

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**SCHOOL OF ENGINEERING AND TECHNOLOGY**

**FINAL ASSESSMENT FOR THE BSC (HONS) INFORMATION TECHNOLOGY; BSC (HONS) COMPUTER SCIENCE; BACHELOR of SOFTWARE ENGINEERING (HONS)YEAR 2**

**ACADEMIC SESSION 2023; SEMESTER 3**

**PRG2104: OBJECT ORIENTED PROGRAMMING**

**Project DEADLINE: Week 14**

**INSTRUCTIONS TO CANDIDATES**

# 

# This assignment will contribute 50% to your final grade.

* This is an individual assignment.

**IMPORTANT**

# The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

# Coursework submitted after the deadline will be awarded 0 marks

**Lecturer’s Remark** (Use additional sheet if required)

I.......Thien Tze Yea........ (Name) ......22022123.......std. ID received the assignment and read the comments..............*Tzeyea*................ (Signature/date)

**Academic Honesty Acknowledgement**

“I ...............Thien Tze Yea................(student name). verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties *(refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme)* for any kind of copying or collaboration on any assignment.”

…............ *Tzeyea*......20/8/2023......... (Student’s signature / Date)

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **No.** | **Content** | **Page** |
| 1 | Project Introduction | 1 |
| 2 | Class Diagram | 2 |
| 3 | Program Implementation | 3-9 |
| 4 | Personal Reflection | 10 |
| 5 | References | 11 |

**Project Introduction**

To accomplish the aim of this final project, I have developed a daily planner. Built on the foundation of Object-Oriented Programming (OOP) and incorporating a Graphical User Interface (GUI) created with ScalaFX and SceneBuilder, the project adheres to the model-view-controller (MVC) architectural pattern for a structured and modular design.

The core of this project is the ability for users not only to manage their to-do-tasks, but also to write their feelings or reflections of the day in a diary. Diaries in this project cannot be deleted. This is to allow users to face their feelings genuinely whether they are writing a diary or looking back at the past diaries. The integration of to-do list and diary allows users to measure their progress and self-growth.

In summary, the daily planner simplifies task management and diary keeping. Users can effortlessly create, edit, delete to-do-tasks, and write diary through an intuitive interface. The project’s organized class structure adheres to OOP principles, fostering code clarity and ease of maintenance.

1

**Class Diagram**

Figure 1: Class Diagram

The figure above shows the class diagram for the Daily Planner.

2

**Program Implementation**

**A screenshot of a computer

Description automatically generated**

Figure 2: Welcome Page

When user runs the program, this is the first page they would see. The user may choose to click the first button (Start the Day) to write to-do-tasks or the second button (End the Day) to write a diary.

A screenshot of a task

Description automatically generated

Figure 3: To-Do-Tasks Overview Interface (Start the Day)

The above interface is shown when the user clicks on the first button. The list of to-do-tasks is on the left side showing the task and the completion status. The user can view the details by clicking the tasks on the left side. The user may return to the home page, create a new task, edit an existing task, or delete a task by clicking the buttons on the right bottom.

3

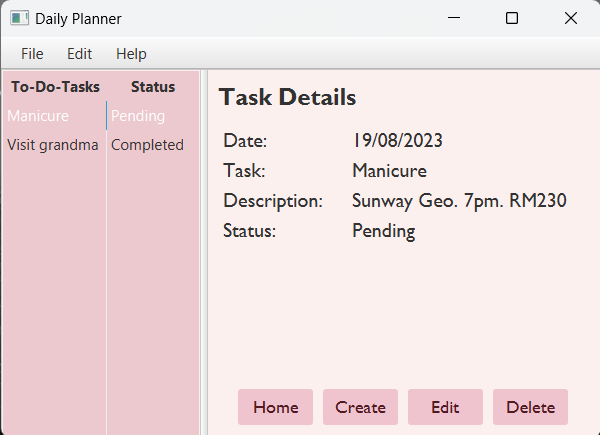


Figure 4: Viewing Task Details

Figure 4 shows the interface when user clicks on the task from the lists. The task details will be shown.

