Task Planner System

Design and implement a task planner system

A task has the following details :

1. Title
2. Creator
3. Assignee (Optional)
4. Status
5. Type
6. Due date

A task can be of the following types with additional information apart from what is mentioned above

1. Feature
    a. Feature summary
    b. Impact (Low, Moderate, High)
2. Bug
    a. Severity (P0, P1 or P2)
3. Story[1]
    a. Story Summary
    b. A story can have multiple Subtasks
        A Subtask has the following details :
            i. Title
            ii. Status
        A Subtask can be created and attached only to an existing story in non-completed status

It should be **easy** to add a new task type to your application

The status can change from a state to any state.
Status field takes one of the following states depending on the task type :
Feature => Open, In progress, Testing, Deployed
Bug => Open, In progress, Fixed
Story => Open, In Progress, Completed

1

Subtask status field takes one of these values
Open, In Progress, Completed
Story can only be marked as Completed, if all of its Subtasks are Completed.

A sprint is defined as a collection of tasks used to track progress. You can add or remove a task from sprints.
A task can be part of only one sprint at a time.

Your task planner should have the following functionalities:

1. Task
    a. Create a task of any type
    b. Create Subtask
    c. Change the status of the task/Subtask
    d. Change assignee of the task
    e. Display tasks assigned to a user categorised by task type
2. Sprint
    a. Create/Delete a Sprint
    b. Add/remove task to/from sprint
    c. Display sprint snapshot. This should display the tasks that are part of the sprint and also display if they are delayed or on track
3. **Bonus Question (Only if time permits) :-**
    a. The transition/change in the status should be based on allowed transitions
    eg :-
    For task type feature, we might want to have only following allowed transitions :
        - Open => In progress
        - In progress => Testing
        - Testing => Deployed
        - In progress => Deployed

The examples below are just to understand the functionalities and **may not necessarily** be used in the same format as input to your driver program.

Let's say we want to add the following tasks to a sprint :

Tasks:

Title	**Creator**	**Assignee**	**Status**	**Due date**	**Type**	**Type attributes (comma separated)**	**Sprint**

Create Dashboard	Brad	Peter	Open	2024-04-12	Feature	Create console for debugging, Low	
Fix mysql issue	Ryan	Ryan	In progress	2024-04-14	Bug	P0	Sprint-1
Create a microservice	Amy	Ryan	Completed	2024-03-12	Story	Add logging to the feature	Sprint-1
Setup console	Ryan	Ryan	In progress	2024-04-14	Feature	Create console for debugging, High	
Console api	Ryan	Ryan	In progress	2024-04-14	Feature	Create api for console , High	

Examples:

- CreateSprint(name="Sprint1")

- CreateTask(title="Create Dashboard", creator="Ryan", dueDate="2024-04-12", type="Feature", attributes={summary: "Create console for debugging", impact:"Low"})

- CreateTask(title="Fix mysql issue", creator="Ryan", dueDate="2024-04-14", type="Bug", attributes={"severity:"P0"})

- AddTaskToSprint(title="Fix mysql issue", sprint="Sprint1")
- AssignTaskToUser(title="Fix mysql issue", user="Ryan")
- UpdateTaskStatus(title="Fix mysql issue", status="In Progress")

Subtask:

Title	Status	Parent task
Development	Open	Create microservice
Unit Test	Open	Create microservice
Integration Test	Open	Create microservice

Example:

CreateSubtask(parentTask=”Create a microservice”, title=”development”)

Display tasks assigned to a user categorised by task type:

Example:

1. Eg: viewAssignedTask(Ryan)
User => Ryan:
    Task Type => Bug
        Title => Fix mysql issue
        Sprint => Sprint-1

    Task Type => Feature
        Title => Setup console
        Sprint =>

        Title => Console api
        Sprint =>

    Task Type => Story
        Title => Create a microservice

Sprint => Sprint-1
Subtask:
Development
Unit Test
Integration Test

2. Eg: viewAssignedTask(Peter)
User => Peter:
Task Type => Feature:
Title => Create Dashboard
Sprint => Sprint Id/Sprint name

3. E.g displayStatus(Sprint-1)

Sprint title => Sprint-1
On track Tasks:
Fix mysql issue
Setup console
Create Dashboard

Delayed Tasks:
Create a microservice

Note: Task will come under “delayed task” if the task (with non-completed status) has crossed the due date

Expectations :
Please make use of in memory data structures and **do not use any external datastore** for storage/querying.

Input/Output can be from terminal or file or taken from some data structure inside the driver program whichever is convenient for you. It should be easy for the evaluator to give new input or change existing input and test new cases.

You can look for API references online.

1. **Demoable code**. Functionalities mentioned above.
2. Clean Interface design for the module along with good readability.
3. Clean internal design and implementation of the library and the application.
4. **Functional completeness and correctness** is a must. Bonus questions are good to have.
5. **Extensibility**
6. Take care of Exception and Corner case handling.
7. Test cases covering various cases are good to have.
8. You are free to use the language of your choice.