/\*\*

\* Deborah Barndt & Kenneth Yan

\* 11-22-16

\* Tickets.java

\* IT

\* This program will hold the ticket system menu in order to create, look up, delete,

\* and modify a ticket.

\* Written by Deborah Barndt & Kenneth Yan. \*/

package IT;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.net.MalformedURLException;

import java.net.URL;

import java.sql.Connection;

import java.sql.Date;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.Calendar;

import javax.swing.ImageIcon;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.UIManager;

import javax.swing.plaf.ColorUIResource;

import javax.swing.plaf.FontUIResource;

// Proj implements one listener and can trigger action off menu item clicked with one action performed event handler

@SuppressWarnings("serial")

public class Tickets extends JFrame implements ActionListener

{

// Class level member vars

private JFrame mainFrame;

// Include main menu objects

private JMenu file = new JMenu("File");

private JMenu tickets = new JMenu("Tickets");

JScrollPane sp = null;

// Include sub menu item objects

JMenuItem ItemNew;

JMenuItem ItemExit;

JMenuItem ItemOpenTicket;

JMenuItem ItemViewTicket;

JMenuItem ItemViewAllTicket;

JMenuItem ItemCloseTicket;

JMenuItem ItemUpdateTicket;

JMenuItem ItemDeleteTicket;

// Include more items below

public Tickets() throws MalformedURLException

{

createTicketTable();

createUserTable();

createContactInfoTable();

createMenu();

prepareGUI();

test();

}

private void createTicketTable()

{

// Vars. for SQL Query create

final String createTicketTable = "CREATE TABLE dbarn\_kyan\_ticket"

+ "(Ticket\_ID INT AUTO\_INCREMENT PRIMARY KEY, "

+ "Subject VARCHAR(30), "

+ "Description VARCHAR(200),"

+ "Status VARCHAR(255),"

+ "Sdate DATETIME,"

+ "Edate DATETIME,"

+ "Priority VARCHAR(255));";

Connection connect = null;

Statement statement = null;

try

{

// This will load the MySQL driver, each DB has its own driver

Class.forName("com.mysql.jdbc.Driver");

// Setup the connection with the DB

connect = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

// Create the table

statement = connect.createStatement();

statement.executeUpdate(createTicketTable);

System.out.println("Created table in given database...");

//end create table

//close connection/statement object

statement.close();

connect.close();

}

catch (Exception e)

{

System.out.println(e.getMessage());

}

}

private void createUserTable()

{

// vars. for SQL Query create

final String createUserTable = "CREATE TABLE dbarn\_kyan\_login"

+ "(User VARCHAR(255), "

+ "Password VARCHAR(60), "

+ "Access VARCHAR(255));" ;

Connection connect = null;

Statement statement = null;

String adduser = "Insert into dbarn\_kyan\_login(User, Password, Access) VALUES"

+ "('jpapa', MD5('thegodfather'), 'user'), "

+ "('kyan', MD5('javarocks'), 'admin'), "

+ "('dbarndt', MD5('designguru'), 'admin'), "

+ "('mengland', MD5('techsavy'), 'user'), "

+ "('lmchugh', MD5('networking'), 'user'), "

+ "('jjerger', MD5('serverboss'), 'user'), "

+ "('lfitzpatrick', MD5('winteriscoming'), 'user');";

try

{

// This will load the MySQL driver, each DB has its own driver

Class.forName("com.mysql.jdbc.Driver");

// Setup the connection with the DB

connect = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

// Create the table

statement = connect.createStatement();

statement.executeUpdate(createUserTable);

statement.executeUpdate(adduser);

System.out.println("Created table in given database...");

//end create table

//close connection/statement object

statement.close();

connect.close();

}

catch (Exception e)

{

System.out.println(e.getMessage());

}

}

private void createContactInfoTable()

{

// vars. for SQL Query create

final String createContactInfoTable = "CREATE TABLE dbarn\_kyan\_user"

+ "(Ticket\_ID INT AUTO\_INCREMENT PRIMARY KEY, "

