CTU training solutions CAREER SUCCESS STARTS AT CTU

Faculty of Information Technology									
I declare that I am familiar with, and will abide to the Examination rules of CTU [illegible] **Signature**	**SUBJECT NAME: Advanced Java** **SUBJECT CODE: JD522**								
	FORMATIVE ASSESSMENT **Duration**: 05 Apr – 24 Apr **Date** 24 April 2024 **Total Marks**: 100 **Total pages**: 19				**Examiner**: Junior Manganyi **Moderator:** Faith Muwishi				
	Student number								
	2	0	2	3	2	7	5	9	
	Surname: Poponi				**Initials**: M.P			/	%

Table of Contents

Project Question(s)

Question 1

Problem: Task Manager Application

You are tasked with creating a Java GUI-based Task Manager application that allows users to manage their tasks, categorize them, and store the information in a SQLite database. The application should provide features such as adding tasks, marking tasks as completed, and viewing tasks based on categories.

Unit 5: I/O and NIO

- Implement a GUI to list tasks from the SQLite database.
- Allow users to save tasks to a text file using OutputStream.
- Implement a mechanism to read tasks from the text file using InputStream.
- Display file properties like size and creation date using NIO.
- Provide an option to export task data to a CSV file using NIO.

Unit 6: Generics and Collections

- Design a task class that uses Generics to store task information.
- Use ArrayList to manage the list of tasks.
- Implement a filter mechanism to search for tasks based on user-defined criteria.
- Categorize tasks using HashMap to organize them based on user-defined categories.

Unit 7: Inner Classes

- Create an inner class to handle GUI components for task entry.
- Design an inner class to manage task categories and their corresponding actions.
- Implement a nested panel structure using inner panels for better organization.

Unit 8: JDBC

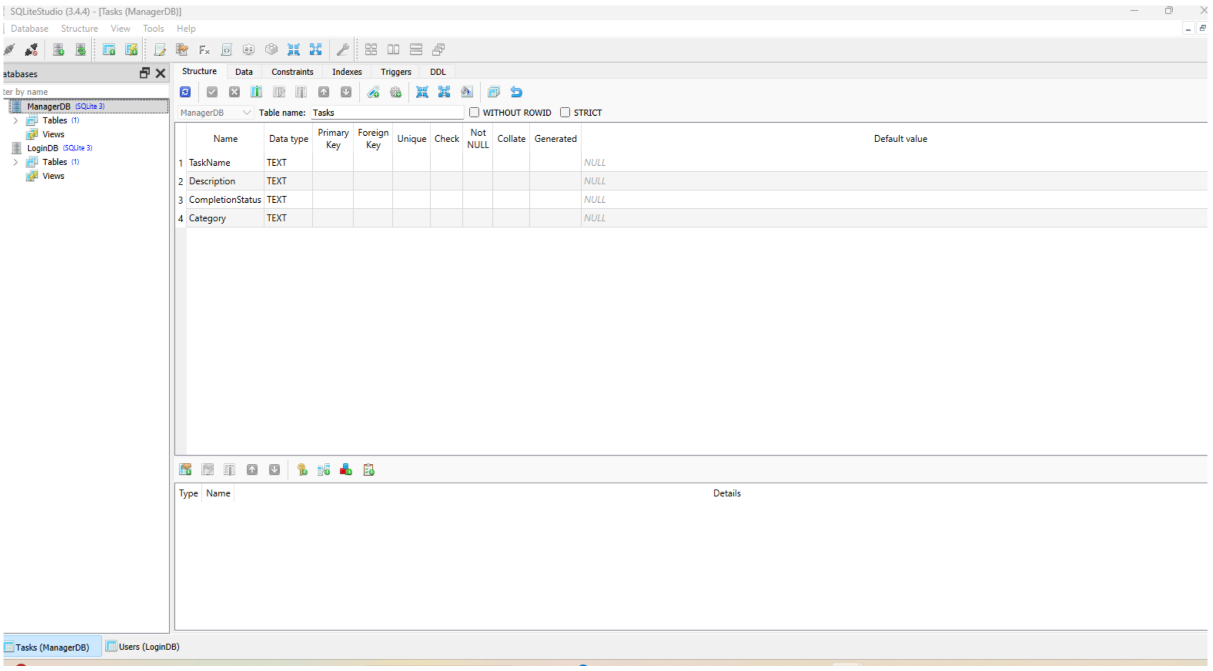
- Integrate a SQLite database with the application using JDBC.
- Design a database schema to store task information, including task name, description, completion status, and category.
- Implement functionalities to insert, update, and retrieve task data from the database.

- Display tasks in the GUI retrieved from the database.

Overall Design and Usability

- Design an intuitive and user-friendly GUI for the Task Manager application.
- Provide appropriate labels, buttons, and input fields for adding, viewing, and managing tasks.
- Implement error handling for input validation and database operations.
- Ensure a smooth and responsive user experience.

