Minh Nguyen

4/24/2018

**FINAL PROJECT**

**PROJECT DESCRIPTION:**

This project goal is creating a ticket system application that allows tracking and monitoring the ticket to solve the problem. The main function of the program is creating, viewing, deleting and updating ticket.

Only admin role can delete and update the ticket. Root admin also can create tables in the database. Normal user role just can only create and view the ticket.

The program has been developed with strict error trapping algorithms which prevents null insertion, SQL injection, unauthorized access.

The program allow user interacts with main database in real time.

3 tables have been created in tickets database for use:

* Mnguye13\_tickets3: main table for existing ticket
* Mnguye13\_deleteLog3: table for tracking deleted ticket
* Mnguye13\_users3: table of stored users’ credentials and their roles

Root credentials for creating tables is:

|  |  |  |
| --- | --- | --- |
| username | password | role |
| minhnguyen | root | root admin |

\*Note: Since tables have been created in the database, using “root admin” account is not advised. Instead, use “admin” account only.

2 additional accounts have been created for accessing database along with existing accounts:

|  |  |  |
| --- | --- | --- |
| username | password | role |
| bill97 | bill2311 | user |
| mnguye13 | minh123 | admin |

Source Code:

Dao.java

/\*\*

\***@author** Minh Nguyen

\*ITMD 411-Final

\*4/24/2018

\*Dao.java

\*This class will allow for database connectivity and CRUD (Create Read Update Delete) like operations.

\*/

**import** java.io.BufferedReader;

**import** java.io.File;

**import** java.io.FileReader;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.List;

**public** **class** Dao {

// instance fields

**static** Connection *connect* = **null**;

Statement statement = **null**;

**static** List<String> *rowData*;

// constructor

**public** **static** Connection getConnection() {

// Setup the connection with the DB

**try** {

*connect* = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** *connect*;

}

**public** **void** createTables() {

// variables for SQL Query table creations

**final** String createTicketsTable = "CREATE TABLE mnguye13\_tickets3(ticket\_id INT AUTO\_INCREMENT PRIMARY KEY, ticket\_issuer VARCHAR(30) NOT NULL, ticket\_description VARCHAR(200) NOT NULL, ticket\_date DATETIME NOT NULL, ticket\_status VARCHAR(10) NOT NULL)";

**final** String createUsersTable = "CREATE TABLE mnguye13\_users3(uid INT AUTO\_INCREMENT PRIMARY KEY, uname VARCHAR(30) NOT NULL, upass VARCHAR(30) NOT NULL, urole VARCHAR(30) NOT NULL)";

**final** String createDeleteLogTable = "CREATE TABLE mnguye13\_deleteLog3(delete\_id INT AUTO\_INCREMENT PRIMARY KEY, ticket\_id INT, ticket\_deleter VARCHAR(30) NOT NULL, delete\_reason VARCHAR(200) NOT NULL, delete\_date DATETIME NOT NULL)";

**try** {

// create table

statement = *getConnection*().createStatement();

statement.executeUpdate(createTicketsTable);

statement.executeUpdate(createUsersTable);

statement.executeUpdate(createDeleteLogTable);

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

// end create table

// close connection/statement object

statement.close();

*connect*.close();

} **catch** (Exception e) {

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

}

// add users to user table

addUsers();

}

@SuppressWarnings("null")

**public** **void** addUsers() {

// add list of users from userlist.csv file to users table

// variables for SQL Query inserts

String sql;

Connection connect = **null**;

Statement statement = **null**;

BufferedReader br;

List<List<String>> array = **new** ArrayList<>(); // array list to hold

// spreadsheet rows &

// columns

// read data from file

**try** {

br = **new** BufferedReader(**new** FileReader(**new** File("./userlist.csv")));

String line;

**while** ((line = br.readLine()) != **null**) {

array.add(Arrays.*asList*(line.split(",")));

}

} **catch** (Exception e) {

System.***out***.println("There was a problem loading the file");

}

**try** {

// Setup the connection with the DB

statement = *getConnection*().createStatement();

// create loop to grab each array index containing a list of values

// and PASS (insert) that data into your User table

**for** (List<String> rowData : array) {

sql = "insert into mnguye13\_users3(uname,upass,urole) " + "values('" + rowData.get(0) + "','" + rowData.get(1) + "','" + rowData.get(2)

+ "');";

statement.executeUpdate(sql);

}

System.***out***.println("Inserts completed in the given database...");

// close connection/statement object

statement.close();

connect.close();

} **catch** (Exception e) {

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

}

}

// add other desired CRUD methods needed like for updates, deletes, etc.

}

ticketsJTable.java

/\*\*

\***@author** Minh Nguyen

\*ITMD 411-Final

\*4/24/2018

\*Dao.java

\*This class will automatically pulling metadata and placing that into the JTable to set up column structure and the grabbing of row data is accomplished automatically as well.

\*/

**import** java.sql.ResultSet;

**import** java.sql.ResultSetMetaData;

**import** java.sql.SQLException;

**import** java.util.Vector;

**import** javax.swing.table.DefaultTableModel;

**public** **class** ticketsJTable {

@SuppressWarnings("unused")

**private** **final** DefaultTableModel tableModel = **new** DefaultTableModel();

**public** **static** DefaultTableModel buildTableModel(ResultSet rs) **throws** SQLException {

ResultSetMetaData metaData = rs.getMetaData();

// names of columns

Vector<String> columnNames = **new** Vector<String>();

**int** columnCount = metaData.getColumnCount();

**for** (**int** column = 1; column <= columnCount; column++) {

columnNames.add(metaData.getColumnName(column));

}

// data of the table

Vector<Vector<Object>> data = **new** Vector<Vector<Object>>();

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

Vector<Object> vector = **new** Vector<Object>();

**for** (**int** columnIndex = 1; columnIndex <= columnCount; columnIndex++) {

vector.add(rs.getObject(columnIndex));

}

data.add(vector);

}

// return data/col.names for JTable

**return** **new** DefaultTableModel(data, columnNames);

}

}

Login.java

/\*\*

\***@author** Minh Nguyen

\*ITMD 411-Final

\*4/24/2018

\*Login.java

\*This class will allow user authentication (identify and verify) for database access.

