}

//Show Main Menu screen

menu.setVisible(true);

//Remove current screen from view

this.dispose();

}

//If data is not found/ does not match data in database

else

{

//Set fields and labels to red to indicate unsuccessful login

pfPwd.setBorder(b);

txtUN.setBorder(b);

pfPwd.setForeground(r);

txtUN.setForeground(r);

txtUN.requestFocus();

lblUNerr.setForeground(r);

lblPwdErr.setForeground(r);

//Message indicating unsuccessful login

JOptionPane.showMessageDialog(null, "Username or Password Incorrect", "Error", JOptionPane.ERROR\_MESSAGE);

}

//close connection to database

con.close();

## **Creating a new Account**

Creating a new account is handled by the New Account screen. The code below uses the accountdetails database and the security database. The user enters their information and the program inserts said data into the database if all the data is valid. If not, all data is valid, appropriate messages are displayed indicating where the errors are. Successful creation of an account is followed by the display of the login screen and a message indicating a successful creation. The user is then prompted to enter a security question along with an answer. This information is stored in the security database.

//Insert into databases

//begin try-catch

try

{

//get connection to database

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

//prompt user to enter new security question and answer

String s = JOptionPane.showInputDialog(null, "Enter new Security Question");

String ss = JOptionPane.showInputDialog(null, "Enter new Answer");

String query2 = "INSERT INTO security (`Username`, `Question`, `Answer`) VALUES(?, ?, ?)";

ps = con.prepareStatement(query2);

//insert question, answer and username into 'Security' table

ps.setString(1, txtUN.getText());

ps.setString(2, s);

ps.setString(3, ss);

//update the database

ps.executeUpdate();

String query = "INSERT INTO `accountdetails`(`Username`, `Password`, `Firstname`, `Lastname`, `Gender`, `DateOfBirth`, `Position`) VALUES (?, ?, ?, ?, ?, ?, ?)";

pst = con.prepareStatement(query);

//insert user details into the table 'accountdetails'

pst.setString(1, txtUN.getText());

pst.setString(2, pfPass.getText());

pst.setString(3, txtFName.getText());

pst.setString(4, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else if(cbFemale.isSelected())

{

gender = "Female";

}

pst.setString(5, gender);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOB.getDate()));

pst.setString(7, cmbPos.getSelectedItem().toString());

pst.executeUpdate();

//close connection to database

con.close();

//display message indicating successful creation of account

JOptionPane.showMessageDialog(null, "New Account Created Successfully");

//set login page to be visible

LoginPage log = new LoginPage();

log.setVisible(true);

//remove current page from view

this.dispose();

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

## **Resetting The Password as the User**

The code below demonstrates how the user can reset their password using the Forgot Password screen. Following a series of steps, the user answers their set security question and upon answering correctly, are granted permission to changer their password.

### **Entering Current Username**

//begin try-catch

try

{

//get connection to database

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

String query = "SELECT \* FROM security WHERE `Username` = '" + txtCUN.getText() + "'";

pst = con.prepareStatement(query);

//obtain result of query from database

ResultSet rs = pst.executeQuery(query);

//if username found

if(rs.next())

{

//endable enter button

btnEnter.setEnabled(true);

//make username field green

txtCUN.setForeground(g);

txtCUN.setBorder(a);

//disable username field and 'Search' button

txtCUN.setEnabled(false);

btnSearch.setEnabled(false);

//enable answer field

txtANS.setEnabled(true);

//display message indicating success

JOptionPane.showMessageDialog(null, "Username found" + "\nAnswer Security Question in the space provided" + "\nThen select \'Enter\'");

//display security question

JOptionPane.showMessageDialog(null, rs.getString("Question"));

}

//if username not found

else

{

//display message indicating error

JOptionPane.showMessageDialog(null, "Username Not Found", "Error", JOptionPane.ERROR\_MESSAGE);

//turn username field red to indicate error

txtCUN.setForeground(r);

txtCUN.setBorder(b);

txtCUN.requestFocus();

}

//close connection to database

con.close();

}

catch(HeadlessException | SQLException ex)

{

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

}

### **Entering Answer to security question**

//begin try-catch

try

{

//get connection to database

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

String query = "SELECT \* FROM security WHERE `Answer` = '" + txtANS.getText() + "'";

pst = con.prepareStatement(query);

//search for answer with regards to question and username

ResultSet rs = pst.executeQuery(query);

