[Date]

FRANCIS ADJEI

Report and Codes on School Management System

**Login Page**

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

This is a front page; we can see this page after launched the application. By using Eclipse, we are creating the Graphical User Interfaces in Java.

We have a Login page in our School Management System.

There are Seven Labels in the Login Page which of three labels hold the username, password and the user type images, the other two labels are for the minimizing and closing the frame whiles the other two labels are the login and the No Account? Create One. All the labels have borders around it. There are also two text fields, a password field and a combo box.

It has a checkbox (show password) and a Button (login). Whenever the user clicks on the checkbox, it shows the password that the user entered and three panels.

In this page, we can login to our School Management System by entering the username, the password and the user type. The user type consists of two modules namely Admin and Teacher.

When the user type is Admin, it directs the user do the Admin main page and likewise. When the user enters incorrect password, username or user type, a dialog box will prompt the user to enter the correct password, username or user type.

The Admin or the Teacher can create an account by clicking on the No Account? Create One! Label.

**Login source code:**

JCheckBox chckbxNewCheckBox = **new** JCheckBox("Show Password");

chckbxNewCheckBox.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if** (chckbxNewCheckBox.isSelected()) {

passwordField.setEchoChar((**char**)0);

}**else** {

passwordField.setEchoChar('\*');

}

}

JButton btnLogin = **new** JButton("login");

btnLogin.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**try** {

String txtusername = username.getText();

String pass = passwordField.~~getText~~();

String txtutype = utype.getSelectedItem().toString();

String query = "Select \* from userTable where username = ? and password = ? and UserType = ?";

PreparedStatement ps = connection.prepareStatement(query);

ps.setString(1, txtusername);

ps.setString(2, pass);

ps.setString(3, txtutype);

ResultSet rs = ps.executeQuery();

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

**int** id = rs.getInt("id");

**if** (txtutype.equals("Admin")) {

MainClass main = **new** MainClass(id, txtusername, txtutype);

main.setVisible(**true**);

main.setLocationRelativeTo(**null**);

main.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.dispose();

}**else** {

TeacherMain bb = **new** TeacherMain(id, txtusername, txtutype);

bb.setVisible(**true**);

bb.setLocationRelativeTo(**null**);

bb.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.dispose();

}

}

**else** {

JOptionPane.*showMessageDialog*(frame, "Invalid Username, password and UserType", "Login Error",3);

}

rs.close();

ps.close();

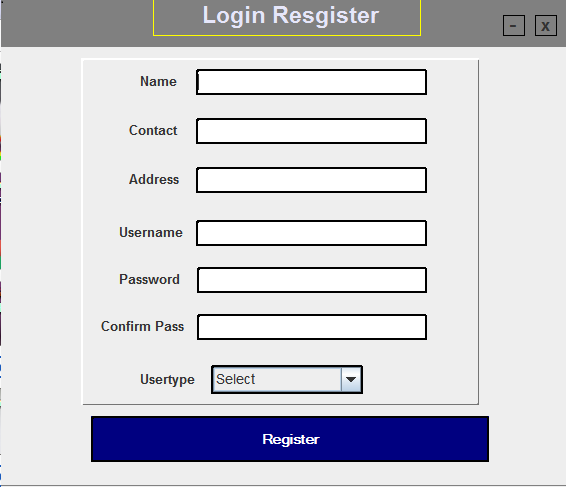
}**catch**(Exception e1) {

JOptionPane.*showMessageDialog*(**null**, e1);

}

}

**User Creation Page**

****

**Fig 2**

This is a page where an Admin and the Teacher will create an account in our School Management System. Here, we have multiple numbers of labels (Login Register, Name, Contact, Address, Username, Password, Confirm Pass, User type, Minimizing and Closing label) and four panels.

And we also have a button, one combo box and six entry fields. I created two methods namely; verifyField() and checkUsername(). The verifyField() method has no argument. It checks whether all the fields and the combo box on the page is not empty. Also, it checks whether the password and the confirm password matches. Whiles the checkUsername() method, checks whether the username, the user enters is already exist. It also has a parameter named Username which has the String data type. These methods were Implemented at the Register button, so when it’s clicked it checks for the two methods.

Moreover, the Contact field contains only numeric values.