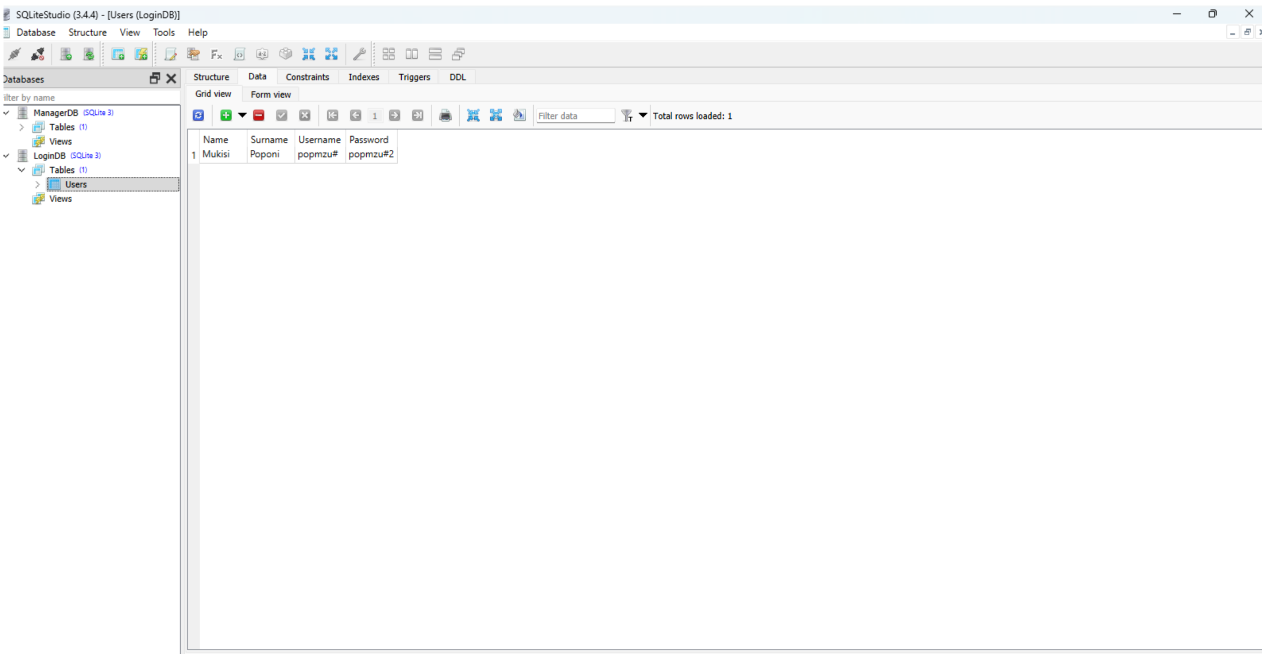
SQLite database


SQLiteStudio (3.4.4) - [Tasks (ManagerDB)]
Database Structure View Tools Help
atabases
ter by name
ManagerDB (SQLite 3)
Tables (1)
Views
LoginDB (SQLite 3)
Tables (1)
Views
Structure Data Constraints Indexes Triggers DDL
Tasks (ManagerDB)
Users (LoginDB)

ManagerDB Table name: Tasks ☐ WITHOUT ROWID ☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	TaskName	TEXT								NULL
2	Description	TEXT								NULL
3	CompletionStatus	TEXT								NULL
4	Category	TEXT								NULL

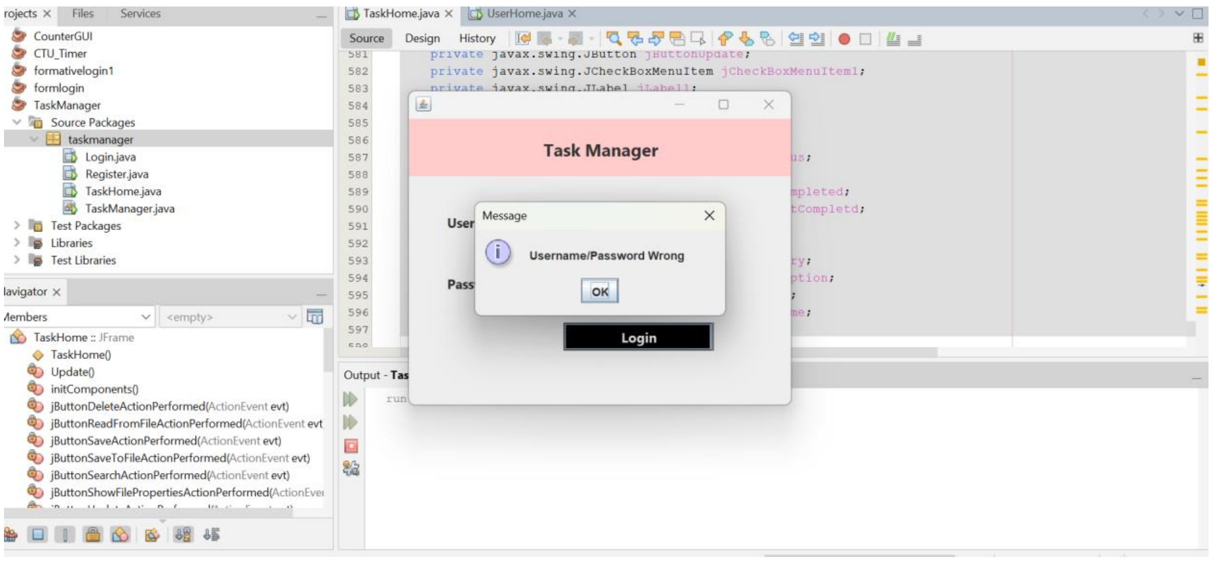
Type	Name	Details



SQLiteStudio (3.4.4) - [Users (LoginDB)]

	Name	Surname	Username	Password
1	Mukisi	Poponi	popmzu#	popmzu#2

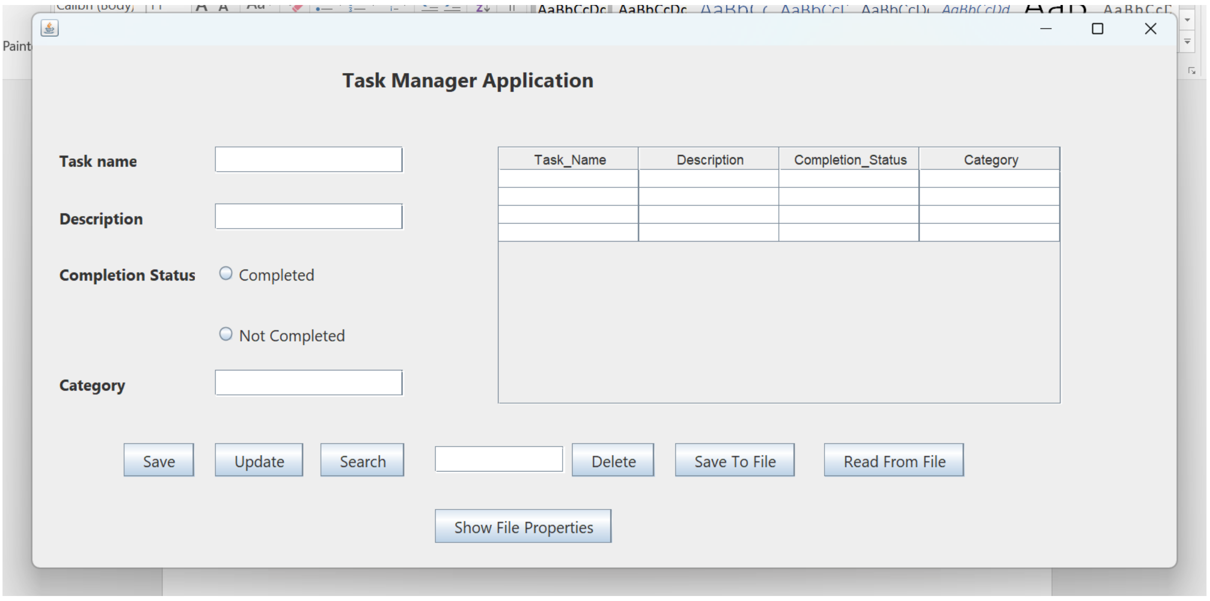
Total rows loaded: 1



```
    private javax.swing.JButton jButtonUpdate;
    private javax.swing.JCheckBoxMenuItem jCheckBoxMenuItem1;
    private javax.swing.JLabel jLabel1;
```

