

EchoText

Software Requirements Specification Version 1.03

Team: Ahmad Ghaddar, Kevin Xing, Harsh Bhagat, Andy Huang

Revision History

Date	Description	Author	Comments
9/21/24	Version 1.00	Andy Huang	First Revision
10/2/24	Version 1.01	Andy Huang	Modified and reorder “3.2 Functional Requirements”, convert “3.3 Non-Functional Requirements” to tables, added images to “3.1.1 User Interfaces”, modified “2.5 Assumptions and Dependencies”, modified “3.4 Design Constraints”
10/15/24	Version 1.02	Andy Huang	Redid all Functional requirements to also include Processing, Error Handling, Forward/Backward Dependency. Moved Reference to the end. Updated Database section.
12/3/24	Version 1.03	Andy Huang	Updated UI section and minor changes to reflect current app development.

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Kevin Xing	Team Lead Meeting Minutes	
	Ahmad Ghaddar	Presentation Lead Frontend Lead UI/UX Lead	
	Harsh Bhagat	GitHub Lead Backend and Data Management Lead	
	Andy Haung	Documentation Lead Testing/QA Lead	

Table of Contents

REVISION HISTORY	I
DOCUMENT APPROVAL	I
1. INTRODUCTION.....	1
1.1 PURPOSE.....	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	2
1.4 OVERVIEW	3
2. GENERAL DESCRIPTION	3
2.1 PRODUCT PERSPECTIVE.....	3
2.2 PRODUCT FUNCTIONS.....	4
2.3 USER CHARACTERISTICS.....	5
2.4 GENERAL CONSTRAINTS	6
2.5 ASSUMPTIONS AND DEPENDENCIES	6
3. SPECIFIC REQUIREMENTS	7
3.1 EXTERNAL INTERFACE REQUIREMENTS	7
3.1.1 User Interfaces	7
3.1.1 Top Toolbar	12
3.1.2 List of Saved Documents.....	12
3.1.3 Import.....	12
3.1.4 Document selection.....	13
3.1.5 Page Selection.....	13
3.1.6 Context Menu	13
3.4 HISTORY VIEW	14
3.1.1 History Counter	14
3.1.2 Clear All	14
3.1.3 Copy and Delete	14
3.5.1 Main Settings Buttons	15
3.5.2 Speech Settings.....	15
3.5.3 Features in the Theme & Display Section.....	16
3.5.3 Theme & Display.....	16
3.5.4 Features in the Theme & Display Section.....	16
3.5.5 Security.....	16
3.5.6 Features in the Security Section.....	17
3.1.2 Hardware Interfaces	18
3.1.3 Software Interfaces.....	18
3.1.4 Communications Interfaces	19
3.2 FUNCTIONAL REQUIREMENTS.....	19
3.3 NON-FUNCTIONAL REQUIREMENTS	36
3.3.1 Performance	36
3.3.2 Reliability	37
3.3.3 Availability.....	38
3.3.4 Security.....	38
3.3.5 Maintainability.....	39
3.4 DESIGN CONSTRAINTS	40
3.5 LOGICAL DATABASE REQUIREMENTS	41
4. ANALYSIS MODELS	42
4.1 DATA FLOW DIAGRAMS (DFD)	42

A. APPENDICES.....	43
A.1 APPENDIX 1.....	43
A.2 APPENDIX 2.....	43
B. REFERENCES	44

1. Introduction

The Software Requirements Specification (SRS) document provides a detailed outline of the functional and non-functional requirements for the development of an on-device, transformer-based Text-to-Speech (TTS) application for iOS. The aim is to ensure that all technical and functional aspects of the system are clearly specified for software engineers, designers, and testers to develop, maintain, and test the product effectively.

This document serves as a comprehensive reference guide for the entire project lifecycle. This SRS addresses specific requirements, including TTS inference, audio management, user interface features, privacy considerations, and system constraints. The primary focus is on creating a locally run application using Sherpa TTS, optimized for iPhone XR and higher devices. It ensures the app operates independently of cloud-based services, offering privacy-focused and offline TTS functionality to users.

1.1 Purpose

The purpose of this SRS document is to specify the requirements for developing an iOS application that provides an offline, transformer-based TTS. This document is intended for software developers, testers, and project stakeholders to ensure clarity and alignment in the project's design, implementation, and testing phases.

The SRS focuses on:

- Defining the functionality the app will deliver, including TTS inference, audio file management, and customizable settings.
- Outlining constraints such as performance limitations and platform dependencies.
- Detailing non-functional requirements like privacy, security, and performance standards.
- By adhering to the requirements in this document, the development team will create an efficient, reliable, and user-friendly application that serves as a private, locally run alternative to cloud-based TTS solutions.

1.2 Scope

The EchoText application is a self-contained TTS system designed for iOS devices. Its primary function is to convert user provided text into natural-sounding speech without relying on external servers or cloud-based services. The application will function entirely offline, with all the text and audio processing and storage on the device.

The key features of EchoText include:

- Support multiple text input methods, including typing, copy-pasting, and importing documents (PDF). Text input will be limited to English only.

- Real-time conversion of text-to-speech using a transformer-based TTS model, ensuring low-latency output. Generated speech will be limited to English only.
- User customization options for voice, pitch, and speed.
- The ability to export audio files via iOS sharing options.
- The ability to manage the audio and text files via the application.
- Optional password protection for accessing the application, ensuring potentially sensitive documents and generated audio remain secure.
- The application is running fully offline, with all the processing happening on-device, with no ability to connect to the internet or communicate with outside servers or services.

The EchoText application is optimized for iPhones running iOS 17 and later, with hardware requirements starting from iPhone XR to ensure real-time performance. While EchoText primarily targets iOS users in the United States and other English-speaking countries, it incorporates global best practices for data privacy and security, referencing GDPR principles for secure handling of user data.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
User	In the context of this document, a user is an individual that interacts with the application, accessing features and functionalities.
TTS	Text-to-speech, a form of speech synthesis used to create a spoken version of the text.
ONNX	Open Neural Network Exchange, an open-source AI framework that allows models to be transferred between different machine learning frameworks.
Model	In the context of machine learning, a model is a mathematical representation trained on data, used to make decisions based on input data.
API	Application Programming Interface, a set of protocols and tools used for building software applications and facilitating interaction between different software systems.
iOS	Apple's mobile operating system, used for iPhones and iPads.
VITS	Variational Inference Text-to-speech, a type of transformer-based TTS model used to generate speech from text.
GDPR	General Data Protection Regulation, a legal framework that sets guidelines for the collection and processing of personal information from individuals who live in the European Union (EU). GDPR is widely recognized as a comprehensive international standard for data privacy.
SQLite	A lightweight, self-containing SQL database engine.
Sherpa-ONNX	An open-source, cross-platform framework used for model inference.

Swift	A programming language developed by Apple, used to interact with the OS and hardware of Apple products.
SwiftUI	A user interface toolkit provided by Apple to design and build iOS and other Apple OS applications.
.wav	A standard audio file format that stores sound data.
Transformer-based	A deep learning architecture designed for processing sequences of data, especially in natural language processing tasks.
iCloud	A cloud storage and services platform provided by Apple, enabling data synchronization, backup, and access across devices for Apple users.
Open source	Software for which the original source code is made freely available and may be redistributed and modified.
GUI	A graphical user interface (GUI) is a visual method of interacting with a computer/device.

1.4 Overview

This document outlines the software requirements for the EchoText application. It is divided into four major sections. The Introduction provides context and background for the project, explaining its purpose and scope. The General Description section offers an overview of the system, its interactions with users, and relevant assumptions or constraints. The Specific Requirements section details the functional and non-functional requirements that define the application's behavior and quality standards. Finally, the Analysis Model section presents diagrams and models that illustrate the flow of data and the system's internal processes. Together, these sections provide a comprehensive guide for requirement specification of the EchoText application.

2. General Description

This section will provide an overview of the EchoText application, explaining how the system operates and how it interacts with its user. It will talk about the target userbase on which this application is designed for. It will also outline the key constraints and assumptions, such as hardware and operating system requirements and general functional requirements it has.

2.1 Product Perspective

This TTS system is designed as a self-contained application running entirely on an iOS device. This system will make use of the sherpa-onnx framework and the onnx runtime to convert user provided text into natural sounding speech.

The core feature of the application is the ability to process various types of text input methods and transform and synthesize speech locally on the device. Users will provide multiple text input methods, including directly typing it, copy-pasting, and importing

document (PDF). This flexibility allows the user the freedom to convert the text into the format they want.

Since the system and application will run purely on device and offline after the initial install, all data processing, text parsing, and speech synthesis will occur locally on device. This ensures that any document or generated voices are private, and any sensitive documents will not go through any cloud or external servers.

In contrast to other TTS applications that depend on cloud services for voice synthesis, this application is designed to provide real-time feedback and reliable voice output in any environment, regardless of network availability. This makes it particularly suitable for users who prioritize privacy, data security, or have limited access to consistent internet service.

2.2 Product Functions

The EchoText application provides several key functions that facilitate the seamless conversion of text into speech. These functions enable users to input text, customize the output, manage audio playback, and export audio files while maintaining privacy and performance through on-device processing.

The first core function of EchoText is text input and processing. Users can input text by typing, copy-pasting, or importing text files such as PDF documents. The application processes and formats this text to ensure compatibility with the TTS engine. EchoText parses the imported text automatically, converting the text into speech.

The second function is text-to-speech conversion, which forms the core of the application. Using transformer-based TTS models, such as VITS, the app converts text into natural-sounding, human-like speech. This function is performed locally on the device, providing quick and secure processing without the need for an internet connection.