+ "Email VARCHAR(255), "

+ "First\_Name VARCHAR(255), "

+ "Last\_Name VARCHAR(255), "

+ "Phone\_Num VARCHAR(200));" ;

Connection connect = null;

Statement statement = null;

String adduser = "Insert into dbarn\_kyan\_user(Email, First\_Name, Last\_Name, Phone\_Num) VALUES"

+ "('jpapa@gmail.com', 'James', 'Papademas', '312-112-6595'), "

+ "('kyan@gmail.com', 'Kenneth', 'Yan', '312-546-8756'), "

+ "('dbarndt@gmail.com', 'Deborah', 'Barndt', '312-584-7856'), "

+ "('mengland@gmail.com', 'Madeleine', 'England', '312-867-5309'), "

+ "('lmchugh@gmail.com', 'Louis', 'McHugh', '312-346-5789'), "

+ "('jjerger@gmail.com', 'James', 'Jerger', '312-582-6435'), "

+ "('lfitzpatrick@gmail.com', 'Lenard', 'Fitzpatrick', '312-648-4285');";

try

{

// This will load the MySQL driver, each DB has its own driver

Class.forName("com.mysql.jdbc.Driver");

// Setup the connection with the DB

connect = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

// Create the table

statement = connect.createStatement();

statement.executeUpdate(createContactInfoTable);

statement.executeUpdate(adduser);

System.out.println("Created table in given database...");

//end create table

//close connection/statement object

statement.close();

connect.close();

}

catch (Exception e)

{

System.out.println(e.getMessage());

}

}

// Column names

private void createMenu()

{

// Set up mnemonics for main menu items (triggered by alt keys)

file.setMnemonic('F');

// Initialize the sub menu items

ItemNew = new JMenuItem("New");

ItemNew.setMnemonic('N');

file.add(ItemNew);

ItemExit = new JMenuItem("Exit");

ItemExit.setMnemonic('X');

file.add(ItemExit);

ItemOpenTicket = new JMenuItem("Open Ticket");

ItemOpenTicket.setMnemonic('O');

tickets.add(ItemOpenTicket);

ItemViewTicket = new JMenuItem("View Ticket");

ItemViewTicket.setMnemonic('V');

tickets.add(ItemViewTicket);

ItemDeleteTicket = new JMenuItem("Delete Ticket");

ItemDeleteTicket.setMnemonic('D');

tickets.add(ItemDeleteTicket);

ItemViewAllTicket = new JMenuItem("View All Tickets");

ItemViewAllTicket.setMnemonic('V');

tickets.add(ItemViewAllTicket);

ItemUpdateTicket = new JMenuItem("Update Ticket");

ItemUpdateTicket.setMnemonic('U');

tickets.add(ItemUpdateTicket);

ItemCloseTicket = new JMenuItem("Close Ticket");

ItemCloseTicket.setMnemonic('C');

tickets.add(ItemCloseTicket);

// Include more sub menu items below

//add listeners for each desired menu item

ItemExit.addActionListener(this);

ItemOpenTicket.addActionListener(this);

ItemViewTicket.addActionListener(this);

ItemViewAllTicket.addActionListener(this);

ItemCloseTicket.addActionListener(this);

ItemUpdateTicket.addActionListener(this);

ItemDeleteTicket.addActionListener(this);

// Add more listeners for any additional sub menu items

}

private void prepareGUI()

{

// Initialize the frame object

mainFrame = new JFrame("Geekbit IT Help Desk");

// Create the jmenu bar

JMenuBar bar = new JMenuBar();

// Set menu orders

bar.add(file);

bar.add(tickets);

// Add menu bar component to frame

mainFrame.setJMenuBar(bar);

mainFrame.setSize(600,600);

mainFrame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e)

{

// TODO Auto-generated method stub

// Add desired coding to trigger each sub menu action as examples shown below

// Exit program

if(e.getSource() == ItemExit)

{

System.exit(0);

}

// New ticket

/\* else if(e.getSource() == ItemNew)