\*/

**import** java.awt.Color;

**import** java.awt.FlowLayout;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**import** java.sql.Connection;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

**import** com.mysql.jdbc.PreparedStatement;

**public** **class** Login {

// create instance fields for class

**private** JFrame mainFrame;

**private** JLabel headerLabel;

**private** JLabel statusLabel;

**private** JLabel namelabel;

**private** JLabel passwordLabel;

**private** JTextField userText;

**private** JPasswordField passwordText;

**private** JButton loginButton;

**private** JPanel controlPanel;

**static** **private** String *role*= **null**;

**public** Login() {

prepareGUI();

showTextFields();

}

**private** **void** prepareGUI() {

// instantiate objects

mainFrame = **new** JFrame("Ticket System"); // title of window form

headerLabel = **new** JLabel("", JLabel.***CENTER***);

statusLabel = **new** JLabel("", JLabel.***CENTER***);

controlPanel = **new** JPanel();

// window frame settings

mainFrame.setSize(400, 400);

mainFrame.setLayout(**new** GridLayout(3, 1));

mainFrame.getContentPane().setBackground(Color.***GRAY***);

mainFrame.setLocationRelativeTo(**null**);

// frame object settings

headerLabel.setText("Account Access");

statusLabel.setSize(350, 100);

// add frame objects to mainframe

mainFrame.add(headerLabel);

mainFrame.add(controlPanel);

mainFrame.add(statusLabel);

mainFrame.addWindowListener(**new** WindowAdapter() {

**public** **void** windowClosing(WindowEvent wE) { // define a window close

// operation

System.*exit*(0);

}

});

}

**private** **void** showTextFields() {

// instantiate controls

namelabel = **new** JLabel("User ID: ", JLabel.***RIGHT***);

passwordLabel = **new** JLabel("Password: ", JLabel.***CENTER***);

userText = **new** JTextField(6);

passwordText = **new** JPasswordField(6);

loginButton = **new** JButton("Login");

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

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

/\*

\* Check credentials for various users

\*

\* Administrator is a super user to the ticket system

\* Code below uses a default (temporary) hard coded admin user

\* name/password for verification. You can change this setting

\* if you like.

\*/

// Create user friendly variable name to store user name from text box

String userName = userText.getText();

// convert characters from password field to string for input validation

String password = **new** String(passwordText.getPassword());

**boolean** adminFlag = **false**;

**if** (userName.equals("minhnguyen") && password.equals("root")) {

adminFlag = **true**;

*role*= "admin";

// close of Login window

mainFrame.dispose();

// open up ticketsGUI file upon successful login

**try** {

**new** ticketsGUI("Admin", *role*);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

} // establish role as admin via constructor call

}

/\*

\* match credentials from text fields with users table for a

\* match for regular users

\*/

**else** **if** (!adminFlag) {

Connection connect = Dao.*getConnection*();

String queryString = "SELECT uname, upass, urole FROM mnguye13\_users3 where uname=? and upass=?";

PreparedStatement ps;

ResultSet results = **null**;

**try** {

// set up prepared statements to execute query string cleanly and safely

ps = (PreparedStatement) connect.prepareStatement(queryString);

ps.setString(1, userName);

ps.setString(2, password);

results = ps.executeQuery();

**if** (results.next()) { // verify if a record match exists

// in table

JOptionPane.*showMessageDialog*(**null**, "Username and Password exist");

// close of Login window

mainFrame.dispose();

// open up ticketsGUI file upon successful login

*role*= results.getString(3);

**new** ticketsGUI(userName,*role*); // establish role as

// regular user via

// constructor call

//System.out.println(role);

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Please Check Username and Password ");

}

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

} **finally** {

**try** {

results.close();

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**try** {

connect.close();

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

}

}

});

// add layout type /background color to control panel

controlPanel.setLayout(**new** FlowLayout());

controlPanel.setBackground(Color.***GRAY***);

// add controls to control panel

controlPanel.add(namelabel);

controlPanel.add(userText);

controlPanel.add(passwordLabel);

controlPanel.add(passwordText);

controlPanel.add(loginButton);

// lastly set visibility of Window as all controls are instantiated for

// frame

mainFrame.setVisible(**true**);

}

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

**new** Login();

}

}

ticketsGUI.java

/\*\*

\***@author** Minh Nguyen

\*ITMD 411-Final

\*4/24/2018

\*ticketGUI.java

\*This class will create main GUI for user to access and interact with the database.

\*Admin can create, view, update, delete tickets but normal user only can view and update ticket.

\*/

**import** java.awt.BorderLayout;

**import** java.awt.Color;

**import** java.awt.Dimension;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowAdapter;

**import** java.awt.event.WindowEvent;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** javax.swing.BoxLayout;

**import** javax.swing.ButtonGroup;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JRadioButton;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextArea;

**import** javax.swing.JTextField;

**public** **class** ticketsGUI **extends** JFrame **implements** ActionListener {

**public** **static** **final** String[] ***status*** = { "Requested", "Opened", "Closed","Awaiting","Responded","Escalated", "Resolved" };

**public** **static** **final** String[] ***viewOption*** = { "ticket\_id", "ticket\_issuer", "ticket\_description","ticket\_date","ticket\_status"};

**static** java.sql.Timestamp *RealTime* = **new** java.sql.Timestamp(**new** java.util.Date().getTime());

// class level member objects

Dao dao = **new** Dao(); // for CRUD operations

String chkIfAdmin = **null**;

**private** JFrame mainFrame, layoutFrame;

JScrollPane sp = **null**;

// Main menu object items

**private** JMenu mnuFile = **new** JMenu("File");

**private** JMenu mnuAdmin = **new** JMenu("Admin");

**private** JMenu mnuTickets = **new** JMenu("Tickets");

/\* add any more Main menu object items below \*/

// Sub menu item objects for all Main menu item objects

JMenuItem mnuItemExit;

JMenuItem mnuItemUpdate;

JMenuItem mnuItemDelete;

JMenuItem mnuItemOpenTicket;

JMenuItem mnuItemViewTicket;

JMenuItem mnuItemChangeStatus;

JLabel loginNote;

JTable jt;

JButton CreateTicket, ViewTicket, UpdateTicket, DeleteTicket, LogOut, switchUser, submitUpdate,submitCreate;

JPanel sub, sub1,sub2,sub3,sub4;

JRadioButton asc, desc;

ButtonGroup order;

JComboBox statCb, viewCb;

JTextField UpdateID;

JTextArea Udescription;

JLabel UpdateId, UpdateInfo, UpdateStat;

JTextField CreateName;

JTextArea Cdescription;

JLabel CreateIssuer, Createinfo, CreateStat;

/\* add any more Sub object items below \*/

// constructor

**public** **void** refresh()

{

}

**public** ticketsGUI(String verifyRole, String role) **throws** SQLException {

chkIfAdmin = verifyRole;

JOptionPane.*showMessageDialog*(**null**, "Welcome " + verifyRole);

**if** (chkIfAdmin.equals("Admin") || role.equals("admin"))

{

System.***out***.println("ADMIN GUI");

dao.createTables(); // fire up table creations (tickets / user

} // tables)

**else** **if**(role.equals("user"))

{

System.***out***.println("USER GUI");

}

/\*

\* else do something else if you like

\*

\*/

createMenu();

prepareGUI(role);

}

**private** **void** createMenu() {

/\* Initialize sub menu items \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// initialize sub menu item for File main menu

mnuItemExit = **new** JMenuItem("Exit");

// add to File main menu item

mnuFile.add(mnuItemExit);

// initialize first sub menu items for Admin main menu

mnuItemUpdate = **new** JMenuItem("Update Ticket");

