“To-do list” app requirements

1. Have you collected some requirements?

Functional Requirements:

1 User Authentication: Users must be able to create and manage their accounts securely.

2 Task Creation: Users should be able to create, edit and delete tasks.

3 Task Prioritization: Users should be able to prioritize their tasks by assigning them different levels of importance.

4 Task Categorization: Users should be able to categorize their tasks to help them stay organized.

5 Task Reminders: Users should be able to set reminders for their tasks.

6 Task Filtering: Users should be able to filter their tasks based on various criteria such as priority, due date, category, etc.

7 Collaboration: Users should be able to share their tasks or task lists with others.

8 Notifications: Users should receive notifications for their tasks, reminders, and other important events.

Non-Functional Requirements:

1 Performance: The app should be able to handle a large number of users and tasks without slowing down.

2 Security: The app should ensure the security of user data and prevent unauthorized access.

3 Usability: The app should be easy to use and navigate.

4 Reliability: The app should be reliable and not lose data or crash frequently.

5 Availability: The app should be available to users at all times, without downtime or server errors.

6 Scalability: The app should be designed to scale up or down based on user demand.

7 Compatibility: The app should be compatible with different operating systems and devices.

2. Did you map the requirements to written use-cases? Feel free to choose epics/user stories.

Epic 1: User Authentication

As a user, I want to be able to create account and register via email or SSO, so that I could log-in and track tasks via several devices

As a user, I want to modify my account and add personal information (Name, Surname, phone number), so that provided information was up-to-date

As a user, I want to have an option to log-off from all devices, so be secure in non-authorized sign-ins

Epic 2: Task creation, editing and prioritization

As a user, I want to create, edit and delete tasks (task category, priority [from 1 to 5], name, description and timeline), so that I could keep tasks status up-to-date

As a user, I want to attach PDF or IMG files to the tasks via “attach” button and “ctrl C” command, so that I could easily add additional info for the tasks

As a user, I want to format font type, font size, font color and bold type of task name and description

As a user, I want to be able to delete task so that I could keep only relevant ones

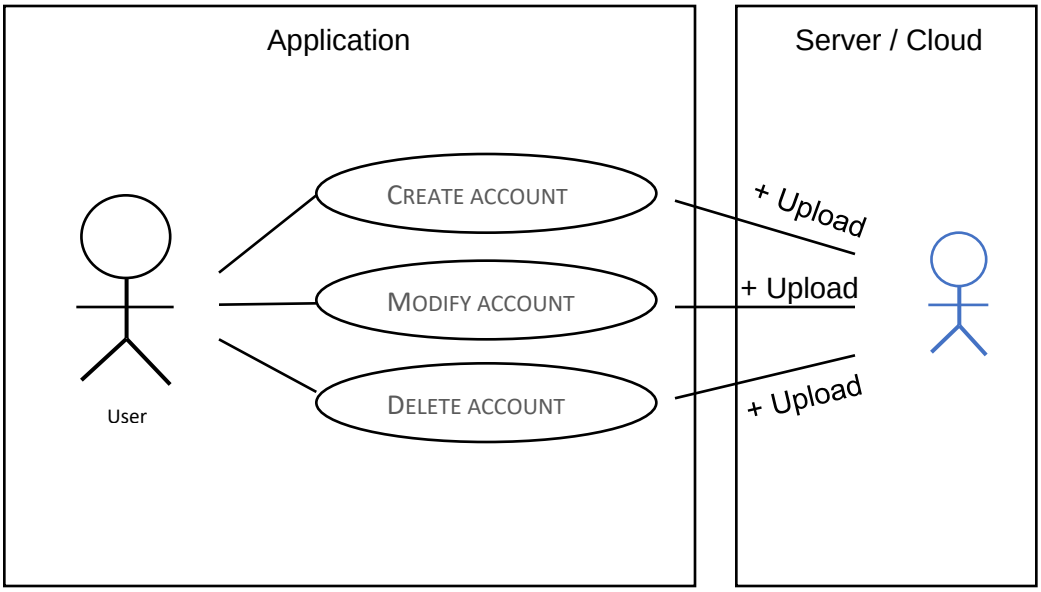
As a user, I want to filter and organize tasks by priority, due date, category so that I could track them in organized way

