Requirements

1) Collecting Requirements

- Create and edit text-based notes.
- Attach photos.
- Capture hand-drawn sketches.
- Allo protected/sensitive notes.
- Sync app data to Dropbox, iCloud, or Google Drive

2) Functional Requirements

- We need to build a note-taking app. Users can create and edit text-base notes.
- A note may also include images or hand-drawn Sketches
- Sensitive notes can be protected from prying eyes using a password.
- The app automatically uploads changes to pre-configured servers.
- The app should support all major platforms: Dropbox, iCloud, or Google Drive

2) NonFunctional Requirements

- Release our app for Android first. We'll support Android 9 and newer versions.
- The app needs to run on the Android Phone & Tablet
- We'll create a dedicated support website and include the link in the app description and its "About" page.

User Storys

Note Creation and Editing (epic)

- As a user, I want to create and edit notes so that I can quickly jot down my thoughts.
- As a user, I want to attach photos to a note so that I can keep my memories in one place.
- As a user, I want to add handwritten sketches so that I can insert funny toons into my notes
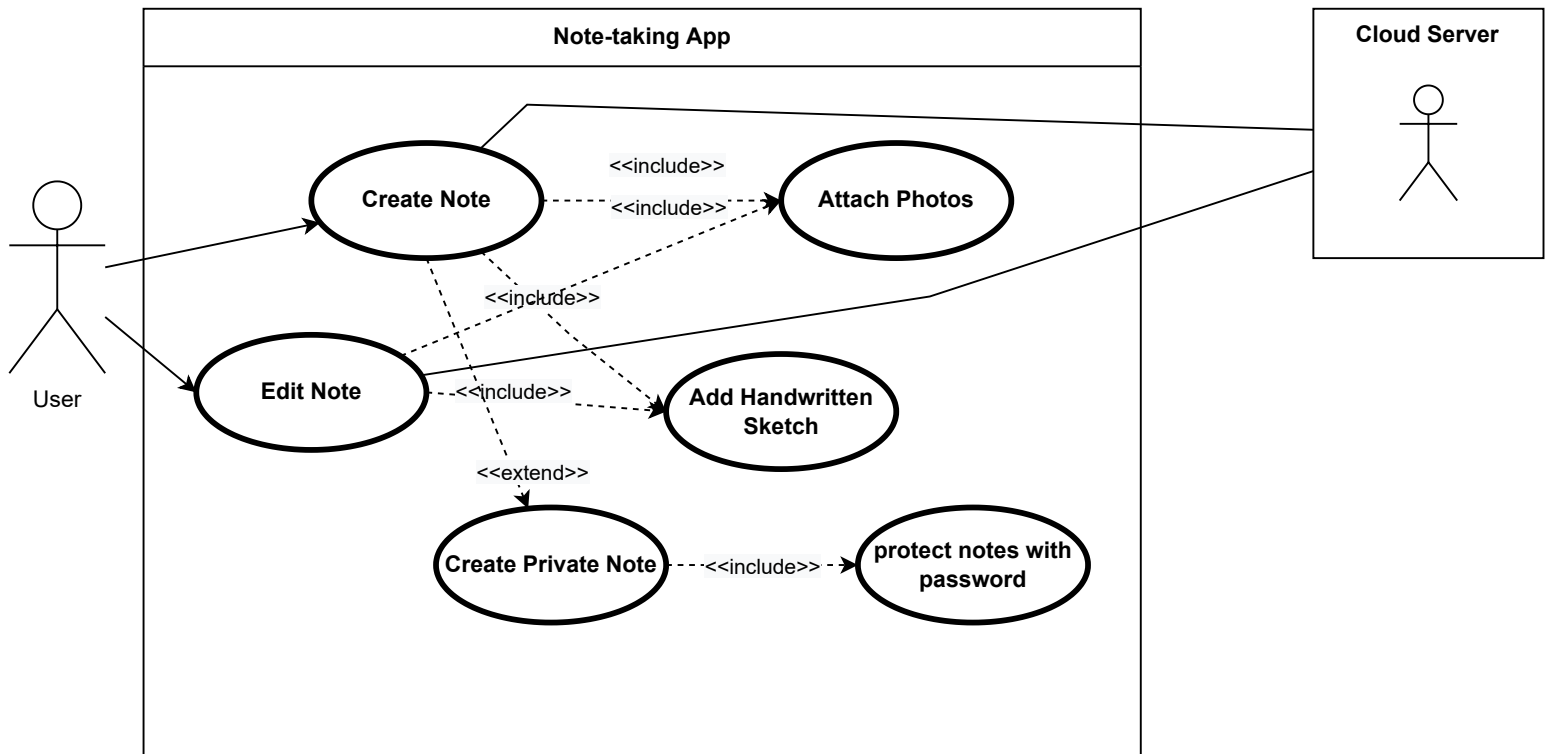
Privacy (epic)

- As a user, I want to create private notes so that only I can access them.
- As a user, I want to protect my sensitive notes with a password.

