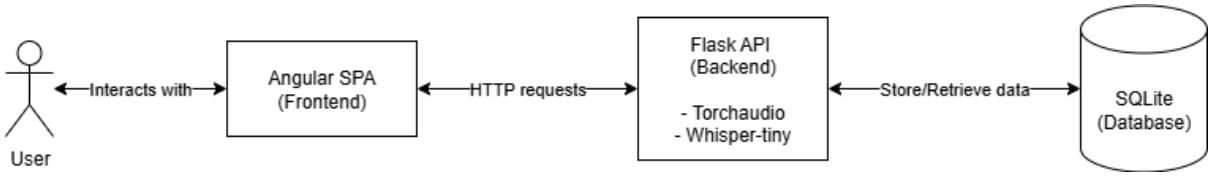

Component Breakdown

1. **User**: Interacts with the application.
2. **Angular SPA (Frontend):** Client-side app handling user interactions and communicating with backend via HTTP.
3. **Flask API (Backend):** Server-side app handling HTTP requests, managing database interactions, and processing audio using:
    - **Torchaudio:** Preprocesses audio (resamples to 16kHz, converts to mono-channel WAV).
    - **Whisper-tiny:** Performs speech-to-text transcription.
4. **SQLite (Database):** Stores and retrieves transcription metadata (filename, transcribed text, creation date).

Assumptions

1. Security measures like authentication and authorization are not critical for this phase.
2. Users will upload non-malicious audio files securely.
3. SQLite3's single-file database is suitable for this small-scale project.
4. Synchronous processing is sufficient.
5. The Whisper Model will provide adequate transcription accuracy.

Considerations

1. **Whisper Model Limitations:** Limited to 50 requests per minute.
2. **SQLite Database Suitability:** Suitable for small applications with basic CRUD operations, but not for high concurrency.
3. **Flask:** Ideal for prototyping but lacks native async support.
4. **Angular Framework:** Chosen for its structured, maintainable architecture and scalability benefits. May be unnecessary but was intentionally selected to demonstrate knowledge to build structured, maintainable architecture.