Hong Kong Metropolitan University

COMP2090SEF/S209F

Data Structures, Algorithms And Problem Solving

Course Project Report

Group 49

Submission date: 15/4/2024

We declare that:

(i) all members of the group have read and checked that all parts of the project (including proposal, code programs, and reports), irrespective of whether they are contributed by individual member or all members as a group, here submitted is original except for source material explicitly acknowledged;

(ii) the project, in parts or in whole, has not been submitted for more than one purpose without declaration;

(iii) we are aware of the University's policy and regulations on honesty in academic work and understand the possible consequence when breaching such policy and regulations;

(iv) we confirm that we have declared in the report about the usage of AI and other generative models, including but not limited to ChatGPT, LLaMA, Gemini, Mistral, and Stable Diffusion, and complied with the instructions provided by HKMU; and

(v) we are aware that all members of the group should be held responsible and liable to disciplinary actions, irrespective of whether he/she has signed the declaration and whether he/she has contributed, directly or indirectly, to the problematic contents. We confirm that we have read through and understood the project requirements. We understand that failure to comply with the project requirements will result in score deduction.

Group members' full name and SID (the eight-digit number)

Lee Chung Hei 13496898

Chung Ming Lung 13490175

Cheng Long Hin 13492410

Lee Shing Chun 13489717

**1. Introduction**

In this group project, we decided to build a to-do list so people can use it to organize their daily activities more efficiently. We do acknowledge that many to-do lists are available for people to use, however, most of them do not have a weather forecast function.

**2. Methodology**

**2.1 Core functions and techniques**

There are 3 core functions in the project, including Task, Weather, and Shopping List. Task is a function for users to note down their task with a due date and details. The Weather function is a secondary function to help users plan their task order according to different weather. Shopping List is a branch of the Task function, it allows users to note down what they need to buy with price and quantity and calculate the total price for the user.

Task has implemented class for hash table, linked list and each task, file input and output for saving record, Tim sort and heap sort for sorting the task, and string operations for task editing.

The weather part of our app relies on API to get the required weather data, so we use the request module to send HTTP requests to the server. The JSON module is used to convert the fetched data into a Python dictionary.

Shopping List has implemented class for add, remove, view and sort items, save and load data and calculate total price. Shell sort and radix sort are used for sorting items.

**2.2 Class and data structure**

**Task:**

Custom hash table, custom linked list, custom task class, list, dictionary, built-in datetime module object

**Weather:**

List and dictionary.

**Shopping list:**

List and string

**2.3 GUI**

The GUI of our app is built by using the tkinter module. We use it use create different widgets that serve their own purpose and place it into one another. Grid geometry manager has been used to organize the layout of the GUI.

**3. Reflection**

There are some limitations and possible improvements that we could make in our program.

- The task page cannot have two tasks with identical task titles.

- The task page does not save the sorted result.

- The weather page of our app requires Wi-Fi to run properly.

- The shopping list requires users to manually click the save button to save the data.

-Task can only be removed one at a time.

**4. Declaration**

We, Group 49, declare that all our code in this project is written by us with no use of AI-generated code. All the knowledge we applied to this project is from what we learned from this course and online learning materials. We did ask AI to teach us about concepts or ideas we don’t fully understand and help us to test our data structure by generating test code which will not be run in actual usage. However, it is solely to help us solidify our understanding of the related knowledge we need to use in the project.

The weather images used in the weather page of our project are provided by the Hong Kong Observatory.

**5. References**

**App reference:**

**Microsoft Todo:**

idea of shopping list

own enhancement of weather information

**Algorithm reference:**

GeeksforGeeks: https://www.geeksforgeeks.org/

**6. Appendix**

**6.1 Modules that need to be installed before testing:**

1) tkcalendar

2) requests

**6.2 Demonstration**

Here are the steps for running this program:

1. Firstly, please ensure that all the necessary modules listed above (6.1) have been installed before executing the program. After they are installed, you are ready to run the main.py, and a window will pop up after a few seconds (Figure 1.1). In the window, there are three pages that you can choose from by clicking on the tabs, which are “Task”, “Weather”, and “Shopping List”. Let’s first focus on the task page.
2. On the task page, users can add a task by clicking the “Add Task” button, a window that allows users to add the task title, select the due date, and add the task detail will pop up (Figure 1.2). After inputting all the necessary information, click the “Create Task” button and a task will be created. Users can select the task that they want to check or edit and click the “Check task” button, a similar window will pop up, however, this time it allows users to save the edit with the “Save Edit” button (Figure 1.4). Users can delete a task by clicking on the task they want to delete and click the “Delete” button.
3. After having multiple tasks on the list, users can sort them by choosing the sorting options in the combobox, the options include sorting by name (A-Z), sorting by date, and sorting by pinned (Figure 1.4). Once the sorting option is selected, users can click “start sort” and the tasks will be sorted according to the chosen sorting method. An example of sorting by date is shown in (Figure 1.5). Other than that, users can also pin or unpin each of the tasks and search for them by typing in their name in the text box.
4. On the weather page, users can see the basic weather information of the day (Figure 2.1). Click the “9-day forecast” button to view the weather prediction for the upcoming 9 days (Figure 2.2).
5. In the shopping list, users can type in the item name, per item price, and item quantity. After inputting the information, click the “Add Item” button and a task will be created (Figure 3.1). Select the item and click the “Remove Item” button if you want to delete it. When there are many items created, users can choose to click the “Clear List” button to remove all the items.
6. As the number of items increases, sorting could be useful. Users can select whether to sort the items by letter or price in the combobox (Figure 3.2). Here is an example of items sorted by price (Figure 3.3). Click the “Calculate Total Price” button to check the total price of the items in the shopping list, and a window that shows the total price will pop up (Figure 3.4).

