Criterion C Development Aditya Dadia; 004974-0023 **Criterion C: Development Complex Techniques used for GUI and JDBC MySQL Queries** I used MySQL Queries and connected my forms to the database on MySQL using the MySQLConnect class as shown in figure 1. * To change this license header, choose License Headers in Project Properties. * To change this template file, choose Tools | Templates * and open the template in the editor. */ package gui; import java.sql.*; import javax.swing.*; 뒨 Ę /** * @author aditya L public class MySqlConnect {
 static Connection conn = null;
 public static Connection ConnectionDB() throws SQLException Ę { try { String url = "jdbc:mysql://localhost:3306/employee"; String username = "root"; String password = ""; conn = DriverManager.getConnection(url, username, password); //JOptionPane.showMessageDialog(null, "Connect to database"); return conn; -} catch(Exception e) { JOptionPane.showMessageDialog(null,e); return null; 3 } } Figure 1

| Criterion C Development | | | | | |
|-------------------------|--------|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Heermanne | | | | | |
| Username | | | | | |
| Password | | | | | |
| | | | | | |
| | | | | | |
| | Submit | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Figure 2 shows us the login form, wherein the username and the password have to be typed to proceed. There is another login table in the database of employee.

| # | Name | Age | Gender | Address |
|---|--------|------|--------|---------|
| 1 | asdfgh | 6542 | jhgf | hgfd |
| 2 | test | 45 | Male | Home |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Figure 3

Figure 3 is the SQL table that stores the username and password. In case of any of the two being incorrect, an error message is shown; else a welcome user dialog is shown which then moves on to the menu screen.

The example below shows an incorrect username and password being typed in the box.

| Criterion C Development | | | | |
|-------------------------|------------------------------|--|--|--|
| | | | | |
| Username Password | incorrect_user ****** Submit | | | |
| | | | | |
| (0 0 0 | Access Denied | | | |
| | alid Username or password | | | |
| | ОК | | | |

Figure 4 shows the error message that is displayed for the incorrect login.

| Criterion C Develop | oment |
|----------------------|------------------------------------|
| | |
| Username Password | sandeepdadia ******** Submit |
| | Message |
| U Wel | lcome User |
| | ОК |

Figure 5 shows us the correct fields for login, and the success message. I have used a similar system of verification for all of my other forms, and used JOptionPane to display these messages

| Criterion C Development | Aditya Dadia; 004974-0023 |
|--|---|
| <pre>private void bSubmitActionPerformed(java.awt.event.ActionEvent evt) {</pre> | |
| <pre>private void bubbletectometricinet()voidatectorectorectorectorectorectorectorecto</pre> | |
| <pre>{ JOptionPane.showMessageDialog(null,"Welcome User"); this.dispose(); new Menu().setVisible(true); </pre> | |
| <pre>} else { JOptionPane.showMessageDialog(null,"Invalid Username or passwor</pre> | ","Access Denied",JOptionPane.ERROR_MESSAGE); |
| } | |
| } | |
| <pre>catch(SQLException HeadlessException e)</pre> | |
| <pre>JOptionPane.showMessageDialog(null, e); }</pre> | |
| 1 | |
| ſ | |
| Figure 6 | |

Figure 6 shows the code that I used for logging in.

| | • | | | | | | |
|---|------|-----|------|--------|--------|------------|------|
| 1 | File | Add | Edit | Search | Delete | Get Report | Help |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| l | | | | | | | |
| ł | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Figure 7

Figure 7 is the menu screen, the basic form from where all the other operations can be accessed.

Criterion C Development Click on:

Aditya Dadia; 004974-0023

- 1) File to get the exit option.
- 2) Add to get the Add User and Add Employee option.
- 3) Edit to get the Edit Employee option.
- 4) Delete to get the delete user and Delete Employee option.
- 5) Get Report to view the data of employees.
- 6) Help to get the salary slip.



Figure 8

Figure 8 shows the code for a part of the menu form, that connects the user to a few other forms.