EchoText also offers voice customization, allowing users to adjust the speech output by modifying parameters such as speed, and pitch. Different voice styles, including male, female, and neutral voices, are available for users to choose from, enabling them to tailor the listening experience to their preferences.

In terms of playback controls, the application includes standard audio controls such as play, pause, and stop, which enable users to listen to the converted speech directly within the app. Users can navigate through the audio by seeking forward or rewinding, allowing for easy playback of longer texts.

EchoText includes a feature for exporting audio, allowing users to save the generated speech as an audio file in the format of WAV. These files can be saved locally on the device or shared through other applications, providing flexibility for later access.

Another important function is document and file management, where users can browse, select, and open text files and audio files stored on their device. EchoText will support the PDF file format.

The application performs all its operations through on-device processing, ensuring that text-to-speech conversions and voice customizations occur locally, maintaining privacy and security. This design also reduces latency, providing real-time feedback without relying on external servers. The application is optimized for offline use allowing the user to listen to any text on the go without worrying about needing an internet connection after the initial installation.

EchoText also includes settings and preferences functionality, where users can configure their preferred voice settings, playback speed, and default audio export format. These settings allow users to personalize their experience and tailor the application to meet their specific needs.

2.3 User Characteristics

EchoText is designed to cater to users who require the ability to convert text into speech on their iOS devices. The general characteristics of the eventual users that affect the specific requirements are as follows:

- **General Users:** Individuals who prefer listening to text content rather than reading it. This includes users who are multitasking, commuting, exercising, or in situations where reading is not convenient.
- **Privacy-Conscious Users:** Users who are concerned about data privacy and security. They prefer applications that process data locally without sending information to external servers, ensuring their personal or sensitive documents remain confidential.
- **Users with Limited Internet Access:** Individuals who require offline functionality due to unreliable, expensive, or unavailable internet connection. They need an application that functions fully without a network connection.
- **Accessibility Needs:** Users with visual impairments, dyslexia, or other reading difficulties who rely on text-to-speech technology to access written content. The application serves as an assistive tool to enhance their ability to consume text-based information.

These user characteristics influence the application's specific requirements by emphasizing:

- **Ease of Use:** A simple and intuitive user interface that accommodates users with different levels of technical expertise.
- **Privacy and Security:** On-device processing to ensure user data remains confidential and is not transmitted to external servers.
- **Offline Functionality:** The ability to operate without internet connectivity to cater to users in various environments.

Understanding these characteristics is crucial for tailoring the application's features and user interface to meet the needs and expectations of its intended audience. The application must balance advanced functionalities with user-friendliness to appeal to users seeking efficient and private text-to-speech solutions on their iOS devices.

2.4 General Constraints

The General Constraints section outlines limitations that could affect the design, development, and operation of the on-device transformer-based TTS application.

Hardware Constraints:

- The iOS devices targeted by this project have specific hardware constraints, such as limited CPU power and RAM compared to desktop environments. Transformer-based models can be resource-intensive, which may impact real-time performance on lower-end devices. Therefore, the application will be limited to iPhone XR and later models.

Operating System Constraints:

- The application is being developed for iOS, which imposes strict guidelines on app design, privacy, and resource usage. Adhering to Apple's Human Interface Guidelines and App Store Review Guidelines will constrain certain aspects of app functionality and design.
- The app will be limited to iOS 17 for compatibility with features and optimizations specific to that version.

Data Privacy:

- Since the application handles user input (e.g., text-to-speech conversions), it must comply with privacy regulations such as GDPR. The app will encrypt user data stored on the device and will not send any data to external servers.

Real-Time Processing:

- The application is expected to provide near real-time TTS conversion. Careful management of latency constraints is essential to optimize the models for faster inference, ensuring the application meets performance requirements without suggesting it is meant for live conversation.

2.5 Assumptions and Dependencies

The Assumptions and Dependencies section lists the factors that the development of this application is dependent on, and assumptions that have been made about the system and environment.

Assumptions:

- It is assumed that developers working on EchoText will require both an iOS device (e.g., iPhone or iPad) for testing and a macOS device (e.g., MacBook or iMac) for development, as iOS applications can only be developed and deployed through Xcode, which is exclusive to macOS.

Dependencies:

The development and operation of EchoText rely on several third-party libraries and frameworks:

- **Sherpa-ONNX (1.10.27):**
Sherpa-ONNX is a critical dependency for EchoText, providing the text-to-speech (TTS) model inference framework. It handles the transformation of text into speech using transformer-based models, performing all inference locally on the device. Updates or changes in Sherpa-ONNX could impact the performance and compatibility of EchoText.
- **SwiftUI (6.0):**
SwiftUI is used for building the user interface of the application. As the primary framework for UI design, SwiftUI's functionalities and limitations directly affect how the app interacts with users. EchoText relies on SwiftUI to provide a seamless and responsive user experience across iOS devices.
- **Swift (6.0):**
EchoText is built using the Swift programming language, which interacts with iOS APIs and handles application logic, including model inference with Sherpa-ONNX. The Swift version in use must be compatible with iOS system updates and maintain integration with third-party libraries.
- **SQLite (3.42.1):**
SQLite is used for local data storage, managing metadata such as user preferences, and generated audio files. EchoText depends on SQLite to store and retrieve data efficiently without requiring a network connection. Changes in SQLite compatibility or updates may affect how data is managed within the app.

3. Specific Requirements

This section outlines all the functional and quality requirements of the system, providing a comprehensive description of the system and its features in detail.

3.1 External Interface Requirements

The external interfaces required for the EchoText application to function correctly. It covers user interfaces, hardware interfaces, software interfaces, and communications interfaces. These external interfaces are essential for interaction between the software, users, hardware, and other systems.

3.1.1 User Interfaces

The EchoText application is divided into five primary tabs: Main, Generated, Documents, Text History, and Settings. Each tab offers specific functionalities to provide a smooth and intuitive user experience for text-to-speech conversion and audio management.

3.1.1.1 Main Page

This is the primary interface where you can input text and generate speech.

1. Header

- **Logo and App Name:**
 - Displays the app name ("EchoText") alongside the app logo for easy identification.
- **PDF counter:**
 - During batch document processing, a PDF counter will show on the far-right side that displays the current pdf being generated out of the total within the batch.

2. Text Input Area

- **"Please input your text below":**
 - Instructional text that prompts you to type or paste your desired text for text-to-speech generation.
- **Word Counter:**
 - Located in the top-right corner above the text box. Tracks the number of words entered, with a maximum limit of 10,000 words.
- **Text Input Box:**
 - The large box in the center of the screen where you can type or paste your text. This is the main area for entering content that will be converted into speech.

3. Action Buttons and Progress Bar (Bottom Section)

- **Speaker Icon (Mute/Unmute):**
 - Toggles the audio output of the current text-to-speech generation on or off.
 - Use this to quickly mute or unmute the playback during or after generation.
- **Microphone Icon (Generate/Terminate):**
 - Starts the text-to-speech generation process when pressed.
 - During generation, the microphone icon switches to a red circle with a square inside it, allowing you to terminate the ongoing process if needed.
- **Trash Icon (Clear Text) / Play-Pause:**



- Clears all text from the input area when not generating.
- During text-to-speech generation, this icon changes to a Play/Pause button, allowing you to play and pause the playback of the currently generating audio.
- **Progress Bar:**
 - Appears below the action buttons during text-to-speech generation.
 - Visually indicates the progress of the generation process, helping you monitor completion in real-time.

4. Navigation Bar (Bottom Menu)

- **Main:** The current page. It is marked with a blue home icon to indicate that you are on the main text input interface.
- **Generated:** Access previously generated audio files.
- **Documents:** Import and manage documents for generation or later use.
- **Text History:** Review your recent text inputs.
- **Settings:** Customize your experience and manage your preferences.

3.1.1.2 Generated Page

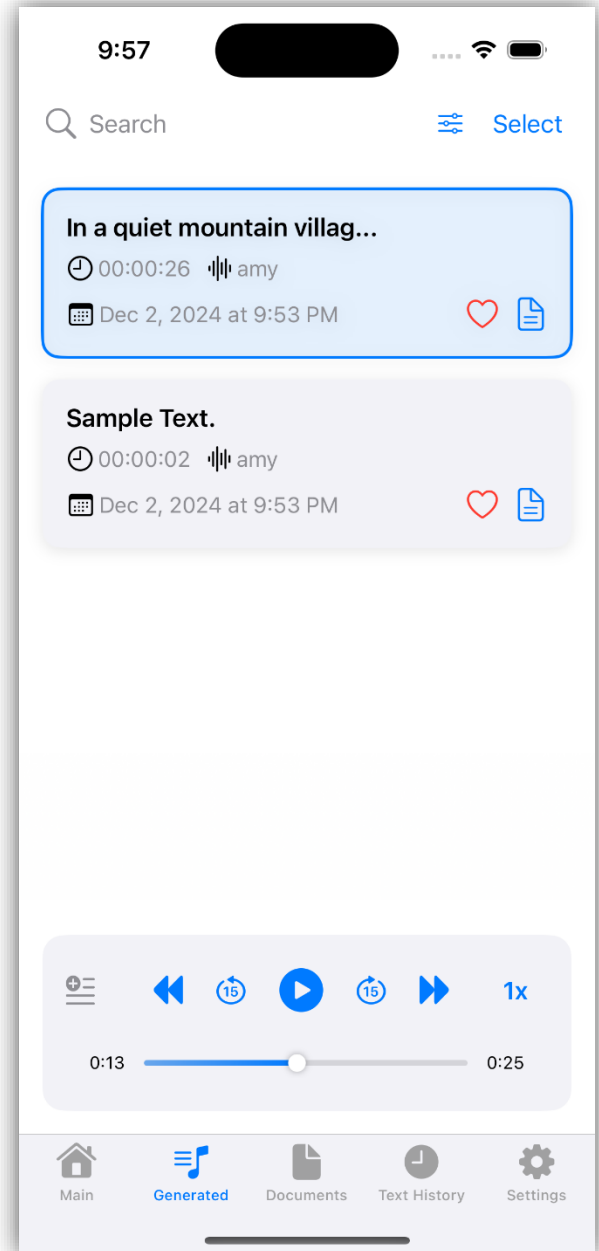
The Generated Page is where all your saved audio files are displayed and managed. This page provides a variety of tools for sorting, playback, and managing your generated audio files. Below is a breakdown of its features and functionality.

