EchoText

Design Document
Version 1.02

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

Revision History

Date	Description	Author	Comments
10/17/24	Version 1.00	Andy Huang	First Revision
10/15/24	Version 1.01	Andy Huang	Updated 2.2 Constraints, 3.2 Software Architecture, 3.4 Performance, 4.1 Use-Case Diagram, 4.2 Use-Case Tables, 4.3 Sequence Diagram, 4.5 Database Design, 4.7 Application Program Interfaces, 4.8 User Interface Design
12/3/24	Version 1.02	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

1. Introduction

1.1 Purpose of the Product Design Specification Document

The EchoText Product Design Specification outlines the technical requirements and design approach for the development of a transformer-based, on-device text-to-speech (TTS) application for iOS platforms. This document serves as a comprehensive guide for the development team, providing detailed information on the architecture, design principles, and implementation guidelines to ensure the application meets performance, privacy, and usability goals.

The purpose of this document is to define the design specifications for the EchoText application, including its system architecture, software components, and user interface. By offering a clear framework, this document aims to facilitate the development of a scalable, maintainable, and high-performance TTS solution that runs entirely offline on mobile devices. It will also serve as a reference point for all stakeholders to ensure alignment with the project's objectives and technical requirements.

This document is primarily intended for the development team, specifically the engineers responsible for building and maintaining the EchoText application. While its main audience is developers, it may also be shared with key technical stakeholders who require a detailed understanding of the design specifications. The document provides essential information for developers involved in the technical implementation, testing, and deployment processes.

2. General Overview and Design Guidelines/Approach

This section describes the principles and strategies to be used as guidelines when designing and implementing the system.

2.1 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. Developers will use Xcode for iOS development, as it is an integrated development environment (IDE) tailored specifically for Apple platforms. However, the choice of Xcode goes beyond convenience; it provides robust tools for efficient code debugging, seamless integration with Swift, and compatibility with Apple frameworks, including Sherpa-ONNX, which is optimized for Swift development.

- The EchoText application is explicitly designed for modern iOS devices, including:
    - iPhone Models: iPhone XR and newer, to leverage hardware capabilities like the A12 Bionic chip and above for optimal performance.
    - iPad Models: iPad models with equivalent or superior hardware specifications to the A12 Bionic chip.

2.2 Constraints

Hardware Constraints:
- The iOS devices targeted by this project have specific hardware constraints, such as:
    - Limited CPU power and RAM compared to desktop environments:
        - **CPU**: The app will require a minimum of an A12 Bionic chip to handle the computational load of the TTS engine efficiently.
        - **Memory**: A minimum of 3 GB of RAM is required to ensure smooth processing, especially for transformer-based models used in Sherpa-ONNX. The target RAM usage for the application itself is 1.5 GB.
        - **Storage**: At least 2 GB of free storage must be available on the device to accommodate model files, user data, and generated audio outputs.
    - 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.
- Xcode is essential for the development of EchoText because Sherpa-ONNX, the chosen TTS framework, only provides native support for Swift. This ensures high performance and compatibility with Apple's hardware and software ecosystem.

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.

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.

2.3 Standards

- **Development Standards**: The application will adhere to Apple's Human Interface Guidelines to ensure a consistent and intuitive user experience across iOS devices.
- **Data Protection Standards**: The application will comply with relevant data protection regulations (e.g., GDPR) to safeguard user information.
- **Testing Standards**: Comprehensive testing methodologies will be employed, including unit testing, integration testing, and user acceptance testing, to validate functionality and performance.
- **Model Compliance**: All models utilized in the application will be developed and optimized in compliance with industry best practices for TTS systems and neural networks.

3. Architecture Design

This section outlines the system and hardware architecture design of the system that is being built.

3.1 Hardware Architecture

The hardware architecture for the EchoText application is designed to support efficient on-device Text-to-Speech (TTS) processing while ensuring optimal performance across various iOS devices. The following subsections outline the hardware components and configurations necessary for the application.

3.1.1 Target Devices

EchoText is designed to run on iOS devices, specifically iPhones and iPads. The following minimum hardware specifications are recommended to ensure smooth operation:

- **Processor:** A12 Bionic chip or newer
- **Memory (RAM):** 3 GB or more
- **Storage:** At least 2GB of free storage for model files, text history, and audio outputs
- **Operating System:** iOS 17.0 or later

These specifications ensure that the device has sufficient processing power and memory to run Sherpa-ONNX-based TTS models while handling text input, playback, and other functionalities.

3.1.2 External Connections

The following hardware interfaces may be used for importing documents and exporting audio files:

- **Wireless Connectivity:** AirDrop, Wi-Fi, and Bluetooth may be used to transfer text documents or audio files to and from the application.