**User Creation Source Code:**

**public** **boolean** verifyField() {

String fname = textField\_1.getText();

String contact = textField\_2.getText();

String address = textField\_3.getText();

String uname = textField\_4.getText();

String pass1 = String.*valueOf*(passwordField.getPassword());

String pass2 = String.*valueOf*(passwordField\_1.getPassword());

String txtutype = comboBox.getSelectedItem().toString();

**if** (fname.trim().equals("") || contact.trim().equals("") || address.trim().equals("") || uname.trim().equals("")

|| pass1.trim().equals("") || pass2.trim().equals("") || txtutype.trim().equals("") ) {

JOptionPane.*showMessageDialog*(**null**, " One or More fields Are Empty", "Empty Field", 2);

**return** **false**;

}**else** **if** (!pass1.equals(pass2) ){

JOptionPane.*showMessageDialog*(**null**, " Password Doesn't Match", "Confirm Password", 2);

**return** **false**;

}**else** {

**return** **true**;

}

}

**public** **boolean** checkUsername(String Username) {

**boolean** username\_exixt = **false**;

**try** {

String query = "select \* from UserTable where Username = ? ";

PreparedStatement ps = connection.prepareStatement(query);

ps.setString(1, Username);

ResultSet rs = ps.executeQuery();

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

username\_exixt = **true**;

JOptionPane.*showMessageDialog*(**null**, " This Username Is Already Taken, Choose Another One", "Username Failed", 2);

}

rs.close();

ps.close();

} **catch** (Exception e1) {

e1.printStackTrace();

}

**return** username\_exixt;

}

JButton btnSave = **new** JButton("Register");

btnSave.setBorder(**new** LineBorder(**new** Color(0, 0, 0), 2, **true**));

btnSave.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String pass1 = String.*valueOf*(passwordField.getPassword());

**if** (verifyField()) {

**if** (!checkUsername( textField\_4.getText())) {

**try** {

String query = "insert into UserTable (Name, Contact, Address, Username, Password, Usertype) values (?,?,?,?,?,?)";

PreparedStatement ps = connection.prepareStatement(query);

ps.setString(1, textField\_1.getText());

ps.setString(2, textField\_2.getText());

ps.setString(3, textField\_3.getText());

ps.setString(4, textField\_4.getText());

ps.setString(5, pass1);

ps.setString(6, comboBox.getSelectedItem().toString());

ps.execute();

JOptionPane.*showMessageDialog*(**null**, " Account Created");

ps.close();

} **catch** (Exception e1) {

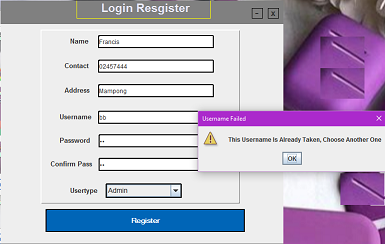
e1.printStackTrace();

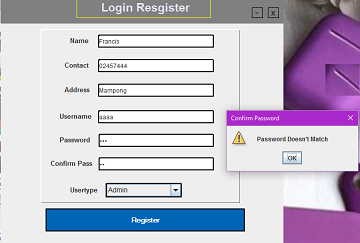
}

}

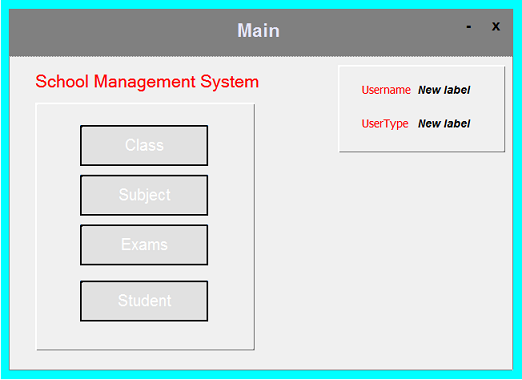
}

}

****

****

**Admin Main Page**



Thispage consists of four buttons namely; Class, Subject, Exams and Student, four panels and eight labels. When the user login and the user type is “Admin” and clicked on the login button, it directs you to this page and the username and the user type he/she entered will appear at the new label respectively. When the Admin click on the Class button, it goes to the Class page and the same applies for the other buttons on this page. When you look at the source you will see that this page has constructors which is overloading.