String query2 = "SELECT \* FROM security WHERE `Username` = '" + txtCUN.getText() + "'";

ps = con.prepareStatement(query2);

ResultSet r = ps.executeQuery(query2);

//if answer matches question and username

if(rs.next())

{

//make field green to indicate correct answer

txtANS.setBorder(a);

txtANS.setForeground(g);

//display message indicating next step

JOptionPane.showMessageDialog(null, "Correct Answer.\n" + "Enter new Password\n" + "Then select 'Enter'");

//enable password fields and disable answer field

pfNPass.setEnabled(true);

pfCPass.setEnabled(true);

txtANS.setEnabled(false);

}

//if answer is wrong

else

{

//set field to red to indicate error

txtANS.setBorder(b);

txtANS.setForeground(re);

//display message explaining error

JOptionPane.showMessageDialog(null, "Incorrect Answer", "Error", JOptionPane.ERROR\_MESSAGE);

//display question again

if(r.next())

{

JOptionPane.showMessageDialog(null, r.getString("Question"));

}

}

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try catch

### **Resetting the password**

LoginPage log = new LoginPage();

//begin try-catch

try

{

//get connection to database

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

String query = "UPDATE accountdetails SET `Password` = ? WHERE `Username` = '" + txtCUN.getText() + "'";

pst = con.prepareStatement(query);

//if new password and confirm password are the same

if(pfNPass.getText().equals(pfCPass.getText()))

{

//update security database

pst.setString(1, pfCPass.getText());

pst.executeUpdate();

//set both fields to green to indicate success

pfCPass.setBorder(a);

pfNPass.setBorder(a);

pfCPass.setForeground(g);

pfNPass.setForeground(g);

//display message indicating success

JOptionPane.showMessageDialog(null, "Password Changed Successfully", "Success", JOptionPane.INFORMATION\_MESSAGE);

//display login form

log.setVisible(true);

//close current screen

this.dispose();

}

//if passwords do not match

else

{

//make both fields red to indicate error

pfCPass.setBorder(b);

pfNPass.setBorder(b);

pfCPass.setForeground(re);

pfNPass.setForeground(re);

//display message explaining error

JOptionPane.showMessageDialog(null, "Passwords Do Not Match", "Failure", JOptionPane.ERROR\_MESSAGE);

}

}

catch(HeadlessException | SQLException ex)

{

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

} //end of try-catch

## **Accessing the different forms from main menu**

Upon entering the main menu, the user can access different forms using the buttons provided. The code below demonstrates how each form is accessed.

### **Student Log**

//set student log screen to be visible

StudentLog stulog = new StudentLog();

stulog.setVisible(true);

//close current screen

this.dispose();

### **Student Offence Register**

//set consultations form to be visible

StudentOffenceRegister con = new StudentOffenceRegister();

con.setVisible(true);

//close current screen

this.dispose();

### **Edit Profile**

//set edit profile screen to be visible

EditProfile ep = new EditProfile();

ep.setVisible(true);

//close current screen

this.dispose();

### **Parent Log**

//set parent log form to be visible

ParentLog pl = new ParentLog();

pl.setVisible(true);

//close current screen

this.dispose();

### **Offences Screen**

//set offence log form to be visible

OffenceLog off = new OffenceLog();

off.setVisible(true);

//close current screen

this.dispose();

### **Help Screen**

//set help screen to be visible

Help h = new Help();

h.setVisible(true);

## **Updating and Deleting user data**

The user is able to update all their information using the Edit Profile Screen. The code below demonstrates how the user is able to update or even delete their information.

### **Update**

//Update databases

try

{

//get connection to database

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

//read temp.txt file

try(Scanner sc = new Scanner(new FileReader("temp.txt")))

{

String line = sc.nextLine();

try(Scanner ac = new Scanner(line).useDelimiter("#"))

{

String un = ac.next();

String p = ac.next();

String pos = ac.next();

//create login object

Login log = new Login(un, p, pos);

//message asking user if they wish to update their security question and answer

int i = JOptionPane.showConfirmDialog(null, "Do you wish to update your security question?", "Security Question", JOptionPane.YES\_NO\_OPTION);

//if user clicks yes

if(i == 0)

{

//ask for new question and answer

String s = JOptionPane.showInputDialog(null, "Enter new Security Question");

String ss = JOptionPane.showInputDialog(null, "Enter new Answer");

