**Trouble Ticketing System:**

**Arif Mukhthar Umarkayam Shakilabanu**

**CS602**

**Final Project – Trouble Ticketing System**

A trouble ticket is a mechanism used in an organization to track the detection, reporting, and resolution of some type of problem. Trouble ticketing systems originated in manufacturing as a paper-based reporting system; now most are Web-based and associated with customer relationship management ([CRM](http://searchcrm.techtarget.com/definition/CRM)) environments, such as [call center](http://searchcrm.techtarget.com/definition/call-center)s or [e-business](http://searchcio.techtarget.com/definition/e-business) Web sites, or with high-level technology environments.

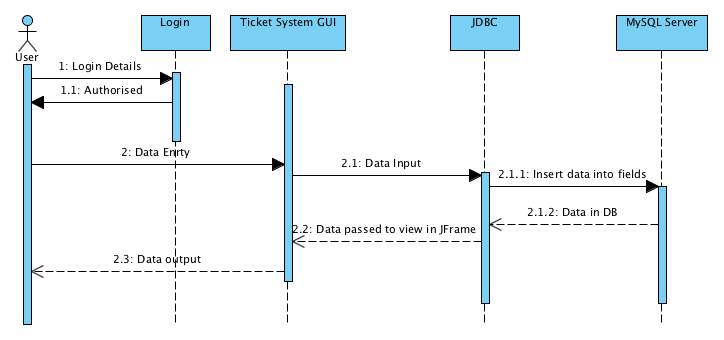
This project aims to implement a robust Trouble Ticketing System with all kinds of possible and anticipated errors handled using Java’s JFrame as the front end and MySQL’s database as the backend. The front end and the backend are connected by the JDBC driver.

**Overview:**

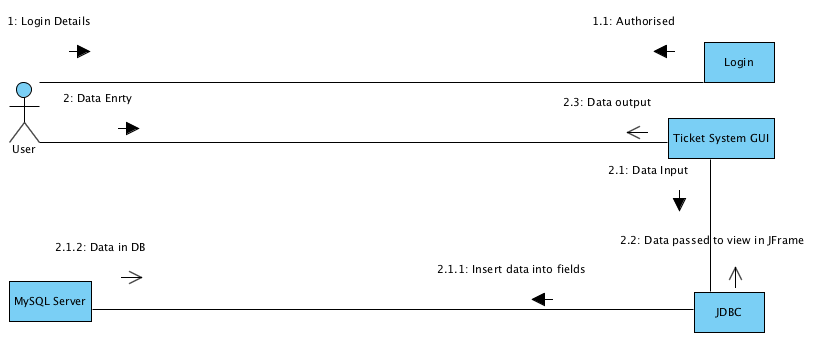
**Features Implemented:**

* **Tabbed GUI front end** using JFrame.
* **MySQL’s Database** as the backend to store user’s data.
* All the possible and anticipated **errors handled** and pop dialog boxes are presented to the user incase an error occurs.
* Implemented **all database manipulation functions** such as Create, View, Update, Search and Delete.
* The **view function** makes use of **JTable** to show the database in JFrame GUI.
* Has a **login screen** for allowing authorized access only.
* Functionalities are **JUnit tested**.
* **Interface concept used**.
* The industry standard **JavaDocs** format is used in the code and JavaDoc is created.
* Completely **developed by adhering to the SDLC – Waterfall Model**.

**Sequence Diagram:**

****

**Communication Diagram:**

****

**CODE:**

**Main Class(login.java):**

**import** java.awt.Graphics;

**import** javax.swing.JOptionPane;

**import** javax.swing.JFrame;

**import** java.lang.String;

**public** **class** login {

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

**boolean** access = **false**;

// dialogue box which prompts the username and password of the login

// user

String message = "User name:" + "\n";

String name = JOptionPane.*showInputDialog*(message);

String password;

name = name.trim();

name = name.toUpperCase();

// if username and password is correct as the given string then the

// trouble\_ticket.java is run

**if** (name.equals("ARIF")) {

JOptionPane.*showMessageDialog*(**null**, "Hello " + name +"!");

message = "Password";

password = JOptionPane.*showInputDialog*(message);

password = password.trim();

password = password.toUpperCase();

**if** (password.equals("OSX")) {

access = **true**;

} **else**

JOptionPane.*showMessageDialog*(**null**, "incorrect password");

} **else**

// if username and password is incorrect then the window is closed

{

JOptionPane.*showMessageDialog*(**null**, "incorrect login name");

System.*exit*(1);

}

**if** (access == **true**) {

**try** {

// this sets the size of the dialogue box that is displayed..

JFrame m = **new** trouble\_ticket();

m.setSize(500, 350);

m.setVisible(**true**);

}

**catch** (Exception e) {

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

}

}

}// end main

}// end class

**Front End – JFrame (trouble\_ticket.java):**

**import** java.awt.\*;

**import** javax.swing.\*;

**import** java.awt.event.\*;

**import** java.util.\*;

**import** java.io.\*;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.text.SimpleDateFormat;

**import** javax.swing.table.\*;

**import** java.sql.\*;

**import** java.util.List;

**import** java.util.Vector;

/\*\*

\* Program : Ticket Tool.java

\* Author : Arif

\* Version : 1.0

\*/

**public** **class** trouble\_ticket **extends** JFrame **implements** ActionListener{

**private** JTabbedPane jtabbedPane;

**private** JPanel create;

**private** JPanel view;

**private** JPanel update;

**private** JPanel close;

**private** JPanel purge;

JTextField create\_emp\_id,create\_t\_desc,create\_sev, widthText, depthText, volumeText,

close\_ticket\_id,close\_t\_status,purge\_ticket\_id,update\_ticket\_id,update\_t\_desc,view\_ticket\_id;

**static** DB *objDB* = **new** DB();

**public** trouble\_ticket() **throws** Exception{

setTitle("Trouble Ticket Tool");

setSize(300, 200);

**try** {

*objDB*.createDataBase();

} **catch** (Exception e) {

e.printStackTrace();

}

JPanel topPanel = **new** JPanel();

topPanel.setLayout( **new** BorderLayout() );

getContentPane().add( topPanel );

createcreate();

createview();

createupdate();

createclose();

createpurge();

jtabbedPane = **new** JTabbedPane();

jtabbedPane.addTab("create", create);

jtabbedPane.addTab("view", view);

jtabbedPane.addTab("update", update);

jtabbedPane.addTab("close", close);

jtabbedPane.addTab("purge", purge);

topPanel.add(jtabbedPane, BorderLayout.***CENTER***);

}

**public** **void** createcreate(){

create = **new** JPanel();

create.setLayout( **null** );