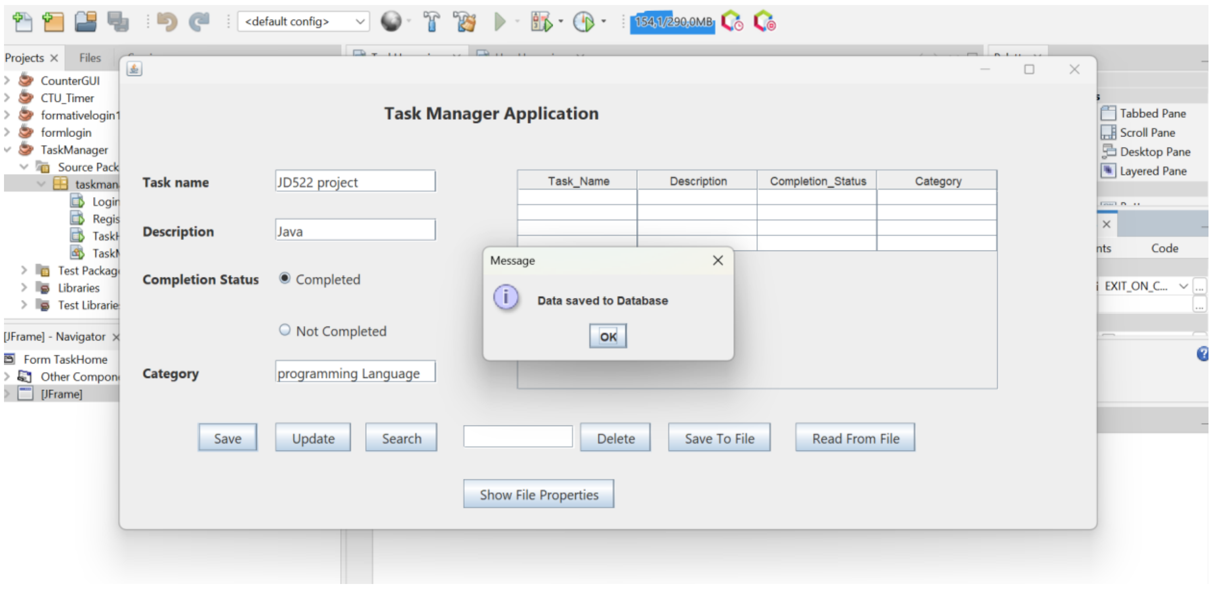
Message

Username/Password Wrong



Save Button (Method)

```
private void jButtonSaveActionPerformed(java.awt.event.ActionEvent evt) {

         try{

//            Connection conn = DriverManager.getConnection("jdbc:sqlite:C:\\Users\\popmz\\OneDrive\\Documents\\NetBeansProjects\\TaskManager\\ManagerDB");

        Connection conn = DriverManager.getConnection("jdbc:sqlite:C:\\Users\\popmz\\OneDrive\\Documents\\NetBeansProjects\\TaskManager\\ManagerDB");

        String taskname,description,category;
        taskname = jTextFieldTaskName.getText();
        description = jTextFieldDescription.getText();
        category = jTextFieldCategory.getText();


        String query ="INSERT INTO Tasks(TaskName,Description,CompletionStatus,Category)VALUES(?,?,?,?)";

        PreparedStatement ps = conn.prepareStatement(query);
        ps.setString(1, taskname);
        ps.setString(2, description);
        ps.setString(3, completionstatus);
        ps.setString(4, category);


        int rowsInserted =ps.executeUpdate();
        if(rowsInserted >0)
        {
            JOptionPane.showMessageDialog(rootPane,"Data saved to
Database");
        }else
        {
            JOptionPane.showMessageDialog(rootPane,"Data NOT saved");
        }

        }catch(SQLException e)
        {
        e.printStackTrace();
        }
    }
```


<default config>

Projects | Files
- CounterGUI
- CTU_Timer
- formativelogin1
- formlogin
- TaskManager
  - Source Pack
    - taskman
      - Login
      - Regis
      - TaskH
      - TaskM
  - Test Packag
  - Libraries
  - Test Librarie

JFrame] - Navigator
- Form TaskHome
- Other Compon
- [JFrame]

Task Manager Application

Task name	JD522 project
Description	Java
Completion Status	[✓ Completed] Not Completed
Category	programming Language

Task_Name	Description	Completion_Status	Category

Message: Data saved to Database [OK]

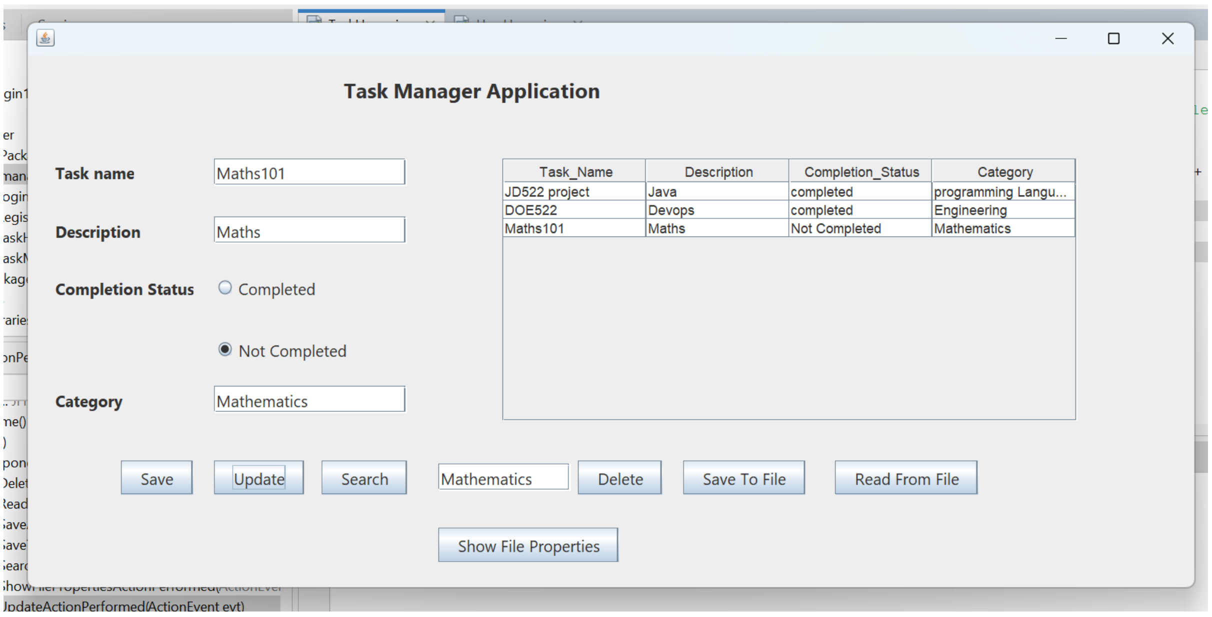
Update Button (method)

```
private void Update(){
    String query = "SELECT *FROM Tasks";
    try{
        ps=conn.prepareStatement(query);
        rs=ps.executeQuery();

        //Clear what is in table
        while(tmodel.getRowCount()>0){
        tmodel.removeRow(0);
        }

        //ADD INFO to table
        while(rs.next()){
```

```
        Object[]row ={
        rs.getString("TaskName"),
        rs.getString("Description"),
        rs.getString("CompletionStatus"),
        rs.getString("Category"),
        };
        tmodel.addRow(row);
        }
        rs.close();
        ps.close();

    }catch(Exception ex)
    {
    JOptionPane.showMessageDialog(rootPane, ex);
    }
```