String query2 = "UPDATE security SET `Username` = ?, `Question` = ?, `Answer` = ? WHERE Username = '" + log.getUsername() + "'";

ps = con.prepareStatement(query2);

//update table 'security'

ps.setString(1, txtUN.getText());

ps.setString(2, s);

ps.setString(3, ss);

ps.executeUpdate();

}

String query = "UPDATE accountdetails SET `Username`= ?, `Password`= ?, `Firstname`= ?, `Lastname`= ?, `Gender`= ?, `DateOfBirth`= ?, `Position`= ? WHERE `Username` = '" + log.getUsername() + "'";

pst = con.prepareStatement(query);

//update data in textfile

log.setUsername(txtUN.getText());

log.setPassword(pfPass.getText());

log.setPosition(cmbPos.getSelectedItem().toString());

try (PrintWriter px = new PrintWriter(new FileWriter(new File("temp.txt"))))

{

String q = log.getUsername() + "#" + log.getPassword() + "#" + log.getPosition();

px.print(q);

}

//update data in database 'accountdetails'

pst.setString(1, txtUN.getText());

pst.setString(2, pfPass.getText());

pst.setString(3, txtFName.getText());

pst.setString(4, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else

{

gender = "Female";

}

pst.setString(5, gender);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOB.getDate()));

pst.setString(7, cmbPos.getSelectedItem().toString());

pst.executeUpdate();

//display after successful update

JOptionPane.showMessageDialog(null, "Profile Updated Successfully");

//set menu page visible

MainMenu menu = new MainMenu();

menu.setVisible(true);

//remove current page from view

setVisible(false);

}

}

}

catch(HeadlessException | SQLException | IOException ex)

{

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

}

//end of try-catch

### **Delete**

//begin try-catch

try

{

//get connection to database

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

String query = "DELETE FROM accountdetails WHERE Username = '" + txtUN.getText() + "'";

String query2 = "DELETE FROM security WHERE Username = '" + txtUN.getText() + "'";

//delete data for user in both tables 'security' and 'accountdetails'

pst = con.prepareStatement(query);

ps = con.prepareStatement(query2);

pst.executeUpdate();

ps.executeUpdate();

clear("", yellow, a);

JOptionPane.showMessageDialog(null, "Deleted Successfully");

//remove data from text file

//begin try-catch

try

{

try (PrintWriter px = new PrintWriter(new FileWriter(new File("temp.txt"))))

{

px.print("username#password#position");

}

}

catch(Exception ex)

{

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

}

//end of try-catch

LoginPage log = new LoginPage();

//set login page visible

log.setVisible(true);

//remove current screen from view

setVisible(false);

}

catch(HeadlessException | SQLException ex)

{

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

}

## **Inserting, Updating and Deleting Student Data**

The user is able to update, delete and insert data regarding any student in the system. The code below demonstrates how the user can perform all of these actions

### **Insert**

//begin try-catch

try

{

String query = "INSERT INTO `studentLog`(`StudentID`, `FirstName`, `LastName`, `Gender`, `Grade`, `DateOfBirth`) VALUES (?, ?, ?, ?, ?, ?)";

//get connection to database

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

pst = con.prepareStatement(query);

//insert data into database

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtFName.getText());

pst.setString(3, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else if(cbFemale.isSelected())

{

gender = "Female";

}

pst.setString(4, gender);

pst.setString(5, cmbGrade.getSelectedItem().toString());

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOB.getDate()));

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display data on jtable

JOptionPane.showMessageDialog(null, "Registered Successfully");

clear("", bl, c);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Update**

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query = "UPDATE studentLog SET `StudentID` = ?, `FirstName` = ?, `LastName` = ?, `Gender` = ?, `Grade` = ?, `DateOfBirth` = ? WHERE `StudentID` = '" + value + "'";

String query2 = "UPDATE parents SET `StudentID` = '" + txtStudentID.getText() + "' WHERE `StudentID = '" + value + "'";

String query3 = "UPDATE conLog SET `StudentID` = '" + txtStudentID.getText() + "' WHERE `StudentID = '" + value + "'";

pst = con.prepareStatement(query);

pst2 = con.prepareStatement(query2);

pst2 = con.prepareStatement(query3);

//update database

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtFName.getText());

pst.setString(3, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else if(cbFemale.isSelected())

{

gender = "Female";

}

pst.setString(4, gender);

pst.setInt(5, Integer.valueOf(cmbGrade.getSelectedItem().toString()));

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOB.getDate()));

pst.executeUpdate();

pst2.executeUpdate();

pst3.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display message indicating success