{

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

}\*/

// Open ticket

else if(e.getSource() == ItemOpenTicket)

{

try

{

// Add a ticket information

String ticketSubject = JOptionPane.showInputDialog(null,"Please enter the ticket subject: ", "Open Ticket",

JOptionPane.PLAIN\_MESSAGE);

String ticketDesc = JOptionPane.showInputDialog(null,"Please enter the ticket description: ", "Open Ticket",

JOptionPane.PLAIN\_MESSAGE);

Object[] possibilities = {"High", "Medium", "Low"};

String ticketPriority = (String) JOptionPane.showInputDialog(null,"Please enter the ticket priority: ",

"Open Ticket", JOptionPane.PLAIN\_MESSAGE, null, possibilities, "Low");

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

int result = statement.executeUpdate("Insert into dbarn\_kyan\_ticket(Subject, Description, Status, Priority, Sdate) "

+ "values('"+ ticketSubject +"','"+ ticketDesc +"', 'Open', '" + ticketPriority + "' , NOW())");

if (result != 0)

{

System.out.println("Ticket Created Successfully!!!");

}

else

{

System.out.println("Ticket cannot be Created!!!");

}

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

// Fill in with id value newly created in table

JOptionPane.showMessageDialog(null, "Ticket ID: created",

"Ticket Created", JOptionPane.INFORMATION\_MESSAGE, icon);

statement.close();

dbConn.close(); //close connections!!!

}

catch (SQLException ex)

{

// TODO Auto-generated catch block

ex.printStackTrace();

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (MalformedURLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

// View a ticket

else if(e.getSource() == ItemViewTicket)

{

//retrieve ticket information

//ticketsJDBC dobj = new ticketsJDBC();

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

String see= JOptionPane.showInputDialog(null, "Please enter the number of the ticket you would like to see: ",

"View Ticket", JOptionPane.PLAIN\_MESSAGE);

ResultSet results = statement.executeQuery("SELECT \* FROM dbarn\_kyan\_ticket WHERE Ticket\_ID = '" + see +"' ;");

// Use JTable built in functionality to build a table model and display the table model

//off a resultset!!!

JTable jt = new JTable(ticketsJTable.buildTableModel(results));

jt.setBounds(30,40,200,300);

sp=new JScrollPane(jt);

mainFrame.add(sp);

mainFrame.setVisible(true); //repaint frame on screen

statement.close();

dbConn.close(); //close connections!!!

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

// Close a ticket

else if(e.getSource() == ItemCloseTicket)

{

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

String ID= JOptionPane.showInputDialog(null, "Please enter the number of the ticket you want to close: ", "Close Ticket",

JOptionPane.PLAIN\_MESSAGE);

String results = "UPDATE dbarn\_kyan\_ticket SET Status = 'CLOSED'" +

"WHERE Ticket\_ID = '" + ID + "';";

statement.executeUpdate(results);

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

// System.out.println("Ticket Removed Successfully!!!");

JOptionPane.showMessageDialog(null, "Ticket number "+ ID +" has been Closed Successfully!!!",

"Close Ticket", JOptionPane.INFORMATION\_MESSAGE, icon);

statement.close();

dbConn.close(); //close connections!!!

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (MalformedURLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

// Delete a ticket

else if(e.getSource() == ItemDeleteTicket)

{

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

String ID = JOptionPane.showInputDialog(null, "Please enter the number of the ticket you want to delete: ",

"Delete Ticket", JOptionPane.PLAIN\_MESSAGE);

// String result1 = JOptionPane.showInputDialog(null,"Are you sure that you want to delete ticket NUMBER " + ID + " ? " + "\n" + "(YES, NO)");

if (JOptionPane.showConfirmDialog(null, "Are you sure you want to delete this ticket?",

"WARNING", JOptionPane.YES\_NO\_OPTION) == JOptionPane.YES\_OPTION)

{

// yes option

String results = "DELETE FROM dbarn\_kyan\_ticket WHERE Ticket\_ID = '" + ID + "';";

statement.executeUpdate(results);

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

// System.out.println("Ticket Removed Successfully!!!");

JOptionPane.showMessageDialog(null, "Ticket number "+ ID +" has been Removed Successfully!!!",

"Delete Ticket", JOptionPane.INFORMATION\_MESSAGE, icon);

}

else

{

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

// no option

JOptionPane.showMessageDialog(null, "Ticket number "+ ID +" has NOT been Removed Successfully!!!",

"Delete Ticket", JOptionPane.INFORMATION\_MESSAGE, icon);

}

statement.close();

dbConn.close(); //close connections!!!

