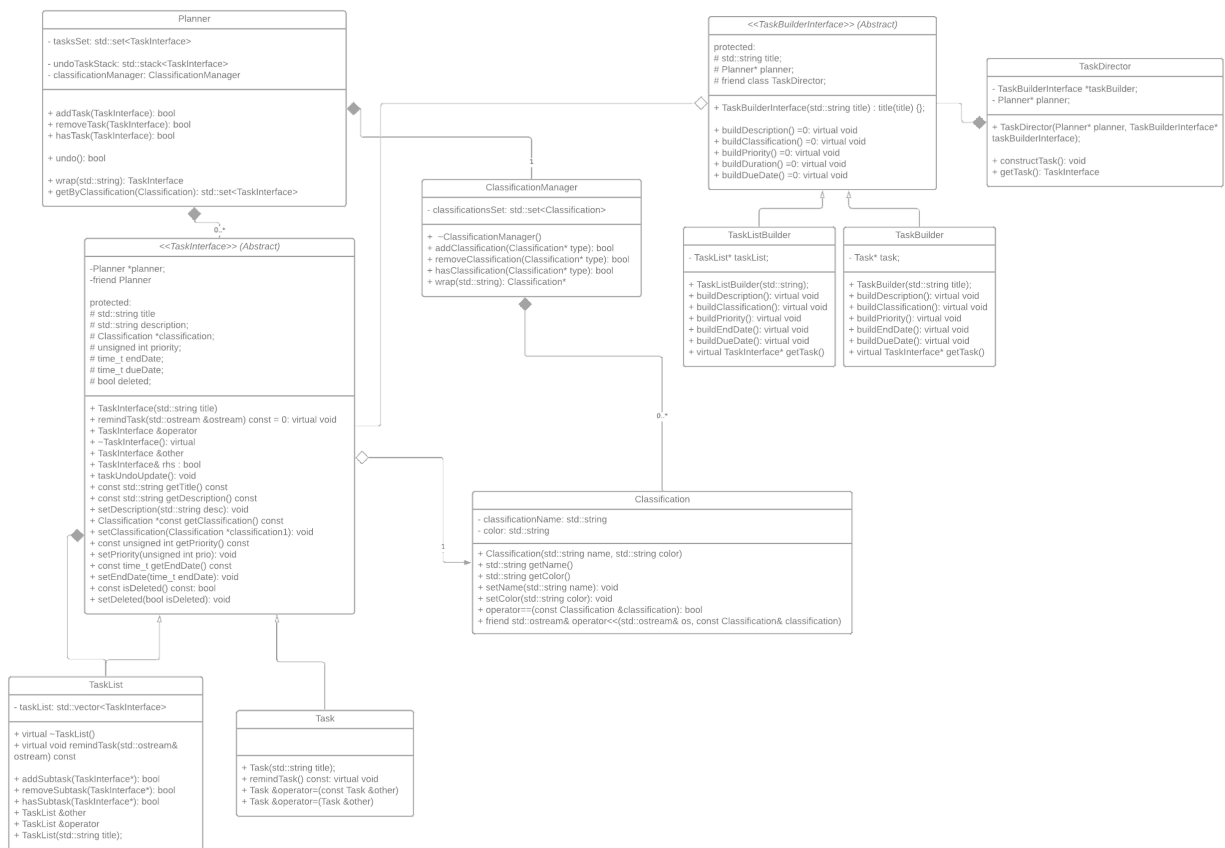


Project Description

- Why is it important or interesting to you?
 - As students, a Task Scheduler allows us to organize our daily lives. Being able to create detailed tasks with due dates and descriptions keeps us on top of our busy schedules. Keeping all of our tasks in a single application lets us catalog our lives in a convenient workspace.
- What languages/tools/technologies do you plan to use?
 - Used C++, github, PuTTY, and gtest. C++ for an objected oriented programming workspace. GitHub for version control system and for a backup repository. PuTTY to connect to Hammer. GTest to create unit tests and make sure the program's classes work as intended.
- What will be the input/output of your project?
 - Input will be taken from the users to generate visual blocks of Tasks with due dates, descriptions, and further details. Output will display a calendar with the tasks in a user-friendly manner. The program will notify users if their task is due today. Output will display a calendar with the tasks in a user-friendly manner. The program will notify users if their task is due today.
- What are the two design patterns you will be using?
 - Implemented the Structural-Composite design pattern as a part of our Task Scheduler. The composite structure, which focuses on creating multiple instances of a type of object, fits with the idea of creating multiple Tasks of varying details. Are able to put the tasks into categories as well as organize them in a certain way. This will allow the Task Scheduler to categorize based on the type of work or priority.
 - Used the Creational-Builder design pattern to handle the varying types of tasks. This pattern allows us to categorize our tasks with different types of object-creating paths. While the tasks will inherit from our base class, the Builder pattern enables us to create different types of the original object. This comes into consideration while creating different task templates for the different classifications of tasks: Work, Personal and Study.
 - So basically these Builder pattern, Creational are design patterns about class instantiation or object creation. These patterns can be further categorized into Class-creational patterns and object-creational patterns. While class-creation patterns use inheritance effectively in the instantiation process, object-creation patterns use delegation effectively to get the job done.

Class Diagram



