



TASK SCHEDULER DOCUMENTATION

Assignment 01



CS/2017/017
S. A. P. B. MAHEEPALA

Introduction

Maintenance is an essential aspect of ensuring optimal functionality and longevity of assets and systems. The Task Scheduler System is designed to streamline maintenance administration by providing a complete and simple solution for planning and managing maintenance work. This application enables maintenance staff to efficiently create, schedule, assign, and track tasks, leading to reduced downtime and improved maintenance performance.

Proposal vs. Implementation

The proposal for the Task Scheduler System aimed to develop a task management application that allows users to add, assign, complete, and delete tasks. The implementation successfully fulfills this proposal, providing a robust and efficient task scheduling tool. In the proposal I have mentioned, User roles and task repetitions and in the implementation, I removed those 2 key features instead of those, I added task completion and see all tasks that are retrieved from the saved file features.

Features

The Task Scheduler System offers the following features:

1. Task Creation: Users can create new tasks with details like name, description, due date, and priority.
2. Task Assignment: Tasks can be assigned to specific individuals, with their names displayed in the task table.
3. Task Completion: Users can mark tasks as completed.
4. Task deletion: can delete a task.
5. Priority Levels: Users can set task priorities as High, Medium, or Low during task creation.

Technology Stack

As mentioned in the proposal as the tech stack, I used,

- Python as the programming language
- Data storage – text file (task_scheduler.txt)
- Development Environment as Visual Studio Code
- Version Controlling – GitHub
 - Link for the repository - https://github.com/Praveen-98cs/Task_Scheduler_Maintenance

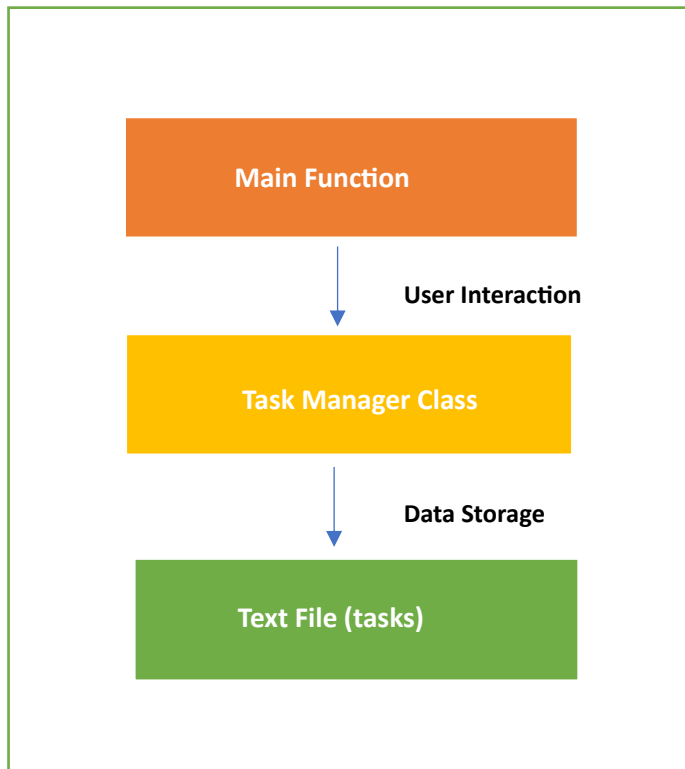
Executing

There are two ways to approach the final output

1. Open the `task_scheduler.exe` file.
Go to the file directory and go to dist and open the .exe file.
2. Open the `task_scheduler.py` file.
 - a. Open the command prompt in the respective directory of the file **task_scheduler.py** and run this command
`"python task_scheduler.py"`
 - b. Open the file **task_scheduler.py** using VS code and run the command
`"python task_scheduler.py"`

The interface is created on CLI.

Architecture



- The main function serves as the entry point of the application and provides a console-based menu for user interaction.

- The Task Manager class manages the list of tasks, including creation, assignment, completion, and deletion. It loads tasks from the text file at the start and saves tasks back to the file after modifications.
- The data storage is achieved through a text file named "task_scheduler.txt," where each task is represented as a line with comma-separated values for its attributes.

Interaction Flow

Step 1: Start the Task Scheduler application.

Step 2: The application loads existing tasks from the "task_scheduler.txt" file, if available.

Step 3: Display the main menu with options for various tasks.

```
=====
===== Task Scheduler Menu =====

1. Create Task
2. Assign Task
3. Complete Task
4. View All Tasks
5. Delete Task
0. Exit

=====
Enter your choice:
```

Figure 1 - The task scheduler Menu

Step 4: User chooses an option from the menu and enter the number of it.