JLabel employeeLabel = **new** JLabel("employee\_id");

employeeLabel.setBounds(10, 15, 150, 20);

create.add( employeeLabel );

create\_emp\_id = **new** JTextField();

create\_emp\_id.setBounds(260, 15, 150, 20);

create.add( create\_emp\_id );

JLabel employeeLabel1 = **new** JLabel("Ticket desc:");

employeeLabel1.setBounds(10, 60, 260, 20);

create.add( employeeLabel1 );

create\_t\_desc = **new** JTextField();

create\_t\_desc.setBounds(260, 60, 150, 20);

create.add( create\_t\_desc );

JLabel employeeLabel2 = **new** JLabel("severity:");

employeeLabel2.setBounds(10, 105, 370, 20);

create.add( employeeLabel2 );

create\_sev = **new** JTextField();

create\_sev.setBounds(260, 105, 150, 20);

create.add( create\_sev);

JButton Close = **new** JButton("Close");

Close.setBounds(20,250,80,20);

Close.addActionListener(**this**);

Close.setBackground(Color.***white***);

create.add(Close);

JButton createButton = **new** JButton("Create");

createButton.setBounds(350,250,80,20);

createButton.addActionListener(**this**);

createButton.setBackground(Color.***white***);

create.add(createButton);

}

/\* CREATE view \*/

**public** **void** createview() **throws** Exception{

view = **new** JPanel();

view.setLayout( **null** );

JLabel employeeLabel1 = **new** JLabel("Ticket id:");

employeeLabel1.setBounds(10, 15, 150, 20);

view.add( employeeLabel1 );

view\_ticket\_id = **new** JTextField();

view\_ticket\_id.setBounds(260, 15, 150, 20);

view.add( view\_ticket\_id );

JButton view\_ticket = **new** JButton("view ticket");

view\_ticket.setBounds(20,250,80,20);

view\_ticket.addActionListener(**this**);

view\_ticket.setBackground(Color.***white***);

view.add(view\_ticket);

JButton displaybutton = **new** JButton("display all tickets");

displaybutton.setBounds(350,250,120,20);

displaybutton.addActionListener(**this**);

displaybutton.setBackground(Color.***white***);

view.add(displaybutton);

}

**public** JTable FillTable(JTable table, ResultSet rs)

{

**try**

{

rs = *objDB*.display();

//To remove previously added rows

**while**(table.getRowCount() > 0)

{

((DefaultTableModel) table.getModel()).removeRow(0);

}

**int** columns = rs.getMetaData().getColumnCount();

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

{

Object[] row = **new** Object[columns];

**for** (**int** i = 1; i <= columns; i++)

{

row[i - 1] = rs.getObject(i);

}

((DefaultTableModel) table.getModel()).insertRow(rs.getRow()-1,row);

}

rs.close();

}

**catch**(Exception e)

{

e.printStackTrace();

}

**return** table;

}

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

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

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

ResultSetMetaData md = rs.getMetaData();

**int** columns = md.getColumnCount();

// Get column names

**for** (**int** i = 1; i <= columns; i++) {

columnNames.addElement(md.getColumnName(i));

}

// Get row data

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

Vector<Object> row = **new** Vector<Object>(columns);

**for** (**int** i = 1; i <= columns; i++) {

row.addElement(rs.getObject(i));

}

tableData.addElement(row);

}

DefaultTableModel ticketTable = **new** DefaultTableModel(tableData, columnNames);

**return** ticketTable;

}

//close

**public** **void** createclose(){

close = **new** JPanel();

close.setLayout( **null** );

JLabel employeeLabel1 = **new** JLabel("Ticket id:");

employeeLabel1.setBounds(10, 15, 260, 20);

close.add( employeeLabel1 );

close\_ticket\_id = **new** JTextField();

close\_ticket\_id.setBounds(260, 15, 150, 20);

close.add( close\_ticket\_id );

JLabel employeeLabel2 = **new** JLabel("Ticket status:");

employeeLabel2.setBounds(10, 60, 370, 20);

close.add( employeeLabel2 );

close\_t\_status = **new** JTextField();

close\_t\_status.setBounds(260, 60, 150, 20);

close.add( close\_t\_status );

JButton ok = **new** JButton("ok");

ok.setBounds(20,250,80,20);

ok.addActionListener(**this**);

ok.setBackground(Color.***white***);

close.add(ok);

}

//purge

**public** **void** createpurge() **throws** Exception{

purge = **new** JPanel();

purge.setLayout( **null** );

JLabel employeeLabel1 = **new** JLabel("Ticket id:");

employeeLabel1.setBounds(10, 15, 260, 20);

purge.add( employeeLabel1 );

purge\_ticket\_id = **new** JTextField();

purge\_ticket\_id.setBounds(260, 15, 150, 20);

purge.add( purge\_ticket\_id );

JButton delete = **new** JButton("delete");

delete.setBounds(20,250,80,20);

delete.addActionListener(**this**);

delete.setBackground(Color.***white***);

purge.add(delete);

Integer ticketId = 0;

**if**(purge\_ticket\_id.getText().equals(""))

ticketId = 999;

**else**

ticketId = Integer.*parseInt*(purge\_ticket\_id.getText());

System.***out***.println(ticketId);

*objDB*.deleteTicket(ticketId);

}

//update

**public** **void** createupdate(){

update = **new** JPanel();

update.setLayout( **null** );

JLabel employeeLabel1 = **new** JLabel("Ticket id:");

employeeLabel1.setBounds(10, 15, 260, 20);

update.add( employeeLabel1 );

update\_ticket\_id = **new** JTextField();

update\_ticket\_id.setBounds(260, 15, 150, 20);

update.add( update\_ticket\_id );

JLabel employeeLabel2 = **new** JLabel("Ticket desc:");

employeeLabel2.setBounds(10, 60, 370, 20);

update.add( employeeLabel2 );

update\_t\_desc = **new** JTextField();

update\_t\_desc.setBounds(260, 60, 150, 20);

update.add( update\_t\_desc );

JButton ok1 = **new** JButton("ok1");

ok1.setBounds(20,250,80,20);

ok1.addActionListener(**this**);

ok1.setBackground(Color.***white***);

update.add(ok1);

}

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

JButton button = (JButton)event.getSource();

String buttonLabel = button.getText();

**if**("Create".equalsIgnoreCase(buttonLabel)){

System.***out***.println("Create button clicked");

String empId = create\_emp\_id.getText().toString();

String ticketDescription = create\_t\_desc.getText().toString();

String severity = create\_sev.getText().toString();

ticketVO objTicketVO = **new** ticketVO();

**if**(empId.equals(""))

empId = "100";

objTicketVO.setEmpId(Integer.*parseInt*(empId));

objTicketVO.setTicketDescription(ticketDescription);

objTicketVO.setSeverity(severity);

**try** {

**int** ticketId = *objDB*.insert(objTicketVO);

JOptionPane.*showMessageDialog*(**null**, "inserted ticket id :"+ticketId);