Syncing to could servers (epic)

- As a user, I want to sync my notes across my Android devices so that my data is up-to-date on all of them.
- As a user, I want my notes automatically uploaded to cloud servers so that I have a backup of all my data
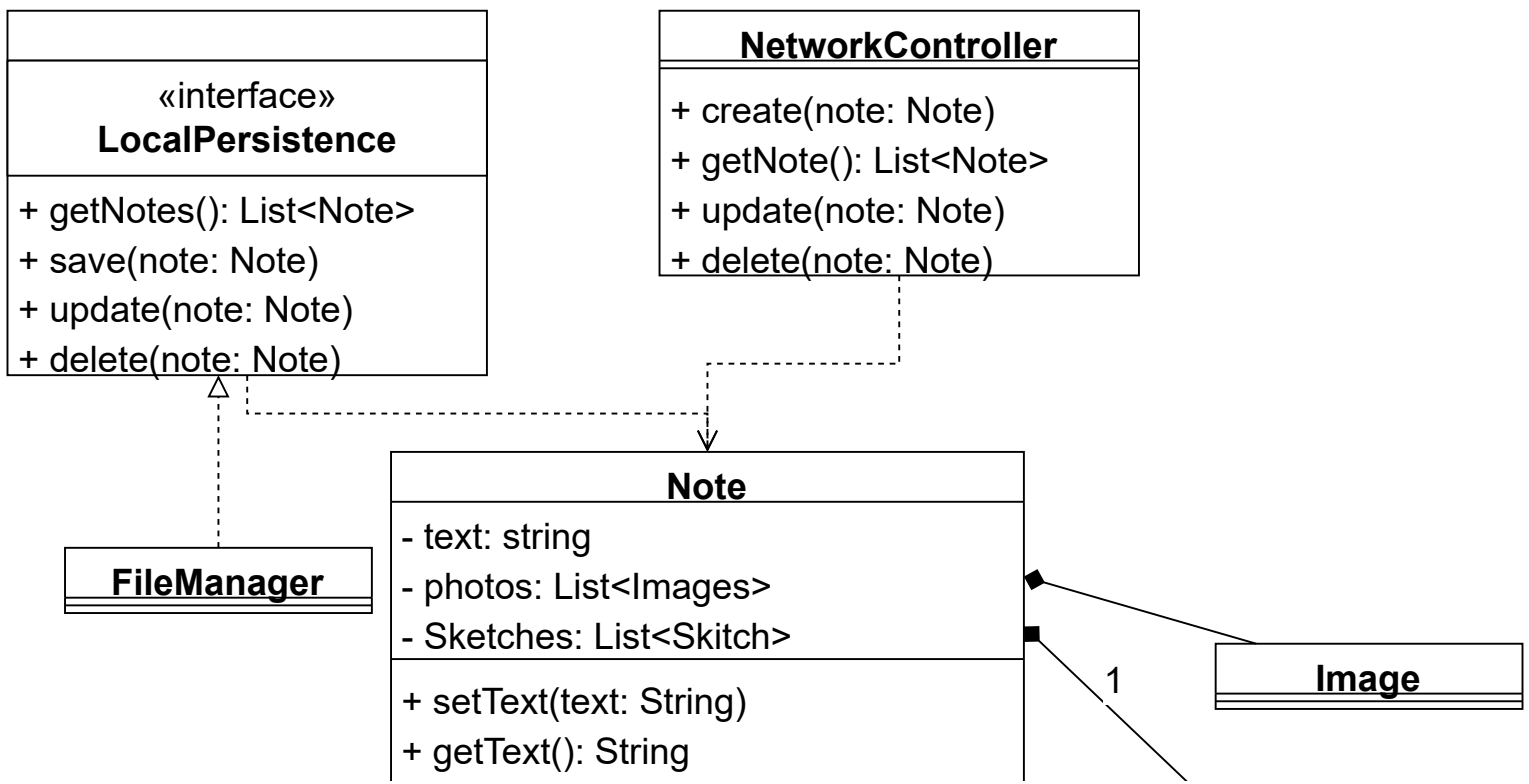
----------------------- Note Diagrams -------------------------

Use-Cases Diagram

Note-taking App

Create Note

Attach Photos

<<include>>

<<include>>

<<include>>

Edit Note

<<include>>

Add Handwritten Sketch

<<extend>>

Create Private Note

<<include>>

protect notes with password

User

Cloud Server

Keep your user case diagrams clean and simple !