Epic 3: Notifications and collaboration

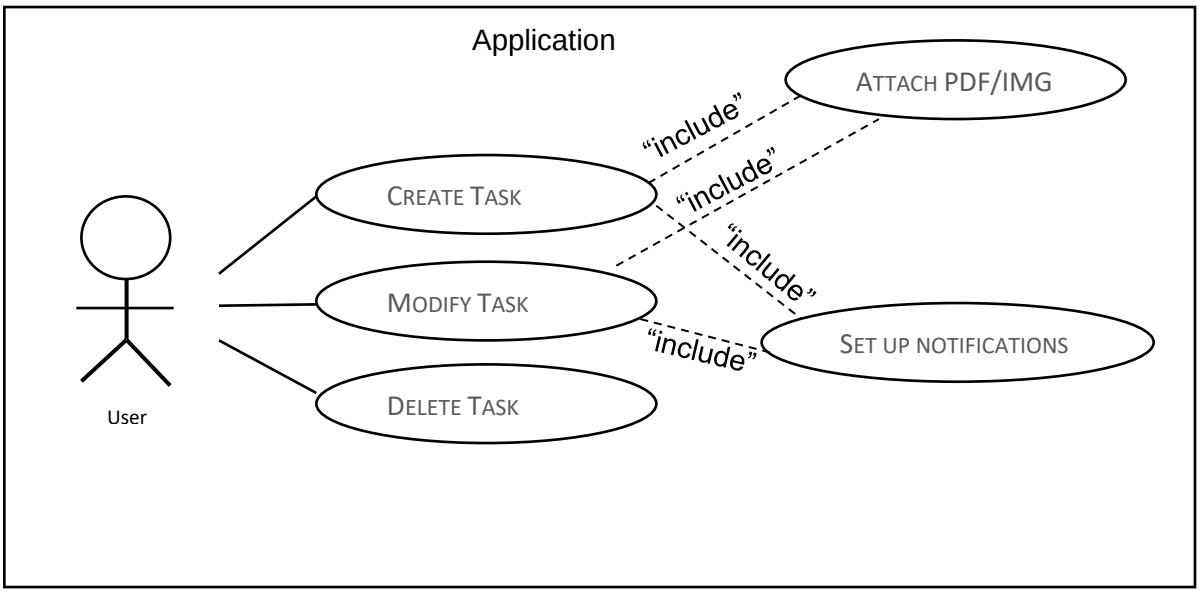
As a user, I want to set up notifications to email and choose on how many days before due date I want to receive reminder (e.g. “Notify if Task due date is in 1-30 Days, could be set as multiply selection), so that I could be warned and be on-track

As a user, I want to share tasks and task list with other in a form of pdf, JPG or via app itself so that I could collaborate with other people

