LH_ARCH_PUBLISHADUIO

Class Diagram

Objective: Define classes, attributes, methods, and relationships for the audio publishing system.

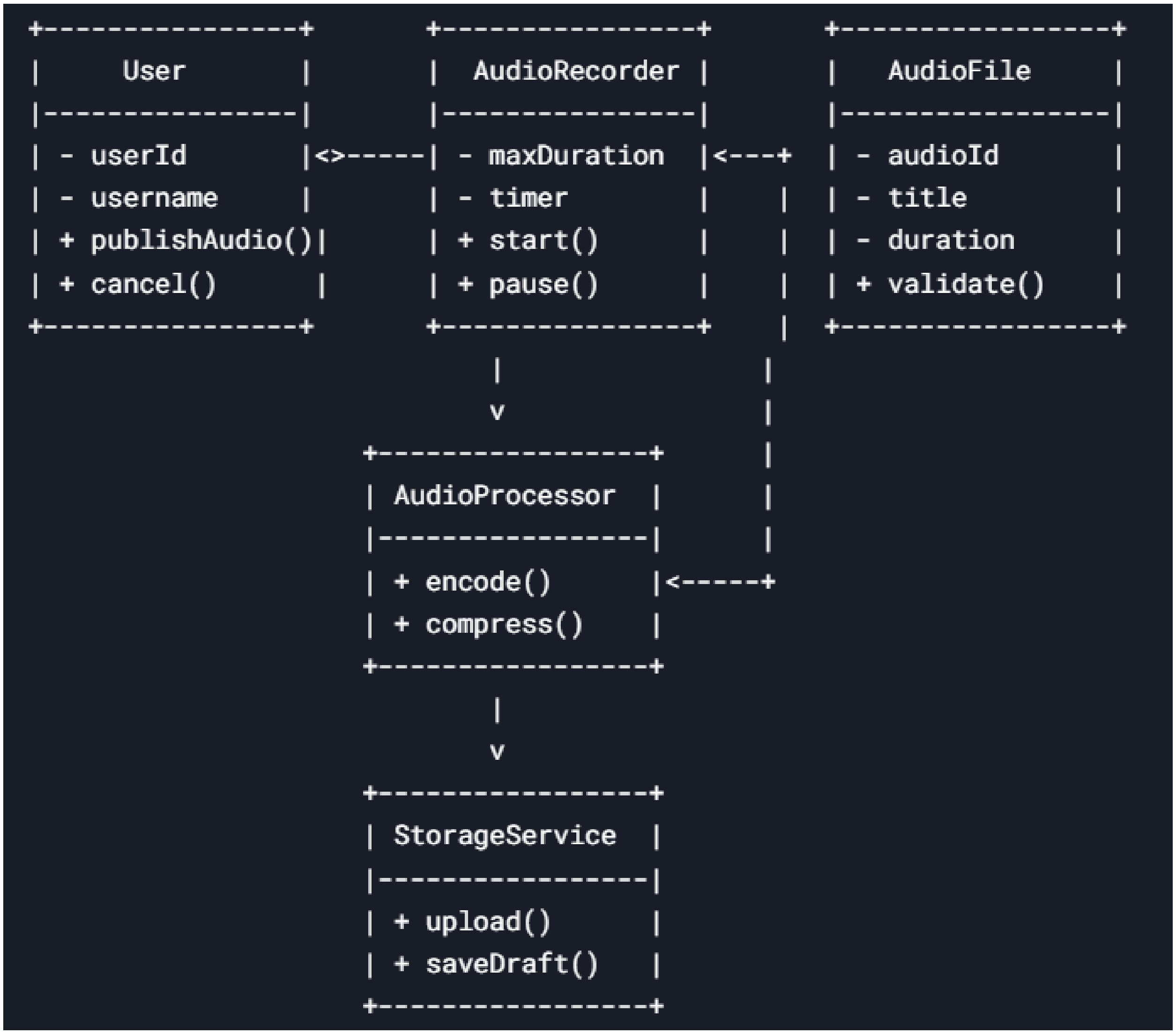
Key Classes:

1. User:
   - Attributes: userId, username, role (e.g., admin, standard user).
   - Methods: publishAudio(), cancelRecording().
2. AudioFile:
   - Attributes: audioId, title, duration, format, status (draft/published).
   - Methods: validateDuration(), convertFormat().
3. AudioRecorder:
   - Attributes: recordingTimer, maxDuration (5:00).
   - Methods: startRecording(), pauseRecording(), stopRecording().
4. AudioProcessor:
   - Methods: encode(), compress(), checkFormatCompatibility().
5. StorageService:
   - Methods: uploadToCloud(), saveDraft().
6. NotificationService:
   - Methods: sendConfirmation(), showError().