3.1.3 Development Hardware Requirements

To develop, test, and deploy the EchoText application, the following hardware is required:

- **macOS Device**: A Mac computer (MacBook, iMac, or Mac mini) running macOS is essential for development as Apple's iOS development tools (Xcode) are exclusive to macOS.
    - **Operating System**: macOS 15.0 (Sequoia) or later, to ensure compatibility with the latest version of Xcode and iOS SDKs.

3.2 Software Architecture

The software architecture for EchoText is based on the Model-View-Controller (MVC) pattern, which enhances modularity, scalability, and maintainability for developing that application. By separating functionality across distinct layers, this architecture enables easier updates, streamlined debugging, and a clear structure for future development.

- **Model:** This layer handles data-related operations, including TTS model configuration, audio file storage, and document management. It is responsible for interfacing with the database and sending data to the view and controller.

- **View:** The View layer manages all user interface components, ensuring a responsive and intuitive user experience. It provides visual elements such as the text editor, generated audio list, and playback controls, dynamically updating based on changes in the Model.

- **Controller:** Acting as an intermediary between the Model and View, the Controller processes user inputs and commands, triggering appropriate updates in the Model and updating the View in response.
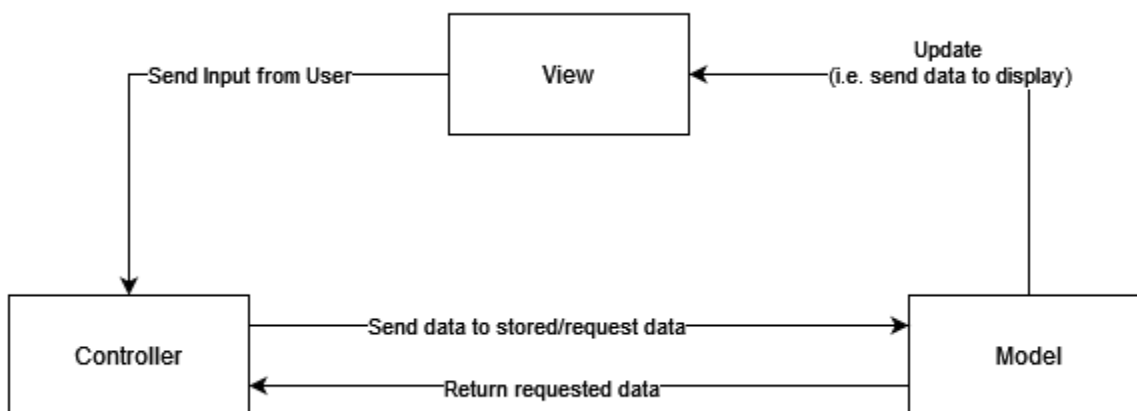
The MVC architecture is ideal for EchoText's local-only design, as it enables efficient data flow between components while ensuring modularity and a clear separation of responsibilities across the application. This separation simplifies maintenance and allows

future upgrades, such as enhancements to TTS capabilities or UI updates, without disrupting the entire application structure.

3.3 Security Architecture

The application is designed to ensure the confidentiality and security of all data processed and stored, including all generated audio files, generated metadata, generated text files, and important text files. The app will operate entirely offline, to minimize attack vectors. The primary goal in terms of security is to protect user data from unauthorized access, even if the device, in both locked and unlocked forms, or iCloud backups are compromised.

3.3.1 Threat modeling

- Device Compromised: If a phone is lost or stolen, either in a locked or unlocked state on the device level, sensitive documents and audio files must remain inaccessible.
- iCloud Compromised: If the iCloud service is breached or the user's Apple account is compromised, backed-up data must remain encrypted and unusable without the decryption keys tied to the original device hardware.

3.3.2 Authentication

- Device-Level Security: iOS's NSFileProtectionCompleteUntilFirstUserAuthentication will be used to secure application data and files until the phone is unlocked by the user.
- App-Level Authentication: The application will give the user the option to set a password at the application level. If enabled, the application will prompt a password upon launch and won't be accessible until the password is entered. This password will be different than the device unlock code, providing security if the phone in an unlocked state is compromised. The password will be stored using the Keychain service with kSecAttrAccessibleWhenUnlocked. The mitigate brute-force attacks on this password, the following measures will be in place: After a few failed attempts, the app will increase the time delay between failed attempts exponentially, capped at one attempt per 24 hours.

3.3.3 Data Encryption

All data, including generated temporary and persistent audio files, generated and imported text files, and generated metadata, will be encrypted using NSFileProtectionComplete to protect the data until the phone is unlocked.

- **Secure Enclave Integration:** Sensitive data, such as encryption keys used to protect user documents and files, will be stored in the Secure Enclave. This ensures that encryption keys are hardware-tied and never exposed to the broader system or iCloud. Even in the event of an iCloud breach, the encrypted data would remain inaccessible without the corresponding hardware to retrieve the keys from Secure Enclave.