1. Top Toolbar

- **Search Icon:** Opens a search bar where you can quickly find specific audio files by name.
- **Filter Icon:** Opens the **Sorting and Filters Sheet**, where you can:
 - Sort by date, length, or name.
 - Filter audio files based on parameters like speed, pitch, length, favorites, includes document, date range and Voice Model.
- **Select Button:** Allows you to select multiple audio files for bulk actions like sharing or deleting.

2. List of Saved Audios

- Audio files are displayed in a **card-like style**, with the following information:
 - **Name:** The title of the audio file.
 - **Length:** Duration of the audio (e.g., 00:26).
 - **Date and Time:** When the audio was generated.
 - **Voice Model:** The name of the voice used for generation (e.g., "Amy").
- Each card includes:
 - **Heart Icon:** Tap to mark the audio as a favorite.
 - **Document Viewer Icon:** Tap to view the text used to generate the audio.



3. Playback Controls

Located below the audio list, the playback controls allow for seamless navigation and playback of your audio files. The controls include:

1. **Queue Icon:** Opens the queue view to manage upcoming audio playback.
2. **Previous Audio:** Skips to the previous audio file in the list.
3. **Rewind 15 Seconds:** Rewinds the playback by 15 seconds.
4. **Play/Pause:** Toggles between playing and pausing the current audio.
5. **Forward 15 Seconds:** Advances the playback by 15 seconds.
6. **Next Audio:** Skips to the next audio file in the list.
7. **Speed Control:** Allows you to adjust the playback speed (e.g., 1x, 1.5x, 2x).

4. Seek Bar

- **Seek Bar:** Displays the progress of the currently playing audio.
- **Current Time:** Shown on the left side of the seek bar, indicating the elapsed time.
- **Total Time:** Shown on the right side of the seek bar, indicating the total duration of the audio.

5. Context Menu

- **Accessing the Menu:** Hold down on any audio file to bring up the context menu.
- **Menu Options:**
 - **Rename:** Edit the name of the audio file.
 - **View Document:** Opens the text used for generating the audio.
 - **Add to Queue:** Adds the audio file to the playback queue.
 - **Share:** Allows you to share the audio file with others.
 - **More Info:** Provides detailed metadata about the audio file.
 - **Delete:** Permanently removes the audio file.

3.1.1.3 Documents Page

The Documents View is designed to help users manage, search, and interact with text-based documents for text-to-speech conversion. This view provides a user-friendly interface for importing, searching, and converting documents.

3.1.1 Top Toolbar

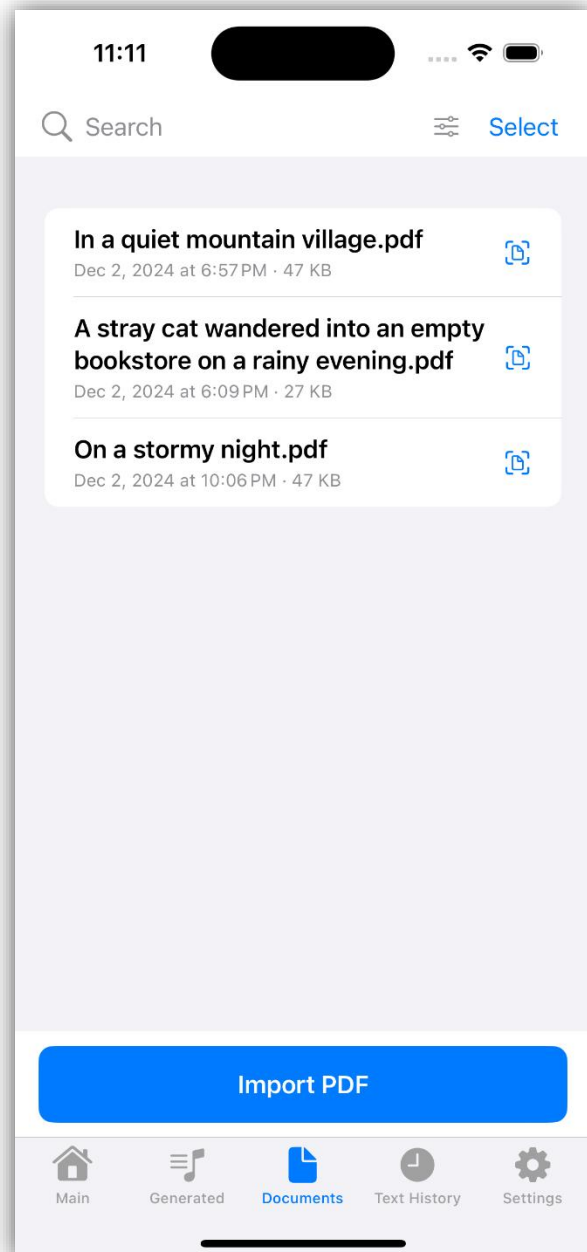
- **Search Icon:** Opens a search bar where you can quickly find specific audio files by name.
- **Filter Icon:** Opens the Sorting and Filters Sheet, where you can:
 - Sort by date, length, or name.
 - Filter audio files based on parameters like speed, pitch, length, favorites, includes document, date range and Voice Model.
- **Select Button:** Allows you to select multiple PDF files for bulk actions like deleting and batch conversion.

3.1.2 List of Saved Documents

- documents files are displayed in a **list-like style**, with the following information:
 - **Name:** The name of the imported PDF.
 - **Date and Time:** The date the PDF was imported.
 - **File Size:** the file size of the PDF.
- **Each PDF includes:**
 - **Document Viewer Icon:** Tap to view icon at the right side to view the PDF that was imported.

3.1.3 Import

At the bottom of the screen, a blue “Import PDF” is where you would press to import PDF documents for conversion. By pressing said button, it will pull up Apple Files app which you can select your desired PDF to import.



3.1.4 Document selection

- **Select all:** Select all documents that are displayed to the user.
- **Done:** to exit out of selected mode.
- **Select document:** the user can press on the individual document to select. A blue icon with a checkmark will be displayed to let the user know that the document has been selected.
- **Delete all Selected:** By pressing on the red trash icon at the bottom of the screen, the user can delete all selected documents. A confirmation popup will show to ensure if the user wants to perform said action.
- **Batch Conversion:** By pressing on the blue wave icon at the bottom of the screen, the user can convert all the documents selected to audio. The app will queue each document to be converted displayed in Main View and the audio file will be saved and displayed in the Generated view.

3.1.5 Page Selection

In the document preview page, the user can press on Select Pages button at the bottom of the app to open up a page selection page for conversion.

- **Select Range:** The user can select a page range to convert said pages to text. There is a dropdown menu next to the “From” text which indicate the start. There is another dropdown menu next to the “To” text which indicates the end. At the bottom, it shows the total number of pages, selected pages, and page range.
- **Select Individual:** The user can select individual pages displayed by a grid view in the middle of the app. The user can click on each box to select which is indicated by a blue box and checkmark. At the bottom show total pages and selected pages.
- **Cancel:** at the top right there is a cancel button to exit out of Select Page view.
- **Generate:** at the top left there is a Generate button to convert the pages to audio.

3.1.6 Context Menu

- **Accessing the Menu:** Press and hold down on any document file to bring up the context menu.
- **Menu Options:**
 - **Generate:** To convert the document to audio.
 - **Rename:** Edit the name of the document file.
 - **Share:** Allows you to share the document file with others.
 - **Delete:** Permanently removes the document file.

3.4 History View

The History View in EchoText provides a user-friendly interface for managing all your previously generated audio files. This feature ensures that users can quickly access, organize, and manage their TTS (Text-to-Speech) history. Whether you need to delete unnecessary files or copy files. History View offers seamless functionality for an optimal user experience. Below is a breakdown of its features and functionality.