Class Relationships:

- User *uses* AudioRecorder and AudioFile.
- AudioRecorder *depends on* AudioProcessor for encoding.
- StorageService *stores* AudioFile.

Visual Example:



```
+----------------+       +----------------+       +-----------------+
|     User       |       |  AudioRecorder |       |   AudioFile     |
|----------------|       |----------------|       |-----------------|
| - userId       |<>-----| - maxDuration  |<---+  | - audioId       |
| - username     |       | - timer        |    |  | - title         |
| + publishAudio()|      | + start()      |    |  | - duration      |
| + cancel()      |      | + pause()      |    |  | + validate()    |
+----------------+       +----------------+    |  +-----------------+
                             |                |
                             v                |
                     +-----------------+      |
                     | AudioProcessor  |      |
                     |-----------------|      |
                     | + encode()      |<-----+
                     | + compress()    |
                     +-----------------+
                             |
                             v
                     +-----------------+
                     | StorageService  |
                     |-----------------|
                     | + upload()      |
                     | + saveDraft()   |
                     +-----------------+
```

FIG1.0 Class Diagram

High-Level Design (HLD) Schema

Objective: Illustrate the system architecture and component interactions.

Components:

1. **Frontend (UI):**
    - **User interacts with the interface to:**
        - **Input audio title.**
        - **Record/pause audio.**
        - **Publish or cancel.**
2. **Audio Processor:**
    - **Encodes and compresses audio files.**
    - **Validates duration (5-minute limit).**
3. **Storage:**
    - **Saves processed audio files (e.g., local storage or cloud).**
4. **Database:**
    - **Stores metadata (title, user ID, duration, format).**
5. **Notification:**
    - **Show success/error messages (e.g., "Published!" or "Recording too long").**

Interaction Flow:

Frontend → Audio Processor → Storage → Database

↓

Notification

Visual Example:

- **Frontend** directly communicates with **Audio Processor** (no API layer).
- **Storage** and **Database** are separate for scalability.
- **Notifications** are triggered by validation results (e.g., errors).

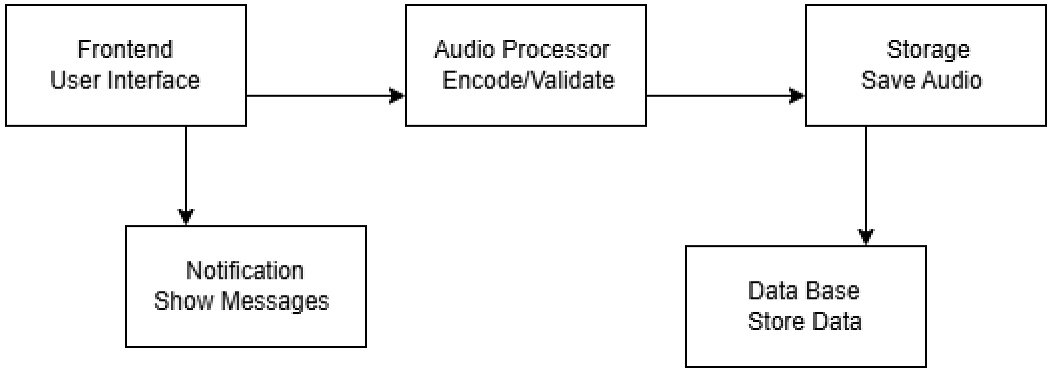


FIG1.1 High Level Diagram

Low-Level Design (LLD) Flowchart

Objective: Map the step-by-step workflow for publishing audio.

