Design and implement a task planner system

Task Model

A **Task** has the following attributes:

1. **Title**: The title of the task.
2. **Creator**: Name of the person who created the task.
3. **Assignee** *(Optional)*: Name of the person assigned to the task.
4. **Status**: Current status of the task (can change based on the task type).
5. **Type**: The type of the task (Feature, Bug, Story, or SubTask).
6. **Due Date**: The date by which the task is due.

Task Types:
A task can be one of the following types, each with additional attributes:

1. **Feature**:
    - Feature Summary
    - Impact (Low, Moderate, High)
2. **Bug**:
    - Severity (P0, P1, P2)
3. **Story**:
    - Story Summary
    - **SubTasks**: A **Story** can have multiple **SubTasks** defined within it.
    - A SubTrack has the following details :
        1) Title
        2) 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

Task Status Transitions

The status can change from a state to any state. The 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

- **SubTask**: Open, In Progress, Completed

A **Sprint** is a collection of tasks used to track progress. A task can be part of only one sprint at a time, but tasks can be added to or removed from sprints.

Your task planner should have the following functionalities:

Task

1. **Create a task**: Create a task of any type (Feature, Bug, Story, or SubTask).
2. **Create a subtask**: A subtask can only be created as part of a **Story**.
3. **Change the status**: Update the status of a task or subtask.
4. **Change assignee**: Reassign a task to another user.
5. **Display tasks assigned to a user**: Display tasks assigned to a specific user, categorized by task type (Feature, Bug, Story).

Sprint

1. **Create/Delete a Sprint**.
2. **Add/Remove a task to/from a Sprint**.
3. **Display Sprint Snapshot**: Show the tasks within a sprint and their current status (On Track or Delayed). A task is considered delayed if it is not completed and has crossed its due date.

Bonus Question (If time permits)

- Implement **status transitions** based on allowed states. For example, for a **Feature**, the allowed transitions could be:
    - Open → In Progress
    - In Progress → Testing
    - Testing → Deployed
    - In Progress → Deployed

Example

Tasks

1. **Task: Create Dashboard**
    - **Creator**: Brad
    - **Assignee**: Peter
    - **Status**: Open

    - **Due Date**: 2024-12-12
    - **Type**: Feature
    - **Feature Summary:** Create a console for debugging
    - **Impact**: Low
    - **Sprint**: Sprint-1
2. **Task: Fix MySQL Issue**
    - **Creator**: Ryan
    - **Assignee**: Ryan
    - **Status**: In Progress
    - **Due Date**: 2024-12-14
    - **Type**: Bug
    - **Severity**: P0
    - **Sprint**: Sprint-1
3. **Task: Create a Microservice**
    - **Creator**: Amy
    - **Assignee**: Ryan
    - **Status**: Completed
    - **Due Date**: 2024-12-18
    - **Type**: Story
    - **SubTasks**: Development, Unit Test, Integration Test
    - **Sprint**: Sprint-1
4. **Subtask:**
    - **Title**: Development
    - **Status**: Open
    - **Parent Task**: Create a microservice
5. **Subtask:**
    - **Title**: Unit Test
    - **Status**: Open
    - **Parent Task**: Create a microservice
6. **Subtask:**
    - **Title**: Integration Test
    - **Status**: Open
    - **Parent Task**: Create a microservice

Display tasks assigned to a user categorized by task type:

For example, if the assignee is **Ryan**:

- **Bug**:
    - Task: Fix MySQL Issue, In progress, 2024-12-14, Bug, P0, Sprint-1

- **Story**:
    - Task: Create a Microservice, completed, 2024-12-18, Sprint-1
    - SubTasks:
        - Development
        - Unit Test
        - Integration Test

For **Peter**:

- **Feature**:
    - Task: Create Dashboard, Open, 2024-12-12, Sprint-1

Sprint Status

For **Sprint-1**:

- **On Track Tasks**:
    - Fix MySQL Issue
    - Setup Console
    - Create Dashboard
- **Delayed Tasks**:
    - Create a Microservice

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

Expectations

1. **In-Memory Data Structures**: Do not use external databases or data stores; the application should function with in-memory data structures.
2. **Flexible Input/Output**: Input and output can come from the terminal, a file, or in-memory structures. The system should be easy to extend with new data or test cases.
3. **Clean and Demoable Code**: Ensure the code is functional and easy to demonstrate.
4. **Extensibility**: The system should be easily extendable to add new task types and features.
5. **OOPs & Design Patterns:** Apply OOPs and design patterns wherever necessary
6. **Exception Handling**: Properly handle exceptions and corner cases.
7. **Test Cases**: Provide test cases to cover various scenarios.
8. **Language Choice**: You are free to choose any programming language you prefer.

Submission Instructions

- Save your code with your name.
- Share your code via the provided Google form link or email.