// add to Admin main menu item

mnuAdmin.add(mnuItemUpdate);

// initialize second sub menu items for Admin main menu

mnuItemDelete = **new** JMenuItem("Delete Ticket");

// add to Admin main menu item

mnuAdmin.add(mnuItemDelete);

//

mnuItemChangeStatus = **new** JMenuItem("Change Ticket Status");

// add to Admin main menu item

mnuAdmin.add(mnuItemChangeStatus);

// initialize first sub menu item for Tickets main menu

mnuItemOpenTicket = **new** JMenuItem("Open Ticket");

// add to Ticket Main menu item

mnuTickets.add(mnuItemOpenTicket);

// initialize second sub menu item for Tickets main menu

mnuItemViewTicket = **new** JMenuItem("View Ticket");

// add to Ticket Main menu item

mnuTickets.add(mnuItemViewTicket);

// initialize any more desired sub menu items below

/\* Add action listeners for each desired menu item \*\*\*\*\*\*\*\*\*\*\*\*\*/

mnuItemExit.addActionListener(**this**);

mnuItemUpdate.addActionListener(**this**);

mnuItemDelete.addActionListener(**this**);

mnuItemOpenTicket.addActionListener(**this**);

mnuItemViewTicket.addActionListener(**this**);

mnuItemChangeStatus.addActionListener(**this**);

// add any more listeners for any additional sub menu items if desired

}

**private** **void** prepareGUI(String role) **throws** SQLException {

// initialize frame object

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

mainFrame = **new** JFrame("Tickets System-Login as "+role.toUpperCase()+"@ "+*RealTime*);

// create jmenu bar

JMenuBar bar = **new** JMenuBar();

bar.add(mnuFile); // add main menu items in order, to JMenuBar

**if**(role.equals("admin"))

{

bar.add(mnuAdmin);

}

bar.add(mnuTickets);

// add menu bar components to frame

mainFrame.setJMenuBar(bar);

mainFrame.addWindowListener(**new** WindowAdapter() {

// define a window close operation

**public** **void** windowClosing(WindowEvent wE) {

System.*exit*(0);

}

});

// set frame options

mainFrame.setSize(1500, 500);

mainFrame.getContentPane().setBackground(Color.***LIGHT\_GRAY***);

mainFrame.setLocationRelativeTo(**null**);

mainFrame.setVisible(**true**);

//loginNote=new JLabel("You are login as "+role);

//loginNote.setHorizontalAlignment(JLabel.CENTER);

sub = **new** JPanel();

sub1= **new** JPanel();

sub3= **new** JPanel();

sub3.setLayout(**new** BoxLayout(sub3, BoxLayout.***Y\_AXIS***));

sub1.setLayout(**new** BoxLayout(sub1, BoxLayout.***Y\_AXIS***));

sub4= **new** JPanel();

//sub4.setLayout(new BoxLayout(sub4, BoxLayout.PAGE\_AXIS));

sub4.setPreferredSize(**new** Dimension (100, 200));

order = **new** ButtonGroup();

CreateTicket=**new** JButton("New Ticket");

ViewTicket= **new** JButton("View Ticket");

UpdateTicket=**new** JButton("Update Ticket");

DeleteTicket= **new** JButton("Delete Ticket");

LogOut= **new** JButton("Log Out ");

switchUser= **new** JButton("Switch User");

asc = **new** JRadioButton("Ascending ");

desc= **new** JRadioButton("Descending");

//byId= new JRadioButton("ID");

//byName= new JRadioButton("Name");

//byStatus= new JRadioButton("Status");

//byDate = new JRadioButton("Date");

asc.setMaximumSize(desc.getPreferredSize());

sub1.add(CreateTicket);

CreateTicket.setMaximumSize(UpdateTicket.getPreferredSize());

DeleteTicket.setMaximumSize(UpdateTicket.getPreferredSize());

**if**(role.equals("admin"))

{

//UpdateTicket=new JButton("Update Ticket");

//DeleteTicket= new JButton("Delete Ticket");

sub1.add(UpdateTicket);

sub1.add(DeleteTicket);

}

sub3.add(switchUser);

sub3.add(LogOut);

LogOut.setMaximumSize(switchUser.getPreferredSize());

sub.add(ViewTicket);

order.add(asc);

order.add(desc);

sub.add(asc);

sub.add(desc);

//order.add(byStatus);

//order.add(byDate);

//sub3.add(byId);

//sub3.add(byName);

//sub3.add(byStatus);

//sub3.add(byDate);

viewCb= **new** JComboBox(***viewOption***);

viewCb.setMaximumSize(asc.getPreferredSize());

sub.add(viewCb);

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.getContentPane().add(sub, BorderLayout.***NORTH***);

mainFrame.getContentPane().add(sub1, BorderLayout.***WEST***);

mainFrame.getContentPane().add(sub3, BorderLayout.***EAST***);

mainFrame.getContentPane().add(sub4,BorderLayout.***SOUTH***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

LogOut.addActionListener(**this**);

switchUser.addActionListener(**this**);

CreateTicket.addActionListener(**this**);

UpdateTicket.addActionListener(**this**);

DeleteTicket.addActionListener(**this**);

ViewTicket.addActionListener(**this**);

viewCb.addActionListener(**this**);

}

**boolean** valid= **false**;

**private** String ticketName=**null**;

**private** String ticketDesc=**null**;

/\*

\* action listener fires up items clicked on from sub menus with one action

\* performed event handler!

\*/

@Override

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

// **TODO** Auto-generated method stub

// implement actions for sub menu items

**if** (e.getSource() == mnuItemExit || e.getSource() == LogOut) {

System.*exit*(0);

} **else** **if**(e.getSource()== switchUser){

mainFrame.dispose();

**new** Login();

} **else** **if**(e.getSource()== ViewTicket){

String Order=**null**;

**if**(asc.isSelected()){

Order="asc";

}

**else** **if** (desc.isSelected())

{

Order="desc";

}

**else**

{

Order="desc";

}

//System.out.println(viewCb.getSelectedItem());

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

ResultSet results = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by "+ viewCb.getSelectedItem()+" "+ Order+";");

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

// display the table model off your result set!!!