| Vame test Age 45 Address 40 Medical 10 Past Salary 123 New Salary 123 Educational Qualification 11 Designations 46 Remarks 500 | | | | |
|--|---------------------------|----------------------|----------------------|-------------------|
| NametestAge45Age45GenderMaleAddressHorrAddressMr. 1ReferenceMr. 1MedicalDiabEmploymentWorlPast Salary123New Salary123Educational QualificationLLMDesignationsHeanRemarksDual | elopment | | Aditya D | adia; 004974-0023 |
| NametestAge45Age45GenderMaleAddressHorrAddressMr. JReferenceMr. JMedicalDiabEmploymentWorlPast Salary123New Salary123Educational QualificationLLMDesignationsHeatRemarksJnus | ADD EMPLO | YEE | | |
| Age45GenderMaleAddressHorrAddressMr.ReferenceMr.MedicalDialsEmploymentWorlPast Salary123New Salary123Educational QualificationLLMDesignationsHeatRemarksDuals | test | Additional Benefits | Has a company car | |
| GenderMaleAddressHorrAddressMr. 4ReferenceMr. 4MedicalDiabEmploymentWorlPast Salary123New Salary123Educational QualificationLLMDesignationsHeadRemarksDnus | 45 | Reporting To | Dr. XYZ | |
| AddressHorReferenceMr. 4MedicalDiabEmploymentWorldPast Salary123New Salary123Educational QualificationLLMDesignationsHeatRemarksDuus | Male | Reportees | A,B,C,D | |
| ReferenceMr.MedicalDiabEmploymentWorlPast Salary123New Salary123Educational QualificationLLMDesignationsHeadRemarksDuas | Home | Key Responsible Area | Sales | |
| MedicalDiabEmploymentWordPast Salary123New Salary123Educational QualificationLLMDesignationsHeadRemarksDuas | Mr. A | Date of Joining | 18/10/1998 | |
| EmploymentWorldPast Salary123New Salary123Educational QualificationLLMDesignationsHeatRemarksJnust | Diabetes | Contact Details | 2337564, xyz%abc.com | |
| Past Salary123New Salary123Educational QualificationLLMDesignationsHeatRemarksDuss | Worked at XYZ | Add Employee | | |
| New Salary123Educational QualificationLLMDesignationsHeatRemarksJnus | 12345 | Erase All | | |
| Educational Qualification LLM Designations Hear Remarks onus | 123456 | Exit | | |
| Designations Hea Remarks onus | LLM | | Message | |
| Remarks onus | Head of Sales | Cust | omer Added | |
| | onus of 20, loves working | | | |
| | | | ОК | |

Figure 9 displays the Add Employee form, from where an employee can be added to the table on the database. It also shows the success message if the customer is added or not. The data used is very similar to that of the client.

Click on Erase All in order to start with a new employee.

Click on Exit to go back to the menu screen.

Aditya Dadia; 004974-0023

private void addActionPerformed(java.awt.event.ActionEvent evt) { try { // TODO add your handling code here: pst.setString(1,Name.getText()); pst.setInt(2, Integer.parseInt(Age.getText())); pst.setString(3,Gender.getText()); pst.setString(4,Address.getText()); pst.setString(5,Reference.getText()); pst.setString(6,Medical.getText()); pst.setString(7,Employment.getText()); pst.setInt(8,Integer.parseInt(PastSalary.getText())); pst.setString(9,Remarks.getText()); pst.setString(10,EducationalQualification.getText()); pst.setString(11,Designations.getText()); pst.setString(12,AdditionalBenefits.getText()); pst.setInt(13,Integer.parseInt(NewSalary.getText())); pst.setString(14,ReportingTo.getText()); pst.setString(15,ContactDetails.getText()); pst.setString(16,DateOfJoining.getText()); pst.setString(17,Reportees.getText()); pst.setString(18,KeyResponsibleArea.getText()); pst.execute(); JOptionPane.showMessageDialog(null, "Customer Added"); } catch (SQLException ex) { Logger.getLogger(Add.class.getName()).log(Level.SEVERE, null, ex); }

Figure 10

Criterion C Development

Figure 10 shows the code required to add an employee.

| Criterion C Development | | | | |
|-------------------------|--|--|--|--|
| | | | | |
| ADD USER | | | | |
| Username: | | | | |
| Password: | | | | |
| Re-type Password: | | | | |
| Exit Save | | | | |
| Message | | | | |
| User Added Successfully | | | | |
| ОК | | | | |



The user has to type in the new username and the password. Only if the password is verified with the re-entered one will it get saved to the login table on the database.

Figure 11 shows the Add User form with the success message. In case the passwords didn't match, an error message would display.

| Criterion C Development | Aditya Dadia; 004974-0023 |
|---|---------------------------|
| <pre>private void saveActionPerformed(java.awt.event.ActionEvent evt) {</pre> | |
| try { | |
| <pre>String u = username.getlext(); String p = passwords.getText(); String p1 = passwords1.getText();</pre> | |
| <pre>conn = gui.MySqlConnect.ConnectionDB(); String sql = "Insert into Login values (?,?)";</pre> | |
| <pre>pst = (PreparedStatement) conn.prepareStatement(sql); if(p.equals(p1)) //</pre> | |
| <pre>pst.setString(1,u); pst.setString(2,p);</pre> | |
| <pre>pst.execute(); JOptionPane.showMessageDialog(null, "User Added Successfully"); }</pre> | |
| else | |
| <pre>JOptionPane.showMessageDialog(null, "Passwords do not match. Please }</pre> | try again"); |
| <pre>} catch (SQLException ex) { Logger.getLogger(AddUser.class.getName()).log(Level.SEVERE, null, ex);</pre> | |
| } | |
| } | |