3.1.1 History Counter

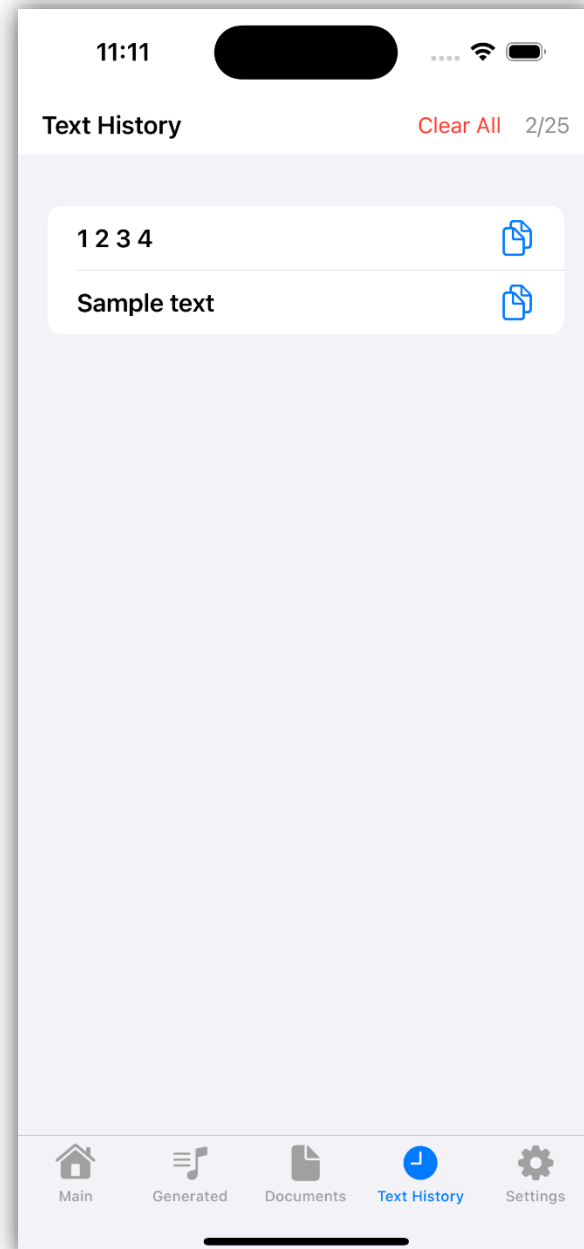
- The counter at the top right corner of the Text History displays the number of files currently in your history. It will save the last 25 generated audio history.

3.1.2 Clear All

- If you want to remove all generated audio files at once, the Clear All button provides a convenient option.
- Tap the Clear All button at the top right of the Text History to delete all files from your history. You will be prompted to confirm this action before the files are removed permanently.

3.1.3 Copy and Delete

- **Copy:** To share or save the audio file elsewhere, you can copy it. Simply tap the copy icon next to the file, and it will be copied to your clipboard, ready to be pasted wherever needed. Alternatively, you can press and hold the file to bring up the copy option.
- **Delete:** To remove a file from your text history you can press and hold the file to bring up the delete option.



3.1.1.5Settings Page

The Settings section in EchoText allows you to customize your experience and manage your preferences. The settings section is divided into five sections for easy navigation.

To access the settings page:

- 1. Open the application.
- 2. Tap the gear icon located in the bottom-right corner of the home screen.

3.5.1 Main Settings Buttons

The main settings page displays the following buttons:

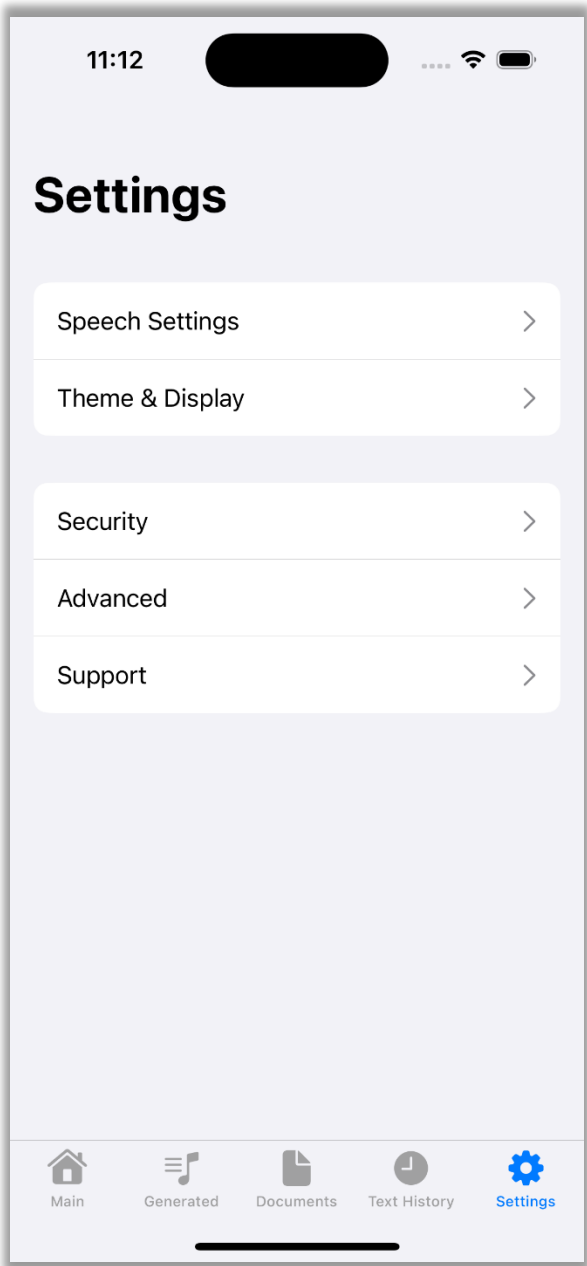
Button	Description
Speech Settings	Adjust text-to-speech options, including voice selection, tone, pitch, and speed.
Theme & Display	Customize the app's appearance, including themes (light/dark) and font settings.
Security	Set up and manage passwords, encryption, and other security features.
Advanced	Access power-user configurations and experimental settings.
Support	Find help, FAQs, and contact information for technical support.

3.5.2 Speech Settings

The Speech Settings section allows you to customize the voice and speaking parameters for text-to-speech (TTS) functionality. It includes options for selecting a voice profile, choosing an inference model, and adjusting playback characteristics such as speed and pitch.

To access the Speech Settings section:

- 1. Open the application.
- 2. Tap the gear icon located in the bottom-right corner of the home screen.



3. Select Speech Settings from the list of options.

3.5.3 Features in the Theme & Display Section

1. Voice Profile:
 - A Button to navigate to the Voice Profile selection and creation screen.
 - Click the “plus” sign on the top left to create and name a new voice profile.
 - Hold an existing voice profile to bring up the options to rename or delete voice profiles.
2. Select Model:
 - A button that brings up a dropdown menu listing available inference models.
3. Speed Slider:
 - A slider to adjust the speaking speed of the TTS engine.
 - Range: 0.5x (slower) to 1.5x (faster), with 1.0x as the default value.
4. Pitch Slider:
 - A button that brings up a dropdown menu listing available inference models.
 - Range: 0.5x (lower pitch) to 2.0x (higher pitch), with 1.0x as the default value.

3.5.3 Theme & Display

The Theme & Display section allows you to customize the app’s visual appearance, providing options to align with your system settings or manually select between light and dark themes.

To access the Theme & Display section:

1. Open the application.
2. Tap the gear icon located in the bottom-right corner of the home screen.
3. Select Theme & Display from the list of options.

3.5.4 Features in the Theme & Display Section

1. Override System Theme:
 - A toggle switch to enable or disable manual theme selection.
2. Light/Dark Theme (Visible only when Override System Theme is enabled):
 - Allows manual switching between light and dark themes.
 - Light Theme: Bright background with dark text.
 - Dark Theme: Dark background with light text.

3.5.5 Security

The Security section helps protect your app data by allowing you to manage a passcode. With options to add, change, or remove a passcode.

To access the Theme & Display section:

1. Open the application.

2. Tap the gear icon located in the bottom-right corner of the home screen.
3. Select Security from the list of options.

3.5.6 Features in the Security Section

1. Add Passcode:
 - Visible Only When No Passcode is Set.
 - A Button to navigate to the Add Passcode screen.
 - Enter a 4~6 digit passcode and confirm by entering it again.
 - Upon successful passcode creation, a Recovery Code will be presented.
 - Save the recovery code in a safe place. It is necessary if you forget your passcode, as the app does not store the passcode for security reasons.
2. Change Passcode:
 - Visible Only When a Passcode is Set.
 - A Button to navigate to the Add Passcode screen.
 - Enter your current passcode and new passcode twice to confirm the change.
3. Remove Passcode:
 - A Button to navigate to the Remove Passcode screen.
 - Enter your current passcode to confirm the action.

3.1.2 Hardware Interfaces

The EchoText application is designed to run on iOS devices, specifically iPhones with an A12 Bionic chip (iPhone XR) or newer to ensure optimal performance of the transformer-based TTS models. The hardware interface requirements include the following:

- **Device Type:** The application is compatible with iPhones running iOS version 17.0 or later. It is optimized for devices with an A12 chip or higher to handle the computational demands of the transformer TTS models.
- **Audio Output:** The app supports both built-in speakers and external audio devices (such as Bluetooth headsets or wired headphones) connected via standard iOS interfaces.
- **Storage Requirements:** EchoText requires at least 2 GB of available storage space on the device for installation and TTS model data. Additional storage may be necessary for storing converted audio files.
- **Memory and Processor:** The application is optimized to run on devices with at least 3 GB of RAM and the processing power to support real-time text-to-speech conversion without significant performance degradation.

3.1.3 Software Interfaces

As the system runs entirely on device, there will be no external interfaces with external software. The application will make use of SwiftUI for UI development. For database, SQLite will be used to store metadata for the documents, texts, and audio files. Below are the key software packages and dependencies for the system.

Sherpa-ONNX Framework:

- Version: 1.10.27
- Source: Sherpa-ONNX is an open-source cross platform model inference framework developed by k2.
- License: Apache-2.0
- The Sherpa-ONNX framework will be bundled with the system and serve as the core for model inference using the ONNX runtime.

Swift Programming Language:

- Version: 6.0
- Source: Swift is an open-source UI toolkit developed by Apple.
- License: Apache-2.0
- Swift language is used to interact with deeper iOS hardware and interfaces, as well as handle logic for Sherpa-ONNX.

SQLite Database:

- Version: 3.46.1
- Source: SQLite is an open-source project maintained by Hwaci.
- License: Public domain
- SQLite will be used to store meta data of the documents, texts, and audio files.

3.1.4 Communications Interfaces

The Communications Interfaces section outlines how the TTS application will communicate with other systems or components, both internally and externally.