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (MalformedURLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

// Update a ticket

else if (e.getSource() == ItemUpdateTicket)

{

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

Object[] options = {"Open", "Close"};

String ud= JOptionPane.showInputDialog(null, "Please enter the number of the ticket you want to edit: ",

"Update Ticket", JOptionPane.PLAIN\_MESSAGE);

String ud1= JOptionPane.showInputDialog(null, "Please enter the new ticket subject: ",

"Update Ticket", JOptionPane.PLAIN\_MESSAGE);

String ud2= JOptionPane.showInputDialog(null, "Please enter the new ticket description: ",

"Update Ticket", JOptionPane.PLAIN\_MESSAGE);

int ud3= (int) JOptionPane.showOptionDialog(null, "Please enter the new ticket status: ",

"Update Ticket", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE, null, options, options[1]);

if(ud3 == 0)

{

String results = "UPDATE dbarn\_kyan\_ticket SET Subject = '" + ud1 +"' ,"

+ "Description = '" + ud2 +"', Status = 'Open' WHERE Ticket\_ID = '" + ud +"';";

statement.executeUpdate(results);

}

// String results = "UPDATE dbarn\_kyan\_ticket SET Subject = '" + ud1 +"' ,"

// + "Description = '" + ud2 +"', Status = '" + ud3 + "', Sdate = NOW()," + "WHERE Ticket\_ID = '" + ud +"';";

else if(ud3 == 1)

{

String results = "UPDATE dbarn\_kyan\_ticket SET Subject = '" + ud1 +"' ,"

+ "Description = '" + ud2 +"', Status = 'Closed', Edate = NOW()" + "WHERE Ticket\_ID = '" + ud +"';";

statement.executeUpdate(results);

}

//String results = "UPDATE dbarn\_kyan\_ticket SET ticket\_name = Replace(ticket\_name, '" + ud1 + "'" + " " , ";

//statement.executeUpdate(results);

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

JOptionPane.showMessageDialog(null, "Ticket Updated Successfully!!!", "Update Ticket",

JOptionPane.INFORMATION\_MESSAGE, icon);

statement.close();

dbConn.close(); //close connections!!!

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (MalformedURLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

// View all tickets

else if(e.getSource() == ItemViewAllTicket)

{

//retrieve ticket information

//ticketsJDBC dobj = new ticketsJDBC();

try

{

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

ResultSet results = statement.executeQuery("SELECT \* FROM dbarn\_kyan\_ticket");

// Use JTable built in functionality to build a table model and display the table model

//off a resultset!!!

JTable jt = new JTable(ticketsJTable.buildTableModel(results));

jt.setBounds(200,300,200,300);

sp=new JScrollPane(jt);

mainFrame.add(sp);

mainFrame.setVisible(true); //repaint frame on screen

statement.close();

dbConn.close(); //close connections!!!

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (SQLException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

}

// Populate the dbarn\_kyan\_ticket table 5 entries

public void test() throws MalformedURLException

{

try

{

// Delete a ticket

// Update a description to 'WH WIFI PROBLEM’

//add a ticket information

Class.forName("com.mysql.jdbc.Driver").newInstance();

Connection dbConn = DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = dbConn.createStatement();

// Add 5 tickets

// 2 high priority

String results = "INSERT INTO dbarn\_kyan\_ticket (Subject, Description, Status, Priority, Sdate, Edate) VALUES"

+ "('Internet', 'Internet down', 'Open', 'High', NOW(), NULL),"