System.***out***.println("inserted ticket id :"+ticketId);

} **catch** (Exception e) {

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

e.printStackTrace();

}

}

**if**("close".equalsIgnoreCase(buttonLabel))

{

System.*exit*(1);

}

**if**("delete".equalsIgnoreCase(buttonLabel)){

Integer ticketId = Integer.*parseInt*(purge\_ticket\_id.getText().toString());

System.***out***.println("Delete button clicked");

Object message = "Are you sure you want to delete this ticket: " + ticketId;

**int** reply = JOptionPane.*showConfirmDialog*(**null**, message, getTitle(), JOptionPane.***YES\_NO\_OPTION***);

**if** (reply == JOptionPane.***YES\_OPTION***) {

JOptionPane.*showMessageDialog*(**null**, "ticket deleted");

}

**else** {

JOptionPane.*showMessageDialog*(**null**, "cancelled");

System.*exit*(0);

}

**try** {

*objDB*.deleteTicket(ticketId);

JOptionPane.*showMessageDialog*(**null**, "deleted ticket no :"+ticketId);

System.***out***.println("deleted ticket id :"+ticketId);

} **catch** (Exception e) {

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

e.printStackTrace();

}

}

**if**("view ticket".equalsIgnoreCase(buttonLabel))

{

ResultSet rs = **null**;

Integer ticketId = Integer.*parseInt*(view\_ticket\_id.getText().toString());

**try** {

rs = *objDB*.queryForTicketId(ticketId);

JTable table = **new** JTable(*buildTable*(rs));

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

JFrame frame = **new** JFrame();

frame.setSize(750, 100);

frame.add(sp);

frame.setVisible(**true**);

} **catch** (Exception e) {

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

e.printStackTrace();

}

}

**if**("display all tickets".equalsIgnoreCase(buttonLabel))

{

ResultSet rs = **null**;

**try** {

rs = *objDB*.display();

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

JTable table = **new** JTable(*buildTable*(rs));

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

JFrame frame = **new** JFrame();

frame.setSize(750, 500);

frame.add(sp);

frame.setVisible(**true**);

} **catch** (Exception e) {

e.printStackTrace();

}

}

**if** ("ok".equalsIgnoreCase(buttonLabel)){

System.***out***.println("Status Update button clicked");

Integer ticketId = Integer.*parseInt*(close\_ticket\_id.getText());

String status = close\_t\_status.getText();

**try** {

*objDB*.updateStatus(ticketId, status);

JOptionPane.*showMessageDialog*(**null**, "closed Ticket ID :"+ticketId);

System.***out***.println("Closed ticket is :"+ticketId);

} **catch** (SQLException e) {

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

e.printStackTrace();

}

}

**if** ("ok1".equalsIgnoreCase(buttonLabel)){

System.***out***.println("Description Update button clicked");

Integer ticketId = Integer.*parseInt*(update\_ticket\_id.getText());

String description = update\_t\_desc.getText();

System.***out***.println(ticketId+" "+description);

**try** {

*objDB*.updateTicketDescription(ticketId, description);

JOptionPane.*showMessageDialog*(**null**, "Updated ticket id :"+ticketId);

} **catch** (SQLException e) {

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

e.printStackTrace();

}

}

}

**private** **void** Exit\_pressed(){

}

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

JFrame frame = **new** trouble\_ticket();

frame.setSize(500, 350);

frame.setVisible(**true**);

}

}

**Back End – MySQL DB (db.java):**

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.Date;

/\*\*

\* Program : JDBC.java

\* Author : Arif

\* Version : 1.0

\*/

**public** **class** DB {

/\*\*

\* Initilaize connect, statement

\*/

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

**private** **static** Statement *statement* = **null**;

// This is used to create database and then create table in those database

**public** **void** createDataBase() **throws** Exception {

**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://localhost/","root","pass");

// creating a table

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

String sql;

String sql1 = "use db;";

*statement*.execute(sql1);

// sql statement for creating a table

sql = "CREATE TABLE if not exists ticketTable "

+ "(Emp\_id INTEGER not NULL, "

+ "Ticket\_id INTEGER not NULL AUTO\_INCREMENT,"

+ "Ticket\_Desc CHAR(255),"

+ "Severity CHAR(55),"

+ "OpenDate DATETIME,"

+ "Status CHAR(55), "

+ " PRIMARY KEY ( ticket\_id ))";

*statement*.execute(sql);

// end create table

} **catch** (Exception e) {

e.printStackTrace();

}

}

**public** **void** updateTicketDescription(Integer ticketId,String ticketDescription) **throws** SQLException

{

**try**

{

String sql = "update ticketTable set Ticket\_Desc = '"+ticketDescription+"' where Ticket\_id = "+ticketId;

*statement*.execute(sql);

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

**public** **void** updateStatus(Integer ticketId,String status) **throws** SQLException

{

**try**

{

String sql = "update ticketTable set Status = '"+status+"' where Ticket\_id = "+ticketId;

*statement*.execute(sql);

}

**catch**(Exception e)

{

e.printStackTrace();

}

}

**public** **int** insert(ticketVO objTicketVO) **throws** Exception

{

**try** {

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

Integer empId = objTicketVO.getEmpId();

Integer ticketId = objTicketVO.getTicketId();

String ticketDescription = objTicketVO.getTicketDescription();

String severity = objTicketVO.getSeverity();

String openDate = objTicketVO.getOpenDate();

String status = objTicketVO.getStatus();

Date date = **new** Date();

String sql1 = "insert into ticketTable (Emp\_id, Ticket\_Desc,Severity,OpenDate,Status) values ("+

empId+",'"+ticketDescription+"','"+severity+"',now(),'"+"Opened"+"');";

System.***out***.println(sql1);

*statement*.executeUpdate(sql1);

// Displays a message that a table is created

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

sql1 = "select last\_insert\_id()";

ResultSet rs = *statement*.executeQuery(sql1);

rs.next();

**return** rs.getInt(1);

// end create table

} **catch** (Exception e) {

e.printStackTrace();

}

**return** 0;

}

**public** **boolean** deleteTicket(Integer ticketId) **throws** Exception

{

**int** outputInteger=0;

**try**

{

String sql = "delete from ticketTable where Ticket\_id = "+ticketId+";";

outputInteger = *statement*.executeUpdate(sql);

}

**catch**(Exception e)

{

e.printStackTrace();

}

**if**(outputInteger!=0)

**return** **true**;

**else**

**return** **false**;

}

**public** ResultSet queryForTicketId(**int** id) **throws** Exception {

ResultSet rs =**null**;

**try** {

String sql = "select \* from ticketTable where Ticket\_id = "+id+"";

rs = *statement*.executeQuery(sql);