Update Button (Action performed)

```
private void jButtonUpdateActionPerformed(java.awt.event.ActionEvent evt) {

        Update();
    }
```



Task Manager Application

Task name: Maths101

Description: Maths

Completion Status: Completed [✓ Not Completed]

Category: Mathematics

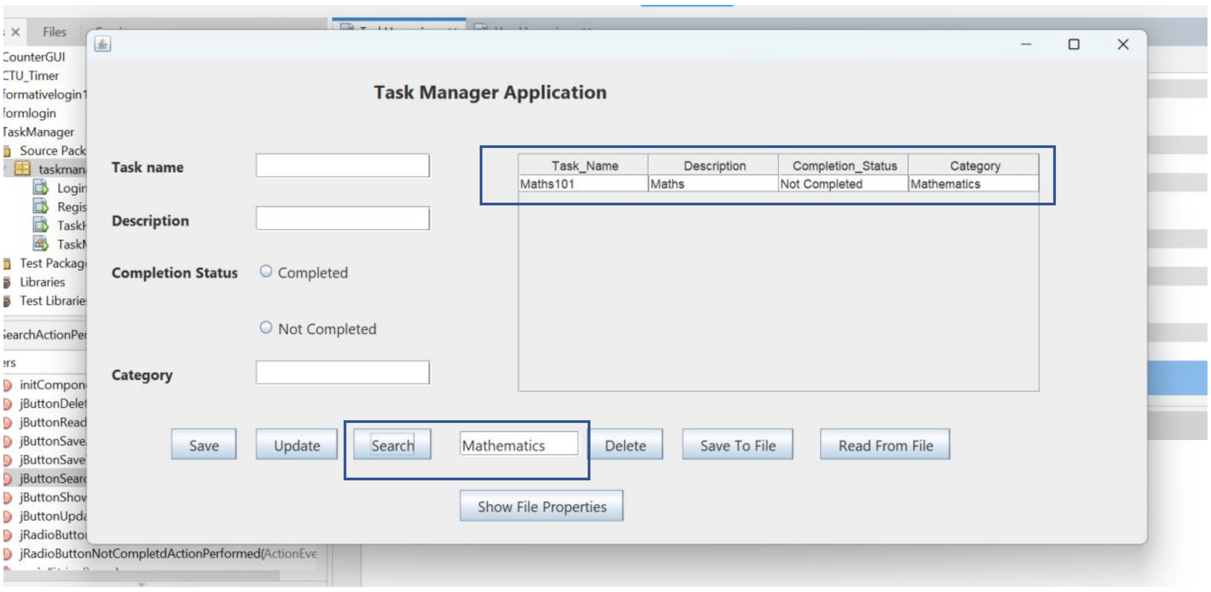
Task_Name	Description	Completion_Status	Category
JD522 project	Java	completed	programming Langu...
DOE522	Devops	completed	Engineering
Maths101	Maths	Not Completed	Mathematics

Search Button (Method)

```java
private void performSearch(String searchValue){
    String query1 = "SELECT * FROM Tasks WHERE Category LIKE ?";

        try {ps=conn.prepareStatement(query1);
        ps.setString(1,"%"+searchValue +"%");
        rs=ps.executeQuery();
        //Clear what is in table
        while(tmodel.getRowCount()>0){
        tmodel.removeRow(0);
        }

        //ADD INFO to table
            while (rs.next()) {
            Object[] row = {
                rs.getString("TaskName"),
                rs.getString("Description"),
                rs.getString("CompletionStatus"),
                rs.getString("Category"),
                };
                tmodel.addRow(row);
            }
            rs.close();
            ps.close();

        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, e);
        }//catch
    }
```

Search Button (Action performed)

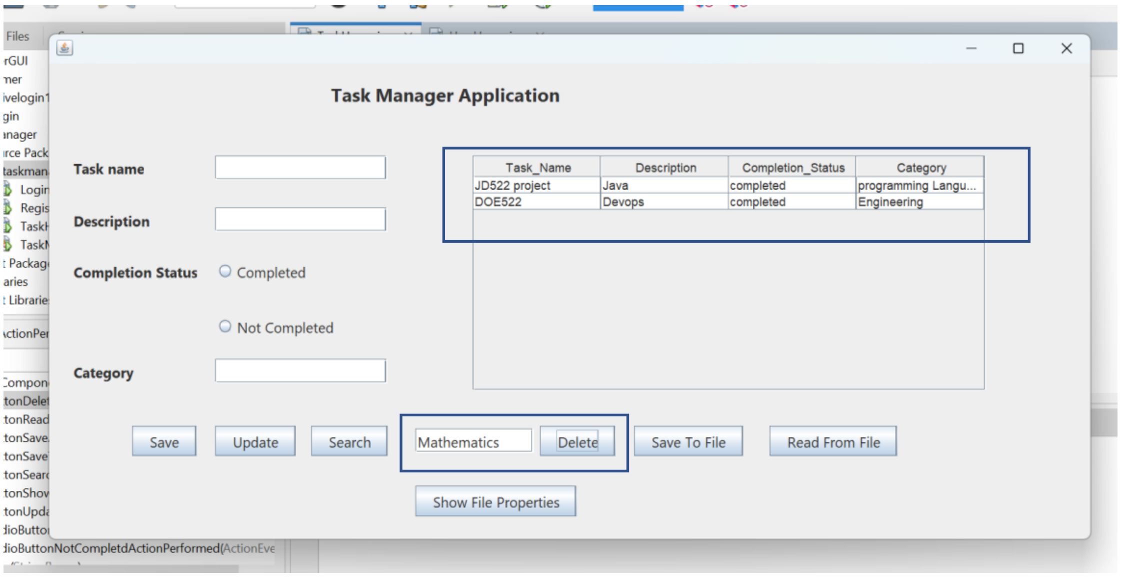
```java
private void jButtonSearchActionPerformed(java.awt.event.ActionEvent evt) {

        String searchValue = jTextFieldSearch.getText();
        performSearch(searchValue);
    }
```



