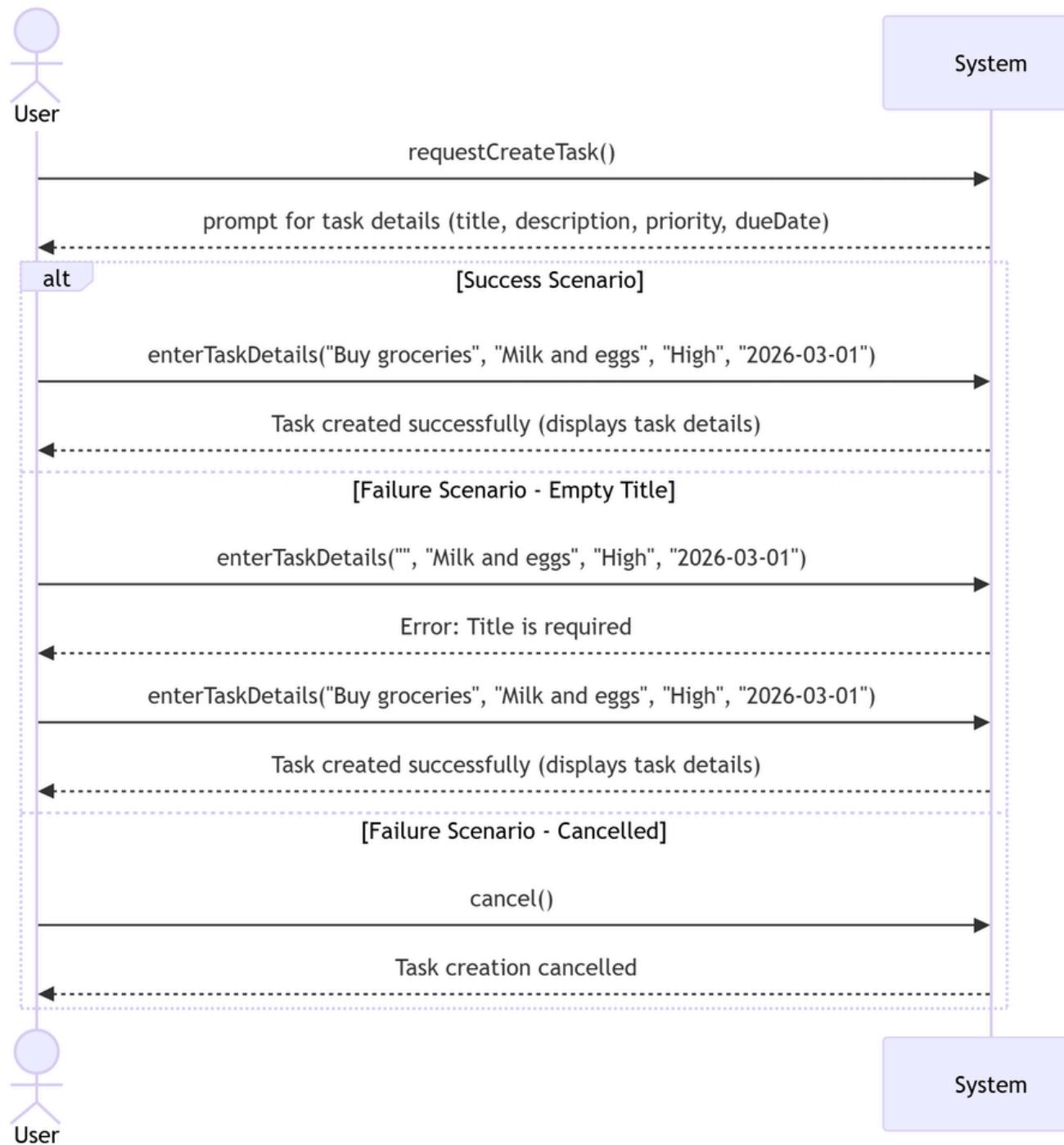
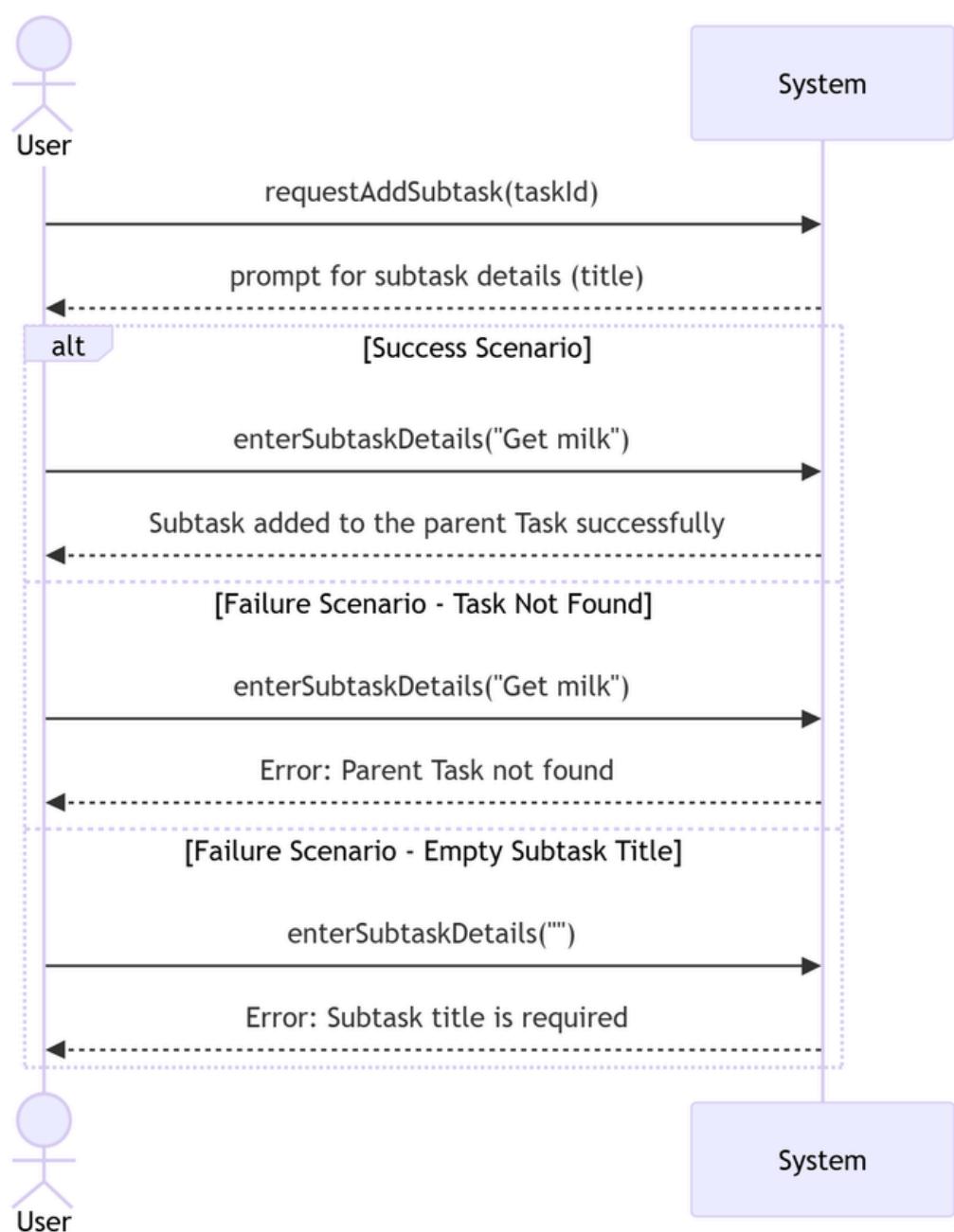