Internal Communication:

- **Sherpa TTS Engine Integration:** The app will interface directly with the locally installed Sherpa TTS engine to perform text-to-speech conversion. The communication between the frontend (developed in SwiftUI) and the backend (Sherpa) will occur through local API calls, ensuring low latency during real-time TTS processing.
- **On-Device Processing:** All interactions between the app's components (e.g., text input and the Sherpa TTS engine) will be handled locally on the device, enhancing user privacy and supporting offline functionality.

External Communication:

- **No Cloud Dependency:** The application does not require communication with external servers for processing text into speech, as the Sherpa model runs entirely on-device. This ensures full offline functionality and eliminates the need for internet access.
- **App Store Requirements:** The app must comply with Apple's communication guidelines, particularly regarding optional features that may require occasional internet access (e.g., updates or user feedback).

File I/O:

- **Storage of TTS Outputs:** Users may have the option to save audio files generated by the Sherpa TTS engine. Communication between the app and the iOS file system will be managed through appropriate iOS APIs for reading and writing files, ensuring that all files are stored locally to protect user privacy.

3.2 Functional Requirements

The functional requirements for the EchoText application define the specific features and behaviors the system must implement to meet user needs. These requirements cover all essential functionalities, including text input, text-to-speech conversion, voice customization, audio playback, file management and GUI. Each requirement is presented with detailed descriptions of inputs, outputs, and dependencies, ensuring a comprehensive understanding of how the system will operate and interact with users. This section serves as the foundation for the development and implementation of EchoText's core features.

3.2.1 TTS Inference

TITLE	TTS Inference
ID	FR1
DESCRIPTION	The system shall convert text input into speech using a selected Text-to-Speech (TTS) model. The inference must process the text locally on the device.
INPUT	Text data entered by the user in the input field (e.g., a sentence or paragraph) and the selected TTS model parameters (voice, pitch, speed, etc.).
PROCESSING	The system processes the input text through the selected TTS model to generate speech. It loads the model and runs inference to convert text to audio.
OUTPUT	The system generates an audio file that can be played back to the user.
ERROR HANDLING	If inference fails (e.g., due to insufficient memory or a corrupted model), the system will display an error message and log the failure.
FORWARD DEPENDENCY	FR6, FR7, FR8, FR26
BACKWARD DEPENDENCY	None
PRIORITY	High

3.2.2 Model Select

TITLE	Model Select
ID	FR2
DESCRIPTION	The system shall allow users to select from various Text-to-Speech (TTS) models that will be used for the TTS inference process.
INPUT	User selects a TTS model from a list of available models displayed in the settings menu.
PROCESSING	The system stores the selected model and ensures it is loaded during the TTS inference process.
OUTPUT	The selected TTS model is saved and used for future TTS conversions.
ERROR HANDLING	If the selected model is corrupted or unavailable, the system will revert to the default model and display a notification to the user.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	None
PRIORITY	High

3.2.3 Adjust Pitch

TITLE	Adjust Pitch
ID	FR3
DESCRIPTION	The system shall allow users to adjust the pitch of the TTS by increasing or decreasing the pitch of the selected voice model before the audio is generated.
INPUT	User adjusts the pitch through a slider or control in the settings menu, selecting a value within a predefined pitch range.
PROCESSING	The system stores the selected pitch value and applies the modification during the TTS inference process by modifying the values of the model before the text is processed.
OUTPUT	The system generates speech audio with the user-defined pitch adjustment applied to the selected voice model.
ERROR HANDLING	If the pitch value is outside the allowed range or cannot be applied, the system resets to the default pitch and notifies the user of the issue.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	None
PRIORITY	High

3.2.4 Import Documents

TITLE	Import Documents
ID	FR4
DESCRIPTION	The system shall allow users to import text-based documents (e.g., PDF, TXT, EPUB) from the device file system
INPUT	The user presses an import document button which will pop a GUI that shows all the documents the user can import.
PROCESSING	The system verifies the file format and stores the document in the local storage.
OUTPUT	The document is displayed within the app, allowing users to view the text and select it for TTS conversion.
ERROR HANDLING	If the file format is unsupported or corrupted, the system notifies the user.
FORWARD DEPENDENCY	FR5
BACKWARD DEPENDENCY	None
PRIORITY	Low

3.2.5 View Documents

TITLE	View Documents
ID	FR5
DESCRIPTION	The system shall allow users to view text-based documents (e.g., PDF, TXT, EPUB) imported into the application, providing basic navigation features for readability.
INPUT	A document selected from the user's list of imported files.
PROCESSING	The system renders the document in a readable format, supporting page scrolling, zooming, and navigation to specific sections of the text.
OUTPUT	The selected document is displayed within the application's document viewer, enabling the user to browse and read the content.
ERROR HANDLING	If the document cannot be opened due to corruption or unsupported format, the system will notify the user and offer to remove the file.
FORWARD DEPENDENCY	FR6
BACKWARD DEPENDENCY	FR4
PRIORITY	High

3.2.6 Batch Inference Documents

TITLE	Batch Inference Documents
ID	FR6
DESCRIPTION	The system shall allow users to select multiple documents for batch processing, converting each document's text to speech sequentially without manual intervention.
INPUT	Multiple text-based documents (e.g., PDF, TXT, EPUB) selected by the user from the document list.
PROCESSING	The system processes each document sequentially through the TTS engine, performing text-to-speech inference on each file and generating corresponding audio outputs.
OUTPUT	An audio file is generated for each document, saved locally or made available for playback.
ERROR HANDLING	If an error occurs with one document (e.g., unsupported format or processing failure), the system logs the error and continues processing the remaining documents.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR5, FR1
PRIORITY	Medium

3.2.7 View Text History

TITLE	View Text History
ID	FR7
DESCRIPTION	The system shall allow users to view a history of previously entered text used in the TTS conversion, enabling them to select past text for reuse.
INPUT	User accesses the text history feature from the settings or main menu, displaying a list of previously entered texts.
PROCESSING	The system retrieves stored text entries from the local database and displays them in reverse chronological order.
OUTPUT	A list of previously entered text entries is displayed, allowing the user to select and reuse any text for new TTS conversions.
ERROR HANDLING	If there is an issue retrieving history data (e.g., data corruption or missing entries), the system will notify the user and log the error.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR1
PRIORITY	Medium

3.2.8 TTS Terminate

TITLE	TTS Terminate
ID	FR8
DESCRIPTION	The system shall allow the user to terminate or stop the Text-to-Speech (TTS) process at any point during the conversion, immediately halting audio generation.
INPUT	The user presses the "Terminate" button while the TTS process is running.
PROCESSING	The system halts the ongoing TTS conversion, stops any audio generation, and clears the remaining tasks related to the current text input.
OUTPUT	The TTS process is terminated, and no further audio is generated.
ERROR HANDLING	If the TTS process cannot be terminated (e.g., due to system delays or failure), the system should notify the user.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR1
PRIORITY	High

3.2.9 Select Voice Profile

TITLE	Select Voice Profile
ID	FR9
DESCRIPTION	The system shall allow users to select a saved voice profile that contains specific TTS settings, including model type, pitch, and speed, to be applied for text-to-speech conversion.
INPUT	The user selects a pre-configured voice profile from the list of saved profiles.
PROCESSING	The system retrieves the settings (TTS model, pitch, speed) from the selected profile and applies them to the upcoming text-to-speech conversion.
OUTPUT	The selected profile is applied, and the TTS process uses the profile settings for speech generation.
ERROR HANDLING	If the profile is corrupted or missing, the system will notify the user and default to the last-used settings or a default profile.
FORWARD DEPENDENCY	FR10, FR11
BACKWARD DEPENDENCY	None
PRIORITY	Medium

3.2.10 Add Voice Profile

TITLE	Add Voice Profile
ID	FR10
DESCRIPTION	The system shall allow users to create and save a custom voice profile by selecting specific TTS settings, including model type, pitch, and speed.
INPUT	The user configures TTS settings (model, pitch, speed) and provides a name for the new voice profile.
PROCESSING	The system stores the configured TTS settings and associates them with the user-provided profile name. The profile is saved to the device for future use.
OUTPUT	The new voice profile is added to the list of saved profiles, available for selection during TTS processes.
ERROR HANDLING	If the profile cannot be saved (e.g., due to insufficient storage), the system will notify the user and request a retry.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR9
PRIORITY	Medium

3.2.11 Delete Voice Profile

TITLE	Delete Voice Profile
ID	FR11
DESCRIPTION	The system shall allow users to delete a previously saved voice profile from the list of available profiles.
INPUT	The user selects a voice profile from the list and confirms the deletion.
PROCESSING	The system removes the selected profile from the local storage and updates the list of available profiles.
OUTPUT	The selected profile is removed from the list, and it is no longer available for selection.
ERROR HANDLING	If the profile cannot be deleted (e.g., due to storage issues), the system notifies the user and logs the issue.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR9
PRIORITY	medium

3.2.12 Select Dark Mode and Light Mode

TITLE	Dark Mode and Light Mode
ID	FR12
DESCRIPTION	The system shall allow users to toggle between Dark Mode and Light Mode to adjust the appearance of the app based on user preference.
INPUT	User selects the theme from the settings menu.
PROCESSING	The system saves the selected theme in the app's persistent storage (NSUserDefaults) and applies the change across all user interfaces.
OUTPUT	The app switches to Dark Mode or Light Mode, reflecting the selected theme throughout the UI.
ERROR HANDLING	If the theme selection fails to apply, the system reverts to the default mode and notifies the user of the error.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	None
PRIORITY	Medium

