CTU training solutions

0861 100 395 | www.ctutraining.co.za | enquiry@ctutraining.co.za

Gabriella Rakgotsoka

20232605

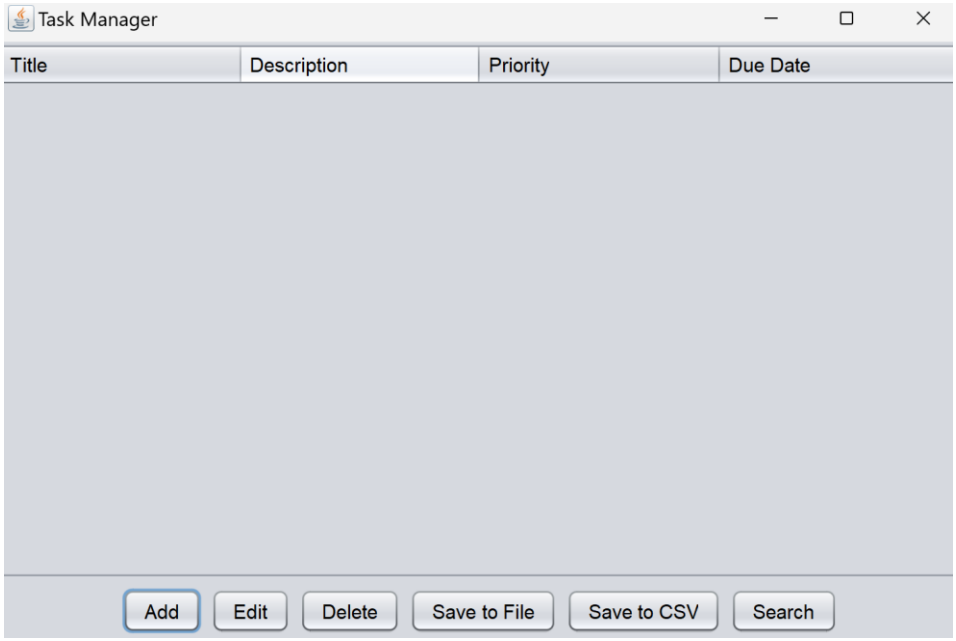
Table of Contents

Question 1

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 (25 marks)

•Implement a GUI to list tasks from the SQLite database. (5 marks)

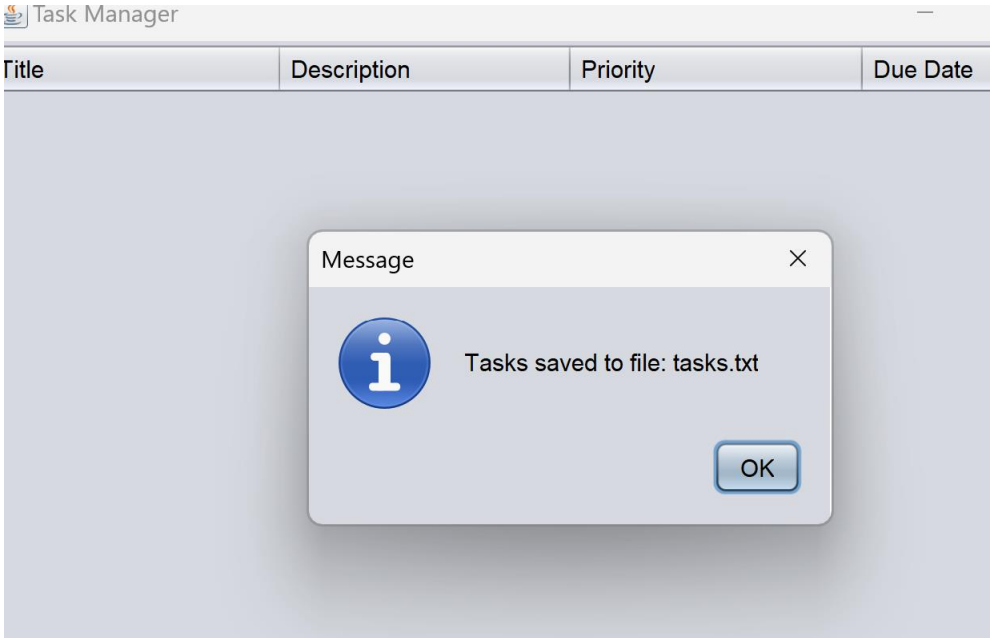


•Allow users to save tasks to a text file using OutputStream. (5 marks)

```
private void saveTasksToFile() {
        try (BufferedWriter writer = new BufferedWriter(new FileWriter(username + ".txt"))) {
            for (Task task : tasks) {
                writer.write(task.title + "," + task.description + "," + task.priority + "," + task.dueDate);
                writer.newLine();
            }
        } catch (IOException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(TaskManager.this, "Error occurred while saving tasks to file.");
        }
```

}

Output:



•Implement a mechanism to read tasks from the text file using InputStream. (5 marks)

```
private void loadTasksFromFile() {
        try (BufferedReader reader = new BufferedReader(new
FileReader(username + ".txt"))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] taskData = line.split(",");
                String title = taskData[0];
                String description = taskData[1];
                String priority = taskData[2];
                String dueDate = taskData[3];
                tasks.add(new Task(title, description, priority, dueDate));
            }
        } catch (IOException e) {
        }
```

•Display file properties like size and creation date using NIO. (5 marks)