3.3.4 Playback Protection

Playback of generated audio files will be handled securely within the app. No data will be exposed or accessible outside of the app, and temporary playback data will also be encrypted while in use.

3.4 Performance

The performance of the EchoText application is critical to providing a seamless and efficient user experience, especially given the resource-intensive nature of on-device Text-to-Speech (TTS) processing. This section outlines the performance considerations, challenges, and optimization strategies related to hardware constraints, text chunking, threading, and other relevant factors.

3.4.1 Hardware Constraints

Device Variability

- **Processor Differences**: iOS devices vary widely in processing capabilities, from older models with A12 Bionic chips to the latest devices with advanced processors. The application must adapt to these differences to ensure acceptable performance across all supported devices.
- **Memory Usage:** To ensure optimal performance, we aim to keep the app's memory usage below 1.5 GB during text-to-speech processing. This allows the app to run smoothly on devices with limited RAM while still delivering efficient performance for resource-intensive operations.
- **Storage Space**: Limited storage on some devices necessitates optimizing the size of TTS models and managing cached data effectively.

3.4.2 Text Chunking

Challenges

- **Large Text Processing:** Processing extensive text inputs in a single operation can lead to high memory consumption and increased processing time, potentially causing the application to become unresponsive.

Solutions

- **Text Segmentation:** The text is split into smaller chunks or sentences, processed sequentially. We first process the **first sentence** independently, ensuring quick initial feedback. Next, we process as many sentences as possible within a **100-word limit**. Once this chunk is processed, the remaining text is handled in **300-word chunks** until the entire text is converted.
- **Memory Management:** Release memory resources immediately after processing each chunk to minimize overall memory usage.
- **User Feedback:** Provide progress indicators during the processing of large texts to keep the user informed and enhance perceived performance.

3.4.3 Multi-threading and Concurrency

Overview

To maintain a responsive user interface (UI) while performing resource-intensive tasks like Text-to-Speech (TTS) processing, EchoText employs multi-threading. By offloading heavy computations to background threads, the application ensures that the main thread remains free to handle user interactions.

Main Thread vs. Background Threads

- Main Thread (UI Thread): Responsible for handling all UI updates and user interactions. Intensive tasks must not block this thread to prevent the application from becoming unresponsive.
- Background Threads: Used for performing heavy computations such as TTS processing, file I/O operations, and data encryption/decryption

3.5 Communication Architecture

The Communication Architecture for EchoText is designed to allow efficient interactions between various components of the application while ensuring low latency and high reliability for real-time Text-to-Speech (TTS) processing. Since EchoText is an offline, on-device application, its communication flow is entirely internal, avoiding reliance on external servers or networks.

3.5.1 Component Communication

EchoText employs an event-driven communication model, where:
- The Controller acts as the primary mediator between the user interface (View) and the underlying logic (Model).
- The Model interacts with the local database (e.g., SQLite) to store and retrieve user-generated content, such as text, audio files, and settings.
- The TTS Engine processes text input by communicating with the Model to fetch the necessary data and returns the generated audio to the Controller.

3.5.2 Local Data Storage and State Management

To provide a seamless user experience, several settings and preferences are stored locally on the device. These include:
- **Voice Profiles:** Custom voice profiles, including pitch, speed, and model settings, are saved locally and applied directly to the app's audio playback functionality.
- **Theme Settings:** Theme preferences (light/dark) are stored locally to provide instant UI adjustments.
- **Password Settings:** The passcode and recovery code are stored securely using **encrypted storage** on the device. When a user sets or changes their password, the local storage is updated immediately.

