

Attendi Microphone Mobile Assessment

Duration: 3 hours

- **Main work:** 2.5 hours
- **Presentation & discussion:** 30 minutes

Deliverable: Code and explanation

Objective

Implement a simplified version of the Attendi Microphone implemented for either the Android platform or the iOS platform, using Jetpack Compose or SwiftUI respectively. You can choose which platform to implement.

The Attendi Microphone is a component that can be used by our partners to easily record audio and upload it to our servers for transcription.

Functional requirements

Microphone Button

A circular button that serves as the main interaction point for recording audio. **For this exercise, we will not actually record audio to keep it simple.**

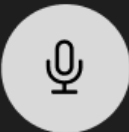
Recording States & Button Behavior

The microphone button should cycle through the following states:

Current State	On Button Tap	Next State
Not Recording	Start recording	Recording
Recording	Pause recording	Paused
Paused	Resume recording	Recording
Paused	Press "Send" button	Processing
Processing		Not Recording

UI States

The expected UI states are shown in the following image:

State	Button 1	Button 2	Timer
Not recording			
Recording			00:00
Paused			00:02
Processing			00:02

Timer Display

When recording starts, a timer should be displayed on screen in **mm:ss** format (e.g., **00:00**, **01:23**). The timer should:

- Start at **00:00** when recording begins
- Update every second while recording
- Pause when the recording is paused
- Resume counting when recording resumes

In the "Not Recording" state, the timer should not be visible.

Send Button

When in the **Paused** state, a **Send** button should appear on screen. When pressed, it transitions to the "Processing" state and initiates a mock upload request.

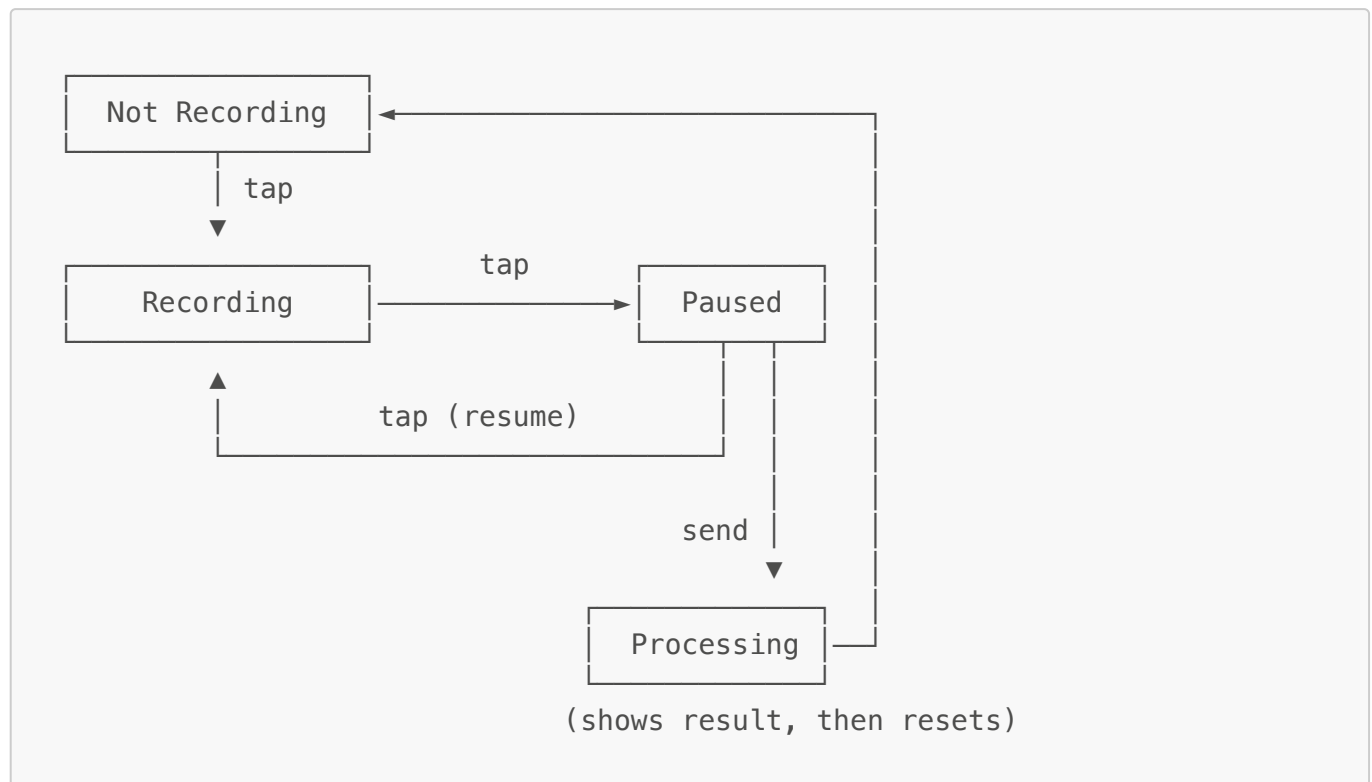
Processing & Result

When **Send** is pressed:

1. The UI should show a "Processing" state
2. A mock request should be made (simulate with a short delay)
3. Once complete, display some dummy transcription text on screen and return to the "Not Recording" state

This can only happen from the **Paused** state.

State Diagram



What we expect from you

1. **Thought process:** Explain your approach and decisions
2. **Clean, idiomatic code:** Follow platform conventions and best practices
3. **Working implementation:** The component should function as described
4. **State management:** Proper handling of the various states and transitions
5. **Demonstrable bonus (optional):** Animations, custom styling, or other enhancements

Good luck – we look forward to your approach!