•Provide an option to export task data to a CSV file using NIO. (5 marks)

```
private void exportToCSV(String filePath) {
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {
        // Write CSV header
        writer.write("Title,Description,Priority,DueDate\n");

        // Write task data
        for (Task task : tasks) {
            writer.write(task.title + "," + task.description + "," + task.priority + "," +
task.dueDate + "\n");
        }

        JOptionPane.showMessageDialog(TaskManager.this, "Task data exported to
CSV successfully!");
    } catch (IOException e) {
        e.printStackTrace();
        JOptionPane.showMessageDialog(TaskManager.this, "Error occurred while
exporting task data to CSV.");
    }
  }
```

Unit 6: Generics and Collections (20 marks)

•Design a task class that uses Generics to store task information. (5 marks)

```
private class Task {
    private String title;
    private String description;
    private String priority;
    private String dueDate;
```

```
    public Task(String title, String description, String priority, String dueDate) {
        this.title = title;
        this.description = description;
        this.priority = priority;
        this.dueDate = dueDate;
    }
```

•Use ArrayList to manage the list of tasks. (5 marks)

```
tasks = new ArrayList<>();
```

•Implement a filter mechanism to search for tasks based on user-defined criteria. (5 marks)

•Categorize tasks using HashMap to organize them based on user-defined categories. (5 marks)

Unit 7: Inner Classes (15 marks)

•Create an inner class to handle GUI components for task entry. (5 marks)

```
class TaskEntryDialog extends JDialog {
    private JTextField titleField;
    private JTextField descriptionField;
    private JComboBox<String> priorityComboBox;
    private JTextField dueDateField;
    private JButton saveButton;
    private JButton cancelButton;

    private Task task;

    public TaskEntryDialog(JFrame parent, Task task) {
        super(parent, "Edit Task", true);
        this.task = task;
```

```
}
```

•Design an inner class to manage task categories and their corresponding actions. (5 marks)

•Implement a nested panel structure using inner panels for better organization. (5 marks)

```
//Inner panel structure
        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new FlowLayout());
        buttonPanel.add(addButton);
        buttonPanel.add(editButton);
        buttonPanel.add(deleteButton);
        buttonPanel.add(saveToFileButton);
        buttonPanel.add(saveToCSVButton);
        buttonPanel.add(searchButton);


        JScrollPane scrollPane = new JScrollPane(taskTable);
```

Unit 8: JDBC (20 marks)

•Integrate a SQLite database with the application using JDBC. (5 marks)

```
private static final String DB_URL = "jdbc:sqlite:Tasks.db";
private void loadTasksFromDatabase() {
        tasks.clear();
        try (Connection conn = DriverManager.getConnection(DB_URL);
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery("SELECT * FROM Tasks")) {
            while (rs.next()) {
                String title = rs.getString("Title");
                String description = rs.getString("Description");
                String priority = rs.getString("Priority");
                String dueDate = rs.getString("DueDate");
                Task task = new Task(title, description, priority, dueDate);
```

```
            tasks.add(task);
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

•Design a database schema to store task information, including task name, description, completion status, and category. (5 marks)

•Implement functionalities to insert, update, and retrieve task data from the database. (5 marks)

```
//method to save data to the Tasks.db database
    private void saveTaskToDatabase(Task task) {
        try (Connection conn = DriverManager.getConnection(DB_URL);
             PreparedStatement pstmt = conn.prepareStatement("INSERT INTO Tasks (Title, Description, Priority, DueDate) VALUES (?, ?, ?, ?)")) {
            pstmt.setString(1, task.getTitle());
            pstmt.setString(2, task.getDescription());
            pstmt.setString(3, task.getPriority());
            pstmt.setString(4, task.getDueDate());
            pstmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


    //method to update data in the Tasks.db database
    private void updateTaskInDatabase(Task task) {
        try (Connection conn = DriverManager.getConnection(DB_URL);
             PreparedStatement pstmt = conn.prepareStatement("UPDATE Tasks SET Title = ?, Description = ?, Priority = ?, DueDate = ? WHERE Title = ?")) {
            pstmt.setString(1, task.getTitle());
            pstmt.setString(2, task.getDescription());
```

```
            pstmt.setString(3, task.getPriority());
            pstmt.setString(4, task.getDueDate());
            pstmt.setString(5, task.getTitle());
            pstmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    //method to delete data from the Tasks.db database
    private void deleteTaskFromDatabase(Task task) {
        try (Connection conn = DriverManager.getConnection(DB_URL);
             PreparedStatement pstmt = conn.prepareStatement("DELETE FROM Tasks
WHERE Title = ?")) {
            pstmt.setString(1, task.getTitle());
            pstmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
```

•Display tasks in the GUI retrieved from the database. (5 marks)

```
loadTasksFromDatabase();
private void displayTasksInTable() {
    tableModel.setRowCount(0);
    for (Task task : tasks) {
        tableModel.addRow(new Object[]{task.getTitle(), task.getDescription(),
task.getPriority(), task.getDueDate()});
    }
    // Allows editing in the table
    taskTable.setModel(tableModel);
}
```

Completed Declaration of Authenticity

I Gabriella Rakgotsoka _ hereby
(FULL NAME)
declare that the contents of this assignment JD522_FA2 is entirely my own work except for the following documents: (List the documents and page numbers of work in this portfolio that were generated in a group)

Activity	Date

Signature:  Date: 2024/04/19