Task_Name	Description	Completion_Status	Category
Maths101	Maths	Not Completed	Mathematics

Delete Button (method/action performed)

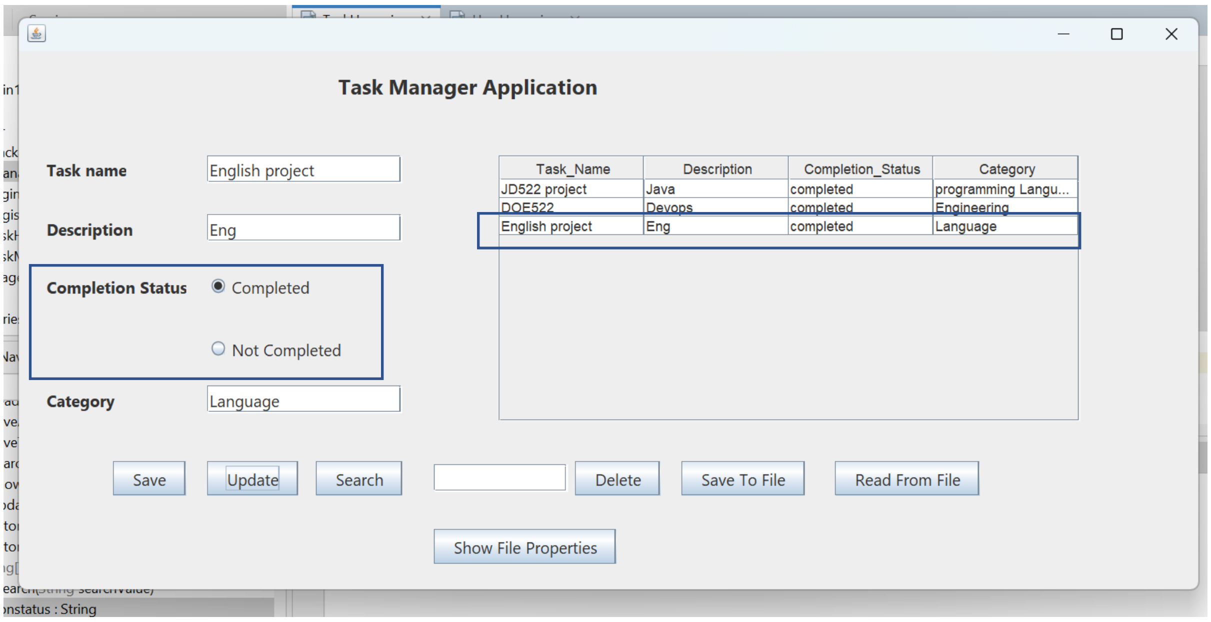
```
private void jButtonDeleteActionPerformed(java.awt.event.ActionEvent evt) {

           try {
            String query = "DELETE FROM Tasks WHERE Category=?";
            ps = conn.prepareStatement(query);
            ps.setString(1, jTextFieldSearch.getText());
            ps.execute();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, e);
        }
        Update();
```



Task_Name	Description	Completion_Status	Category
JD522 project	Java	completed	programming Langu...
DOE522	Devops	completed	Engineering

Completion Status Radio Buttons

```
private String completionstatus; //Object declation


   private void
jRadioButtonCompletedActionPerformed(java.awt.event.ActionEvent evt) {

        completionstatus="completed";
    }

    private void
jRadioButtonNotCompletdActionPerformed(java.awt.event.ActionEvent evt) {

        completionstatus="Not Completed";
    }
```



Task Manager Application

Task name: English project

Description: Eng

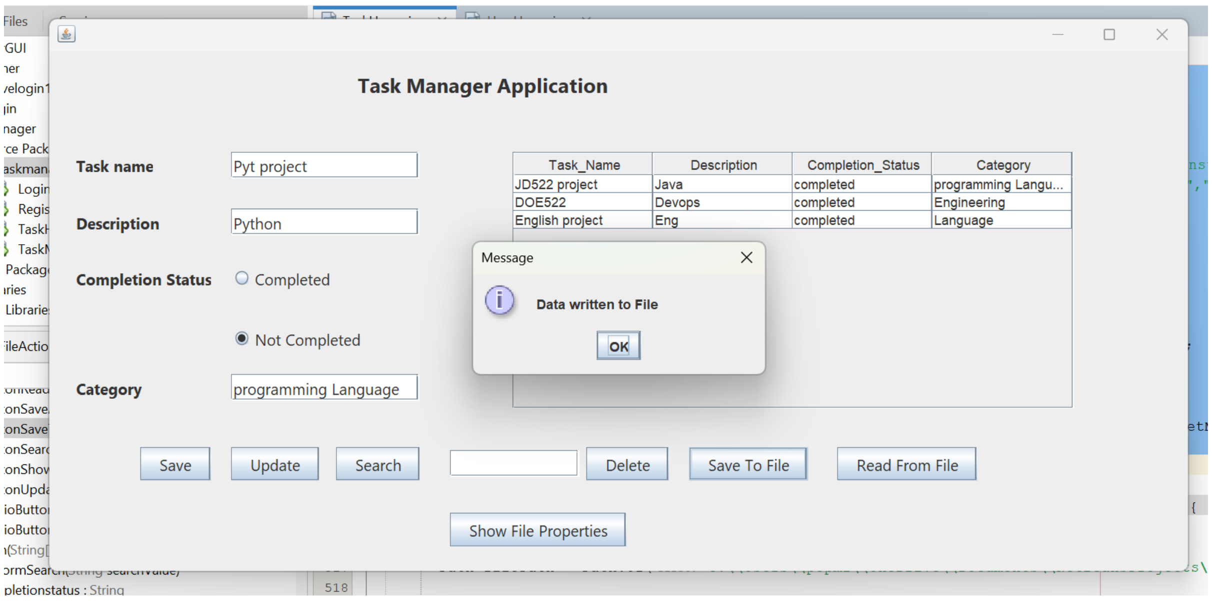
Completion Status: (● Completed) ○ Not Completed