**Source Code:**

**public** MainClass() {

setUndecorated(**true**);

initComponent();

}

**int** iiid;

String usname;

String usertype;

**private** JLabel lblNewLabel\_5;

**public** MainClass(**int** id, String txtusername, String txtutype) {

setUndecorated(**true**);

initComponent();

**this**.usname = txtusername;

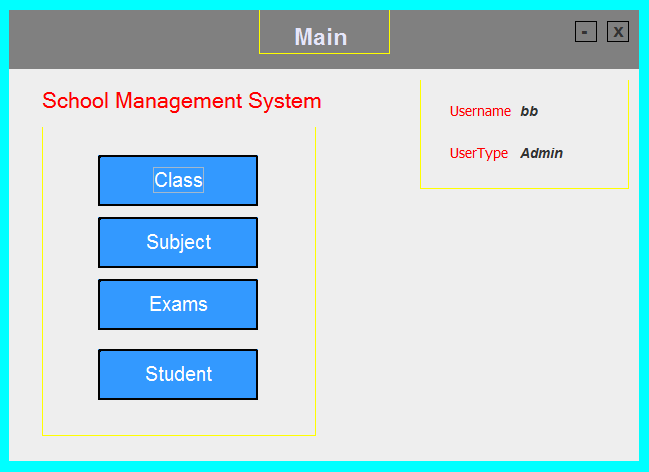
lblNewLabel.setText(usname);

**this**.usertype = txtutype;

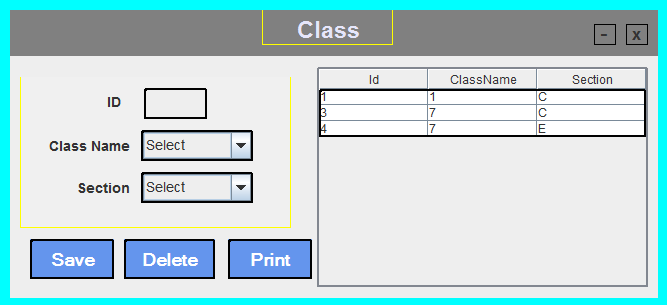
lblNewLabel\_1.setText(usertype);

**this**.iiid = id;

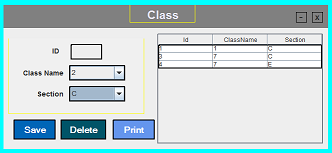
}

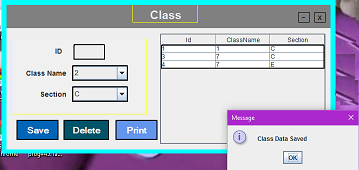
****

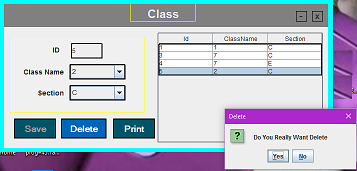
**Class Page**

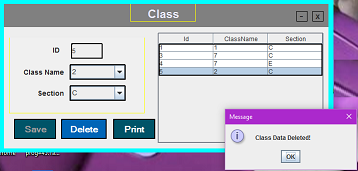
****

This page consists of four panels, six labels, a text field, two combo box, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and ”- “minimizes the page.









When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

The print button prints the data on the table in a word or pdf format.

**Source Code:**

**public** **void** classRefreshTable() {

**try** {

String query = "select \* from ClassTable";

PreparedStatement ps = connection.prepareStatement(query);

ResultSet rs = ps.executeQuery();

table.setModel(DbUtils.*resultSetToTableModel*(rs));

rs.close();

ps.close();

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

table = **new** JTable();

table.setBorder(**new** LineBorder(**new** Color(0, 0, 0), 2, **true**));

table.addMouseListener(**new** MouseAdapter() {

@Override

**public** **void** mouseClicked(MouseEvent e) {

**int** row = table.getSelectedRow();

String ID\_ = (table.getModel().getValueAt(row, 0)).toString();

**try** {

String query = "select \* from ClassTable where ID = '"+ID\_+"'";

PreparedStatement ps = connection.prepareStatement(query);

ResultSet rs = ps.executeQuery();

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

textField.setText(rs.getString("ID"));

comboBoxClass.setSelectedItem(rs.getString("ClassName"));

comboBoxSection.setSelectedItem(rs.getString("Section"));

btnSave.setEnabled(**false**);

}

**public** **void** actionPerformed(ActionEvent e) {

**try** {

String query = "insert into ClassTable (ClassName, Section) values (?, ?)";

PreparedStatement ps = connection.prepareStatement(query);

ps.setString(1, comboBoxClass.getSelectedItem().toString());

ps.setString(2, comboBoxSection.getSelectedItem().toString());

ps.execute();

JOptionPane.*showMessageDialog*(**null**, " Class Data Saved");

ps.close();

} **catch** (Exception e1) {

e1.printStackTrace();

}

classRefreshTable();

btnDelete.setBorder(**new** LineBorder(**new** Color(0, 0, 0), 2, **true**));

btnDelete.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**int** action = JOptionPane.*showConfirmDialog*(**null**, "Do You Really Want Delete", "Delete", JOptionPane.***YES\_NO\_OPTION***);

**if** (action == 0) {

**try** {

String query = "delete from ClassTable where ID = '"+textField.getText()+"' ";