}

**catch** (Exception e) {

e.printStackTrace();

}

**return** rs;

}

**public** ResultSet display() **throws** Exception

{

ticketVO[] objArrTicketVO = **null**;

ResultSet rs = **null**;

**try** {

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

String sql = "select \* from ticketTable";

rs = *statement*.executeQuery("select \* from ticketTable");

rs.last();

**int** size = rs.getRow();

System.***out***.println("size :"+size);

objArrTicketVO = **new** ticketVO[size];

**for**(**int** i=0;i<size;i++)

objArrTicketVO[i] = **new** ticketVO();

rs.beforeFirst();

**int** i=0;

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

{

objArrTicketVO[i].setEmpId(rs.getInt(1));

objArrTicketVO[i].setTicketId(rs.getInt(2));

objArrTicketVO[i].setTicketDescription(rs.getString(3));

objArrTicketVO[i].setSeverity(rs.getString(4));

objArrTicketVO[i].setOpenDate(rs.getDate(5).toString());

// objArrTicketVO[i].setCloseDate(rs.getDate(6).toString());

objArrTicketVO[i].setStatus(rs.getString(6));

System.***out***.println(objArrTicketVO[i]);

i++;

}

rs.beforeFirst();

}

**catch** (Exception e) {

e.printStackTrace();

}

**return** rs;

}

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

DB con = **new** DB();

con.createDataBase();

ticketVO obj = **new** ticketVO();

obj.setEmpId(100);

con.insert(obj);

con.display();

}

}

**Interface: (TicketVO.java)**

**import** java.util.Date;

/\*\*

\* Program : ticketVO.java

\* Author : Arif

\* Version : 1.0

\*/

**public** **class** ticketVO {

Integer empId;

Integer ticketId;

String ticketDescription;

String severity;

String openDate;

String status;

**public** String toString()

{

**return** empId+" "+ticketId+" "+ticketDescription+" "+severity+" "+openDate+" "+status;

}

**public** Integer getEmpId() {

**return** empId;

}

**public** **void** setEmpId(Integer empId) {

**this**.empId = empId;

}

**public** Integer getTicketId() {

**return** ticketId;

}

**public** **void** setTicketId(Integer ticketId) {

**this**.ticketId = ticketId;

}

**public** String getTicketDescription() {

**return** ticketDescription;

}

**public** **void** setTicketDescription(String ticketDescription) {

**this**.ticketDescription = ticketDescription;

}

**public** String getSeverity() {

**return** severity;

}

**public** **void** setSeverity(String severity) {

**this**.severity = severity;

}

**public** String getOpenDate() {

**return** openDate;

}

**public** **void** setOpenDate(String openDate) {

**this**.openDate = openDate;

}

**public** String getStatus() {

**return** status;

}

**public** **void** setStatus(String status) {

**this**.status = status;

}

}

**JUnit test: (runTest.java)**

**import** **static** org.junit.Assert.*assertEquals*;

**import** org.junit.Test;

/\*\* This class is used to JUnit test the Circle class

\* **@author** Arif

\* **@version** v1.0

\*\*/

**public** **class** runTest {

@Test

**public** **void** testrun() {

ticketVO v = **new** ticketVO();

String expectedOutput = "null"+" "+"null"+" "+"null"+" "+"null"+" "+"null"+" "+"null";

String actualOutput=

v.toString();

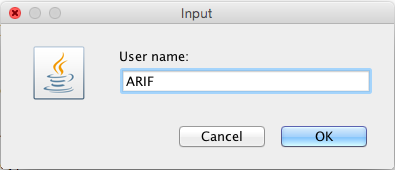
*assertEquals*(expectedOutput, actualOutput);