1. User should be able to create, modify or delete Account

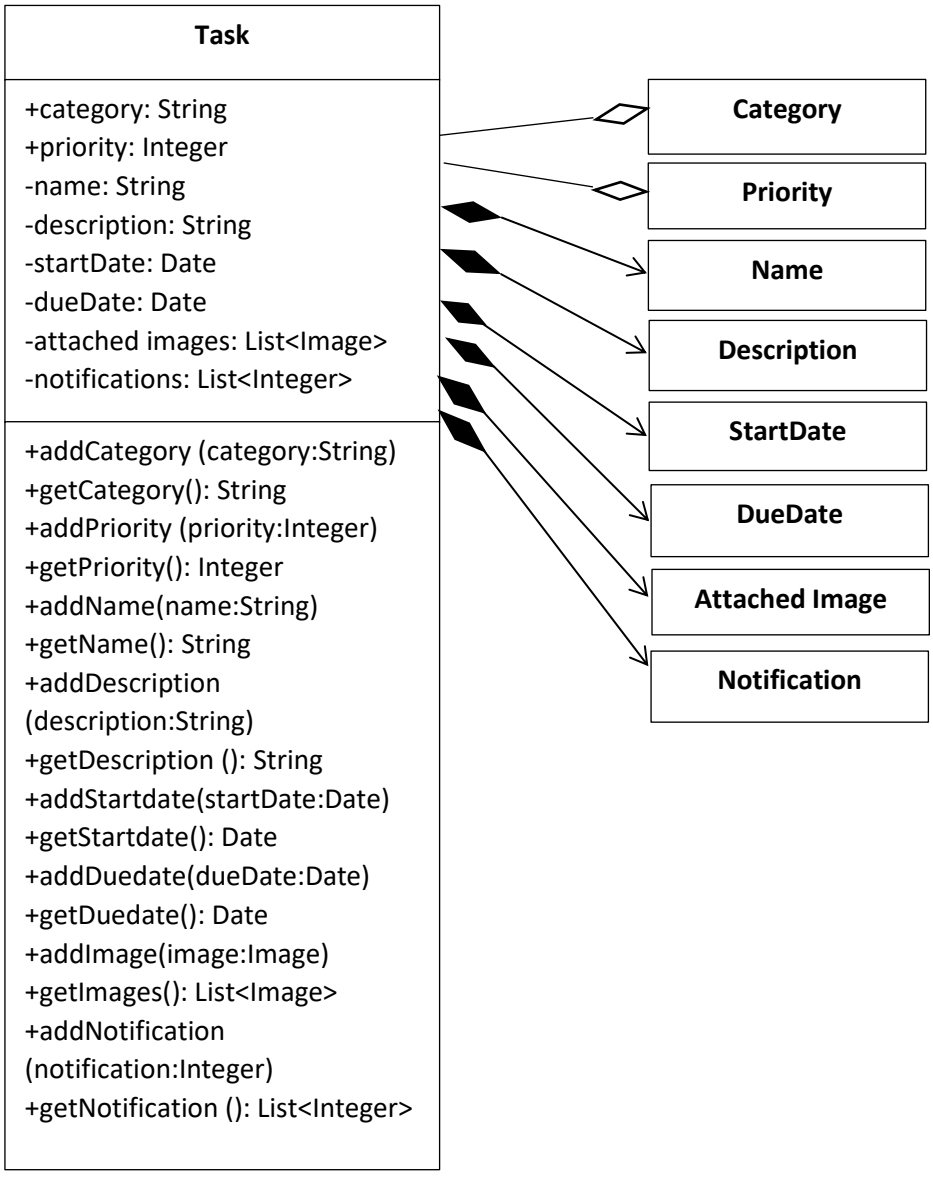


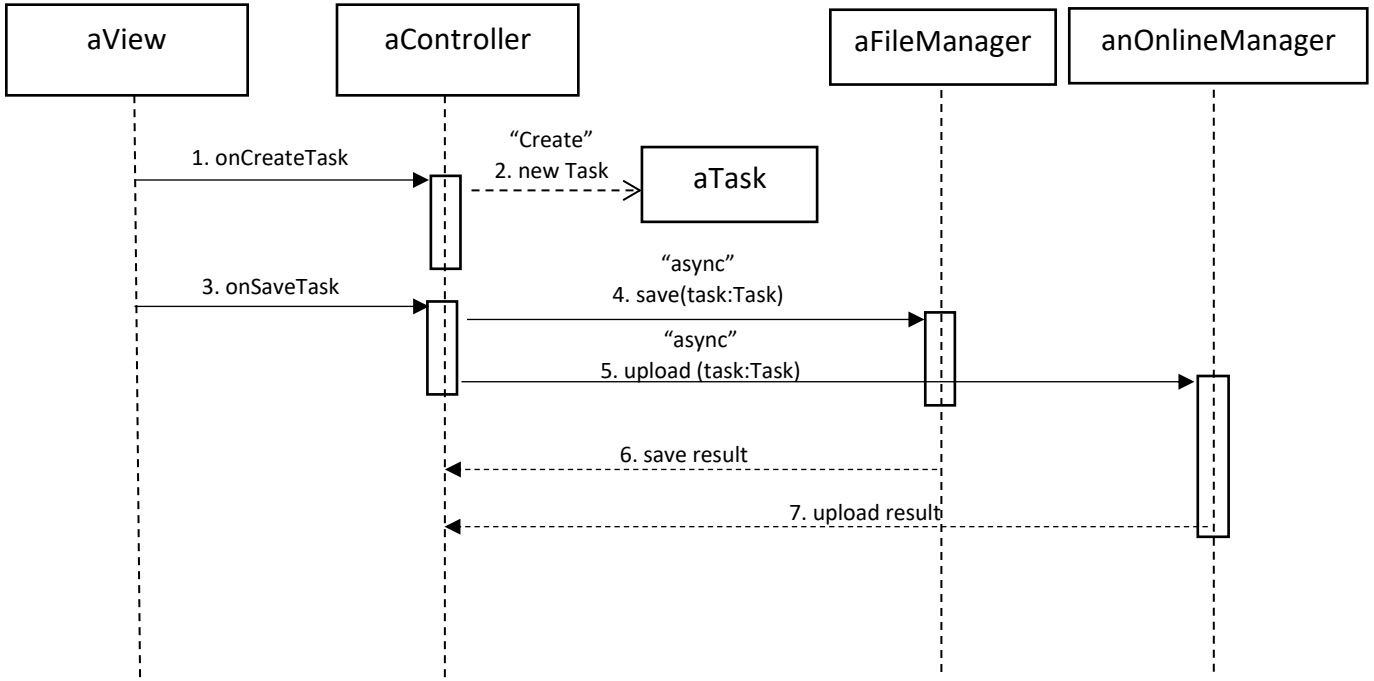
2. User should be able to create, modify or delete Tasks, attach PDF or IMG files to them and set up notifications