PreparedStatement ps = connection.prepareStatement(query);

ps.execute();

JOptionPane.*showMessageDialog*(**null**, " Class Data Deleted!");

btnSave.setEnabled(**true**);

ps.close();

} **catch** (Exception e1) {

e1.printStackTrace();

}

}

classRefreshTable();

JButton btnBack = **new** JButton("Print");

btnBack.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

MessageFormat header = **new** MessageFormat("Printing in Progress");

MessageFormat footer = **new** MessageFormat("Page {0, number, interger}");

**try** {

table.print(JTable.PrintMode.***NORMAL***, header, footer);

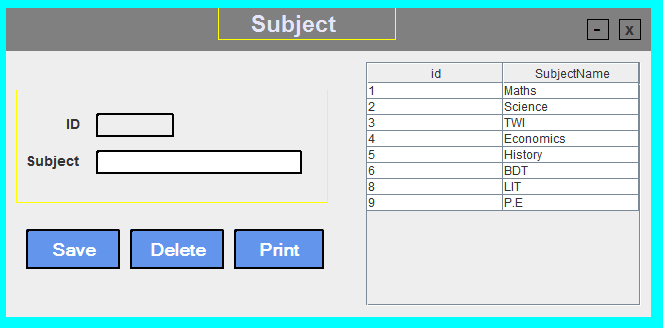
}**catch** (java.awt.print.PrinterException e4){

System.***err***.format("No Printer Found", e4.getMessage());

}

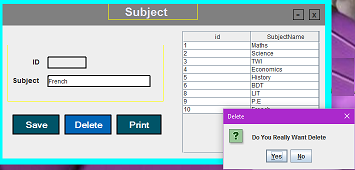
}

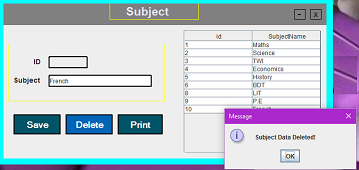
**Subject Page**

****

This page consists of four panels, six labels, two text field, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and ”- “minimizes the page.

|  |  |
| --- | --- |
|  |  |

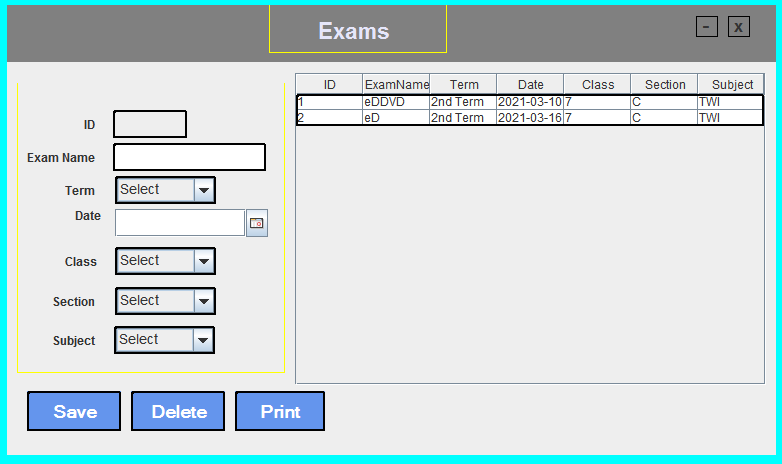
****

****

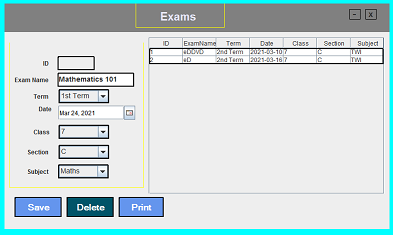
When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

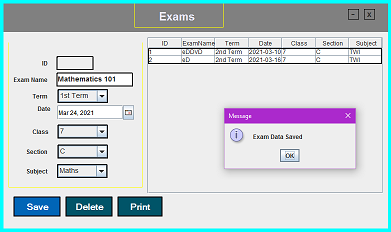
The print button prints the data on the table in a word or pdf format.

