

View.java

```
package TODO;

import java.awt.BorderLayout;
import java.awt.EventQueue;

import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.DefaultListModel;
import javax.swing.JButton;
import javax.swing.JList;
import javax.swing.JTable;
import javax.swing.ListModel;
import javax.swing.SwingConstants;
import javax.swing.JLabel;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.net.http.WebSocket.Listener;
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

import com.mysql.*;
import com.mysql.jdbc.*;
```

```
import javax.swing.ListSelectionModel;

import javax.swing.table.DefaultTableModel;
```

```
public class View extends JFrame implements ActionListener{
```

```
    private JPanel contentPane;
```

```
    JButton addButton;
```

```
    JButton removeButton;
```

```
    JButton refreshButton;
```

```
    JButton editButton;
```

```
    DefaultListModel<String> listmodel;
```

```
    JList<String> list ;
```

```
    ViewModel viewmodel = new ViewModel();
```

```
    /**
```

```
     * Launch the application.
```

```
     * @throws ClassNotFoundException
```

```
     * @throws SQLException
```

```
     */
```

```
    public static void main(String[] args) throws ClassNotFoundException, SQLException {
```

```
        EventQueue.invokeLater(new Runnable() {
```

```
            public void run() {
```

```
                try {
```

```
                    View frame = new View();
```

```
                    frame.setVisible(true);
```

```
                } catch (Exception e) {
```

```
                    e.printStackTrace();
```

```
                }
```

```
            }
```

```
        });
```

```

}

/**
 * Create the frame.
 * @throws SQLException
 * @throws ClassNotFoundException
 * @throws NumberFormatException
 */
public View () throws NumberFormatException, ClassNotFoundException, SQLException {
    //components
    setTitle("TODO");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 776, 480);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    JPanel buttonsPanel = new JPanel();
    JPanel titlePanel = new JPanel();

    setContentPane(contentPane);
    // init

    addButton = new JButton("Add Task");
    buttonsPanel.add(addButton);
    addButton.addActionListener(this);

    removeButton = new JButton("Remove Task");
    buttonsPanel.add(removeButton);
    removeButton.addActionListener(this);

```

```
refreshButton = new JButton("Refresh Data");
buttonsPanel.add(refreshButton);
refreshButton.addActionListener(this);
```

```
editButton = new JButton("Edit Task");
buttonsPanel.add(editButton);
editButton.addActionListener(this);
```

```
contentPane.add(buttonsPanel,BorderLayout.SOUTH);
```

```
JLabel title = new JLabel("TODO");
titlePanel.add(title);
contentPane.add(titlePanel,BorderLayout.NORTH);
```

```
listmodel = new DefaultListModel<String>();
list = new JList<>(listmodel);
contentPane.add(list, BorderLayout.CENTER);
```

```
displayData();
```

```
}
```

```
public void displayData() throws NumberFormatException, ClassNotFoundException,
SQLException {
```

```
Thread thread = new Thread(new Runnable()
{
    public void run()
    {
        // this will be run in a separate thread
```

```

String[] results;

listmodel.clear();

String init = "ID Description Date";

listmodel.addElement(init);

String result="";

try {

    result = viewmodel.refreshView();

} catch (NumberFormatException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

} catch (ClassNotFoundException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

} catch (SQLException e) {

    // TODO Auto-generated catch block

    e.printStackTrace();

}

results = result.split("\n");

List<String> al = new ArrayList<String>();

al = Arrays.asList(results);

for(String s: al) {

    listmodel.addElement(s);

}

}

});

// start the thread

thread.start();

}

```

@Override

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

```
    if(e.getActionCommand().contentEquals("Add Task")) {  
        try {  
            try {  
                viewmodel.addView();  
                displayData();  
            } catch (ClassNotFoundException e1) {  
                // TODO Auto-generated catch block  
                e1.printStackTrace();  
            }  
        } catch (NumberFormatException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        } catch (SQLException e1) {  
            // TODO Auto-generated catch block  
            e1.printStackTrace();  
        }  
    }  
}
```

```
    if(e.getActionCommand().contentEquals("Remove Task")) {  
        try {  
            try {  
                viewmodel.deleteView();  
                displayData();  
            } catch (ClassNotFoundException e1) {  
                // TODO Auto-generated catch block  
                e1.printStackTrace();  
            }  
        }  
    }
```

```

    } catch (NumberFormatException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

```

if(e.getActionCommand().contentEquals("Refresh Data")) {
    try {
        try {
            displayData();
        } catch (ClassNotFoundException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    } catch (NumberFormatException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    } catch (SQLException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}

```

```

if(e.getActionCommand().contentEquals("Edit Task")) {
    try {
        try {
            viewmodel.updateView();

```

```
        displayData();
    } catch (ClassNotFoundException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
} catch (NumberFormatException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
} catch (SQLException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
}

}

}
```


ViewModel.java

```
package TODO;

import java.sql.SQLException;

import javax.swing.JFrame;
import javax.swing.JOptionPane;

public class ViewModel {

    Model model = new Model();

    public void addView() throws NumberFormatException, SQLException,
    ClassNotFoundException{