Figure 12 shows the code for Add User

| C | riterion C Develo | pment | | Aditya D | adia; 004974-0023 |
|---|---------------------------|---------------------------|----------------------|-------------------------|-------------------|
| | | EDIT | EMPLOYEE | | |
| | Name | test | Raise Percent | | |
| | Age | 45 | Additional Benefits | Has a company car | |
| | Gender | Male | Reporting To | Dr. XYZ | |
| | Address | Place | Reportees | A,B,C,D | |
| | Reference | Mr. A | Key Responsible Area | Sales | |
| | Medical | Diabetes | Date of Joining | 18/10/1998 | |
| | Employment | Worked at XYZ | Contact Details | 9892337564, xyz%abc.com | |
| | | | | | |
| | Past Salary | 12345 | Display | | |
| | New Salary | 123456 | Edit | | |
| | Educational Qualification | LLM | Erase Exi | t | |
| | Designations | Head of Sales | | | |
| | Remarks | onus of 20, loves working | | | |
| | | | | | |
| | \mathbf{O} | Message | | | |
| ł | | | | | |
| | | istomer Editted | | | |
| Ì | | | | | |
| | | | ОК | | |
| L | | | | | |

Figure 13 shows the edit form. The edit form first searches for the employees by name, when the user types in the name and clicks display. The textboxes then get filled and the user can make changes to all the data and click on edit to save those changes.

Criterion C Development Aditya Dadia; 004974-0023 private void editActionPerformed(java.awt.event.ActionEvent evt) { try { conn = gui.MySqlConnect.ConnectionDB(); String sql = "Update employee set Name=?,Age=?,Gender=?,Address=?,Reference=?,Medical=?,Employment=?," + "PastSalary=?,Remarks=?,EducationalQualification=?,Designations=?,AdditionalBenefits=?," + "NewSalary=?,ReportingTo=?,ContactDetails=?,DateOfJoining=?,Reportees=?,KeyResponsibleArea=?"; pst = (PreparedStatement) conn.prepareStatement(sql); pst.setString(1,Name.getText()); pst.setInt(2, Integer.parseInt(Age.getText())); pst.setString(3,Gender.getText()); pst.setString(4,Address.getText()); pst.setString(5,Reference.getText()); pst.setString(6,Medical.getText()); pst.setString(7,Employment.getText()); pst.setInt(8,Integer.parseInt(PastSalary.getText())); pst.setString(9,Remarks.getText()); pst.setString(10,EducationalQualifications.getText()); pst.setString(11,Designations.getText()); pst.setString(12,AdditionalBenefits.getText()); pst.setInt(13,Integer.parseInt(NewSalary.getText())); pst.setString(14,ReportingTo.getText()); pst.setString(15,ContactDetails.getText()); pst.setString(16,DateOfJoining.getText()); pst.setString(17,Reportees.getText()); pst.setString(18,KeyResponsibleArea.getText()); pst.execute(); JOptionPane.showMessageDialog(null, "Customer Editted"); } catch (SQLException ex) { Logger.getLogger(Edit.class.getName()).log(Level.SEVERE, null, ex); 3

Figure 14

Figure 14 shows the code for edit.

| Criterion C Development | Aditya Dadia; 004974-0023 |
|--|---------------------------|
| | |
| DELETE USER | |
| Enter the username and password of the employee to be dele | ted |
| Username: | |
| Password: | |
| Delete Exit | |

Figure 15 shows the code for delete user, wherein the user has to simply enter the username and password of the user to be deleted and click on delete.

| Criterion C Development | Aditya Dadia; 004974-0023 |
|---|--|
| Delete | |
| Enter Name to be deleted | |
| Delete | Exit |
| Figure 16 | |
| Figure 16 shows the form to delete an employee fi | rom the database, by entering the name and |
| clicking delete. | |
| <pre>private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { try { // TODO add your handling code here: conn = MySqlConnect.ConnectionDB(); } catch (SQLException ex) { Logger.getLogger(Delete.class.getName()).log(Level.SEVERE, null,</pre> | ex); |

Figure 17

}

}

3

try {

catch(Exception e)
{

String sql = "delete from employee where name = ?";

pst.execute(); JOptionPane.showMessageDialog(null,"Employee Deleted");

pst = conn.prepareStatement(sql);
pst.setString(1,jTextField1.getText());
ret events();

JOptionPane.showMessageDialog(null,e);

Figure 17 shows the code to delete an employee. The code to delete a user is similar.