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

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**else** **if** (e.getSource() == mnuItemOpenTicket) {

**try** {

ticketName=**null**;

ticketDesc=**null**;

valid=**false**;

**while**(valid==**false**)

{

// get ticket information

ticketName = JOptionPane.*showInputDialog*(**null**, "Enter your name");

**if** (ticketName.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid name! Try again!");

}

**else**

{

valid=**true**;

}

}

valid=**false**;

**while**(valid==**false**)

{

ticketDesc = JOptionPane.*showInputDialog*(**null**, "Enter a ticket description");

**if** (ticketDesc.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid description! Try again!");

}

**else**

{

valid=**true**;

}

}

java.sql.Timestamp date = **new** java.sql.Timestamp(**new** java.util.Date().getTime());

String ticket\_status = (String) JOptionPane.*showInputDialog*(**null**,

"Enter ticket status",

"ticket status",

JOptionPane.***QUESTION\_MESSAGE***,

**null**,

***status***,

***status***[0]);

// insert ticket information to database

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

**if** (ticketName != **null** && !ticketName.isEmpty() && ticketDesc !=**null** && !ticketDesc.isEmpty() ) {

// doSomething

**int** result = statement

.executeUpdate("Insert into mnguye13\_tickets3(ticket\_issuer, ticket\_description, ticket\_date, ticket\_status) values(" + " '"

+ ticketName + "','" + ticketDesc + "','" + date+ "','" + ticket\_status + "')", Statement.***RETURN\_GENERATED\_KEYS***);

// retrieve ticket id number newly auto generated upon record

// insertion

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id = 0;

**if** (resultSet.next()) {

id = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (result != 0) {

System.***out***.println("Ticket ID : " + id + " created successfully!!!");

JOptionPane.*showMessageDialog*(**null**, "Ticket id: " + id + " created");

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE CREATED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

}

**else**

{

JOptionPane.*showMessageDialog*(**null**, "Invalid Insertion! Ticket cannot be created");

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

} **catch** (SQLException ex) {

// **TODO** Auto-generated catch block

ex.printStackTrace();

}

} **else** **if** (e.getSource() == mnuItemViewTicket) {

// retrieve ticket information for viewing in JTable

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

ResultSet results = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

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

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

} **else** **if** (e.getSource() == mnuItemDelete || e.getSource()== DeleteTicket)

// delete ticket

{

valid=**false**;

String id=**null**;

String reason = **null**;

String deleter= **null**;

JFrame frame = **new** JFrame("DELETE TICKET");

**while** (valid==**false**)

{

id = JOptionPane.*showInputDialog*(

frame,

"Enter ticket ID to continue",

"DELETE TICKET",

JOptionPane.***WARNING\_MESSAGE***);

**if** (id.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid ID! Try again!");

}

**else**

{

valid=**true**;

}

}

valid=**false**;

**while** (valid==**false**)

{

deleter = JOptionPane.*showInputDialog*(

frame,

"Enter your name",

JOptionPane.***WARNING\_MESSAGE***);

**if** (deleter.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid description! Try again!");

}

**else**

{

valid=**true**;

}

}

valid=**false**;

**while** (valid==**false**)

{

reason = JOptionPane.*showInputDialog*(

frame,

"Enter reason to delete the ticket",

JOptionPane.***WARNING\_MESSAGE***);

**if** (reason.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid description! Try again!");

}

**else**

{

valid=**true**;

}

}

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

java.sql.Timestamp date = **new** java.sql.Timestamp(**new** java.util.Date().getTime());

**int** result = statement.executeUpdate("Insert into mnguye13\_deleteLog3(ticket\_id, ticket\_deleter, delete\_reason, delete\_date) values(" + " '"+ id + "','" + deleter + "','" + reason+ "','" + date + "')", Statement.***RETURN\_GENERATED\_KEYS***);

**int** results = statement.executeUpdate("DELETE FROM mnguye13\_tickets3 WHERE ticket\_id = "+id+";",Statement.***RETURN\_GENERATED\_KEYS***);

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id1 = 0;

**if** (resultSet.next()) {

id1 = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (results != 0) {

System.***out***.println("Ticket ID : " + id + " deleted successfully!!!");

//JOptionPane.showMessageDialog(null, "Ticket id: " + id1 + " deleted");

JOptionPane.*showMessageDialog*(**null**,"DELETE TICKET [" +id+ "] SUCCESSFULLY!","TICKET DELETED",JOptionPane.***INFORMATION\_MESSAGE***);

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE DELETED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**else** **if** (e.getSource()==mnuItemUpdate)

{

valid=**false**;

String description = **null**;

String id=**null**;

JFrame frame = **new** JFrame("UPDATE TICKET");

**while** (valid==**false**)

{

id = JOptionPane.*showInputDialog*(

frame,

"Enter ticket ID to continue",

"UPDATE TICKET",

JOptionPane.***WARNING\_MESSAGE***);

**if** (id.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid ID! Try again!");

}

**else**

{

valid=**true**;

}

}

valid=**false**;

**while** (valid==**false**)

{

description = JOptionPane.*showInputDialog*(

frame,

"Enter ticket description to change",

JOptionPane.***WARNING\_MESSAGE***);

**if** (description.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid description! Try again!");

}

**else**

{

valid=**true**;

}

}

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

**int** results = statement.executeUpdate("UPDATE mnguye13\_tickets3 SET ticket\_description = '"+description+"' WHERE ticket\_id ="+id+";",Statement.***RETURN\_GENERATED\_KEYS***);

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id1 = 0;

**if** (resultSet.next()) {

id1 = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (results != 0) {

System.***out***.println("Ticket ID : " + id + " updated successfully!!!");

JOptionPane.*showMessageDialog*(**null**,"UPDATE TICKET ["+id+"] SUCCESFULLY!","TICKET UPDATE",JOptionPane.***INFORMATION\_MESSAGE***);

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE UPDATED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

//JOptionPane.showMessageDialog(null,"UPDATE TICKET ["+id+"] SUCCESFULLY!","TICKET UPDATE",JOptionPane.INFORMATION\_MESSAGE);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}

**else** **if** (e.getSource()==mnuItemChangeStatus)

{

valid=**false**;

String id=**null**;

JFrame frame = **new** JFrame("CHANGE TICKET STATUS");

**while** (valid==**false**)

{

id = JOptionPane.*showInputDialog*(

frame,

"Enter ticket ID to continue",

"CHANGE TICKET STATUS",

JOptionPane.***WARNING\_MESSAGE***);

**if** (id.isEmpty()){

JOptionPane.*showMessageDialog*(**null**, "Invalid ID! Try again!");

}

**else**

{

valid=**true**;

}

}

String ticket\_status = (String) JOptionPane.*showInputDialog*(frame,

"Enter ticket status",

"ticket status",

JOptionPane.***QUESTION\_MESSAGE***,

**null**,

***status***,

***status***[0]);

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

**int** results = statement.executeUpdate("UPDATE mnguye13\_tickets3 SET ticket\_status = '"+ticket\_status+"' WHERE ticket\_id ="+id+";",Statement.***RETURN\_GENERATED\_KEYS***);

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id1 = 0;

**if** (resultSet.next()) {

id1 = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (results != 0) {

System.***out***.println("Ticket ID : " + id + " updated successfully!!!");

JOptionPane.*showMessageDialog*(**null**,"UPDATE TICKET ["+id+"] SUCCESFULLY!","TICKET UPDATE",JOptionPane.***INFORMATION\_MESSAGE***);

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE UPDATED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

//JOptionPane.showMessageDialog(null,"UPDATE TICKET ["+id+"] SUCCESFULLY!","TICKET UPDATE",JOptionPane.INFORMATION\_MESSAGE);