**Test cases:**

**Task:**

(please don’t delete any record or skip any steps while trying the test case, since some of the error test cases are due to duplicate tasks)

**Add Task: (task title is case sensitive)**

**Normal test case:**

1)title: Task1, (date on or after today), detail: Detail...

2)title: Task2, (date on or after today), detail: Task2 (newline) Detail

**error test case:**

1)title: Task1, (date on or after today), detail: (empty)--> (duplicate task)

2)title: (empty), (date on or after today), detail:(empty) --> (mandatory title)

3)title: Task3, (date before today), detail(empty) --> (due date can only be on today or after)

**Search:**

**Normal test case**

1)Task -- (result: task with the word “Task” as a substring)

2)Task1 -- (result: task 1)

3)Task3 -- (result: empty)

4)(empty) -- (result: all tasks)

**Check task:**

**Normal test case**

1)select task1-> press check task:

result: title: Task1, (date on or after today), detail: Detail...

2)change title: Task1 -> Task3 -> Save Edit

**Error test case**

1) select Task3 change to Task2-> Save Edit -> (duplicate task)

2) change Task3 due date to date before today -> (invalid due date)

**Sort:**

**By Name:**

Order: Task2->Task3

**By Date:**

Order: ascending date order

**By Pinned:**

Select a task to pin or unpin

Order: pinned task will be on top

**Delete:**

(error will occur if no task was selected)

Select task -> Delete -> confirm

**Weather:**

**Check Temperature:**

Select a location in the list-> temperature updated

**9 days forecast:**

Select 9 date forecast-> switch page to Task or Shopping list (window will change back to normal size)

**Shopping List:**

**Check Add items:**

Item: Apple

Price: 10

Quantity: 5

--> 5 Apple $10 added to the shopping list

**Check Remove items:**

* 1. Click the Apple in the shopping list
  2. Click the Delete item button

--> The Apple should be removed from the shopping list

**Check Clearing the list:**

Add 2 items

For example:

Item: Apple Item: Banana

Price: 10 Price: 20

Quantity: 5 Quantity: 10

And then click Clear List, all items in the list box should be empty

**Check Sort by price:**

Add 3 items:

For example:

Item: Apple Item: Banana Item: Orange

Price: 10 Price: 20 Price: 15

Quantity: 5 Quantity: 10 Quantity: 20

Click sort by price,

The list should sort in ascending order.

5 Apple $10

20 Orange $15

10 Banana $20

**Check sort by letter:**

Click sort by letter,

The list should be sorted alphabetically by their names.

5 Apple $10

10 Banana $20

20 Orange $15

**Check saves and load data:**

After listing the items, click on save data.

The data should be saved to file.

Exit and rejoin the app and click load data.

The data should be displayed in the list box.

**Check calculates total price:**

The total price should be $340.

**7. Workload allocation**

Lee Chung Hei (13496898): Weather, heap sort, GUI

Chung Ming Lung (13490175): Task, Tim sort, GUI

Cheng Long Hin (13492410): Shopping List, Radix sort, GUI

Lee Shing Chun (13489717): Shopping List, Shell sort GUI

Task Page

A screenshot of a computer

Description automatically generated

(Figure 1.1)

A screenshot of a computer

Description automatically generated

(Figure 1.2)

A screenshot of a computer

Description automatically generated

(Figure 1.3)

A computer screen shot of a computer

Description automatically generated

(Figure 1.4)

A screenshot of a computer

Description automatically generated

(Figure 1.5)

Weather Page

A screenshot of a weather forecast

Description automatically generated

(Figure 2.1)

A screenshot of a weather forecast

Description automatically generated

(Figure 2.2)

Shopping List Page

A screenshot of a computer

Description automatically generated

(Figure 3.1)

A screen shot of a computer

Description automatically generated

(Figure 3.2)

A screenshot of a computer

Description automatically generated

(Figure 3.3)

A screenshot of a computer

Description automatically generated

(Figure 3.4)