}

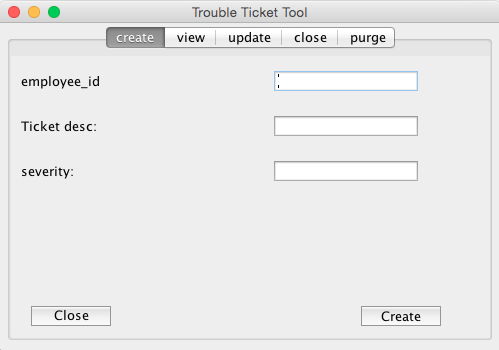
}

**Screen Shots:**

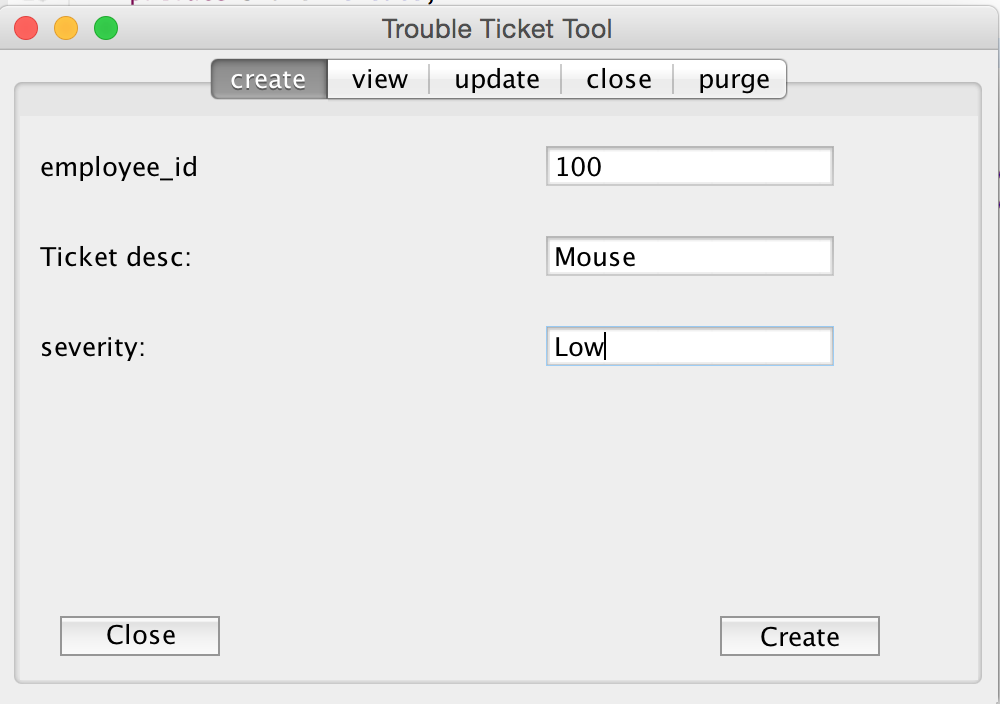
1. **Login Screen:**

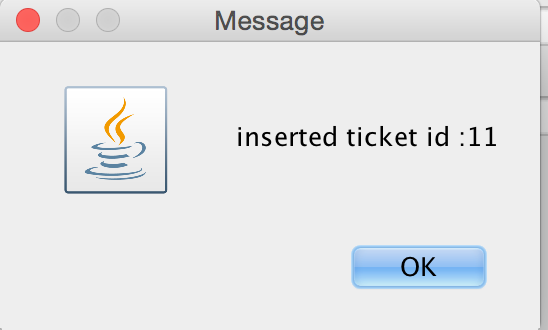
****

****

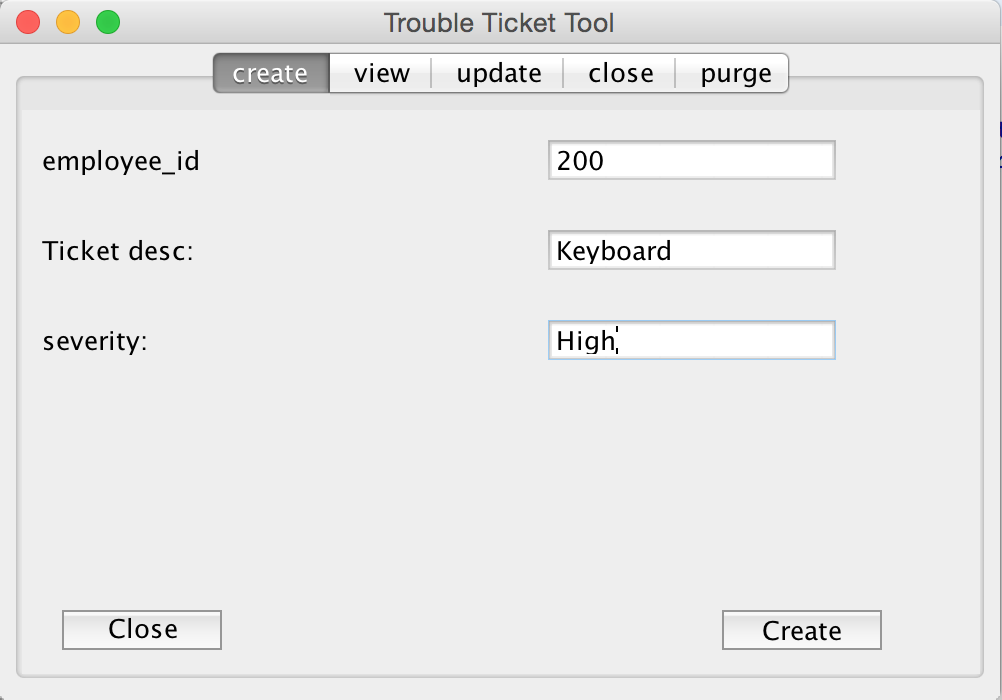
****

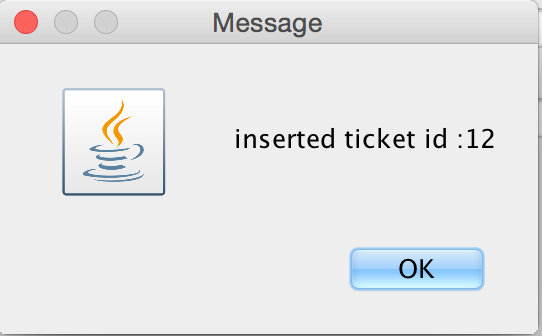
1. **Create Tickets:**

****

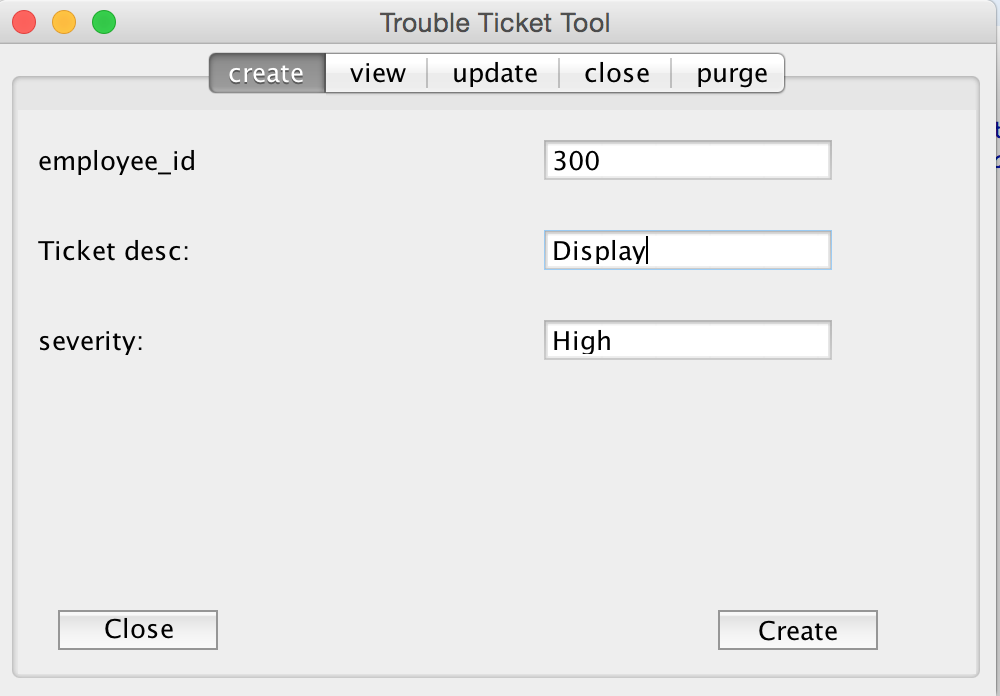
****

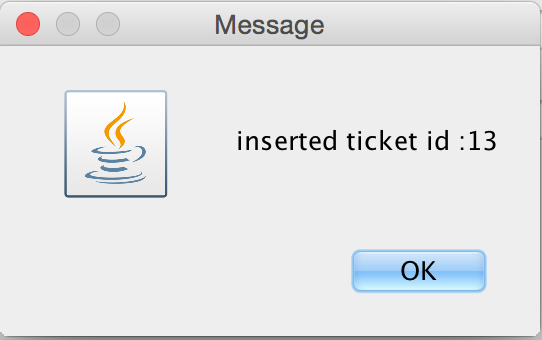
**Creating Ticket #11**

****

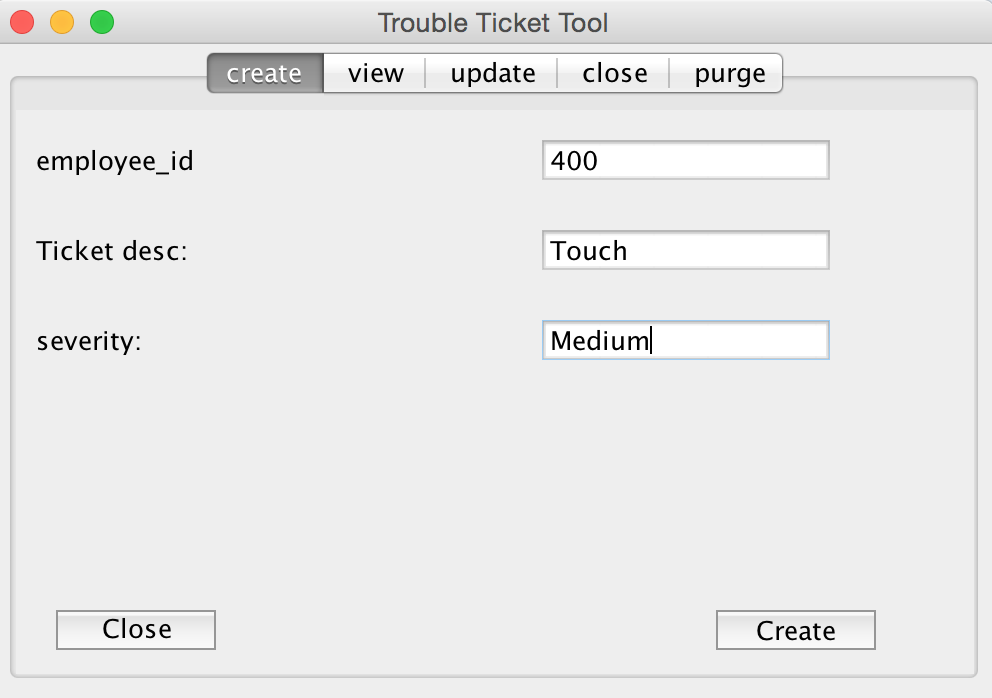
****

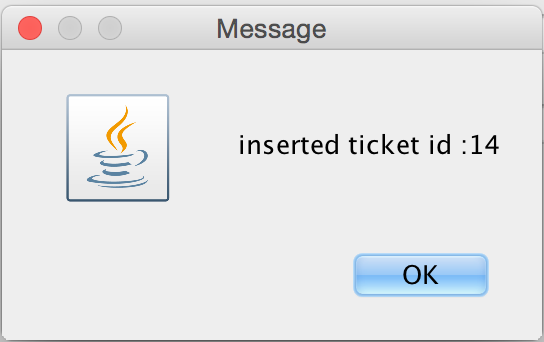
**Creating Ticket #12**

****

****

**Creating Ticket #13**

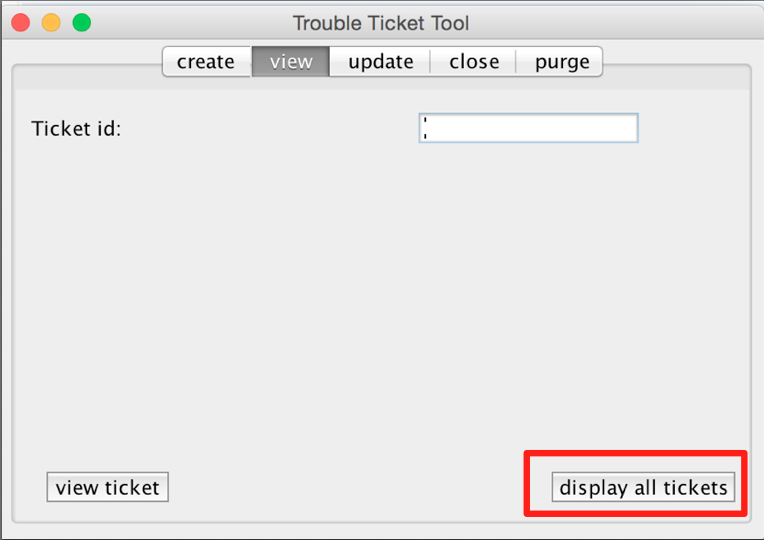