Step 5: If the user selects "Create Task" (Option 1):

- Prompt the user to enter task details: name, description, due date, and priority.
- Validate the input for date format and priority level.
- Create a new task with the provided details and add it to the list of tasks.
- Save the updated tasks to the "task_scheduler.txt" file.
- Display a success message indicating that the task has been created.

```
=====
Enter your choice: 1
Enter task name: task_1
Enter task description: checking task 1
Enter due date (YYYY-MM-DD): 2023-10-11
Enter priority (High/Medium/Low): High
Task created successfully.
Hit Enter to go to Main Menu
```

Figure 2 – Details to add in creating a task

Step 6: If the user selects "Assign Task" (Option 2):

- a. Check if there are any tasks available. If not, display a message and return to the main menu.
- b. Display a table with all tasks showing their details.
- c. Prompt the user to enter the number of the task to assign.
- d. If the selected task's status is "Assigned," prompt the user to change the name of the assignee
- e. Otherwise, prompt for the name of the assignee.
- f. Update the task's status to "Assigned" and set the assignee's name.
- g. Save the updated tasks to the "task_scheduler.txt" file.
- h. Display a success message indicating that the task has been assigned.

#	Task Name	Description	Due Date	Priority	Status	Assignee
1	task_1	checking task 1	2023-10-11	High	Assigned	praveen
2	task_2	checking task 2	2023-10-14	medium	Pending	
3	task_3	checking task 3	2023-10-15	low	Pending	

```
Enter the number of the task to assign: 1
Change the name of the assignee: raveen
Task assigned successfully.
Hit Enter to go to Main Menu
```

Figure 3 – if task has already assigned, appoint a new assignee

#	Task Name	Description	Due Date	Priority	Status	Assignee
1	task_1	checking task 1	2023-10-11	High	Assigned	raveen
2	task_2	checking task 2	2023-10-14	medium	Pending	
3	task_3	checking task 3	2023-10-15	low	Pending	

```
Enter the number of the task to assign: 2
Enter the name of the assignee: kaveen
Task assigned successfully.
Hit Enter to go to Main Menu
```

Figure 4 – task is not assigned, assign a new person

Step 7: If the user selects "Complete Task" (Option 3):

- a. Check if there are any assigned tasks available.
- b. If not, display a message and return to the main menu.
- c. Display a table with all assigned tasks showing their details.
- d. Prompt the user to enter the number of the completed task.

- e. If the selected task is assigned, mark it as "Completed."
- f. Save the updated tasks to the "task_scheduler.txt" file.
- g. Display a success message indicating that the task has been completed.

#	Task Name	Description	Due Date	Priority	Status	Assignee
1	task_1	checking task 1	2023-10-11	High	Assigned	raveen
2	task_2	checking task 2	2023-10-14	medium	Assigned	kaveen

Enter the number of the completed task: 1
Task completed successfully.
Hit Enter to go to Main Menu_

Figure 5 – complete task -Table only shows the assigned tasks to assign as completed

Step 8: If the user selects "View All Tasks" (Option 4):

- a. Check if there are any tasks available. If not, display a message and return to the main menu.
- b. Display a table with all tasks showing their details.

===== All Tasks =====

#	Task Name	Description	Due Date	Priority	Status	Assignee
1	task_1	testing 1	2023-10-11	high	Completed	raveen
2	task_2	testing 2	2023-10-13	medium	Assigned	praveen
3	task_3	testing 3	2023-10-17	low	Pending	None

Hit Enter to go to Main Menu

Figure 6 – All the tasks

Step 9: If the user selects "Delete Task" (Option 5):

- a. Check if there are any tasks available. If not, display a message and return to the main menu.
- b. Display a table with all tasks showing their details.
- c. Prompt the user to enter the number of the task to delete.
- d. Delete the selected task from the list of tasks.
- e. Save the updated tasks to the "task_scheduler.txt" file.
- f. Display a success message indicating that the task has been deleted.

#	Task Name	Description	Due Date	Priority	Status	Assignee
1	task 1	testing 1	2023-10-11	high	Completed	raveen
2	task_2	testing 2	2023-10-13	medium	Assigned	praveen
3	task 3	testing 3	2023-10-17	low	Pending	None

Enter the number of the task to delete: 3
Task deleted successfully.
Hit Enter to go to Main Menu

Figure 7 – table of tasks to choose Delete a task

Step 10: If the user selects "Exit" (Option 0):

- a. Save all tasks to the "task_scheduler.txt" file.
- b. Exit the application.

Step 11: The application ends, and the user exits the Task Scheduler.

Conclusion

The Task Scheduler System provides a user-friendly, efficient solution for managing maintenance tasks. It simplifies task scheduling, assignment, and tracking, enhancing productivity and timely task completion. The application is implemented using Python, ensuring compatibility and ease of use. Comprehensive documentation and testing ensure understanding and maintenance of the system, making it a valuable tool for maintenance management.