JOptionPane.showMessageDialog(null, "Updated Successfully");

clear("", bl, c);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Delete**

if(txtStudentID.getText().isEmpty())

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblSIDerr.setText("Select data from table\*");

//set label and field to red to indicate error

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

else

{

try

{

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

String query = "SELECT \* FROM studentLog WHERE StudentID = '" + txtStudentID.getText() + "'";

pst = con.prepareStatement(query);

ResultSet rs = pst.executeQuery(query);

if(rs.next())

{

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query2 = "DELETE FROM studentlog WHERE StudentID = '" + value + "'";

String query3 = "DELETE FROM parents WHERE StudentID = '" + value + "'";

String query4 = "DELETE FROM conlog WHERE StudentID = '" + value + "'";

pst = con.prepareStatement(query2);

pst2 = con.prepareStatement(query3);

pst3 = con.prepareStatement(query4);

//delete data from database

pst.executeUpdate();

pst2.executeUpdate();

pst3.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

clear("", bl, c);

//display message indicating success

JOptionPane.showMessageDialog(null, "Deleted Successfully");

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

}

else

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblSIDerr.setText("SELECT DATA FROM TABLE\*");

//set label and field to red to indicate error

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

}

catch(HeadlessException | SQLException ex)

{

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

}

}

## **Inserting, Updating and Deleting Student Offence Registration Data**

The user is able to update, delete and insert data regarding any student offence data in the system. The code below demonstrates how the user can perform all of these actions

### **Insert**

//begin try-catch

try

{

String query = "INSERT INTO `conLog`(`StudentID`, `OffenceID`, `Sanction`, "

+ "`Subject`, `ParentsTold`, `DateOfOffence`) VALUES (?, ?, ?, ?, ?, ?)";

//get coonection to database

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

//insert data into database

pst = con.prepareStatement(query);

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtOffenceID.getText());

pst.setString(3, cmbSanction.getSelectedItem().toString());

pst.setString(4, cmbSubject.getSelectedItem().toString());

if(cbYes.isSelected())

{

ab = "Yes";

}

else if(cbNo.isSelected())

{

ab = "No";

}

pst.setString(5, ab);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOO.getDate()));

pst.executeUpdate();

//display data in jtable

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display message to indicate success

JOptionPane.showMessageDialog(null, "Registered Successfully", "Register", JOptionPane.INFORMATION\_MESSAGE);

clear();

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Update**

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query = "UPDATE conLog SET `StudentID` = ?, `OffenceID` = ?, `Sanction` = ?, `Subject` = ?, `ParentsTold` = ?, `DateOfOffence` = ? WHERE `StudentID` = '" + value + "'";

//update database

pst = con.prepareStatement(query);

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtOffenceID.getText());

pst.setString(3, cmbSanction.getSelectedItem().toString());

pst.setString(4, cmbSubject.getSelectedItem().toString());

if(cbYes.isSelected())

{

ab = "Yes";

}

else if(cbNo.isSelected())

{

ab = "No";

}

pst.setString(5, ab);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

pst.setString(6, sdf.format(jdcDOO.getDate()));

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

//display data in database on jtable

Default();

//display message indicating success

JOptionPane.showMessageDialog(null, "Updated Successfully", "Update", JOptionPane.INFORMATION\_MESSAGE);

clear();

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Delete**

//Begin try-catch

try

{

//get connection to database

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

if(txtStudentID.getText().isEmpty())

{

changeColor(blue, a);

setLabels("", blue);

//display green stars to indicate correct input

lblSIDerr.setText("Select Data from Table\*");

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

else

{

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query = "SELECT\* FROM studentLog WHERE StudentID = '" + value + "'";

ResultSet rs = (con.createStatement()).executeQuery(query);

//if student ID found in table

if(!rs.next())

{

changeColor(blue, a);

setLabels("", blue);

//display green stars to indicate correct input

lblSIDerr.setText("Select Data from Table\*");

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

else

{

int j = JOptionPane.showConfirmDialog(null, "Student Information: \nName: " + rs.getString("Firstname") + " " + rs.getString("LastName") + "\nGrade: " + rs.getInt("Grade"), "Delete Offence for this student", JOptionPane.YES\_NO\_OPTION);

if(j == 0)

{

//display red stars to indicate correct input

lblSIDerr.setText("\*\*\*\*");

}

else

{

changeColor(blue, a);

setLabels("", blue);