Criterion C Development

| Criterion | C Deve | lopment |
|-----------|--------|---------|
|-----------|--------|---------|

| Sal | ary Slin |
|--------------------------------|----------|
| Sai | |
| Enter the name of the employee | |
| Basic Salary | |
| Dearness Allowance | |
| House Rent Allowance | |
| Travelling Allowance | |
| Provident Fund | |
| Net Pay | |
| Gross Pay | |
| Exit Display Clear Print | |
| igure 18 | |
| Iguie io | |
| | |
| | |

Criterion C Development Aditya Dadia; 004974-0023 Figure 18 shows the blank salary slip. To access the salary of an employee and find out other salary variables, the user need only type the name of the employee, and based on the field of "new salary" his salary figures will be displayed on the click of display. One can print this form as well after future development of this product, as the print option does not work right now.

| • • | | |
|--------------------------------|-------------------|--|
| Salary Slip | | |
| Enter the name of the employee | test | |
| Basic Salary | 123456 | |
| Dearness Allowance | 10493.76 | |
| House Rent Allowance | 18518.39999999998 | |
| Travelling Allowance | 12345.6 | |
| Provident Fund | 98764.8 | |
| Net Pay | 164813.76 | |
| Gross Pay | 66048.96 | |
| Exit Display Clear Print | | |

Figure 19

Figure 19 is an example of how the salary slip works.



Figure 20 is the code for calculating the salary.



Figure 21

Figure 21 displays the code for printing form as a whole, which needs to be edited.

| Criterion C Deve | lopment | | Aditya Dadia; 0049 | 74-0023 |
|--|---|---|---|---------|
| Enter the Employee's Ful | l Name and click display | SEARCH EMPLC | DYEE | |
| Name | asdfgh | Additional Benefits | kjhgfv | |
| Age | 6542 | Reporting To | jhgf | |
| Gender | jhgf | Reportees | jnhbgfd | |
| Address | hgfd | Key Responsible Area | jhgfd | |
| Reference | hgfds | Date of Joining | jhbgfdk | |
| Medical | jhgfds | Contact Details | jhgfd | |
| Employment | jhgfd | Serial Number | | |
| Past Salary | 765 | Display | | |
| New Salary | 7654 | Erase | | |
| Educational Qualification | jhgf | Add to Table | | |
| Designations | jhgf | Exit View All | | |
| Remarks | hgf | Cat Papart | Delate Row | |
| | | Get Report | Delete Kow | |
| Name Age Gen A asdf 6542 jhgf H test 45 Male H | Add Ref Med Em ngfd hgfds jhgfds jhgfd Home Mr. A Dia Wor | Past Re Edu Desi Add 765 hgf jhgf jhgf kjhg 123 Ver LLM Hea Has | I New Rep Con Dat Rep Key frv 7654 jhgf jhgfd jhb jnh jhgfd 123 Dr 989 18/ A,B, Sales | |

Figure 22 shows the search by name form, the most compact form of the database. The user need only type the name and click display, and all the fields will display the corresponding data. If the user wishes to compare data of two or multiple employees, he has to click add to table, and do the whole displaying and adding to table process again. If he wishes to view all the data, he can click on view all and the data will come on the jTable. I have used jTable for the following because it is easy to print. If the user wishes to view a majority of the employees but not all, he can click on view all and delete the employees he doesn't want through delete row. He can also sort them in alphabetical order using sort. I have used an ArrayList and bubble sort for the process of sorting. The user can then print the data by clicking on print.





| <pre>private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {</pre> | |
|--|---------|
| try { | |
| <pre>String sql = "Select * from employee"; pst = (PreparedStatement) conn.prepareStatement(sql); rs = pst.executeQuery(); tblCustomer.setModel(DbUtils.resultSetToTableModel(rs)); catch (SQLException ex) { Logger.getLogger(SearchByName.class.getName()).log(Level.SEVERE, nul }</pre> | l, ex); |
| } | |

Figure 26 shows the print table code.

| Criterion C Development | Aditya Dadia; 004974-0023 |
|-------------------------------|-----------------------------|
| Prin | t |
| General Page Setup Appearance | |
| Print Service | |
| Name: HP Deskjet 3520 series | P <u>r</u> operties |
| Status: Accepting jobs | |
| Type: | |
| Info: HP Deskjet 3520 series | Print To <u>F</u> ile |
| Print Range | Copies |
| . ● A <u>I</u> I | Number <u>o</u> f copies: 1 |
| ○ Pages 1 To 1 | Collate |
| | |
| | Print Cancel |
| Get Report | Sort Delete Row |
| | |

Figure 27 is the print window which opens up when one clicks print.



Figure 28 shows the code for sorting using arraylists. Employee and EmployeeList are classes that I created in the objects package, while the forms designed above are in the gui package.



Figure 29 shows the packages, classes, and forms used for the product.

The clear/erase button on all the forms erases values from all the textboxes.

The exit button on all the forms disposes the forms and takes the user back to the menu.

The exit on the menu form shuts the application.



Figure 30

Word Count: 995

Criterion C Development