The Planner class will contain the entire collection of TaskInterfaces. The TaskInterfaces, while using the Composition strategy pattern, will allow us to remind the user using any class that is subclassed to TaskInterface. Some examples include: TaskList and Task. In addition, in order to categorize each Task, it is required to create the Classification class and are able to assign classifications to each Task. The Builder strategy pattern will be used to create Tasks from scratch. This will allow us to assign defaults and quickly construct objects.

Screenshots

```
Main Menu
1)Create a task
2)View all reminders
3)Exit
1

Task Menu
1)Create a regular task
2)Create a task list
3)Go back
1

Enter a title:Title

Enter a description or press (N) to skip:Description

Enter a classification or press (N) to skip:Classy

Enter a priority or press (-1) to skip:3

Enter due date year or (-1) to skip: 1

Enter due date month or (-1) to skip:-1

Enter due date day or (-1) to skip:-1

Enter due date hour or (-1) to skip: 1

Enter due date minute or (-1) to skip: 1

Enter due date second or (-1) to skip: 1

Enter end date year or (-1) to skip: 1

Enter end date month or (-1) to skip:-1

Enter end date day or (-1) to skip:-1

Enter end date hour or (-1) to skip: 1

Enter end date minute or (-1) to skip: 1

Enter end date second or (-1) to skip: 1

Completed the task!
Main Menu
1)Create a task
2)View all reminders
3)Exit
2

[5 ] (Classy ) "Title          " "Description          " "Tue Mar  9 23:11:44 2021" "Tue Mar  9 23:41:47 2021"

Main Menu
1)Create a task
2)View all reminders
3)Exit
1
```

```

Main Menu
1)Create a task
2)View all reminders
3)Exit
|

Task Menu
1)Create a regular task
2)Create a task list
3)Go back
|

Enter a title:title
Enter a description or press (N) to skip:desc
Enter a classification or press (N) to skip:classy
Enter a priority or press (-1) to skip:3
Enter due date year or (-1) to skip:1975
Enter due date month or (-1) to skip:4
Enter due date day or (-1) to skip:3
Enter due date hour or (-1) to skip:4
Enter due date minute or (-1) to skip:4
Enter due date second or (-1) to skip:4
Enter end date year or (-1) to skip:1988
Enter end date month or (-1) to skip:2
Enter end date day or (-1) to skip:3
Enter end date hour or (-1) to skip:4
Enter end date minute or (-1) to skip:2
Enter end date second or (-1) to skip:3

Completed the task!
Main Menu
1)Create a task
2)View all reminders
3)Exit
|

[5 ] (classy ) "title          " "desc          " "Fri Apr  4 05:04:04 1975" "Wed Feb  3 04:02:05 1988"

Main Menu
1)Create a task
2)View all reminders
3)Exit
|