****

****

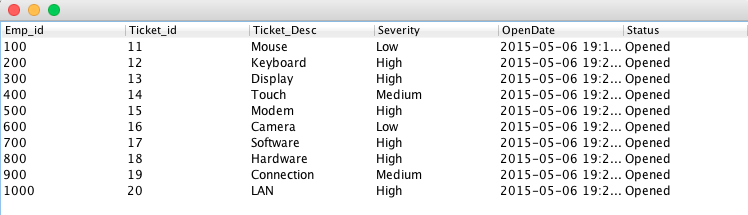
**Creating Ticket #14**

***Tickets #15 - #20 are added as shown above.***

1. **View Tickets Database:**

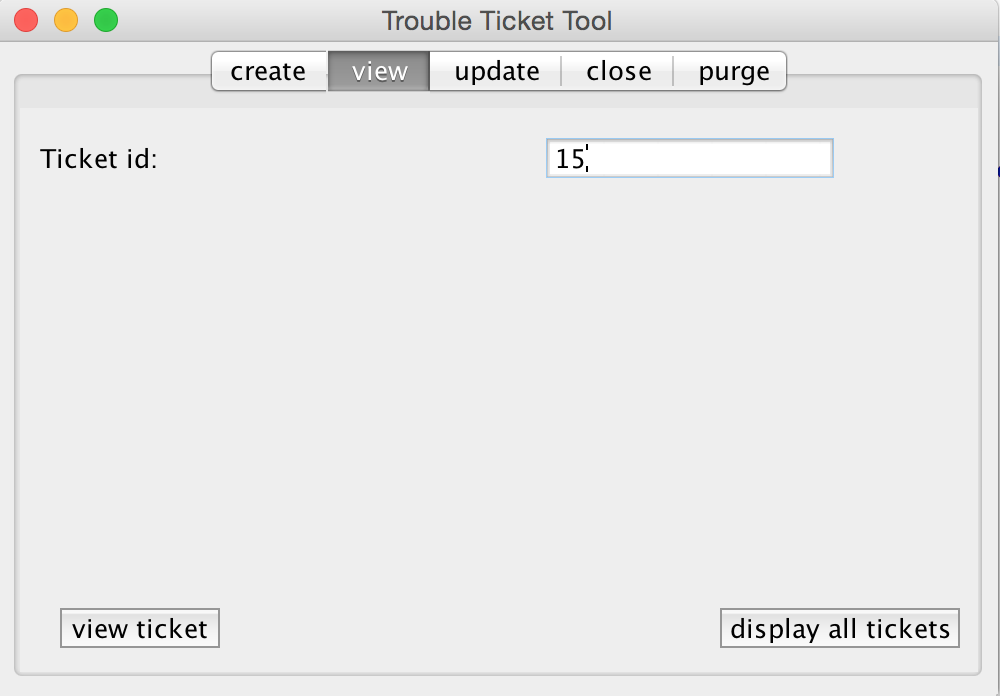
****

**3a. View all Tickets stored in the Database**

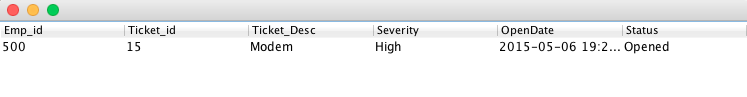
****

**3b. Database showing all created tickets up till Ticket #20.**

1. **Search:**

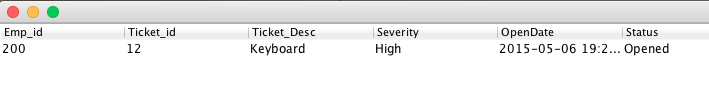
****

**4a. Search Ticket #15 in the Database.**

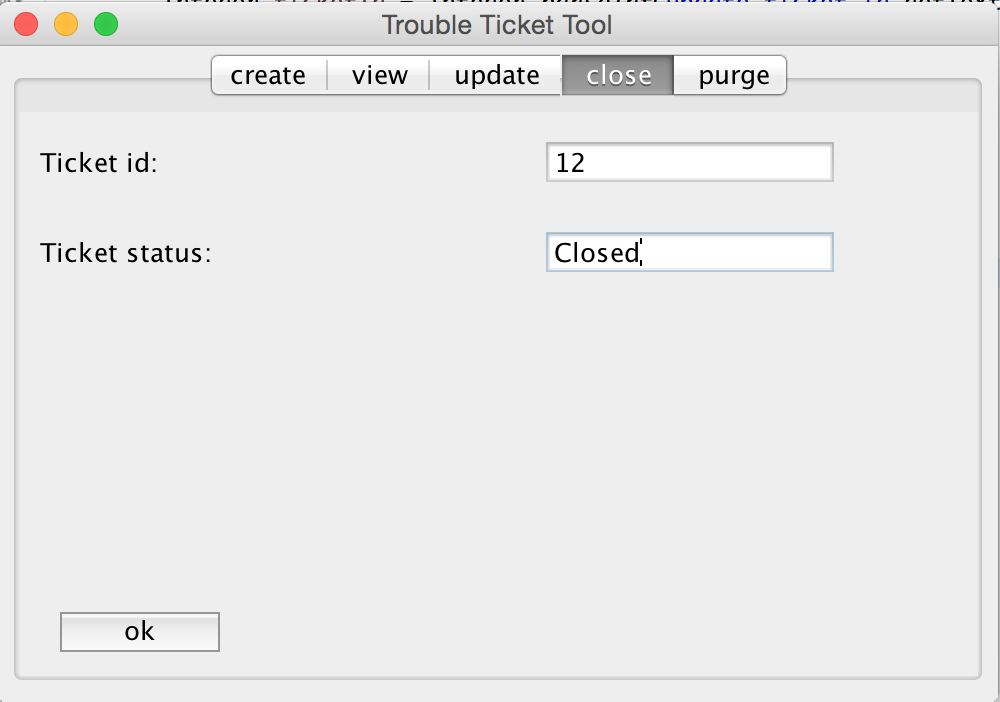