3.2.13 Reset Settings

TITLE	Reset Settings
ID	FR13
DESCRIPTION	The system shall allow users to reset all settings to their default values.
INPUT	User clicks the “Reset” button in the settings menu.
PROCESSING	The system restores all customizable settings (e.g., theme, playback speed, etc.) to their default values and clears any user-defined preferences from persistent storage.
OUTPUT	The app resets all settings and notifies the user that the reset was successful.
ERROR HANDLING	If the reset fails, the system displays an error message and retains the current settings.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR12, FR18
PRIORITY	High

3.2.14 Play Audio

TITLE	Play Audio
ID	FR14
DESCRIPTION	The system shall allow users to play generated audio files.
INPUT	User clicks the “Play” button for an audio file in the app.
PROCESSING	The system retrieves the audio file from local storage and begins playback.
OUTPUT	The audio file plays through the device’s speakers or connected audio output.
ERROR HANDLING	If the file cannot be played (e.g., file is corrupted), the system notifies the user of the error.
FORWARD DEPENDENCY	FR25
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.15 Pause Audio

TITLE	Pause Audio
ID	FR15
DESCRIPTION	The system shall allow users to pause audio playback at any point.
INPUT	User clicks the “Pause” button during audio playback.
PROCESSING	The system pauses the playback and retains the current position for resumption.
OUTPUT	Audio playback stops and remains paused until resumed.
ERROR HANDLING	If the pause function fails, playback continues, and the system prompts the user to try again.
FORWARD DEPENDENCY	FR25
BACKWARD DEPENDENCY	FR26
PRIORITY	Medium

3.2.16 Rewind

TITLE	Rewind
ID	FR16
DESCRIPTION	The system shall allow users to rewind audio playback in a configurable number of seconds (e.g., 10 seconds).
INPUT	User clicks the “Rewind” button during audio playback.
PROCESSING	The system moves the playback position backward by the specified number of seconds.
OUTPUT	The audio starts playing from the new position.
ERROR HANDLING	If rewinding fails, playback resumes from the original position, and the system notifies the user of the error.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	26
PRIORITY	Low

3.2.17 Fast Forward

TITLE	Fast Forward
ID	FR17
DESCRIPTION	The system shall allow users to fast-forward audio playback by a configurable number of seconds (e.g., 10 seconds).
INPUT	User clicks the “Fast Forward” button during audio playback.
PROCESSING	The system moves the playback position forward by the specified number of seconds.
OUTPUT	The audio starts playing from the new position.
ERROR HANDLING	If fast-forwarding fails, playback resumes from the original position, and the system notifies the user of the error.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	Low

3.2.18 Adjust Playback Speed

TITLE	Adjust Playback Speed
ID	FR18
DESCRIPTION	The system shall allow users to adjust the speed of audio playback (e.g., 0.5x, 1x, 1.5x, 2x).
INPUT	User selects the desired playback speed from the settings menu or playback controls.
PROCESSING	The system applies the selected speed to the current or upcoming audio playback.
OUTPUT	Audio plays at the chosen speed without affecting pitch.
ERROR HANDLING	If the speed adjustment fails, playback continues at the default speed, and the system alerts the user.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	Medium

3.2.19 Previous Track

TITLE	Previous Track
ID	FR19
DESCRIPTION	The system shall allow users to skip to the previous audio track in the queue during playback.
INPUT	The user presses the "Previous" button in the audio playback interface.
PROCESSING	The system halts the current track and starts playback from the previous track in the queue.
OUTPUT	The previous audio track begins playing from the start.
ERROR HANDLING	If there is no previous track (e.g., the first track is playing), the system notifies the user or loops back to the last track depending on settings.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.20 Next Track

TITLE	Next Track
ID	FR20
DESCRIPTION	The system shall allow users to skip to the next audio track in the queue during playback.
INPUT	The user presses the "Next" button in the audio playback interface.
PROCESSING	The system halts the current track and starts playback from the next track in the queue.
OUTPUT	The next audio track begins playing from the start.
ERROR HANDLING	If there is no next track (e.g., the last track is playing), the system stops playing audio or loops back to the first track depending on settings.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.21 Seek Bar for Audio

TITLE	Seek Bar for Audio
ID	FR21
DESCRIPTION	The system shall provide a seek bar that allows users to jump to a specific point in the audio track during playback.
INPUT	The user interacts with the seek bar, dragging the control to a new position in the audio track.
PROCESSING	The system adjusts the current playback position to the specified timestamp in the audio track.
OUTPUT	Playback resumes from the new position in the audio track.
ERROR HANDLING	If the user seeks beyond the audio length (e.g., seeks past the end of the track), the system snaps the playback to the end of the track and stops.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.22 Add to Queue

TITLE	Add to Queue
ID	FR22
DESCRIPTION	The system shall allow users to add converted speech files to a queue for continuous playback.
INPUT	The user adds a previously generated audio file from the audio file view to the queue via UI elements.
PROCESSING	The system retrieves the selected audio file's metadata from the database, adds the file to the playback queue, and the queue is updated to reflect the new change.
OUTPUT	The selected speech file is added to the queue allowing the user to manage and play multiple audio files in sequence.
ERROR HANDLING	If the system cannot retrieve the selected audio file from storage, it will display an error message to the user, prompting them to try again.
FORWARD DEPENDENCY	FR23, FR24
BACKWARD DEPENDENCY	FR26
PRIORITY	Low

3.2.23 Delete from Queue

TITLE	Delete from Queue
ID	FR23
DESCRIPTION	The system shall allow the user to remove speech files from the queue. The user can select one or more files from an existing queue and remove them to update the queue's content.
INPUT	The user selects one or more speech files from an existing queue.
PROCESSING	The system removes the item from the queue. The queue is updated.
OUTPUT	The selected speech file(s) are removed from the queue and the updated queue is displayed to the user.
ERROR HANDLING	If the selected audio file(s) cannot be removed from the queue due to a system error, the system will notify the user and suggest retrying.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR22
PRIORITY	Low

3.2.24 Reorder Items in Queue

TITLE	Reorder Items in Queue
ID	FR24
DESCRIPTION	The system shall allow the user to change the order of audio files within the queue. The user can drag and drop items or use interface controls to move files up or down in the queue order.
INPUT	User interaction to change the position of one or more audio files within a queue (e.g., drag and drop, or using up/down controls).
PROCESSING	The system retrieves the current queue and the positions of the items the user wants to reorder. The system adjusts the portions of the selected item in the queue and updates the queue.
OUTPUT	The queue is updated to reflect the new order of audio files as specified by the user and saved within the app.
ERROR HANDLING	If the selected audio file(s) cannot be reorder from the queue due to a system error, the system will notify the user and suggest retrying.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR22
PRIORITY	Low

3.2.25 Automatic Bookmarking

TITLE	Automatic Bookmarking
ID	FR25
DESCRIPTION	When the user closes and quits the app, if the user is playing or paused on an audio file, when the user re-opens the app, the play-back control will automatically pause on where the user left off.
INPUT	User quits the application while an audio file is actively playing or paused.
PROCESSING	The system records the playback position of the audio file and stores it in NSUserDefaults. Upon re-opening the app, the system retrieves the playback position, loads the audio file into the playback system, and remains paused.
OUTPUT	The playback control shows the user's last position on the audio file when the app is reopened.
ERROR HANDLING	If the system fails to save or retrieve the stored position, or if the audio file cannot be accessed, upon re-opening the application, an error is displayed, and no audio will be loaded into the playback system.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR14, FR15
PRIORITY	Low

3.2.26 View Audio Files (Files, Model, Date Created, Length)

TITLE	View Audio Files
ID	FR26
DESCRIPTION	The system shall allow users to view application-generated audio files within the application. The user will be able to see the list of audio files, the model or profile used to generate the audio, the date the audio is generated, and the length of the audio. The audio viewer will provide basic navigation features such as scrolling.
INPUT	Switching to the view audio tab UI via manually switching.
PROCESSING	The system retrieves the metadata for all generated audio files from the database. The metadata list is organized and displayed in the audio file view.
OUTPUT	A list of generated audio files that the user can scroll through.
ERROR HANDLING	If the system is unable to retrieve the audio files' metadata from SQLite, the system will display an error message informing the user that the audio list could not be loaded.
FORWARD DEPENDENCY	FR14, FR15, FR16, FR17, FR18, FR19, FR20, FR21, FR22, FR27, FR28, FR29, FR30
BACKWARD DEPENDENCY	FR1
PRIORITY	High

3.2.27 Rename Audio Files

TITLE	Rename Audio Files
ID	FR27
DESCRIPTION	The system shall enable users to rename previously saved audio files.
INPUT	Current name of the audio file and the new name provided by the user.
PROCESSING	The name of the audio file in the database is updated, and the audio file view is updated to reflect this change.
OUTPUT	Audio file with the updated name.
ERROR HANDLING	Display an error message if there is an issue updating the file name, reverting to the original state.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.28 Delete Audio Files

TITLE	Delete Audio Files
ID	FR28
DESCRIPTION	The system shall provide the functionality to delete saved audio files.
INPUT	Selection of the audio file to be deleted by the user.
PROCESSING	The audio file in local storage and its metadata in the database are deleted, and the audio file view is updated to reflect this change.
OUTPUT	Confirmation of the deleted audio file
ERROR HANDLING	Display an error message if there was an issue removing the audio file or the metadata and reverting to the original state.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	High

3.2.29 Search Audio Files by Name

TITLE	Search Audio Files by Name
ID	FR29
DESCRIPTION	The system shall provide a search functionality that allows users to find saved audio files by entering part or all the file names. This feature enables users to quickly locate specific audio files, especially when managing a large number of files.
INPUT	User enters a keyword or phrase into a search bar within the Generated Audios page.
PROCESSING	The system compares the search query with the names of the audio files and filters and retrieves a list of audio files whose name contains the keyword or phrase.
OUTPUT	The app displays a filtered list of audio files whose names match the entered search query.
ERROR HANDLING	If the search query returns no results, a “No results found” message will be displayed. If there are errors in the search query, an error message will be displayed.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	Medium