Steps:

1. **Start Recording**:
   - User clicks "Record" → Validate permissions → Start timer.
2. **Validate Duration**:
   - If recording exceeds 5:00 → Show error: *"Recording exceeds 5 minutes"*.
3. **Process Audio**:
   - Encode to MP3 → Compress → Save draft (optional).
4. **Publish**:
   - Click "Publish" → Confirm via modal → Upload to cloud → Update database → Notify user.

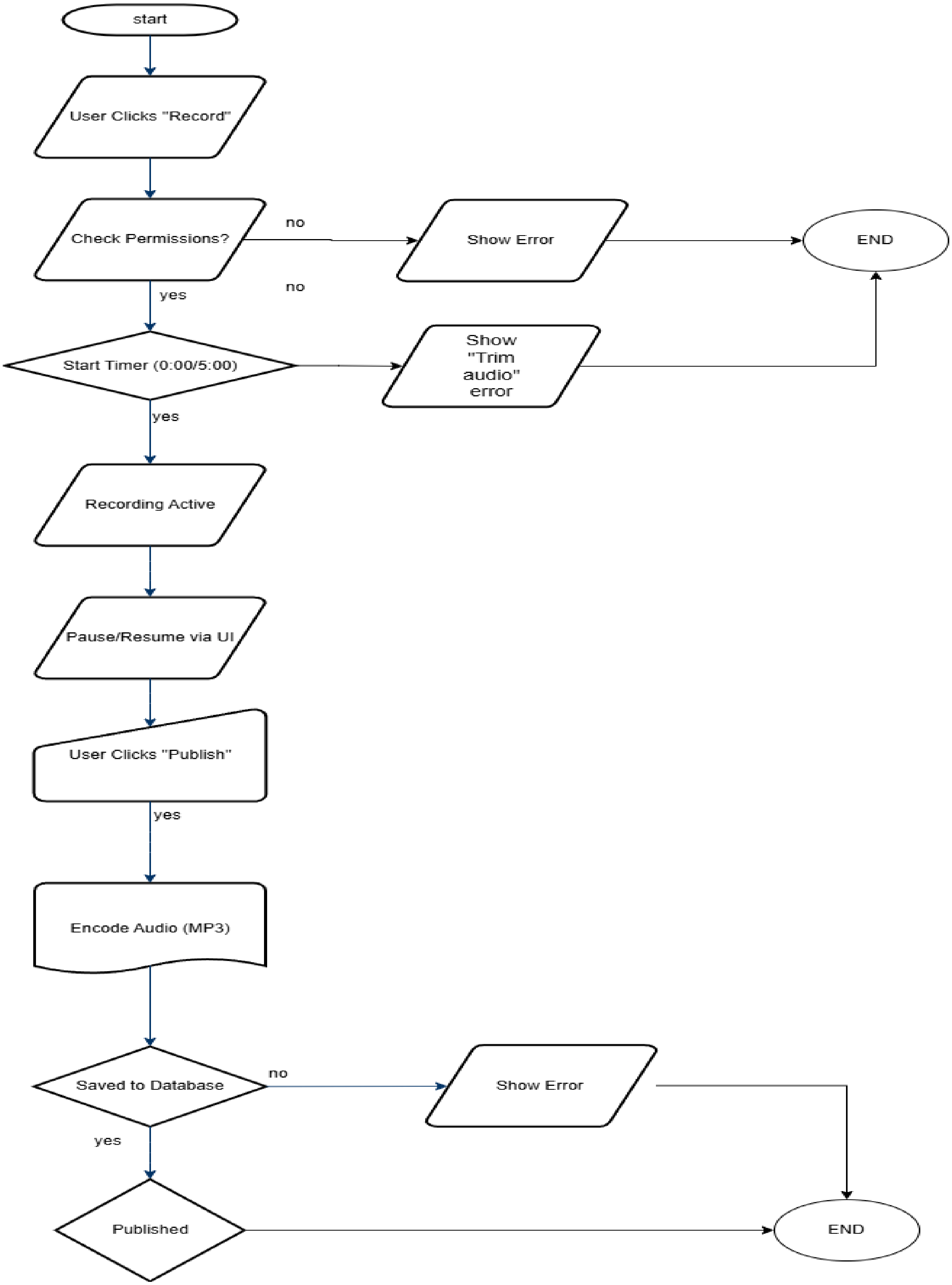


FIG1.2 Low Level Diagram