FaceCap Assignment

Technical choices: Why did you choose this framework/approach and what did you build first and why?

I chose React Native Expo because of multiple reasons. When I built my first mobile app (not FaceCap), I had the freedom to choose my own framework. After a bit of research, I picked React Native since it's one of the most popular and stable libraries out there for mobile app development. All major apps (Instagram, Amazon Shopping, Coinbase, Discord) are built on this library.
Moreover, I have a lot of experience building web applications using React. So learning React Native was quite easy for me.
Finally, Expo is the officially recommend framework by React Native. It comes with great development tooling such as the Expo Go app which allows to test apps without generating native builds. Also, it comes with cross-platform APIs (Camera, File System, etc.) which is a huge productivity boost. Finally, if I need to build a production app, Expo Application Service features signing apps and pushing over the air updates for both AppStore and PlayStation.

What would you improve with more time or implement beyond?

I implemented all the features mentioned in the assignment, including some extra features like selecting different video duration and toggling between front/back camera. So within the scope of this assignment, I can't think of a whole lot of other features to implement. Maybe, a cancel recording button can be implemented during the recording, which I attempted to but implementation required a lot more effort than expected because of how the API works. Therefore I decided to not implement it.

But if the question is to make this into a production level application, then I need to do research on what users are looking to get out of a video diary application. Just off the top of my head, I can think of features like user authentication, date filter, adding tags/descriptions to videos, being able to backup the videos to a drive, and a lot more. Also, this is just a frontend client application. A backend server application needs to be developed for some of the features mentioned above.

Also when it comes to production apps, a lot more needs to be done, such as testing, monitoring, logging, security, scalability, etc.

Challenges: What problems did you face and how did you solve them?

The only issue I faced is a race condition issue when trying to implement the visual countdown timer during recording the video. Expo camera API provides an async function to start recording which makes it difficult to start the visual countdown timer at the same time, leading to

inconsistency between the countdown shown on the screen and the actual video time. I improved the problem by using states and useEffect(). The component state is set to “recording” and then invokes the recordAsync(). There is an useEffect() that runs when the state is updated. Within that useEffect(), a setInterval is started which will update the timer on the screen. However, since the setInterval is called after recordAsync() is called (on the next re-render after state is changed), the app has to wait to update the timer when the JS event-loop is available. This causes a slight delay between the timer and the recording. This can be solved in the future by using native camera modules.

Time Breakdown

I didn’t clock the hours for everything since it’s unnatural to do that. Instead I write what features were implemented each day, which is more realistic. You can consider each day to be 8 hours of work.

Setting up development environment	2 hours
Wireframe development	1 hour
Implement basic UI: Header, Flatlist, and BottomSheet	Day 1
Implement Camera Module	Day 2
Implement Loading/Saving/Deleting videos and Visual Countdown Timer	Day 3
Implement video playback, thumbnails,and finishing UI touches	Day 4

Use of AI

I almost never use AI to write code since it builds up technical debt as the development progresses. It gets harder to debug and develop if I’m not aware and knowledgeable of each and every line of code.

The only exceptions are to write helper utility functions, such as time formatter, and to write tailwind classes for UI design.

My main use of AI is to summarize API documentation such as Expo API modules used in this project. Without the help of AI, it would take me days to read and understand the whole documentation and find the parts I need to know.

Also, I use AI to ask for best practices and design patterns.