****

**4b. Ticket #15 searched and displayed.**

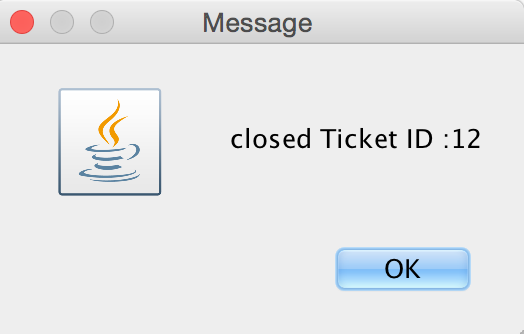
1. **Ticket Status Change:**

****

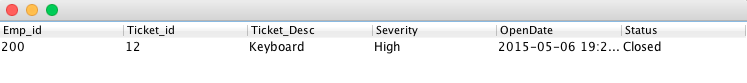
**5a. Ticket Status before Closing.**

****

**5b. Status changed to closed.**

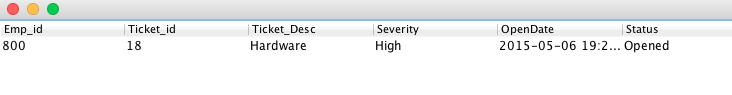
****

**5c. Ticket Closed Notification.**

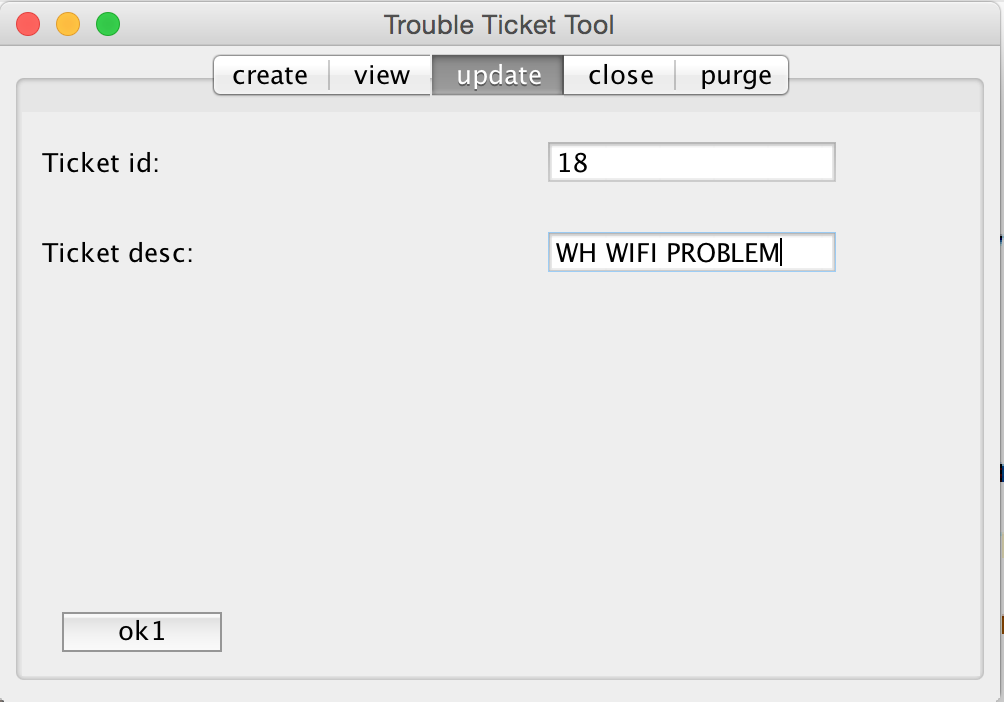
****

**5d. Ticket closed and updated in the Database.**

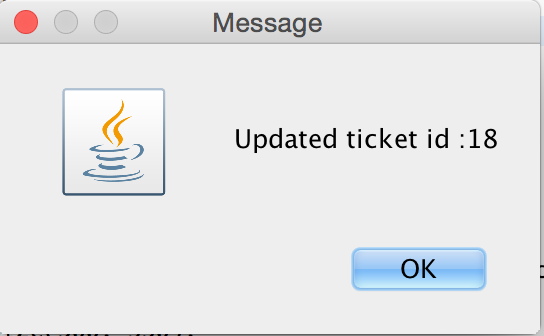
1. **Update Ticket Description:**

****

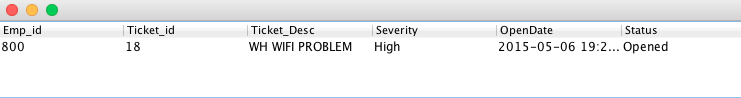
**6a. Ticket Description before Updation.**