Category: Language

Task_Name	Description	Completion_Status	Category
JD522 project	Java	completed	programming Langu...
DOE522	Devops	completed	Engineering
English project	Eng	completed	Language

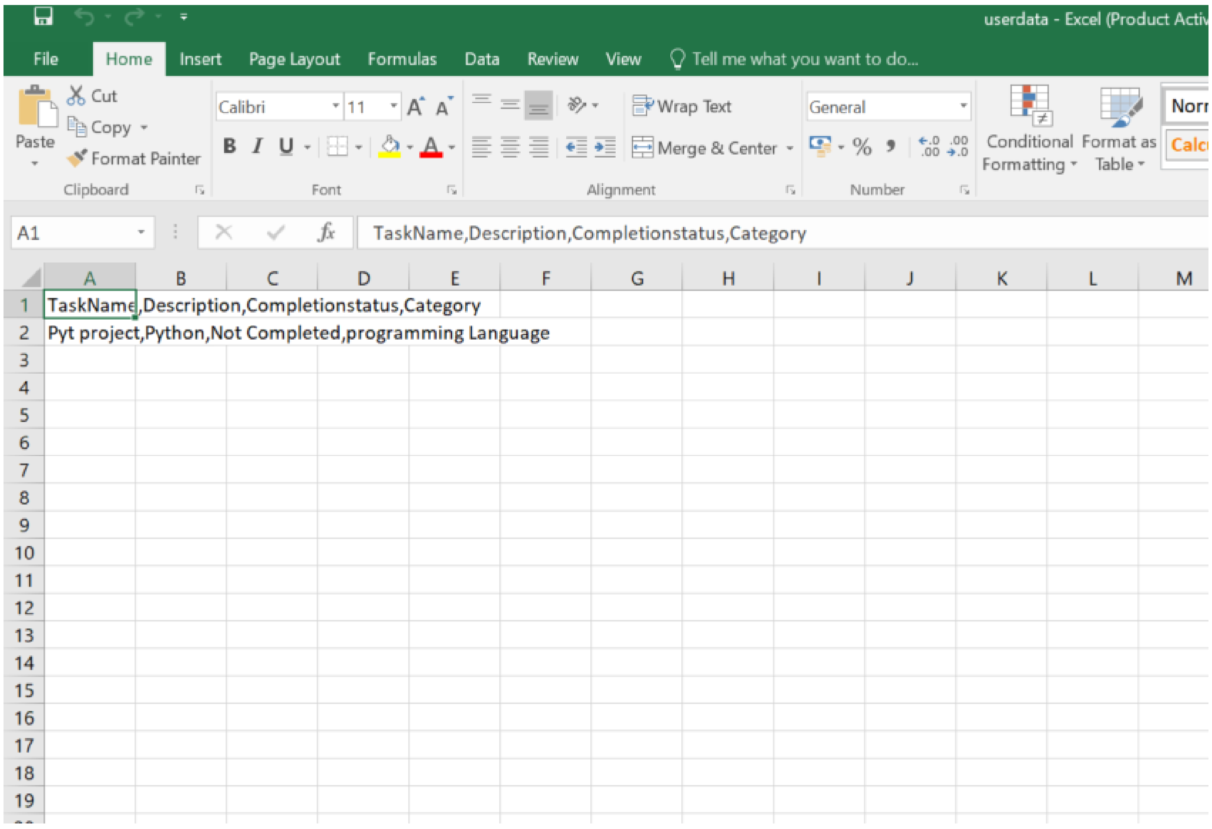
Save To File Button (Method/Action performed)

```
private void jButtonSaveToFileActionPerformed(java.awt.event.ActionEvent
evt) {

        try
       {//Get INFO to save to text file
        String taskname,description,category;
        taskname = jTextFieldTaskName.getText();
        description = jTextFieldDescription.getText();
        category = jTextFieldCategory.getText();

        //String to write to file
        StringBuilder dataToSave = new StringBuilder();

dataToSave.append("TaskName,").append("Description,").append("Completionsta
tus,").append("Category\n");

dataToSave.append(taskname).append(",").append(description).append(",").app
end(completionstatus).append(",").append(category).append("\n");

          FileOutputStream outputStream = new
FileOutputStream("userdata.csv");

        outputStream.write(dataToSave.toString().getBytes());

        outputStream.close();

        JOptionPane.showMessageDialog(rootPane,"Data written to File");

       }catch(Exception e)
       {
           JOptionPane.showMessageDialog(rootPane, "Error saving to
File"+e.getMessage());
       }
    }
```



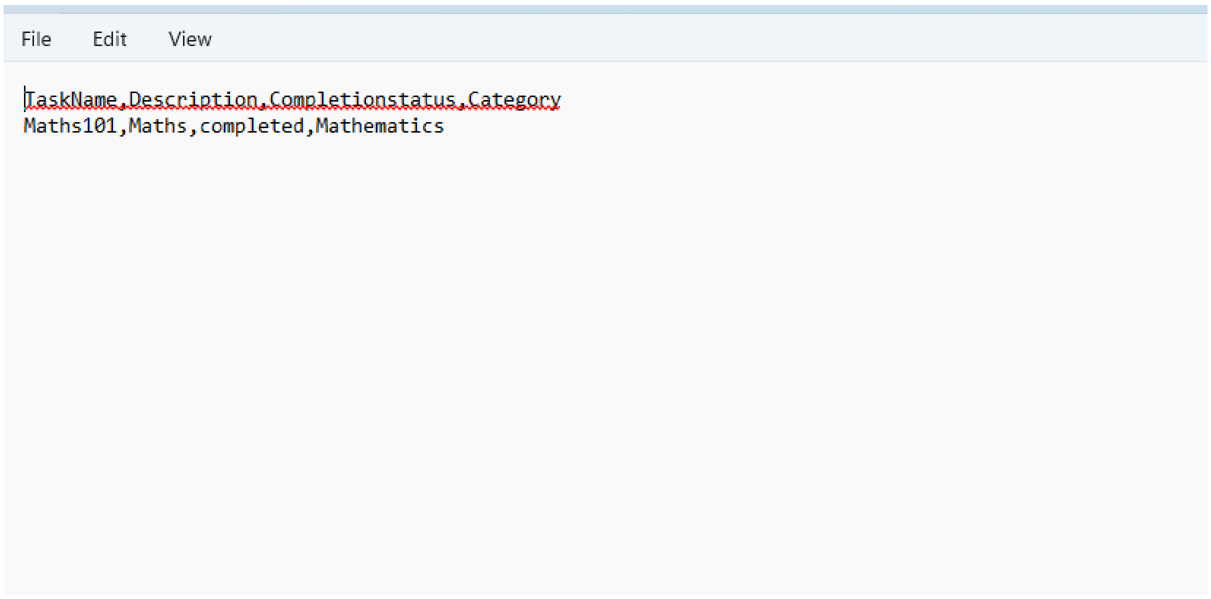
Task_Name	Description	Completion_Status	Category
JD522 project	Java	completed	programming Langu...
DOE522	Devops	completed	Engineering
English project	Eng	completed	Language