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}**else** **if** (e.getSource()==CreateTicket)

{

sub4.setVisible(**false**);

//jt.setVisible(false);

//sub4.setVisible(false);

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

//sp.setVisible(true);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

System.***out***.println("Button Create");

CreateName = **new** JTextField(10);

Cdescription = **new** JTextArea(5,20);

CreateIssuer= **new** JLabel("Issuer Name: ");

Createinfo= **new** JLabel("Description: ");

CreateStat= **new** JLabel("Status: ");

submitCreate=**new** JButton("CREATE");

sub4= **new** JPanel();

//sub4.setLayout(new BoxLayout(sub4, BoxLayout.PAGE\_AXIS));

sub4.setPreferredSize(**new** Dimension (100, 200));

sub4.add(CreateIssuer);

sub4.add(CreateName);

sub4.add(Createinfo);

sub4.add(Cdescription);

sub4.add(CreateStat);

statCb=**new** JComboBox(***status***);

//cb.setMaximumSize( asc.getPreferredSize() );

sub4.add(statCb);

sub4.add(submitCreate);

//mainFrame.getContentPane().add(sp, BorderLayout.CENTER);

mainFrame.getContentPane().add(sub4,BorderLayout.***SOUTH***);

mainFrame.setVisible(**true**);

submitCreate.addActionListener(**this**);

}

**else** **if** (e.getSource()==UpdateTicket)

{

sub4.setVisible(**false**);

//jt.setVisible(false);

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

//sp.setVisible(true); // screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

System.***out***.println("Button Update");

UpdateID = **new** JTextField(10);

Udescription = **new** JTextArea(5,20);

UpdateId= **new** JLabel("ID: ");

UpdateInfo= **new** JLabel("Description: ");

UpdateStat= **new** JLabel("Status: ");

submitUpdate=**new** JButton("UPDATE");

sub4= **new** JPanel();

//sub4.setLayout(new BoxLayout(sub4, BoxLayout.PAGE\_AXIS));

sub4.setPreferredSize(**new** Dimension (100, 200));

sub4.add(UpdateId);

sub4.add(UpdateID);

sub4.add(UpdateInfo);

sub4.add(Udescription);

sub4.add(UpdateStat);

statCb=**new** JComboBox(***status***);

//cb.setMaximumSize( asc.getPreferredSize() );

sub4.add(statCb);

sub4.add(submitUpdate);

submitUpdate.addActionListener(**this**);

//mainFrame.getContentPane().add(sp, BorderLayout.CENTER);

mainFrame.getContentPane().add(sub4,BorderLayout.***SOUTH***);

mainFrame.setVisible(**true**);

} **else** **if** (e.getSource()==submitUpdate) {

System.***out***.println(UpdateID.getText());

System.***out***.println(Udescription.getText());

System.***out***.println(statCb.getSelectedItem());

**if**(UpdateID.getText().isEmpty() || Udescription.getText().isEmpty())

{

System.***out***.println("ERROR");

JOptionPane.*showMessageDialog*(**null**,"Error! Missing 1 or more field(s)","INVALID",JOptionPane.***ERROR\_MESSAGE***);

}

**else**

{

**try** {

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

**int** results = statement.executeUpdate("UPDATE mnguye13\_tickets3 SET ticket\_description = '"+Udescription.getText()+"', ticket\_status = '"+statCb.getSelectedItem() +"' WHERE ticket\_id ="+UpdateID.getText()+";",Statement.***RETURN\_GENERATED\_KEYS***);

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id1 = 0;

**if** (resultSet.next()) {

id1 = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (results != 0) {

System.***out***.println("Ticket ID : " + UpdateID.getText() + " updated successfully!!!");

JOptionPane.*showMessageDialog*(**null**,"UPDATE TICKET ["+UpdateID.getText()+"] SUCCESFULLY!","TICKET UPDATE",JOptionPane.***INFORMATION\_MESSAGE***);

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE UPDATED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

statement.close();

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

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

UpdateID.setText("");

Udescription.setText("");

statCb.setSelectedIndex(0);

}

} **else** **if** (e.getSource()==submitCreate) {

System.***out***.println(CreateName.getText());

System.***out***.println(Cdescription.getText());

System.***out***.println(statCb.getSelectedItem());

**if**(CreateName.getText().isEmpty() || Cdescription.getText().isEmpty())

{

System.***out***.println("ERROR");

JOptionPane.*showMessageDialog*(**null**,"Error! Missing 1 or more field(s)","INVALID",JOptionPane.***ERROR\_MESSAGE***);

}

**else**

{

**try** {

java.sql.Timestamp date = **new** java.sql.Timestamp(**new** java.util.Date().getTime());

// insert ticket information to database

Connection dbConn = DriverManager

.*getConnection*("jdbc:mysql://www.papademas.net/tickets?autoReconnect=true&useSSL=false"

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

Statement statement = dbConn.createStatement();

// doSomething

**int** result = statement

.executeUpdate("Insert into mnguye13\_tickets3(ticket\_issuer, ticket\_description, ticket\_date, ticket\_status) values(" + " '"

+ CreateName.getText() + "','" + Cdescription.getText() + "','" + date+ "','" + statCb.getSelectedItem() + "')", Statement.***RETURN\_GENERATED\_KEYS***);

// retrieve ticket id number newly auto generated upon record

// insertion

ResultSet resultSet = **null**;

resultSet = statement.getGeneratedKeys();

**int** id = 0;

**if** (resultSet.next()) {

id = resultSet.getInt(1); // retrieve first field in table

}

// display results if successful or not to console / dialog box

**if** (result != 0) {

System.***out***.println("Ticket ID : " + id + " created successfully!!!");

//JOptionPane.showMessageDialog(null, "Ticket id: " + id + " created");

JOptionPane.*showMessageDialog*(**null**,"CREATE TICKET ["+id+"] SUCCESFULLY!","TICKET CREATED",JOptionPane.***INFORMATION\_MESSAGE***);

} **else** {

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

JOptionPane.*showMessageDialog*(**null**,"TICKET CANNOT BE CREATED","ERROR",JOptionPane.***ERROR\_MESSAGE***);

}

ResultSet view = statement.executeQuery("SELECT \* FROM mnguye13\_tickets3 order by ticket\_date desc");

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

// display the table model off your result set!!!

jt = **new** JTable(ticketsJTable.*buildTableModel*(view));

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

sp = **new** JScrollPane(jt);

//mainFrame.add(sp);

mainFrame.getContentPane().add(sp, BorderLayout.***CENTER***);

mainFrame.setVisible(**true**); // refreshes or repaints frame on

// screen

} **catch** (SQLException ex) {

// **TODO** Auto-generated catch block

ex.printStackTrace();

}

CreateName.setText("");

Cdescription.setText("");

statCb.setSelectedIndex(0);