EchoText

3.2.30 Share Audio via iOS Features

TITLE	Share Audio via iOS Features
ID	FR30
DESCRIPTION	The system shall allow users to share audio files using Apple's built-in sharing features. Users can share audio files via Messages, Mail, AirDrop, or other apps and services supported by the iOS Share Sheet. This functionality enables users to easily distribute their generated audio files to others.
INPUT	User selects an audio file to share from within the app. User taps the Share button and selects the desired sharing method from the Share Sheet.
PROCESSING	The selected audio file is retrieved and packaged by the system, then shared via the iOS Share Sheet.
OUTPUT	The audio file is packaged and sent via the chosen method. The app displays a confirmation of successful sharing or handles any errors.
ERROR HANDLING	If the audio file cannot be retrieved, or the sharing process fails due to a network issue, the system will display an error prompting the user to retry.
FORWARD DEPENDENCY	None
BACKWARD DEPENDENCY	FR26
PRIORITY	Low

3.3 Non-Functional Requirements

The non-functional requirements for the EchoText application define the quality attributes and system-wide constraints that the software must meet to ensure a high-quality application. These requirements focus on aspects such as performance, reliability, availability, security, maintainability, and portability, specifying measurable criteria that the system must achieve. Non-functional requirements ensure that EchoText not only meets functional expectations but also performs efficiently and securely across different environments and usage scenarios.

3.3.1 Performance

The Performance section specifies the performance criteria for the TTS application, focusing on real-time processing with the Sherpa TTS engine. The following requirements must be met to achieve performance:

3.3.1.2 Resource Utilization

ID	NFR2
DESCRIPTION	The app must operate with minimal memory usage with minimal storage.
REQUIREMENT	Memory: The application should operate within a memory usage of no more than 500 MB during normal operations. Storage: The Sherpa models and associated resources should occupy minimal storage, targeting under 2GB for a standard installation.

3.3.1.3 Scalability

ID	NFR3
DESCRIPTION	The app must handle varying text lengths efficiently, from short phrases to longer passages, without compromising performance.
REQUIREMENT	The goal is to maintain real-time processing capabilities (within 500 ms to 1 second regardless of input size.

3.3.2 Reliability

EchoText must be reliable to ensure a consistent and dependable user experience. The following requirements must be met to achieve reliability:

3.3.2.1 Session Stability

ID	NFR4
DESCRIPTION	The application must maintain stability across user sessions, avoiding crashes or unexpected terminations.
REQUIREMENT	The app will avoid crashes or unexpected terminations in at least 95% of user sessions.

3.3.2.2 Error Handling

ID	NFR5
DESCRIPTION	The app must handle errors gracefully, ensuring data integrity in case of failures
REQUIREMENT	The app will handle errors gracefully without crashing, ensuring data integrity in 98% of error cases.

3.3.2.3 Resources Management

ID	NFR6
DESCRIPTION	The application must manage resources efficiently, avoiding resource exhaustion.
REQUIREMENT	The app will prevent resource exhaustion, keeping memory usage under 500 MB in 95% of use cases.

3.3.2.4 Large File Handling

ID	NFR7
DESCRIPTION	The system should reliably process large text files without crashing.
REQUIREMENT	The app will reliably process text files up to 100 MB without crashing in 90% of attempts.

3.3.2.5 Extended Usage Stability

ID	NFR8
DESCRIPTION	The application must function continuously for extended periods without performance degradation.
REQUIREMENT	The app will function continuously for up to 4 hours without performance degradation in 95% of sessions.

3.3.2.6 Data Integrity

ID	NFR9
DESCRIPTION	The app must ensure data integrity across all operations.
REQUIREMENT	The app will ensure no data loss or corruption in 95% of operations involving user data.

3.3.3 Availability

The EchoText application is designed to maintain high availability, ensuring that users can access its text-to-speech functionalities at any time without interruptions. The following requirements must be met to achieve availability:

3.3.3.1 Uptime

ID	NFR10
DESCRIPTION	The system must be available for use at any time, ensuring minimal downtime and access even without internet connectivity.
REQUIREMENT	99.5% availability, with all features working as intended during normal operation.

3.3.4 Security

The TTS iOS application will implement multiple security measures to protect user data. These measures will make sure that only authorized users will have access to the system and its data. The following requirements must be met to achieve security:

3.3.4.1 Data Encryption

ID	NFR11
DESCRIPTION	This encryption must ensure that even if the device or the iCloud account is compromised, unauthorized individuals cannot access the stored data. The system must guarantee data integrity, ensuring that data cannot be tampered with by unauthorized applications.
REQUIREMENT	The system must encrypt all potentially sensitive data, including documents, text files, audio files, and metadata.

3.3.4.2 Optional Password Protection

ID	NFR12
DESCRIPTION	This feature must prevent unauthorized access to the application and its files, even if someone gains access to the unlocked phone.
REQUIREMENT	The system must allow users to optionally set a password to restrict access to the application.

3.3.4.3 Backup and Recovery

ID	NFR13
DESCRIPTION	The system must not create backups of user data outside of the device. The system must notify the user that lost data will not be recoverable.
REQUIREMENT	The system will not have internet connectivity. The system will preemptively notify users that data will not be recoverable.

3.3.4.4 Compliance

ID	NFR14
DESCRIPTION	The system must adhere to best practices for security and privacy during development.
REQUIREMENT	The system will comply with GDPR, although not applicable to our main target region, it is widely considered a global standard.

3.3.5 Maintainability

Maintainability of the EchoText application is essential to ensure the system can be easily modified, updated, or enhanced over time with minimal effort. The following requirements must be met to achieve maintainability:

3.3.5.1 Version Control

ID	NFR15
DESCRIPTION	All code changes must be properly tracked, documented, and organized using version control tools.
REQUIREMENT	All code changes must be tracked using GitHub, with each feature or bug fix developed in a separate branch.

3.3.5.2 Modularity

ID	NFR16
DESCRIPTION	The system should be modular to allow independent updates and minimal impact on unrelated components.
REQUIREMENT	At least 80% of code changes should affect only one module, ensuring isolated updates and ease of maintenance.

3.3.5.3 Manual Testing

ID	NFR17
DESCRIPTION	Manual testing must be conducted before major updates and releases to ensure system stability and usability.
REQUIREMENT	Manual tests should cover UI/UX, integration, and system-level testing, with documented test cases and results for each release.

3.3.5.4 Automated Testing

ID	NFR18
DESCRIPTION	Automated tests should be implemented to quickly detect issues during development and ensure system reliability.
REQUIREMENT	Automate any part of the codebase that can be easily automated using apple XCTest before deployment.

3.4 Design Constraints

The design of the EchoText application is subject to two main constraints, which is IOS development and Sherpa-ONNX. Those constraints are outlined below:

IOS development Limitations:

Due to the nature of iOS development, Apple imposes certain restrictions on development for their closed platform devices, which can only be built and deployed using macOS.

These constraints include:

- **Platform Exclusivity:** iOS applications must be developed and compiled exclusively on macOS using Xcode. This restricts development to Apple environments, limiting the ability for developers to work on other platforms such as Windows or Linux, and increases the reliance on macOS infrastructure for build and deployment processes.

Sherpa-ONNX limitations:

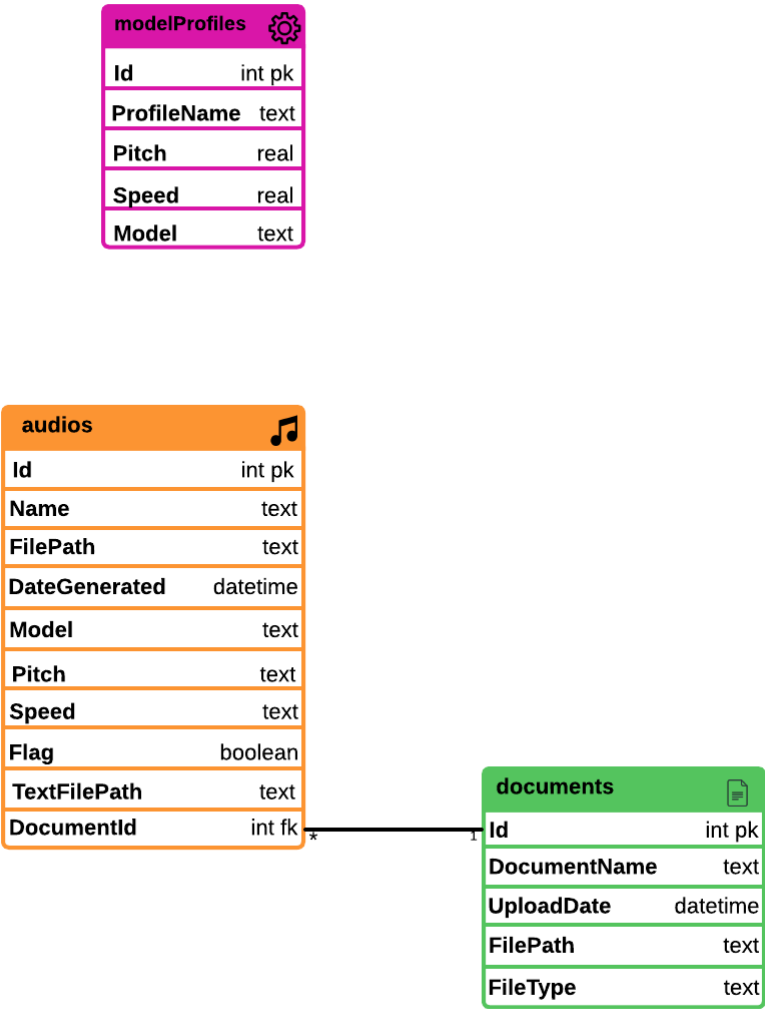
Sherpa-ONNX, the framework used for on-device text-to-speech model inference in EchoText, also introduces its own limitations:

- **Batch Inference Model:** Sherpa-ONNX processes the entire text input in a batch, performing the full inference at once and outputting the complete audio file only

after processing is complete. This design limits real-time streaming capabilities, meaning that EchoText cannot provide speech output progressively as text is being processed. Users must wait until the entire text has been converted before hearing any audio output, which affects the responsiveness of the system, particularly for long text inputs.