CSV file



```
TaskName,Description,Completionstatus,Category
Pyt project,Python,Not Completed,programming Language
```

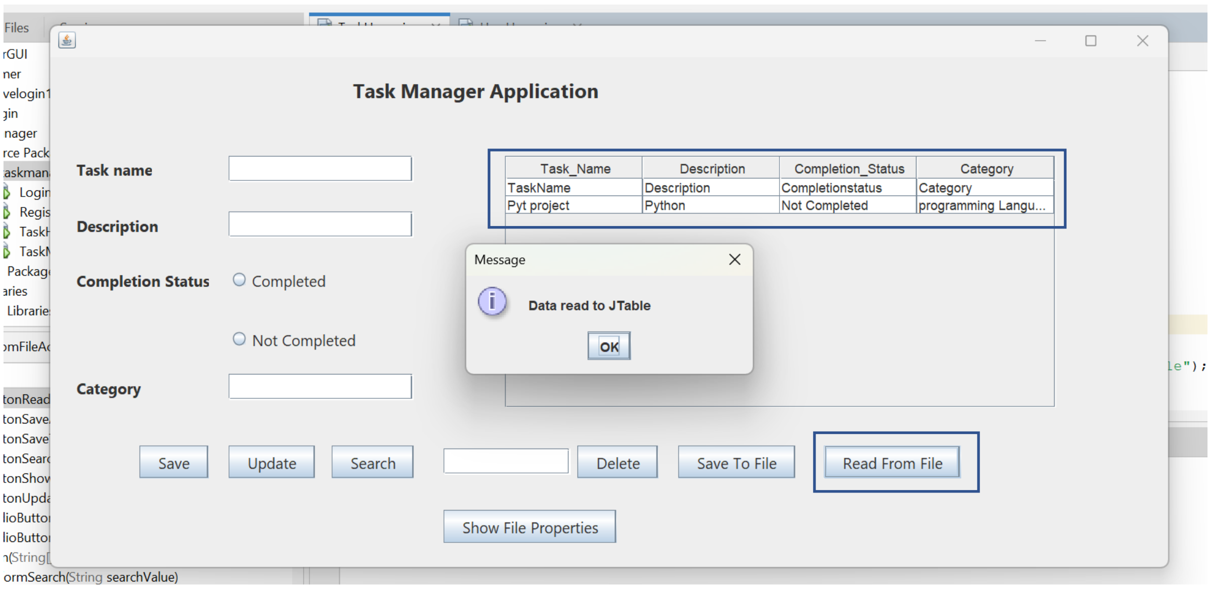
Txt file



```
TaskName,Description,Completionstatus,Category
Maths101,Maths,completed,Mathematics
```

Read From File Button (Method/Action performed)

```
private void jButtonReadFromFileActionPerformed(java.awt.event.ActionEvent
evt) {

         try {
            tmodel.setRowCount(0);//discard what is in the table

            String filename = "userdata.csv";

            FileReader reader = new FileReader(filename);

            BufferedReader breader = new BufferedReader(reader);

            String line;
            while ((line = breader.readLine()) != null) {
                String[] data = line.split(",");

                if (data.length >= 4) {
                    String value1 =data[0].trim();
                    String value2 =data[1].trim();
                    String value3 =data[2].trim();
                    String value4 =data[3].trim();


                    Object[] rowData = {value1,value2,value3,value4};
                    tmodel.addRow(rowData);
                }
            }
            breader.close();
            JOptionPane.showMessageDialog(rootPane, "Data read to JTable");

        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, "Fail to read from
File" + e.getMessage());
        }
    }
```



Task Manager Application

Task name

Description

Completion Status ○ Completed ○ Not Completed

Category

Task_Name	Description	Completion_Status	Category
TaskName	Description	Completionstatus	Category
Pyt project	Python	Not Completed	programming Langu...