```

```

Enter end date year or (-1) to skip: -1
Enter end date month or (-1) to skip: -1
Enter end date day or (-1) to skip: -1
Enter end date hour or (-1) to skip: -1
Enter end date minute or (-1) to skip: -1
Enter end date second or (-1) to skip: -1
Enter a title: test
Enter a description or press (N) to skip: N
Enter a classification or press (N) to skip: N
Enter a priority or press (-1) to skip: -1
Enter due date year or (-1) to skip: -1
Enter due date month or (-1) to skip: -1
Enter due date day or (-1) to skip: -1
Enter due date hour or (-1) to skip: -1
Enter due date minute or (-1) to skip: -1
Enter due date second or (-1) to skip: -1
Enter end date year or (-1) to skip: -1
Enter end date month or (-1) to skip: -1
Enter end date day or (-1) to skip: -1
Enter end date hour or (-1) to skip: -1
Enter end date minute or (-1) to skip: -1
Enter end date second or (-1) to skip: -1
Add more to the task list (Y/N)?:
N

Completed the task list!
Main Menu
1)Create a task
2)View all reminders
3)Exit
N
[5 ] (test ) "test" " " "Tue Mar 9 23:15:24 2021" "Tue Mar 9 23:45:28 2021"
[0 ] (Unnamed) "test" " " "Write a description fo..." "Wed Dec 31 16:00:00 1969" "Tue Mar 9 23:15:45 2021"

```

```

Enter due date month or (-1) to skip: 1
Enter due date day or (-1) to skip: -1
Enter due date hour or (-1) to skip: -1
Enter due date minute or (-1) to skip: -1
Enter due date second or (-1) to skip: -1
Enter end date year or (-1) to skip: 1
Enter end date month or (-1) to skip: 1
Enter end date day or (-1) to skip: -1
Enter end date hour or (-1) to skip: -1
Enter end date minute or (-1) to skip: -1
Enter end date second or (-1) to skip: -1

Completed the task!
Main Menu
1)Create a task
2)View all reminders
3)Edit a Task
4)Exit
2
[0 ] (test ) "test" " " "Tue Mar 9 23:42:41 2021" "Wed Mar 10 00:12:44 2021"

Main Menu
1)Create a task
2)View all reminders
3)Edit a Task
4)Exit
3
Which Task would you like to edit: test1

What would you like to edit: 1)Description
2)Priority
3)Due Date
4)End Date
1

Change description: no more test

Main Menu
1)Create a task
2)View all reminders
3)Edit a Task
4)Exit
2
[0 ] (test ) "test" " " "no more test" " " "Tue Mar 9 23:42:41 2021" "Wed Mar 10 00:12:44 2021"

Main Menu
1)Create a task
2)View all reminders
3)Edit a Task
4)Exit
1

```

Installation/Usage

The program installation requires someone to be able to call a .out file.

- The user is prompted with a main menu. They are shown 5 options which are : “Create a task”, “Remove a task”, “View all reminders”, “Edit a task”, “Undo”, and “Exit”
- Creating a task allows the user to create a task or a tasklist.
 - Task creation asks the user to enter title, description, classification, due date, and end date.
- Removing a task allows the user to remove a task from the planner.
 - Asks the user for the title of the task.
- Viewing all reminders displays each task to the console.
 - Displays simply to the console.
- Edit a task changes the task details
 - Task edit asks the user to edit a title, description, classification, due date, OR end date.
- Undo reverts back to the previous task before editing.
 - Simply undos.
- Exit, exits the program

Testing

Created test cases and tested each individual task, classification, builder, and planner class in order to validate the cases. Also tested examples using the planner for undoing tasks. Tasks and classifications are IMMUTABLE in testing.