+ "('Blackboard', 'Can not access blackboard on ipad', 'Open', 'Med', NOW(), NULL),"

+ "('MyIIT', 'Cannot access MyIIT Portal', 'Closed', 'Low', NOW(), NOW()),"

+ "('Register', 'I need to register PS3', 'Closed', 'Low', NOW(), NOW()),"

+ "('Email', 'Cannot access hawk email', 'Open', 'High', NOW(), NULL);" ;

statement.executeUpdate(results);

// Change icon inside dialog box.

final ImageIcon icon = new ImageIcon(new URL("http://www.clemson.edu/ccit/media/helpbutton\_resp\_sm.jpg"));

// Fill in with id value newly created in table

JOptionPane.showMessageDialog(null, "Ticket ID: created", "Ticket Created",

JOptionPane.INFORMATION\_MESSAGE, icon);

statement.close();

dbConn.close();

}

catch (SQLException ex)

{

// TODO Auto-generated catch block

ex.printStackTrace();

}

catch (InstantiationException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (IllegalAccessException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

catch (ClassNotFoundException e1)

{

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

public static void durationTime()

{

try

{

System.out.println("The duration of time all open tickets have existed:\n");

// Time open

Class.forName("com.mysql.jdbc.Driver");

Connection connect = (Connection) DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

Statement statement = (Statement) connect.createStatement();

String sql = "SELECT Ticket\_ID, Description, Sdate, EXTRACT(MINUTE FROM Sdate) AS Sdate2, EXTRACT(MINUTE FROM NOW()) AS Sdate3 "

+ "FROM dbarn\_kyan\_ticket WHERE Status = 'Open';";

//String sql2 = "SELECT DATEDIFF(Sdate, NOW()) FROM dbarn\_kyan\_ticket AS DifD;";

//"DATEDIFF(Sdate2, Sdate3) AS DifD

ResultSet result5 = statement.executeQuery(sql);

while (result5.next())

{

// Ticket\_ID, Subject, Description, Status, Priority

int ID = result5.getInt("Ticket\_ID");

String Desc = result5.getString("Description");

String date = result5.getString("Sdate");

// String dur = result5.getString("DifD");

String date2 = result5.getString("Sdate2");

int result = Integer.parseInt(date2);

String date3= result5.getString("Sdate3");

int result1 = Integer.parseInt(date3);

int dur1 = (result1 - result);

System.out.println(ID + "\t" + Desc + "\t" + dur1 + " Minutes");

}

statement.close();

connect.close();

}

// Handle the errors for JDBC.

catch(SQLException se)

{

se.printStackTrace();

}

// Handle the errors for Class.forName.

catch(Exception e)

{

e.printStackTrace();

}

}

public static void ratioTickets()

{

try

{

int opencount = 0;

int closecount = 0;

// Closed to open

// Register the JDBC driver.

Class.forName("com.mysql.jdbc.Driver");

// Open a connection to the database.

//System.out.println(" Connecting to a selected database... " + "\n");

Connection connect = (Connection) DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

//System.out.println(" Connected to the database successfully.... " + "\n");

// Execute a query to the database.

//System.out.println(" Creating a statement... " + "\n");

Statement statement = (Statement) connect.createStatement();

Statement statement1 = (Statement) connect.createStatement();

String sql1 ="SELECT COUNT(Status) AS Apple FROM dbarn\_kyan\_ticket WHERE Status = 'Open';";

ResultSet result6 = statement1.executeQuery(sql1);

result6.next();

String Staus = result6.getString("Apple");

String sql = "SELECT COUNT(Status) AS Orange FROM dbarn\_kyan\_ticket WHERE Status = 'Closed';";

ResultSet result5 = statement.executeQuery(sql);

result5.next();

String Staus1 = result5.getString("Orange");

/\* while (result5.next())

{

String Staus = result5.getString("Apple");

String Staus1 = result5.getString("Orange");

if(Staus == "Open")

{

opencount++;

}

else if(Staus == "Closed")

{

closecount++;

}

}

\*/

System.out.println("\nThe ratio of open to closed tickets: " + Staus + " / " + Staus1);

statement.close();

connect.close();

}

// Handle the errors for JDBC.

catch(SQLException se)

{

se.printStackTrace();

}

// Handle the errors for Class.forName.

catch(Exception e)

{

e.printStackTrace();

}

}

//String Priority, Connection connect, Statement statement, ResultSet result

public static void ticketsHigh()

{

try

{

System.out.println("\nThe tickets with high priority:\n");

// Register the JDBC driver.