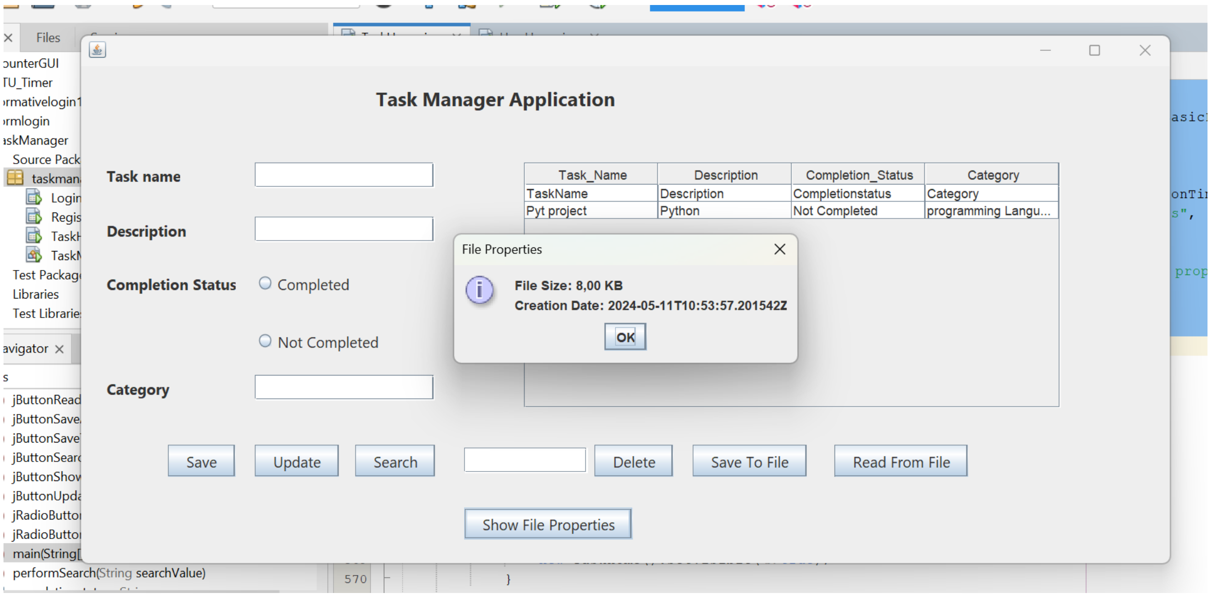
Message

Data read to JTable

Show File Properties Button

```
private void
jButtonShowFilePropertiesActionPerformed(java.awt.event.ActionEvent evt) {


      Path filePath =
Path.of("C:\\Users\\popmz\\OneDrive\\Documents\\NetBeansProjects\\TaskManag
er\\ManagerDB");


        try{
        //Get file size
        long fileSize = Files.size(filePath);
        String fileSizeString = String.format("%.2f KB", (double) fileSize
/ 1024);

        // Get file creation date
            BasicFileAttributes fileAttributes =
Files.readAttributes(filePath, BasicFileAttributes.class);
            FileTime creationTime = fileAttributes.creationTime();

            // Display file properties
            String message = "File Size: " + fileSizeString + "\nCreation
Date: " + creationTime;
            JOptionPane.showMessageDialog(this, message, "File Properties",
JOptionPane.INFORMATION_MESSAGE);
        }catch(IOException ex){
             ex.printStackTrace();
             JOptionPane.showMessageDialog(this, "Error accessing file
properties", "Error", JOptionPane.ERROR_MESSAGE);
        }
```

```
    }
```


Task Manager Application

Task name

Description

Completion Status ○ Completed ○ Not Completed

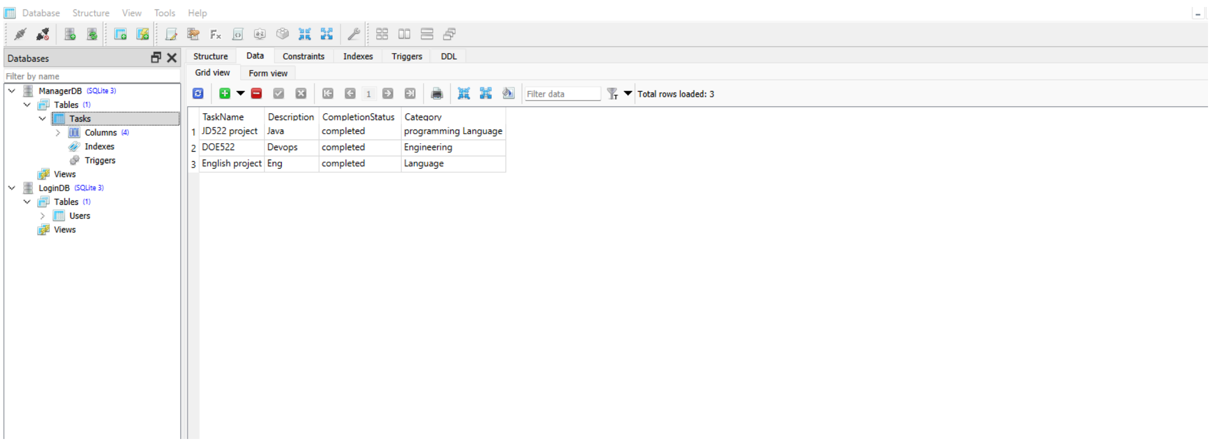
Category

Task_Name	Description	Completion_Status	Category
TaskName	Description	Completionstatus	Category
Pyt project	Python	Not Completed	programming Langu...

File Properties

File Size: 8,00 KB
Creation Date: 2024-05-11T10:53:57.201542Z

SQLite database (updated)



	TaskName	Description	CompletionStatus	Category
1	JD522 project	Java	completed	programming Language
2	DOE522	Devops	completed	Engineering
3	English project	Eng	completed	Language

TaskHome (Inner and outer class)

```java
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.attribute.BasicFileAttributes;
import java.nio.file.attribute.FileTime;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

/**
 *
 * @author popmz
 */
public class TaskHome extends javax.swing.JFrame {

    private Connection conn;
    private PreparedStatement ps;
    private ResultSet rs;
    private DefaultTableModel tmodel;

    /** Creates new form TaskHome */
    public TaskHome() {
        initComponents();

        try
        {
        conn =
DriverManager.getConnection("jdbc:sqlite:C:\\Users\\popmz\\OneDrive\\Docume
nts\\NetBeansProjects\\TaskManager\\ManagerDB");
        }
        catch(SQLException error)
        {
        error.printStackTrace();
        }
        tmodel =(DefaultTableModel)jTable1.getModel();

    }

    private void Update(){
    String query = "SELECT *FROM Tasks";
    try{
        ps=conn.prepareStatement(query);
        rs=ps.executeQuery();

        //Clear what is in table
        while(tmodel.getRowCount()>0){
        tmodel.removeRow(0);
        }

        //ADD INFO to table
        while(rs.next()){
        Object[]row ={
        rs.getString("TaskName"),
        rs.getString("Description"),
        rs.getString("CompletionStatus"),
        rs.getString("Category"),
        };
        tmodel.addRow(row);
        }
        rs.close();
        ps.close();

    }catch(Exception ex)
    {
    JOptionPane.showMessageDialog(rootPane, ex);
    }

    }

    // search method
    private void performSearch(String searchValue){
    String query1 = "SELECT * FROM Tasks WHERE Category LIKE ?";

        try {ps=conn.prepareStatement(query1);
        ps.setString(1,"%"+searchValue +"%");
        rs=ps.executeQuery();
        //Clear what is in table
        while(tmodel.getRowCount()>0){
        tmodel.removeRow(0);
        }

        //ADD INFO to table
            while (rs.next()) {
            Object[] row = {
                rs.getString("TaskName"),
                rs.getString("Description"),
                rs.getString("CompletionStatus"),
                rs.getString("Category"),
                };
                tmodel.addRow(row);
            }
            rs.close();
            ps.close();

        } catch (Exception e) {
            JOptionPane.showMessageDialog(rootPane, e);
        }//catch
    }
```