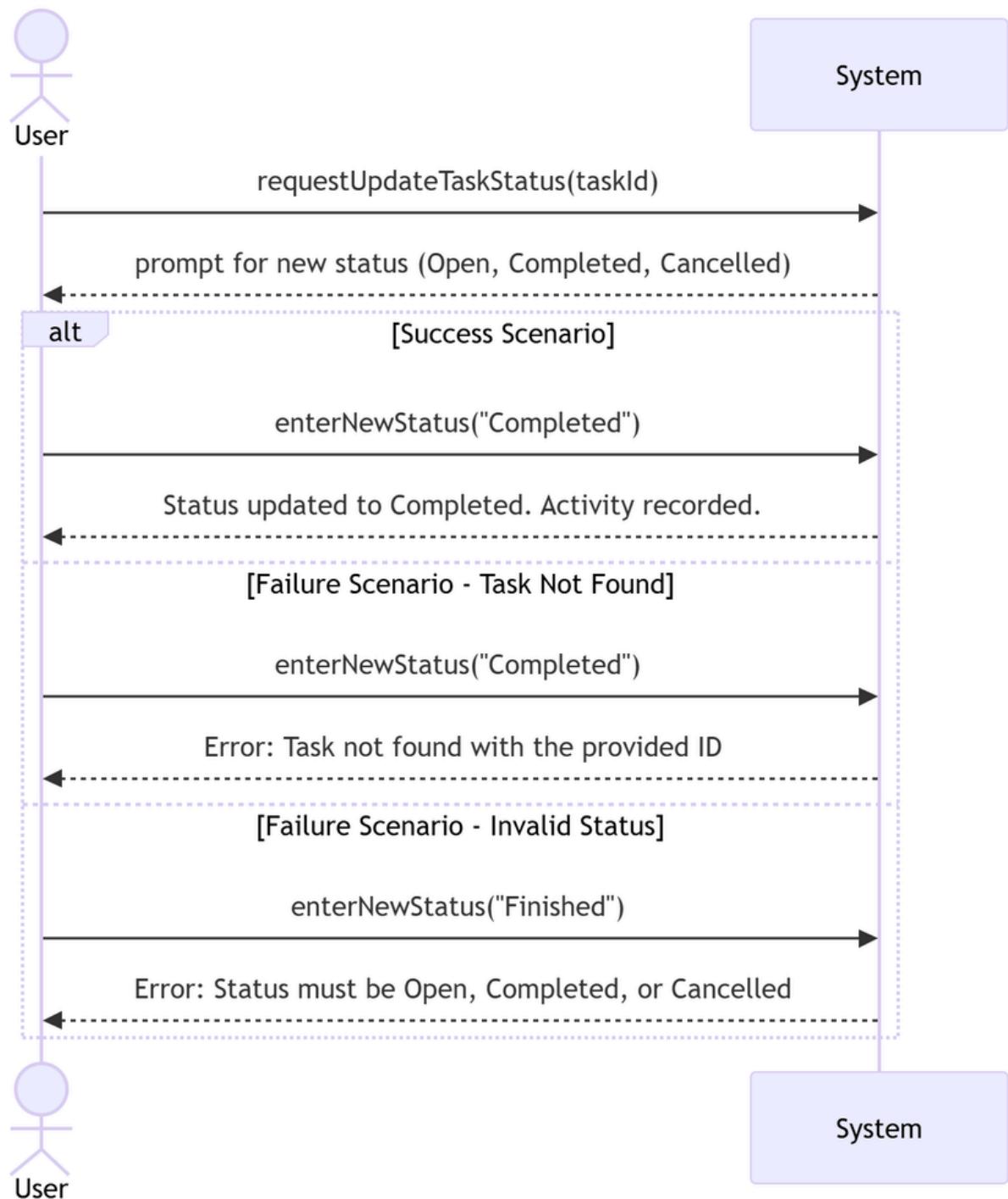
1) Create task SSD :