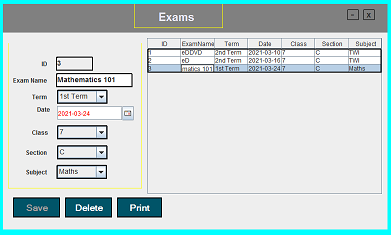
**Exams Page**

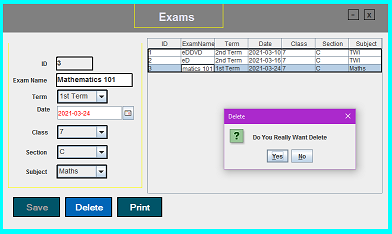
****

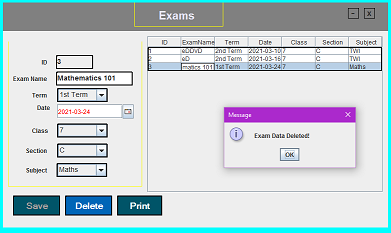
This page consists of four panels, six labels, two text field, four combo boxes, a date chooser, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and”- “minimizes the page. The date chooser component is used to pick a date to the exams page.





****

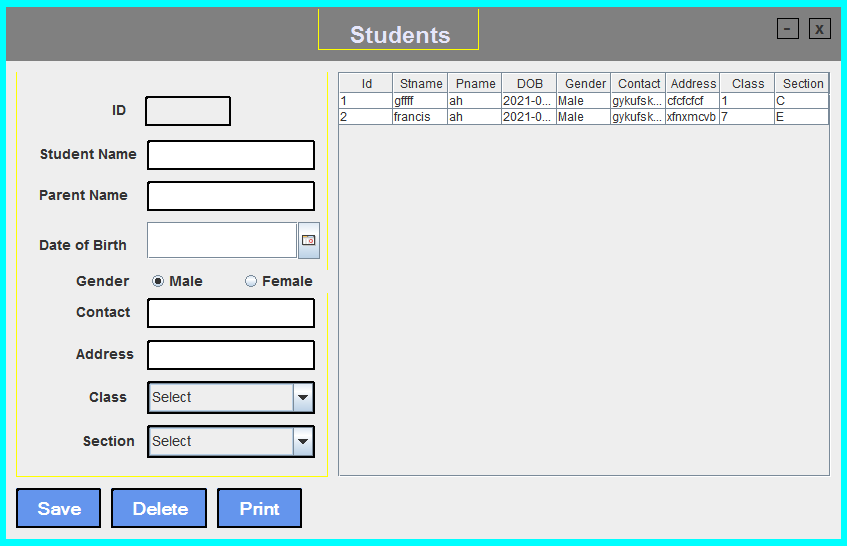
****

****

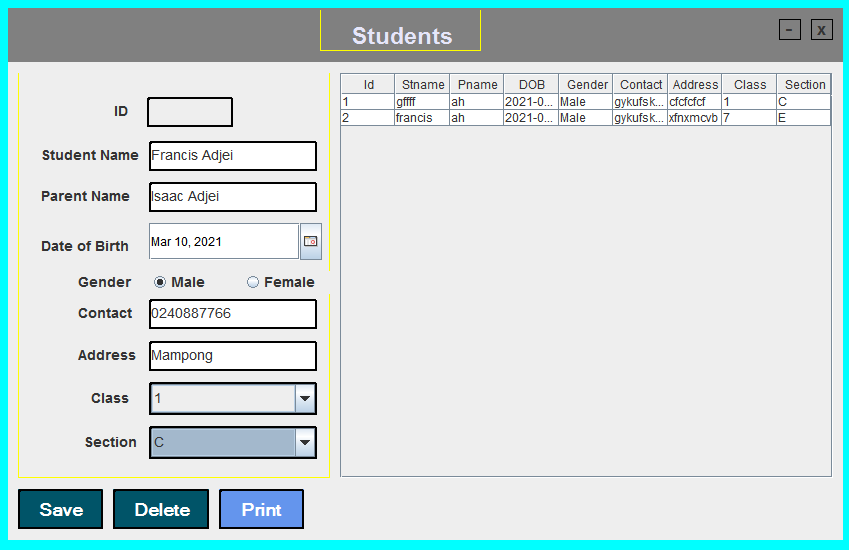
When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

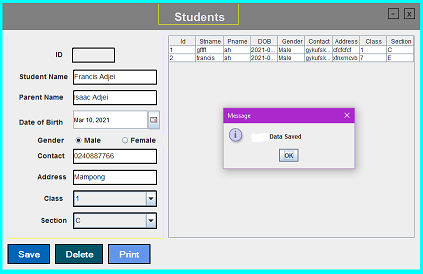
The print button prints the data on the table in a word or pdf format.