3.5 Logical Database Requirements

The application will utilize a database to store metadata and user-generated content related to text-to-speech (TTS) features. The diagram outlines a database where documents are uploaded, processed through various model profiles, and audio files are generated and stored. Each audio file is linked to its corresponding document and stores settings such as model, pitch, and speed.



4. Analysis Models

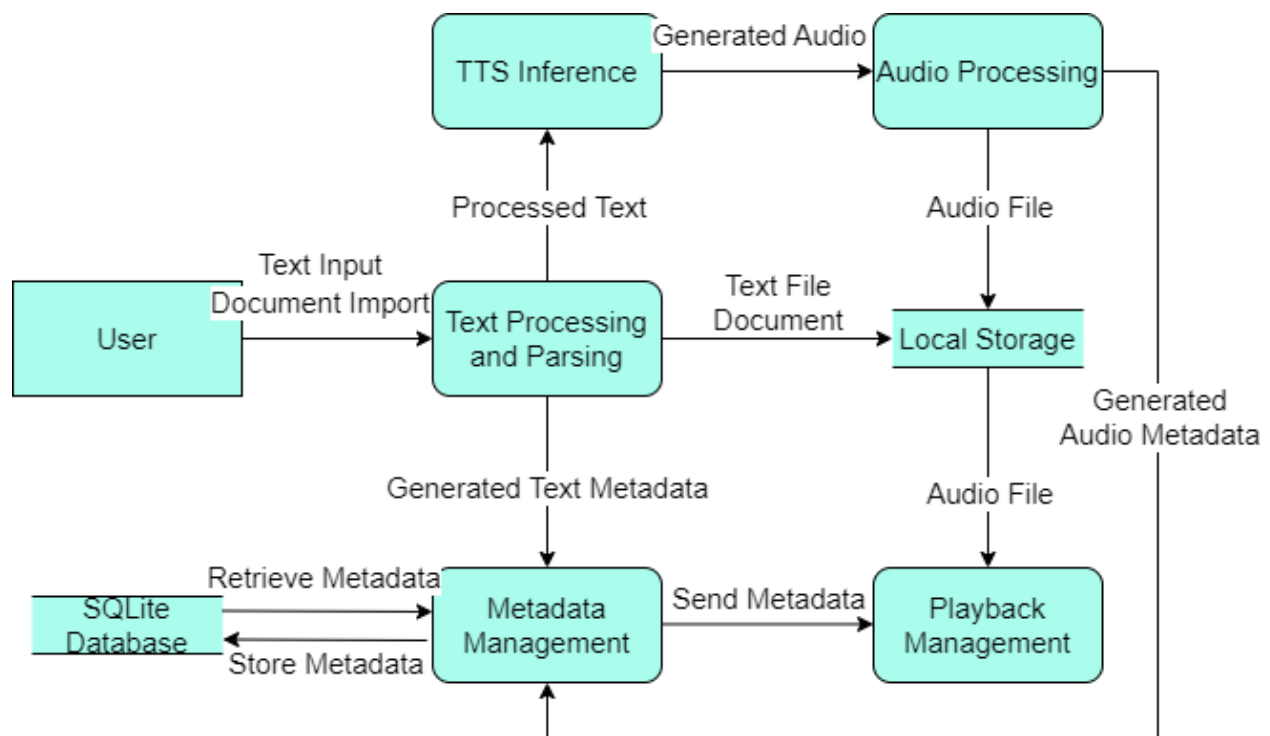
We will use a Data Flow Diagram (DFD) to show the internal workings of our TTS application. It shows how data flows between the user, the main subprocesses of the application, and the data stores.

4.1 Data Flow Diagrams (DFD)

The data flow diagram (DFD) provides a visual representation of how data flows through the application.

Process descriptions:

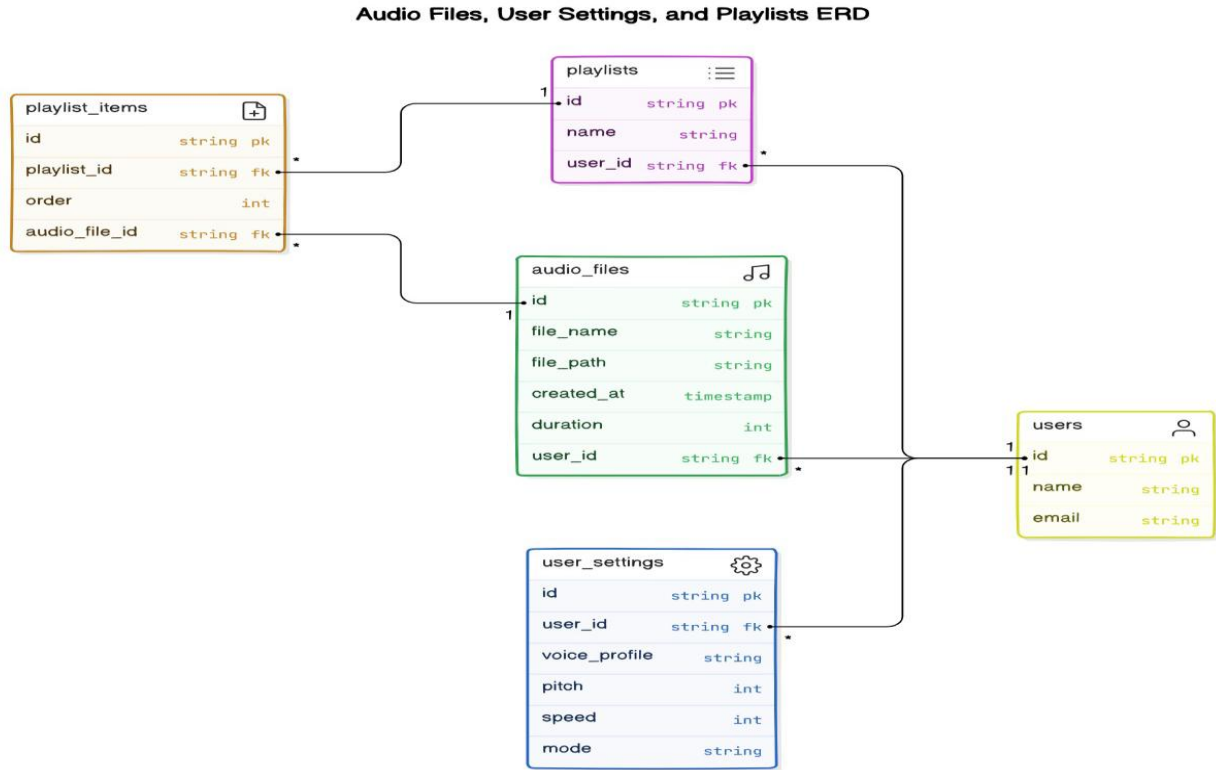
- Text Processing and Parsing: Text is extracted from files or a string, and the text is chunked to allow for more responsive inferencing.
- Audio Processing: The audio's pitch is adjusted if needed, and after inference is complete the audio chunks are stitched together.
- Playback Management: Plays audio chunks via AVAudioPlayer, as well as provide the audio control interface to the user.



A. Appendices

A.1 Appendix 1

Initial Database design diagram



A.2 Appendix 2

Re-numbered FR from V1.00 to V1.01

FR Name	V1.00 ID	V1.01 ID
Text Input	FR1	FR1
Copy-Paste	FR2	FR2
Play/Pause	FR3	FR18
Generated Voice Speed	FR4	FR9
Previous Track	FR5	FR22
Next Track	FR5	FR23
Seek Bar for Audio	FR6	FR24
Import Documents via Computer	FR7	FR3
View Documents	FR8	FR4
View Audio Files	FR9	FR30
Model Select	FR10	FR11
TTS Inference	FR11	FR8
Rename Audio Files	FR13	FR31

Delete Audio Files	FR14	FR32
Clear Text Button	FR15	FR7
Choose Auto Save/Manually Save	FR16	FR33
Batch Inference Documents	FR17	FR5
Automatic Bookmarking	FR18	FR29
Dark Mode and Light Mode	FR20	FR16
Add to Playlist	FR21	FR25
Delete from Playlist	FR22	FR27
Select from Playlist	FR23	FR26
Reorder Items in Playlist	FR24	FR28
Search Audio File by Name	FR25	FR34
Export Audio Files to Computer	FR26	FR35
Share Audio via iOS Features	FR27	FR36
Adjust Pitch	FR28	FR10
Reset Settings	FR29	FR17
Select Voice Profile	FR30	FR12
Add Voice Profile	FR31	FR13
Delete Voice Profile	FR32	FR14
Settings Page	FR33	FR15
View Text History	FR34	FR6
Rewind	FR35	FR19
Fast Forward	FR36	FR20
Adjust Playback Speed	FR37	FR21

B. References

[1] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.