//display green stars to indicate correct input

lblSIDerr.setText("Choose Correct Student\*");

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

}

}

}

catch(SQLException ex)

{

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

}

//end of try-catch

if(lblSIDerr.getText().equals("\*\*\*\*"))

{

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

//get values from each column for specified row

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String value1 = jTDisplay.getModel().getValueAt(num, 1).toString();

String value2 = jTDisplay.getModel().getValueAt(num, 2).toString();

String value3 = jTDisplay.getModel().getValueAt(num, 3).toString();

String value4 = jTDisplay.getModel().getValueAt(num, 4).toString();

String value5 = jTDisplay.getModel().getValueAt(num, 5).toString();

String query = "SELECT \* FROM conLog WHERE StudentID = '" + value + "' AND Offence = '"

+ value1 + "' AND Sanction = '" + value2 + "' AND Subject = '"

+ value3 + "' AND ParentsTold = '" + value4 + "' AND DateOfOffence = '" + value5 + "'";

ps = con.prepareStatement(query);

ResultSet rs = ps.executeQuery(query);

int temp = 0;

if(rs.next())

{

temp = rs.getInt("ID");

}

String query2 = "DELETE FROM conLog WHERE ID = '" + temp + "'";

pst = con.prepareStatement(query2);

//delete data from database

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

//display data in database

Default();

clear();

//display message indicating success

JOptionPane.showMessageDialog(null, "Deleted Successfully", "Delete", JOptionPane.INFORMATION\_MESSAGE);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

## **Inserting, Updating and Deleting Parent Data**

The user is able to update, delete and insert data regarding any student parent data in the system. The code below demonstrates how the user can perform all of these actions

### **Insert**

//begin try-catch

try

{

String query = "INSERT INTO `parents`(`StudentID`, `Firstname`, `Lastname`, `Gender`, `Relationship`, `Cellnumber`) VALUES (?, ?, ?, ?, ?, ?)";

//get connection to database

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

pst = con.prepareStatement(query);

//insert data into database

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtFName.getText());

pst.setString(3, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else if(cbFemale.isSelected())

{

gender = "Female";

}

pst.setString(4, gender);

pst.setString(5, cmbRel.getSelectedItem().toString());

pst.setString(6, number);

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display data on jtable

JOptionPane.showMessageDialog(null, "Registered Successfully");

clear("", bl, c);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Update**

//if all data is valid

if(lblSIDerr.getText().equals("\*\*\*\*") && lblFNerr.getText().equals("\*\*\*\*") && lblLNerr.getText().equals("\*\*\*\*") && lblRELerr.getText().equals("\*\*\*\*") && lblGENerr.getText().equals("\*\*\*\*") && lblCNerr.getText().equals("\*\*\*\*"))

{

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query = "UPDATE parents SET `StudentID` = ?, `Firstname` = ?, `Lastname` = ?, `Gender` = ?, `Relationship` = ?, `Cellnumber` = ? WHERE `StudentID` = '" + value + "'";

String query2 = "UPDATE studentLog SET `StudentID` = '" + txtStudentID.getText() + "' WHERE `StudentID` = '" + value + "';";

pst = con.prepareStatement(query);

ps = con.prepareStatement(query2);

//update database

pst.setString(1, txtStudentID.getText());

pst.setString(2, txtFName.getText());

pst.setString(3, txtLName.getText());

if(cbMale.isSelected())

{

gender = "Male";

}

else if(cbFemale.isSelected())

{

gender = "Female";

}

pst.setString(4, gender);

pst.setString(5, cmbRel.getSelectedItem().toString());

pst.setString(6, number);

pst.executeUpdate();

ps.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display message indicating success

JOptionPane.showMessageDialog(null, "Updated Successfully");

clear("", bl, c);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Delete**

if(txtStudentID.getText().isEmpty())

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblSIDerr.setText("Select data from table\*");

//set label and field to red to indicate error

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

else

{

try

{

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

String query = "SELECT \* FROM parents WHERE StudentID = '" + txtStudentID.getText() + "'";

pst = con.prepareStatement(query);

ResultSet rs = pst.executeQuery(query);

if(rs.next())

{

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query2 = "DELETE FROM parents WHERE StudentID = '" + value + "'";

pst = con.prepareStatement(query2);

//delete data from database

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

clear("", bl, c);

//display message indicating success

JOptionPane.showMessageDialog(null, "Deleted Successfully");