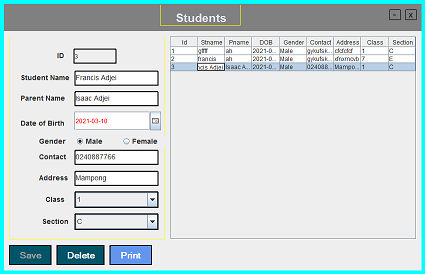
**Student Page**

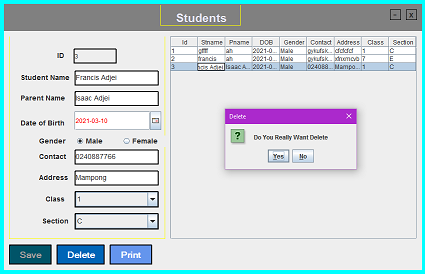
****

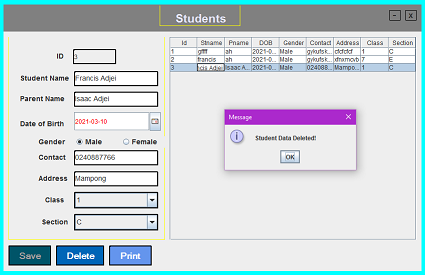
This page consists of four panels, six labels, six text field, two combo boxes, a date chooser, a radio button, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and”- “minimizes the page. The date chooser component is used to pick a date to the exams page.







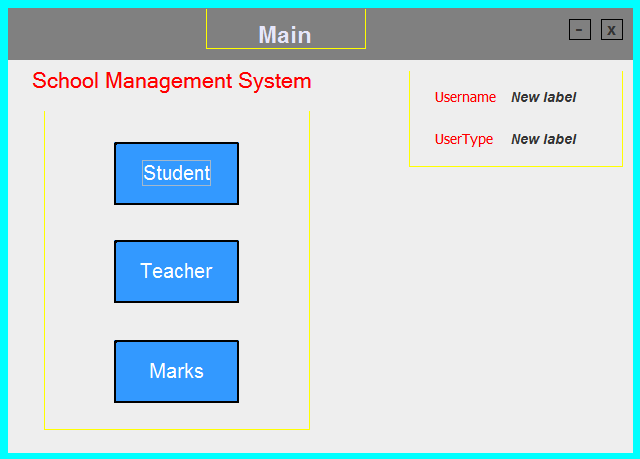




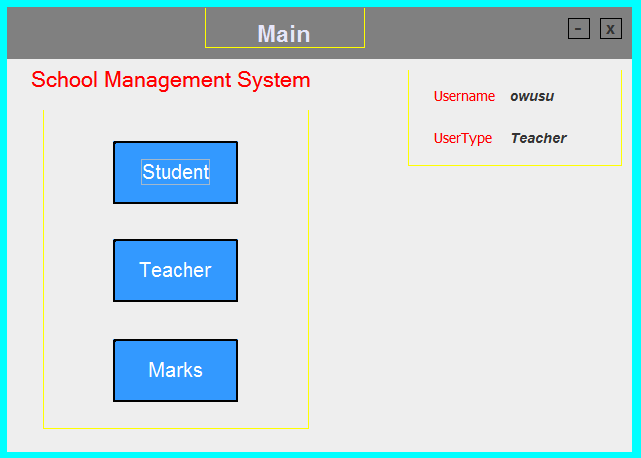
When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

The print button prints the data on the table in a word or pdf format.