****

**6b. Ticket description changed.**

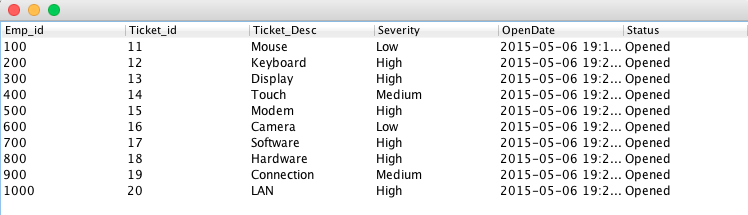
****

**6c. Confirmation for ticket updation.**

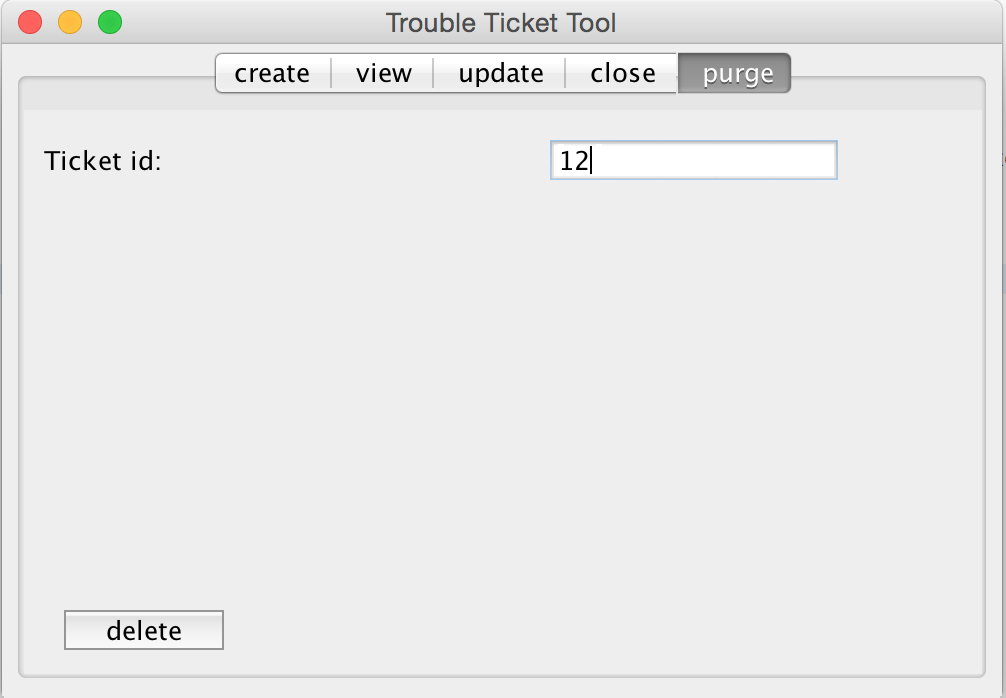
****

**6d. Ticket description updated and changed in Database.**

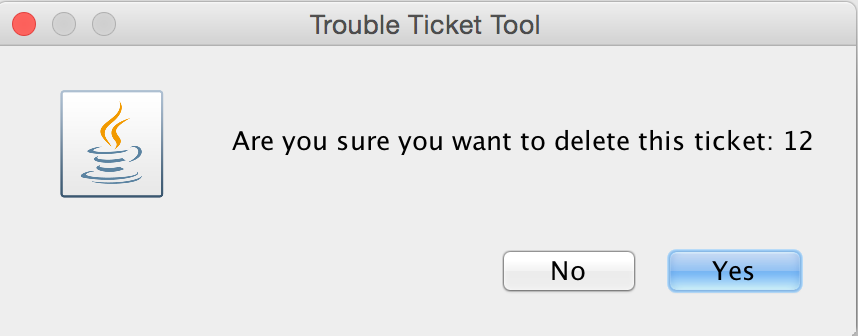
1. **Delete Tickets in Database:**

****

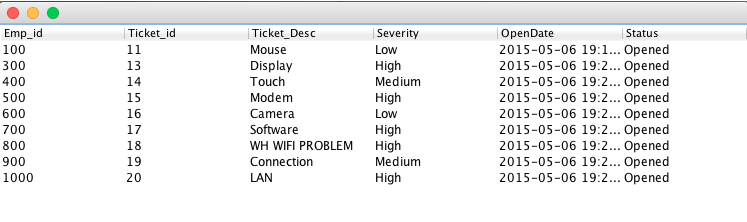
**7a. Ticket #12 in Database before deletion.**

****

**7b. Ticket #12 deleted.**

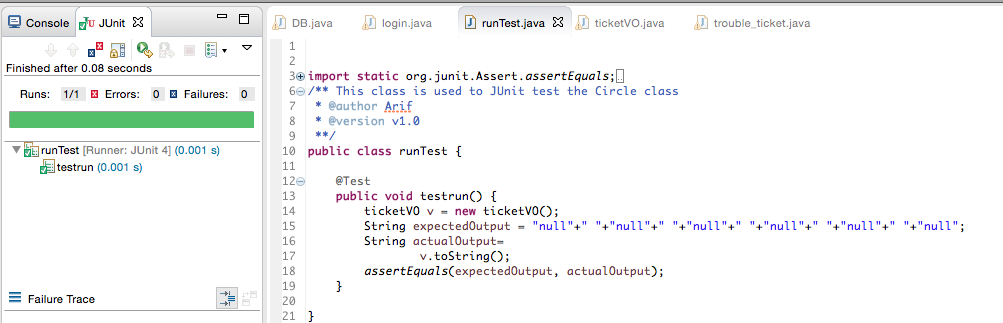
****

**7c. Deletion Confirmation.**

****

**7d. Ticket #12 deleted and Updated in the database.**

**8.JUnit Test:**

****