2) ADD SUBTASK :



3) UPDATE TASK :



4) FULLY DRESSED SCENARIO :

Primary Actor: User

GOAL :

User: Wants to quickly and accurately record a new task with its details so it can be tracked, organized, and managed over time.

Preconditions:

The system is running.
The user is interacting with the command-line interface.

Success Guarantee (Postconditions):

- A new Task instance is successfully created and stored in the system.
- The task has a valid title, an Open status, and the current date as its creationDate.
- An ActivityEntry is recorded reflecting the task's creation with a timestamp.

Main Success Scenario:

1. The User requests to create a new task.
2. The System prompts the User to enter task details (title, optional description, priority, optional due date).
 3. The User enters the task details (e.g., title="Buy groceries", priority="High").
4. The System validates the provided input (ensures the title is not empty, dates are valid, and priority matches PriorityLevel enum: Low, Medium, or High).
5. The System creates a new Task with the provided details, sets its creationDate to the current date, and sets the status to Open.
6. The System creates an ActivityEntry with a timestamp and a short description ("Task created").
7. The System displays a success message confirming the task creation and prints the created task details.

Extensions (Alternative Flows):

a. The User cancels the operation at any prompt:

1. The System cancels the task creation process.
2. The System returns to the main menu without saving any data.

3a. The User provides an empty title:

1. The System displays an error message stating that the title is required.
2. The System prompts the User to enter the title again.
3. The User enters a valid title.
4. The System continues from step 4.

3b. The User provides an invalid priority level:

1. The System displays an error message stating that the priority must be one of: Low, Medium, or High.
2. The System prompts the User to enter the priority again.
3. The User enters a valid priority.
4. The System continues from step 4.

3c. The User provides an invalid due date format:

1. The System displays an error message explaining the correct date format.
2. The System prompts the User to enter the due date again.
3. The User enters a valid due date (or skips by providing an empty input).
4. The System continues from step 4.