        JFrame frame = new JFrame("Input");
        String descAdd = JOptionPane.showInputDialog(frame, "Please insert description");
        String dateAdd = JOptionPane.showInputDialog(frame, "Please insert Date");

        model.addTask( descAdd, dateAdd);

    }

    public void deleteView() throws NumberFormatException, SQLException,
    ClassNotFoundException{

        JFrame frame = new JFrame("Input");
        String id = JOptionPane.showInputDialog(frame, "Please insert id");
```

```
model.deleteTask(Integer.parseInt(id));
```

```
}
```

```
public String refreshView() throws NumberFormatException, SQLException,  
ClassNotFoundException{
```

```
return model.readFromDB();
```

```
}
```

```
public void updateView() throws NumberFormatException, SQLException,  
ClassNotFoundException{
```

```
JFrame frame = new JFrame("Input");
```

```
String id = JOptionPane.showInputDialog(frame, "Please insert task ID");
```

```
String descNew = JOptionPane.showInputDialog(frame, "Please insert new  
Description");
```

```
String dateNew = JOptionPane.showInputDialog(frame, "Please insert new Date");
```

```
model.updateTask(Integer.parseInt(id), descNew, dateNew);
```

```
}
```

```
}
```


Model.java

```
package TODO;

import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.SQLException;

import org.apache.log4j.Logger;
import org.apache.log4j.PropertyConfigurator;

import com.mysql.jdbc.PreparedStatement;

public class Model {
    final static Logger logger = Logger.getLogger(Model.class);

    static private DBHandler dbHandler;
    static private Connection connection;
    static private PreparedStatement preparedStatement;

    public void addTask(String desc, String date) throws SQLException,
    ClassNotFoundException {

        dbHandler = new DBHandler();
        connection = dbHandler.getDbConnection();
        String insert = "INSERT INTO tasks (`desc`, `date`) VALUES (?,?)";
        preparedStatement = (PreparedStatement)
connection.prepareStatement(insert);
        preparedStatement.setString(1, desc);
        preparedStatement.setString(2, date);

        preparedStatement.executeUpdate();
        preparedStatement.close();
        PropertyConfigurator.configure("log4j.properties");
        logger.info("Task added successfully");

    }

    public void deleteTask(int id) throws SQLException, ClassNotFoundException
    {

        dbHandler = new DBHandler();
        connection = dbHandler.getDbConnection();
        String query = "DELETE FROM tasks where `id` = ?";

        try {
            PreparedStatement preparedStatement = (PreparedStatement)
connection.prepareStatement(query);

            preparedStatement.setInt(1, id);
            preparedStatement.execute();
            preparedStatement.close();

            PropertyConfigurator.configure("log4j.properties");
            logger.info("Task deleted successfully");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```

    }

    public String readFromDB() throws SQLException, ClassNotFoundException {
        dbHandler = new DBHandler();
        connection = dbHandler.getDbConnection();

        String query = "SELECT * from tasks";
        PreparedStatement preparedStatement = (PreparedStatement)
connection.prepareStatement(query);

        ResultSet resultSet = preparedStatement.executeQuery();
        /*
        * while (resultSet.next()) { System.out.println("Names: " +
        * resultSet.getInt("id") + " " + resultSet.getString("desc") + " "
+
        * resultSet.getInt("date")); }
        */
        String result = "";
        while (resultSet.next()) {
            result += resultSet.getInt("id") + " " +
resultSet.getString("desc") + " " + resultSet.getString("date")
            + System.LineSeparator();
        }
        PropertyConfigurator.configure("log4j.properties");
        Logger.info("read from database completed successfully");
        return result;
    }

    public void updateTask(int id, String newDesc, String newDate) throws
SQLException, ClassNotFoundException {
        dbHandler = new DBHandler();
        connection = dbHandler.getDbConnection();

        String query = "UPDATE tasks SET `desc` = ?, `date` = ? "
            + " where `id` = ? ";

        try {
            PreparedStatement preparedStatement = (PreparedStatement)
connection.prepareStatement(query);
            preparedStatement.setString(1, newDesc);
            preparedStatement.setString(2, newDate);
            preparedStatement.setInt(3, id);
            preparedStatement.executeUpdate();
            preparedStatement.close();
            PropertyConfigurator.configure("log4j.properties");
            Logger.info("Task updated successfully");
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}

```

Config.java

```
package TODO;

public class Config {
    protected String dbHost = "localhost";
    protected String dbPort = "3306";
    protected String dbUser = "root";
    protected String dbPass = "root";
    protected String dbName = "todo";
}
```

DBHandler.java

```
package TODO;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DBHandler extends Config {
    Connection dbConnection;
    public Connection getDbConnection() throws ClassNotFoundException,
SQLException {
        String connectionString = "jdbc:mysql://" + dbHost + ":" + dbPort +
"/" + dbName ;

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

        dbConnection =
DriverManager.getConnection(connectionString,dbUser,dbPass);

        return dbConnection;
    }
}
```