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

}

else

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblSIDerr.setText("SELECT DATA FROM TABLE\*");

//set label and field to red to indicate error

lblSIDerr.setForeground(r);

txtStudentID.setBorder(b);

txtStudentID.setForeground(r);

}

}

catch(HeadlessException | SQLException ex)

{

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

}

## **Inserting, Updating and Deleting Offence Data**

The user is able to update, delete and insert data regarding any student parent data in the system. The code below demonstrates how the user can perform all of these actions

### **Insert**

//begin try-catch

try

{

String query = "SELECT \* FROM offences WHERE `OffenceID` = '" + txtOffenceID.getText() + "';";

//get connection to database

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

pst = con.prepareStatement(query);

ResultSet rs = pst.executeQuery(query);

//if offence ID found

if(rs.next())

{

//display message indicating error

lblOIDerr.setText("Offence ID already exists\*");

lblOIDerr.setForeground(r);

txtOffenceID.setBorder(b);

txtOffenceID.setForeground(r);

}

//if not found

else

{

//display message indicating correct input

lblOIDerr.setText("\*\*\*\*");

lblOIDerr.setForeground(g);

txtOffenceID.setBorder(a);

txtOffenceID.setForeground(g);

}

}

catch(SQLException ex)

{

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

}

//end of try-catch

### **Update**

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query = "UPDATE offences SET `OffenceID` = ?, `Description` = ?, `Level` = ? WHERE `OffenceID` = '" + value + "'";

String query2 = "UPDATE conLog SET `OffenceID` = '" + txtOffenceID.getText() + "WHERE `OffenceID` = '" + value + "'";

pst = con.prepareStatement(query);

ps = con.prepareStatement(query2);

//update database

pst.setString(1, txtOffenceID.getText());

pst.setString(2, txtDescription.getText());

if(cb1.isSelected())

{

level = 1;

}

else if(cb2.isSelected())

{

level = 2;

}

else if(cb3.isSelected())

{

level = 3;

}

else if(cb4.isSelected())

{

level = 4;

}

pst.setInt(3, level);

pst.executeUpdate();

ps.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

//display message indicating success

JOptionPane.showMessageDialog(null, "Updated Successfully");

clear("", bl, c);

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

### **Delete**

if(txtOffenceID.getText().isEmpty())

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblOIDerr.setText("Select data from table\*");

//set label and field to red to indicate error

lblOIDerr.setForeground(r);

txtOffenceID.setBorder(b);

txtOffenceID.setForeground(r);

}

else

{

try

{

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

String query = "SELECT \* FROM offences WHERE OffenceID = '" + txtOffenceID.getText() + "'";

pst = con.prepareStatement(query);

ResultSet rs = pst.executeQuery(query);

if(rs.next())

{

//begin try-catch

try

{

//get connection to database

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

int num = jTDisplay.getSelectedRow();

String value = jTDisplay.getModel().getValueAt(num, 0).toString();

String query2 = "DELETE FROM offences WHERE OffenceID = '" + value + "'";

pst = con.prepareStatement(query2);

//delete data from database

pst.executeUpdate();

DefaultTableModel model = (DefaultTableModel)jTDisplay.getModel();

model.setRowCount(0);

Default();

clear("", bl, c);

//display message indicating success

JOptionPane.showMessageDialog(null, "Deleted Successfully");

}

catch(HeadlessException | SQLException ex)

{

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

}

//end of try-catch

}

else

{

changeColor("", bl, c);

//error message to be displayed if character(s) not letters

lblOIDerr.setText("SELECT DATA FROM TABLE\*");

//set label and field to red to indicate error

lblOIDerr.setForeground(r);

txtOffenceID.setBorder(b);

txtOffenceID.setForeground(r);

}

}

catch(HeadlessException | SQLException ex)

{

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

}

}

## **Accessing Help Topics**

On the Help screen, the user has access to various help topics that will assist them throughout the program. The code below demonstrates how the user accesses one of the topics. The code is almost identical for all the topics only differing in the names of the text files in which the data is stored.

//begin try-catch

try

{

//open About.txt file for reading

try(Scanner sc = new Scanner(new FileReader("About.txt")))

{

String temp = "";

//while the file has another line

while(sc.hasNextLine())

{

//place line in a string variable until whole file has been read

temp += sc.nextLine() + "\n";

}

//display all the lines

taDisplay.setText(temp);

}

}

catch(Exception ex)

{

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

}

//end of try-catch