}

}

/\*

\* continue implementing any other desired sub menu items (like for

\* update and delete sub menus for example) with similar syntax &

\* logic as shown above\*

\*/

}

**private** JButton JButton(String string) {

// **TODO** Auto-generated method stub

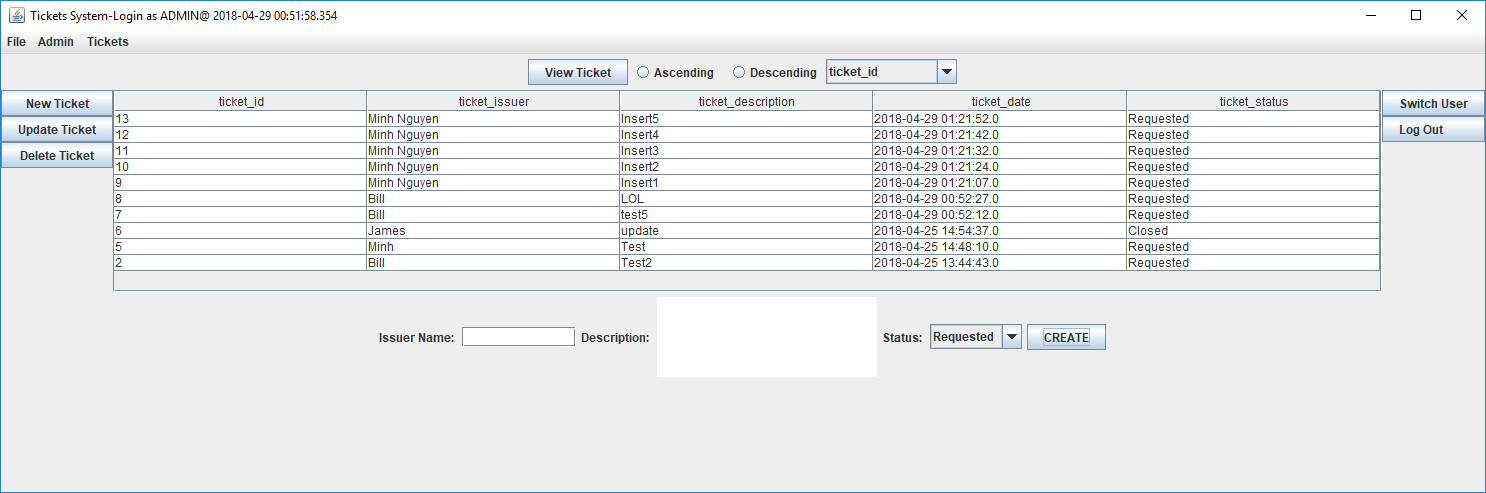
**return** **null**;

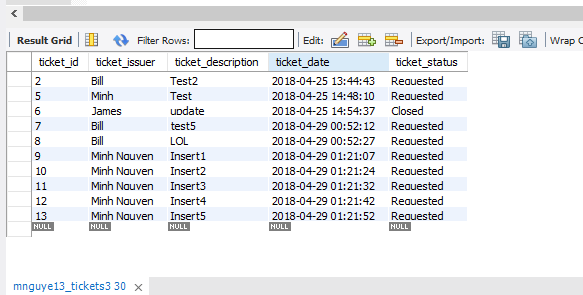
}

}

**OUTPUT:**

1. Please insert at least 5 tickets into the DB. Include a record with your name into the DB.

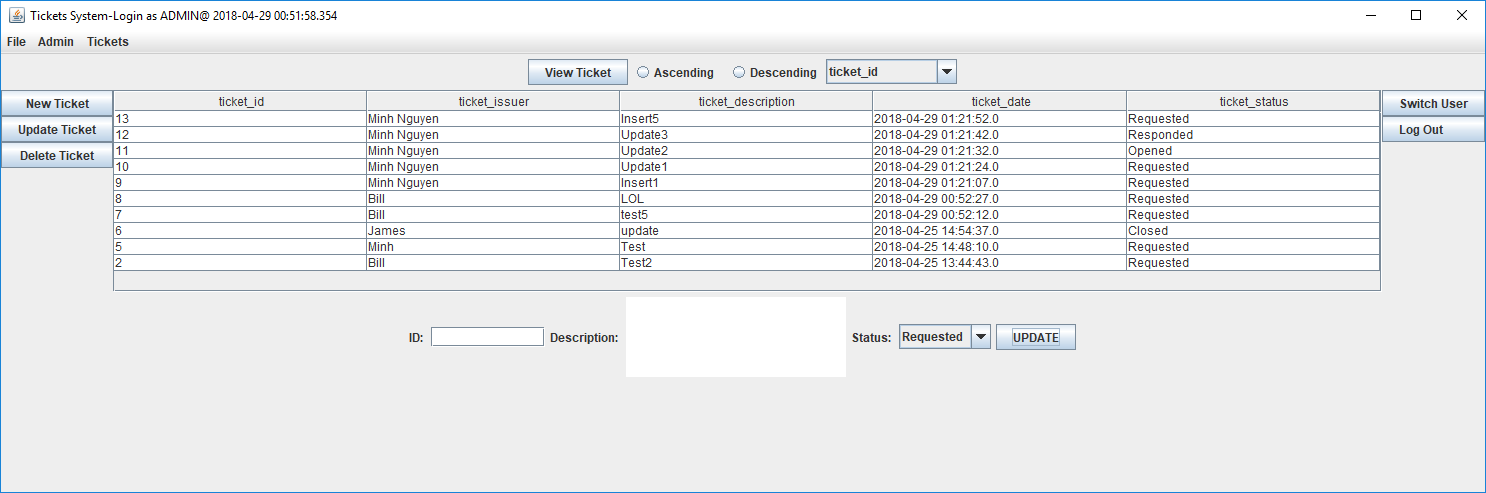


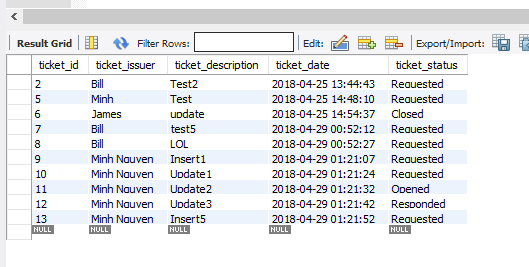


2. Update your record by changing your ticket description.

Ticket #10, #11, #12 have been updated.