Class Diagram

«interface»
LocalPersistence

+ getNotes(): List<Note>
+ save(note: Note)
+ update(note: Note)
+ delete(note: Note)

NetworkController

+ create(note: Note)
+ getNote(): List<Note>
+ update(note: Note)
+ delete(note: Note)

FileManager

Note

- text: string
- photos: List<Images>
- Sketches: List<Skitch>

+ setText(text: String)
+ getText(): String

Image

1

+ addImage(image: image)

+ getImages(): List<Image>

+ addSketch(sketch: Sketch)

+ getAllSketches(): List<Sketch>

Sketch

0...*

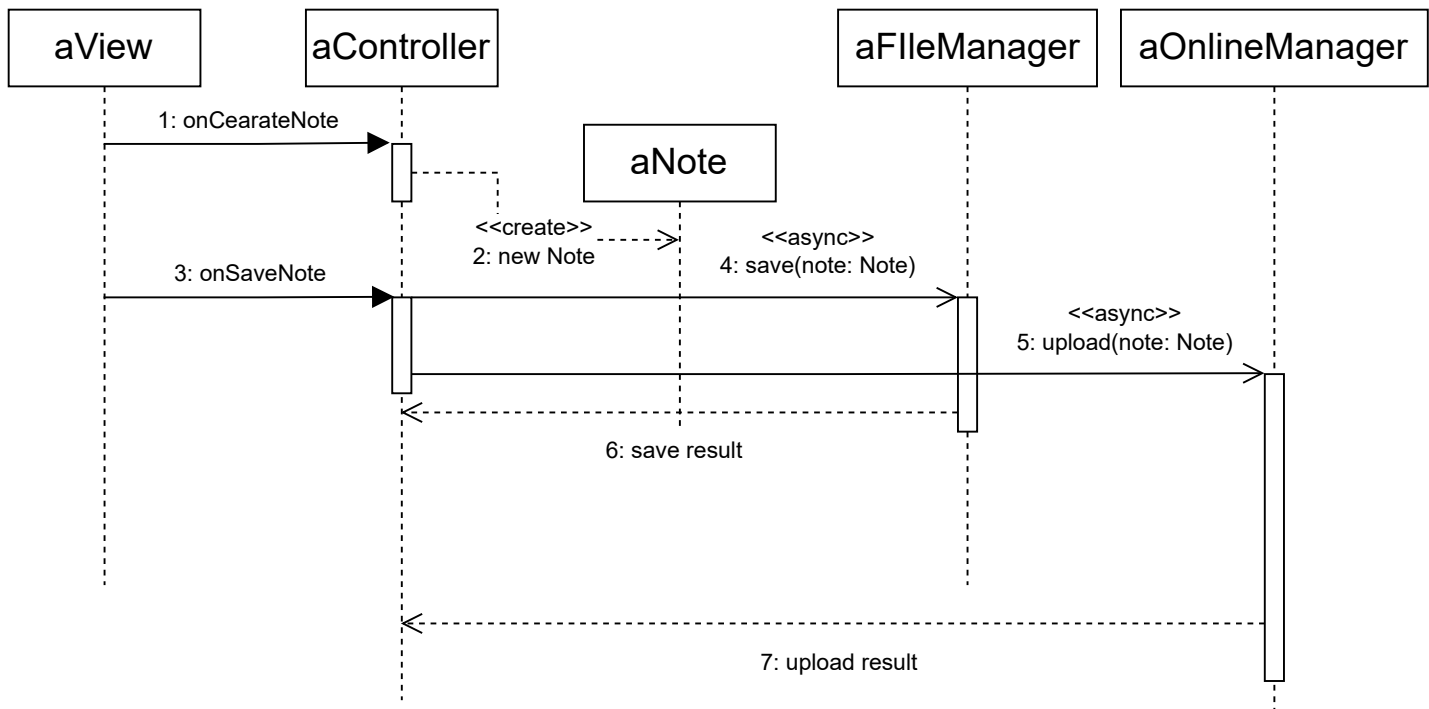| **PrivateNote** |
|---|
| -passwordHash: String |

Crypto
+hash(input: string): String

Sequence Diagram

aView	aController	aFIleManager	aOnlineManager

1: onCearateNote

aNote

<<create>>
2: new Note

<<async>>
4: save(note: Note)

3: onSaveNote

<<async>>
5: upload(note: Note)

6: save result

7: upload result

Statechart Diagram

Archived

app terminated

app started

note created

New

details added

Unsaved Changes

cancle

save note

Save & Upload Pending

save and upload failed

save ok, upload failed

Persisted

save and upload successful

retry upload

upload successful

Persisted & Uploaded

Upload Pending

else