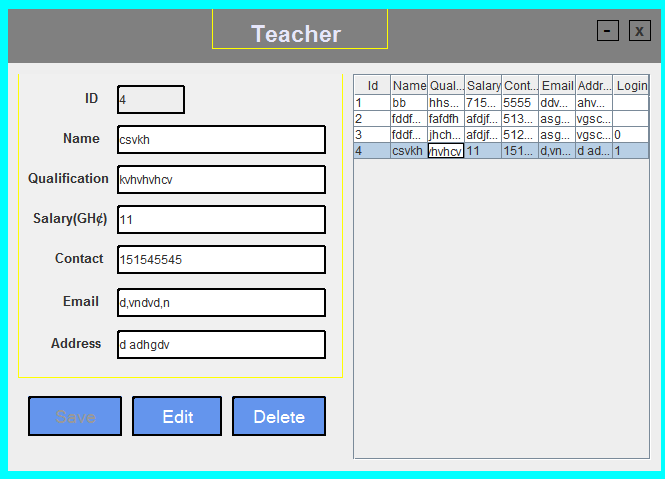
**Teacher Main Page**

****

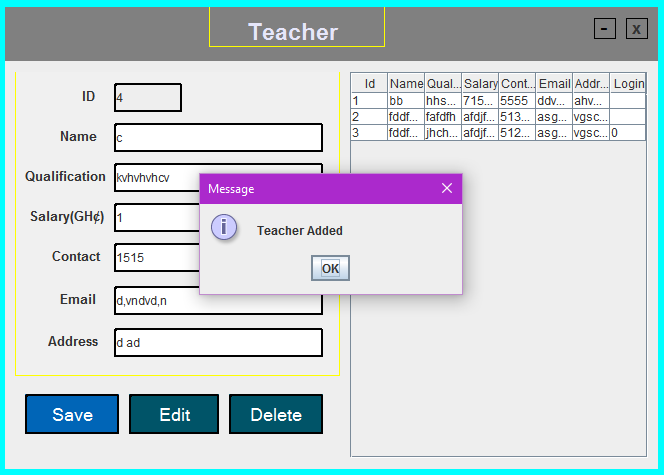
Thispage consists of four buttons namely; Student, Teacher and Marks, four panels and eight labels. When the user login and the user type is “Admin” and clicked on the login button, it directs you to this page and the username and the user type he/she entered will appear at the new label respectively. When the Admin click on the Class button, it goes to the Student page and the same applies for the other buttons on this page. When you look at the source you will see that this page has constructors which is overloading.

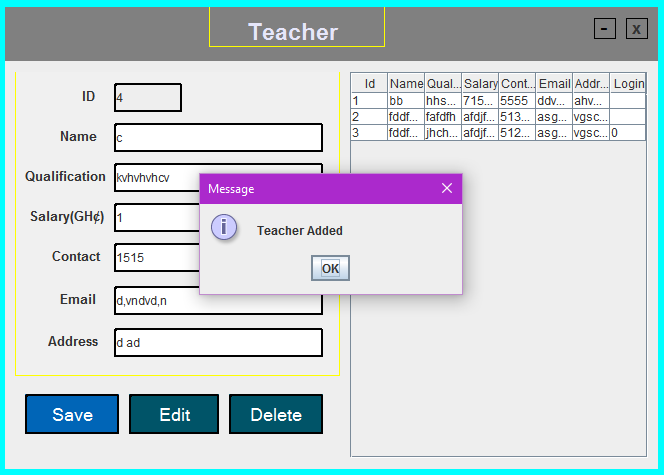
****

**Teacher Page**

****

This page consists of four panels, nine labels, seven text field, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and”- “minimizes the page. The date chooser component is used to pick a date to the exams page.

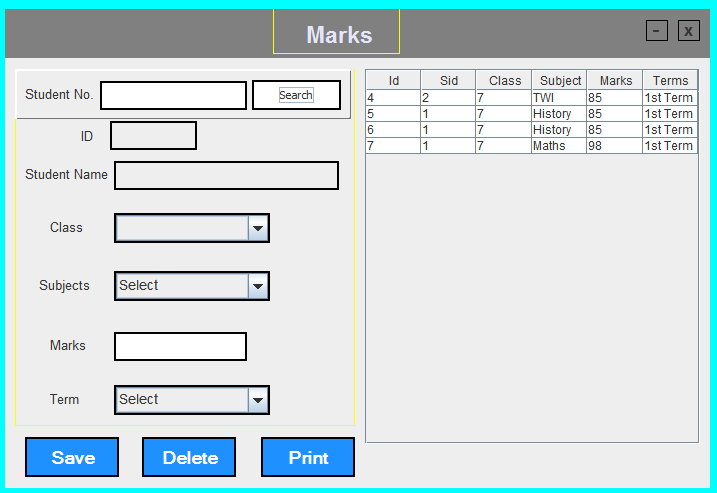


****

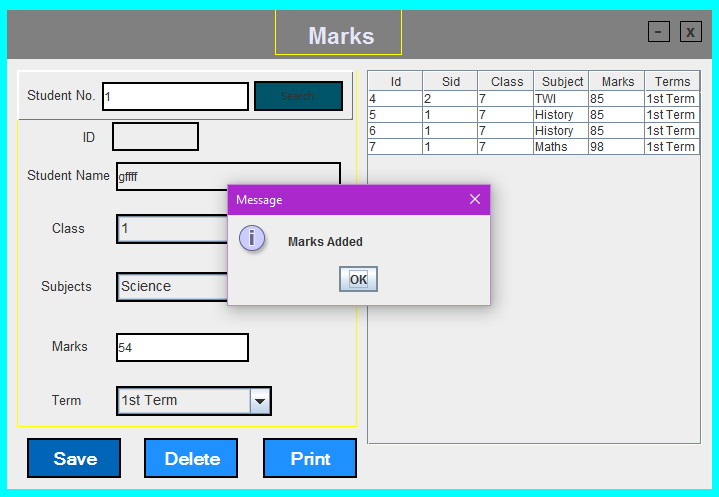
When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