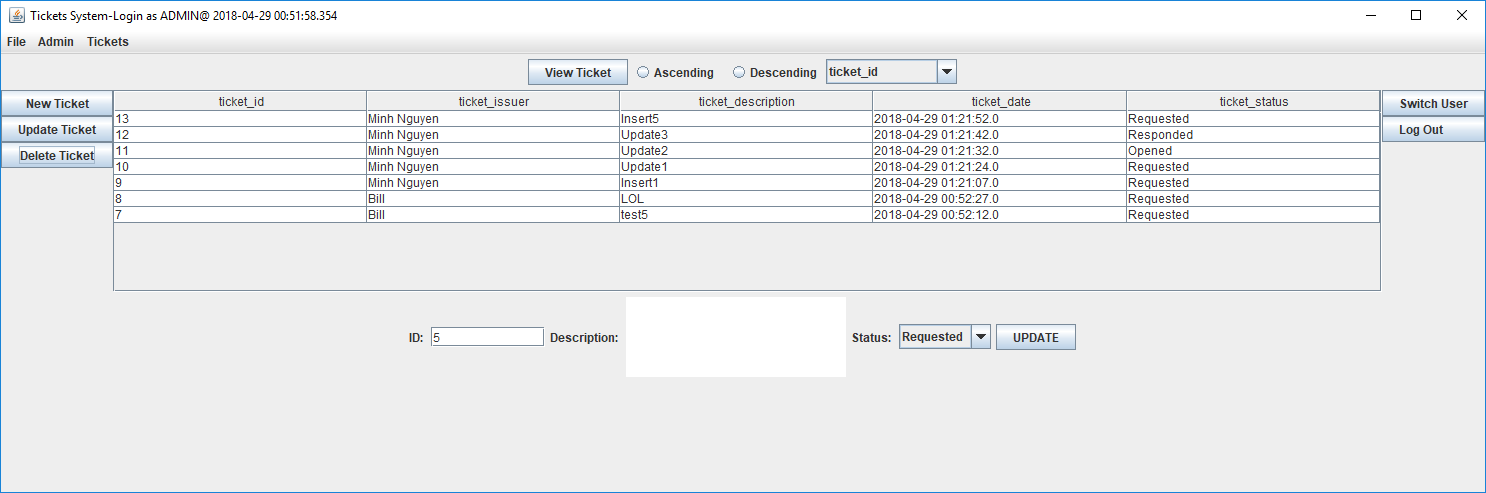
3. Show a view of your updated ticket.



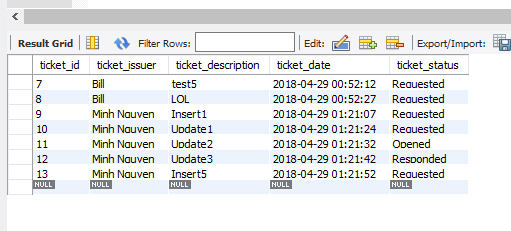


4. Delete your ticket from the DB.

Ticket #2, #5, #6 have been deleted.

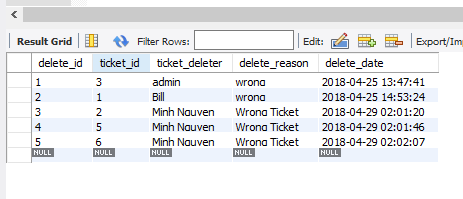


Database view for table mnguye13\_tickets3



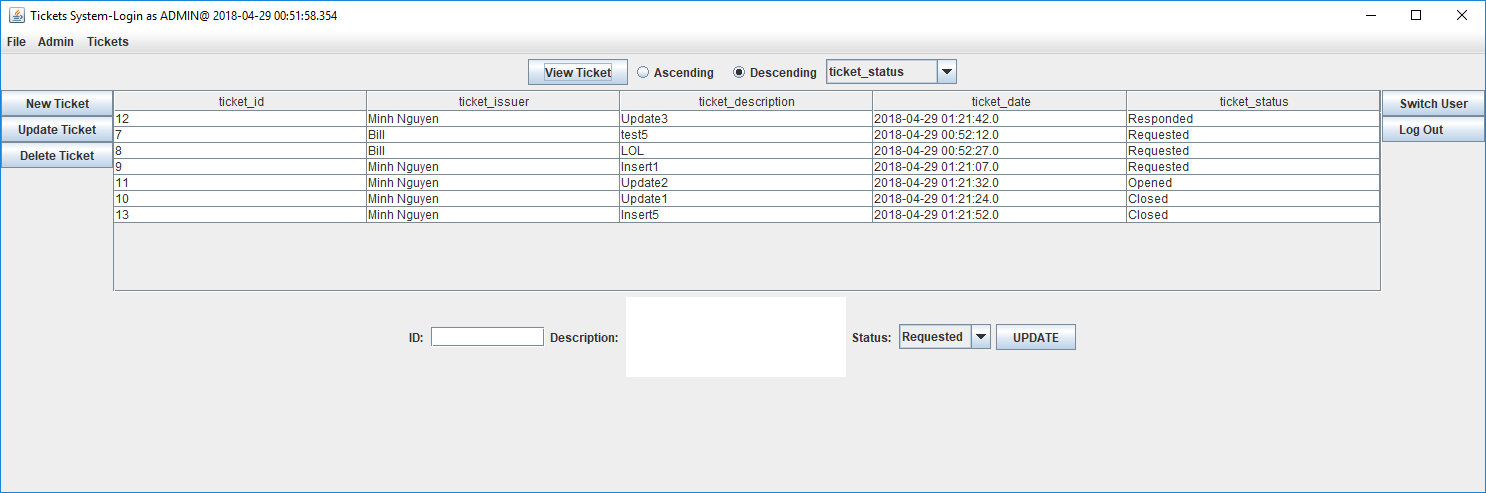
Database view for table mnguye13\_deleteLog3

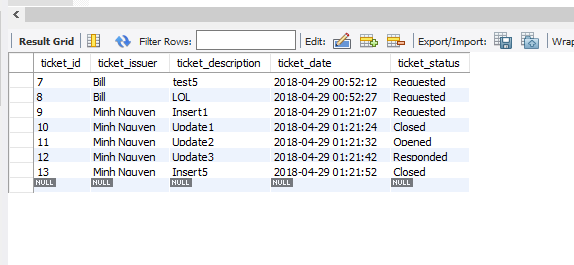
* Shows that ticket #2, #5, #6 has been deleted by Minh Nguyen (reason of wrong ticket)