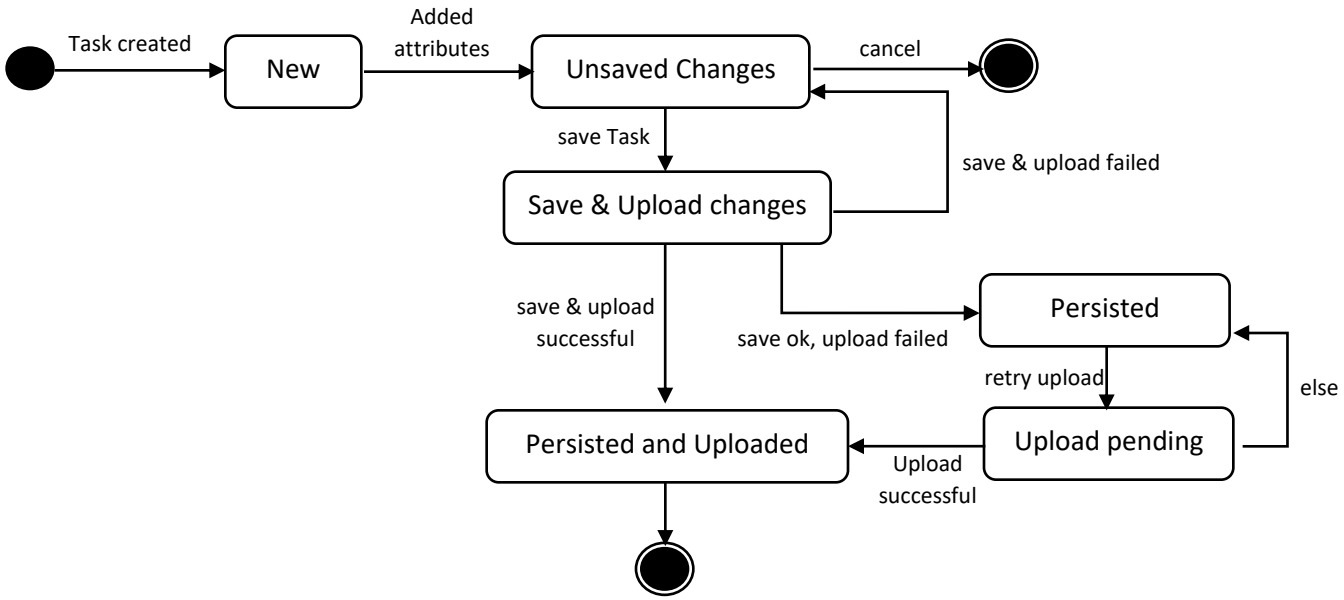
Draw your class diagrams and figure out the relations between your classes.

*Notification can be set as “Remind in X (type:Integer) days before due date”





The user can initiate the Task creation by pressing a button, so we need a view instance that represents the button and controller object, that receives the trigger event. Controller triggers the creation of a new Task instance. Next, the user pressing the save button that will trigger two actions: saving to the local persistence and uploading Task to the cloud storage. The saving to local persistence and uploading the Task to the cloud return instantly, without blocking the caller. Eventually, they return a signal either success or failure.


Task created
New
Added attributes
Unsaved Changes
cancel
save Task
save & upload failed
Save & Upload changes
save & upload successful
save ok, upload failed
Persisted
retry upload
else
Persisted and Uploaded
Upload pending
Upload successful