A screenshot of a computer

Description automatically generated

Figure 5: Creating a New Task

The user needs to enter the date, the task, the description of the task and the completion status of the task to create a new task. Every text field needs to fill in correct format or an error message will pop out. The user may click on “Cancel” and the dialog will be closed.

4

A screenshot of a computer

Description automatically generated

Figure 6: Error Message of Leaving Text Field Blank

Figure 6 shows the error message when a user leaves the Task text field, Description text field and Status text field blank.

A screenshot of a computer

Description automatically generated

Figure 7: Error Message of Entering Wrong Date Format

Figure 7 shows the error message when user enters a date with wrong format.

5

A screenshot of a computer

Description automatically generated

Figure 8: Editing Task

Figure 8 shows the editing dialog. The user may modify the task details in this dialog. After the user clicks on the “Create” button, the edited task will be updated. Error messages will be shown (refer to Figure 6 and Figure 7) as well if the user did not follow the format.

A screenshot of a computer

Description automatically generated

Figure 9: Deleting a Task without Selecting a Task

If the user clicks “Delete” without selecting a task first, error message will be popped out to ask the user to select a task.

6

A screenshot of a computer

Description automatically generated

Figure 10: Delete Confirmation

When a user has selected a task to be deleted, a confirmation message will show. The user may clicks on “OK” button to delete it or clicks on “Cancel” to keep the task.

A screenshot of a computer

Description automatically generated

Figure 11: Diary Overview Interface (End the Day)

This is the diary overview interface. The user may return to the welcome page and click on the “End the Day” button to access it. The saved diaries will be shown on the left side. There are no “Delete” and “Edit” buttons here as I encourage the users to face their feelings honestly.

7

A screenshot of a computer

Description automatically generated

Figure 12: Viewing a Diary

The user may view a diary from the left table by clicking it. The content will be displayed on the right side.

A screenshot of a computer

Description automatically generated

Figure 13: Creating a Diary

A date and diary content are needed to create a new diary. The user must follow the date format as shown in the figure or the error message will pop out. Clicking on the “Save” button with writing in correct format will create a new diary.

8

A screenshot of a computer

Description automatically generated

Figure 14: Saving a Diary with No Content Written in the Diary Text Area

An error message shown in Figure 14 will be displayed if the user did not enter any content in the text area.

9

**Personal Reflection**

In my project, I've used object-oriented programming (OOP) concepts to organize my code into different classes, each handling specific tasks. For instance, I created classes like “**Diary”** and “**Tasks”** to manage diaries and tasks. These classes use various tools like constructors, methods, and properties to handle data and behavior. I've also made sure these classes work together and with user interface parts, showing how things are wrapped up neatly. Also, I've kept different parts of the code separate for simplicity and easy changes without messing up the whole thing.

However, there are a lot of challenges for me during this assignment. Adding a function that can return to the welcome page is the biggest challenge for me. I have encountered many errors when writing the code for it. The code is runnable sometimes, but the functions are not performing well. It took few hours for me to debug and I finally found out the reason it is not functioning: the fx:id was not synchronized in the fxml file and scala file.

The strength of my submitted work is that I have built a database for both tasks and diaries that are recorded in the Daily Planner. The user may use this planner every day to keep track of their daily life as all data will be saved. Besides that, I have made a well-designed user interface that can attract the user. The weakness of this Daily Planner is that it does not have much functionality. A search function is needed if the tasks or the diaries saved is getting more.   
  
Presentation Video Link: https://youtu.be/VugrtPQeNNA

10

**References**

Chin. T.M. (2023) AddressApp source code [Source code].

*Dialogs and Alerts*. (n.d.). Www.scalafx.org. Retrieved August 20, 2023, from <https://www.scalafx.org/docs/dialogs_and_alerts/>

*Free Photo | Aesthetic sky background with moon and clouds in pink*. (n.d.). Freepik. Retrieved August 20, 2023, from https://www.freepik.com/free-photo/aesthetic-sky-background-with-moon-clouds-pink\_15612019.htm#query=pink%20aesthetic&position=23&from\_view=keyword&track=ais

‌

‌

11