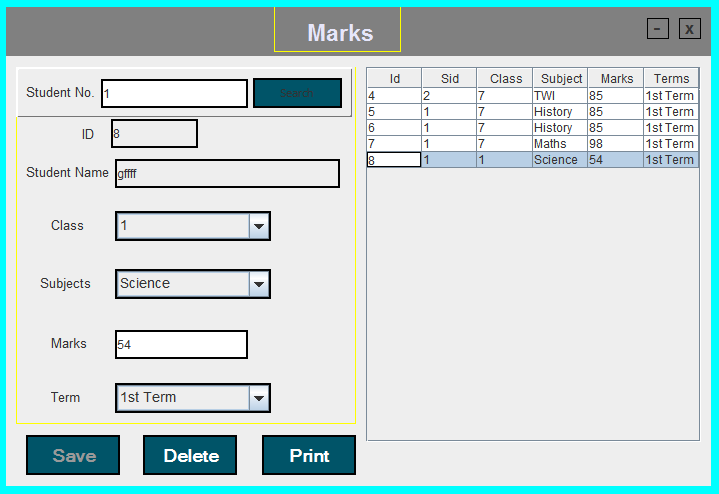
The print button prints the data on the table in a word or pdf format.

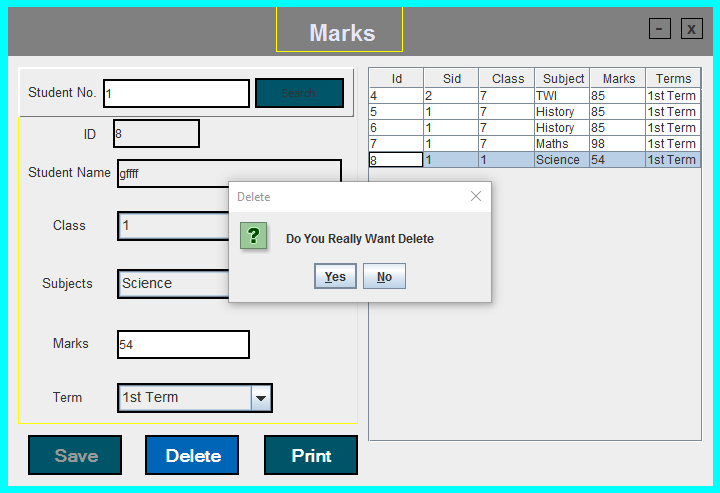
**Marks Page**

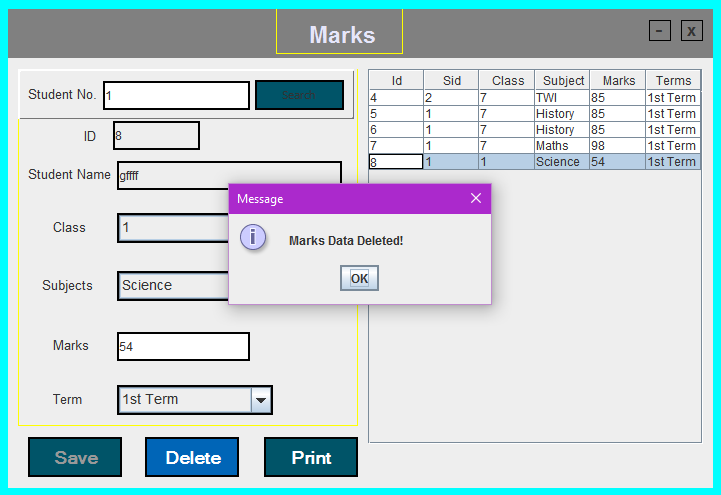
****

This page consists of four panels, ten labels, four text field, three buttons (save, delete and print) and a table. The ID text field cannot be edited. When then user click on the “x” and “-” labels, the “X” closes the page and directs you to the Admin Main Page and”- “minimizes the page. The date chooser component is used to pick a date to the exams page. The user will enter his/her school id and then click on the button. When the user clicked on the search button, his/her name, class and subject will automatically appear on the text fields and combo box.









When the user enters the values and click on the save button, it saves the data into the database and the data in the database will pop up into table. Also, when the user wants to delete a data, he/she must click on the data he/she wants to delete on the table and then clicked on the delete button, there will be a message dialog on the screen asking “Do you really want to delete the data from the database”. If you clicked “yes” the data will be deleted and otherwise. When the user clicks on the table, the save button is set to unenabled until the delete button has performed a task.

The print button prints the data on the table in a word or pdf format.