Class.forName("com.mysql.jdbc.Driver");

// Open a connection to the database.

//System.out.println(" Connecting to a selected database... " + "\n");

Connection connect = (Connection) DriverManager.getConnection("jdbc:mysql://www.papademas.net/tickets?"

+ "user=fp411&password=411");

//System.out.println(" Connected to the database successfully.... " + "\n");

// Execute a query to the database.

//System.out.println(" Creating a statement... " + "\n");

Statement statement = (Statement) connect.createStatement();

// Sort the data for pep in descending order.

String sql = "SELECT Ticket\_ID, Subject, Description, Status, "

+ "Priority FROM dbarn\_kyan\_ticket WHERE Priority = " + "'High';";

ResultSet result = statement.executeQuery(sql);

while (result.next())

{

// Ticket\_ID, Subject, Description, Status, Priority

int ID = result.getInt("Ticket\_ID");

String Sub = result.getString("Subject");

String Desc = result.getString("Description");

String Stat = result.getString("Status");

String Prio = result.getString("Priority");

System.out.println(ID + "\t" + Sub +

"\t" + Desc + "\t" + Stat +

"\t" + Prio);

}

statement.close();

connect.close();

}

// Handle the errors for JDBC.

catch(SQLException se)

{

se.printStackTrace();

}

// Handle the errors for Class.forName.

catch(Exception e)

{

e.printStackTrace();

}

}

public static void main(String args[]) throws MalformedURLException

{

new Tickets();

durationTime();

ratioTickets();

ticketsHigh();

// Change the color of the background of the dialog box

UIManager UI = new UIManager();

UI.put ("OptionPane.background", new ColorUIResource(118, 122, 152));

// Change the color of the panel background of the dialog box

UIManager.put ("Panel.background", new ColorUIResource(241, 235, 228));

// Change the color of the scroll bar background of the dialog box

UIManager.put ("ScrollBar.background", new ColorUIResource(241, 235, 228));

// Change the color of the text area background of the dialog box

UIManager.put ("TextArea.background", new ColorUIResource(241, 235, 228));

// Change the color of the table headers background of the dialog box

UIManager.put ("TableHeader.background", new ColorUIResource(118, 122, 152));

// Change the color of the table background of the dialog box

UIManager.put ("Table.background", new ColorUIResource(241, 235, 228));

// Change the font in the dialog box

UIManager.put ("OptionPane.messageFont", new FontUIResource(new Font("Coda", Font.PLAIN, 14)));

// Change the font in the dialog box

UIManager.put ("TextArea.font", new FontUIResource(new Font("Coda", Font.PLAIN, 14)));

// Change the font in the button

UIManager.put("OptionPane.buttonFont", new Font("Coda", Font.BOLD, 14));

// Change the font in the dialog box

UIManager.put ("OptionPane.messageFont", new FontUIResource(new Font("Coda", Font.PLAIN, 14)));

// Change the font in the table

UIManager.put("TableHeader.font", new Font("Coda", Font.BOLD, 14));

// Change the font in the table box

UIManager.put ("Table.font", new FontUIResource(new Font("Coda", Font.PLAIN, 14)));

}

}

// What to send

// Ticket creation

// All ticket view

// Closed ticket notify

// WH WIFI PROBLEM’ and description

// Ticket delete request

// Total time tickets existed

// Ratio of closed to open tickets

// Tickets open with high priority

// JAR executable file

// Include also in your Word Doc file an opening description

// Summarizing your application, its uses, functionality and

// any basic design principles

// that make your application code worthy!