**1. Create a Task**

**Use Case Name**: Create a Task

**Actor**: User

**Description**: This use case allows the user to create a new task in the to-do app.

**Trigger**: User decides to add a new task.

**Preconditions**: User is logged into the app.

**Postconditions**: The new task is added to the user's to-do list.

**Main Flow of Events**:

User selects the "Add Task" option in the app.

App prompts the user to enter task details such as task name, description, due date, and priority.

User enters the task details.

User confirms and saves the task.

App validates the input and adds the task to the user's to-do list.

App displays a confirmation message to the user.

**Alternative Flows**:

If the user cancels the task creation, the use case ends.

**2. View Task Details**

**Use Case Name**: View Task Details

**Actor**: User

**Description**: This use case allows the user to view detailed information about a specific task.

**Trigger**: User selects a task from the to-do list.

**Preconditions**: User is logged into the app and has tasks in the to-do list.

**Postconditions**: User views the detailed information of the selected task.

**Main Flow of Events**:

User selects a task from the to-do list.

App displays detailed information about the selected task, including task name, description, due date, and priority.

**Alternative Flows**:

If the selected task is deleted by another use case, display an error message and return to the task list.

**3. Mark Task as Complete**

**Use Case Name**: Mark Task as Complete

**Actor**: User

**Description**: This use case allows the user to mark a task as completed in the to-do app.

**Trigger**: User completes a task.

**Preconditions**: User is logged into the app and has tasks in the to-do list.

**Postconditions**: The selected task is marked as completed in the user's to-do list.

**Main Flow of Events**:

User selects a task from the to-do list.

User marks the task as completed.

App updates the task status to "Completed" and moves it to the completed tasks section.

App displays a confirmation message to the user.

**Alternative Flows**:

If the user accidentally marks the task as completed, there should be an option to undo the action and revert the task to its previous status.

**4. Edit Task Details**

**Use Case Name**: Edit Task Details

**Actor**: User

**Description**: This use case allows the user to edit the details of an existing task.

**Trigger**: User decides to edit a task.

**Preconditions**: User is logged into the app and has tasks in the to-do list.

**Postconditions**: The task details are updated in the user's to-do list.

**Main Flow of Events**:

User selects a task from the to-do list to edit.

App displays the current details of the selected task.

User modifies the task details (name, description, due date, or priority).

User saves the changes.

App validates the input and updates the task details in the to-do list.

App displays a confirmation message to the user.

**Alternative Flows**:

If the user cancels the editing process, the original task details remain unchanged.

**5. Delete Task**

**Use Case Name**: Delete Task

**Actor**: User

**Description**: This use case allows the user to delete a task from the to-do app.

**Trigger**: User decides to delete a task.

**Preconditions**: User is logged into the app and has tasks in the to-do list.

**Postconditions**: The selected task is removed from the user's to-do list.

**Main Flow of Events**:

User selects a task from the to-do list to delete.

User confirms the deletion.

App removes the selected task from the to-do list.

App displays a confirmation message to the user.

**Alternative Flows**:

If the user cancels the deletion, the task remains in the to-do list.