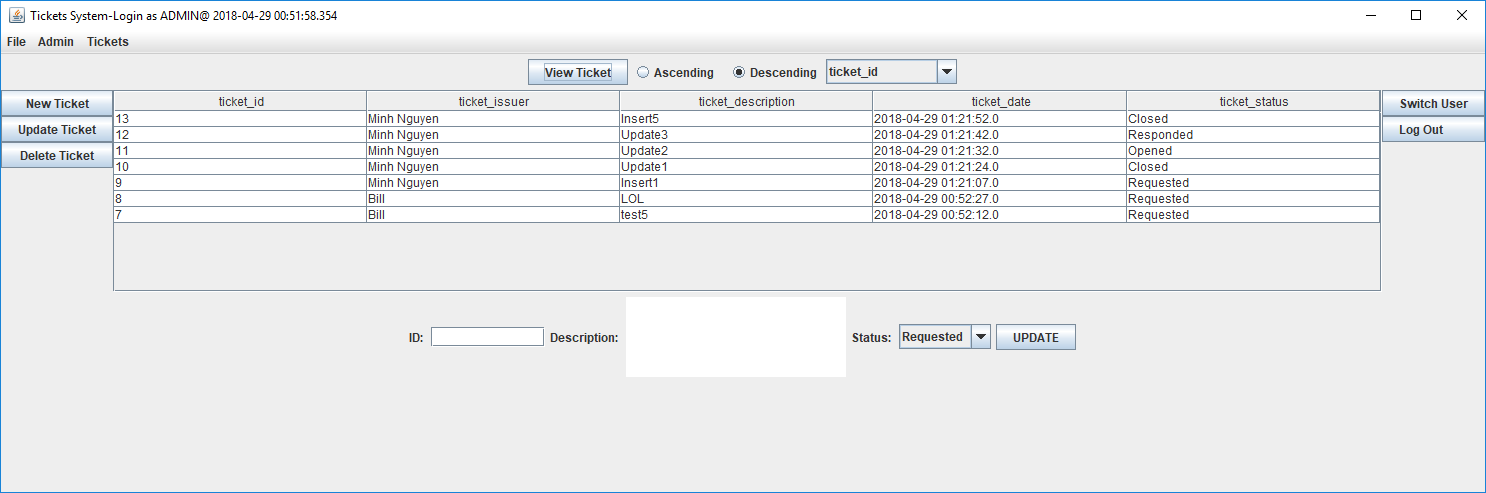
5. Close two existing tickets.

Ticket #10, #13’s ticket statuses have been changed to CLOSED

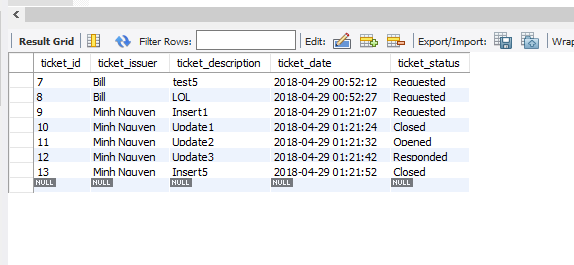




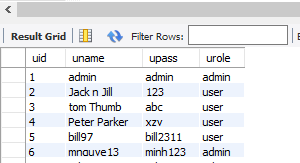
6. Lastly, show a table view of all of your tickets.



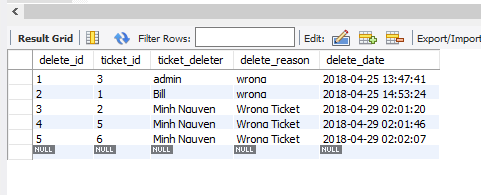
Mnguye13\_tickets3 table



Mnguye13\_users3 table

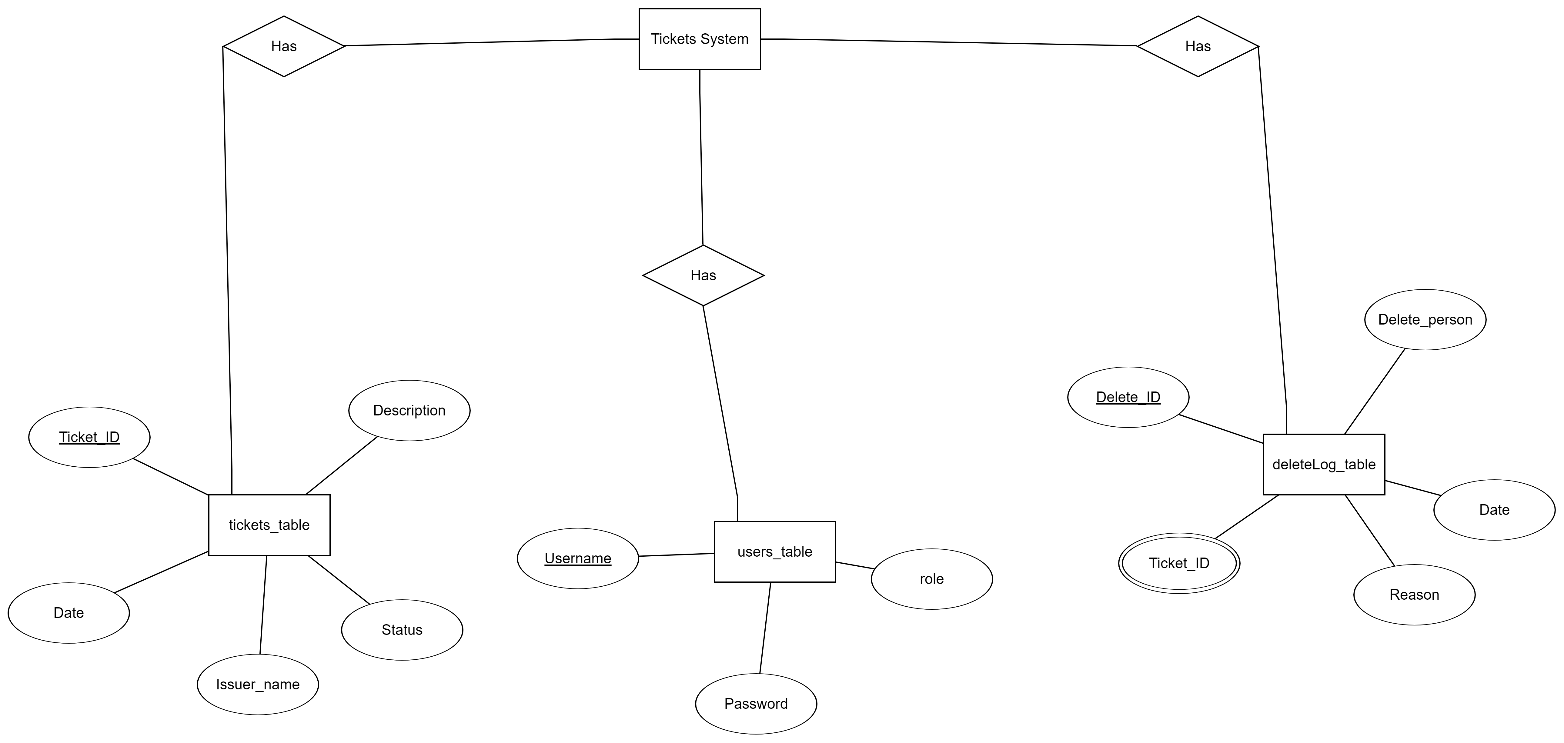


Mnguye13\_deleteLog3 table



**EXTRA CREDIT**

1. GUI ( fluid navigation, buttons , menus, table view actions, event handling)
2. UML, E-R Diargram, use case



Use case:

A ticket system has 3 main tables for ticket, user, and log. The ticket table has ticket\_ID, description, name of issuer, timestamp and status. The user table has username, password and user role. The deleteLod table will keep track of deleted ticket which has delete\_ID, ticket\_ID, timestamp and reason for deleting.

\*\*\*UML Diagram located in project folder as **Mnguye13\_ticketsystem.ucls**