4 System Design

4.1 Use-Case Diagram
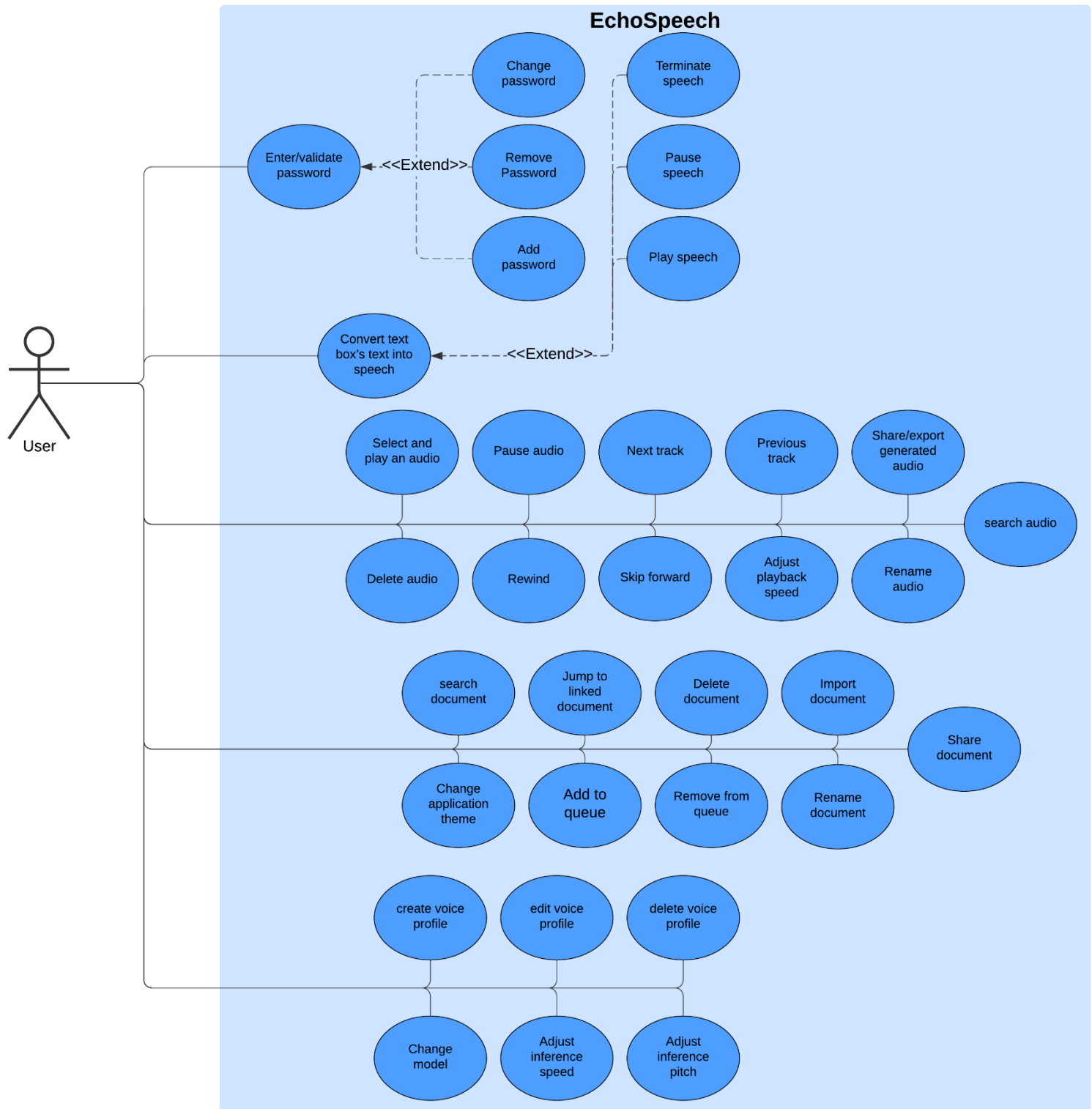
4.2 Use-case tables

4.2.1 Use-Case: Convert text box's text into speech

ID	UC1
Name	Convert Text Box's Text into Speech
Description	This use-case describes how the user converts text input from the text box into speech by triggering the TTS process.
Actors	User (primary actor)
Trigger	The user presses the microphone icon button.
Preconditions	The app displays a text box with user-entered text.
Postconditions	An audio file is generated from the input text and is available for playback.
The Normal Flow	1. The user enters text into the text box. 2. The user presses the microphone icon button. 3. The system converts text to speech using the TTS engine. 4. The system generates the audio file.
Alternative Flow	**Document TTS:** 1. The user clicks on the document tab in the navigation bar 2. The user long press on one of the documents 3. The user presses on "generate" button 4. The system converts document text to speech using the TTS engine. 5. The system generates the audio file.
Exception	Conversion error: if the TTS fails, the system notifies the user
Assumptions	The user has entered text in the text box or uploaded a document for conversion.

4.2.2 Use-Case: Terminate Speech (During Conversion)

ID	UC2
Name	Terminate Speech (During Conversion)
Description	This use-case describes how the user stops an ongoing TTS process, halting the speech generation before completion.
Actors	User (primary actor)
Trigger	The user presses the stop icon button during speech generation.
Preconditions	The app is actively converting text to speech and playing the audio output in real-time.
Postconditions	The ongoing speech generation is terminated.
The Normal Flow	1. The user starts the TTS process by pressing the microphone icon button. 2. The user decides to stop the process and presses the stop icon button. 3. The system stops generating speech and terminates the conversion.
Alternative Flow	N/A
Exception	N/A
Assumptions	The TTS conversion is running.

4.2.3 Use-Case: Pause Speech (During Conversion)

ID	UC3
Name	Pause Speech
Description	This use-case describes how the user pauses the playback of the generated speech during the TTS conversion process.
Actors	User (primary actor)
Trigger	The user presses the pause icon button during audio playback.
Preconditions	The app is actively converting text to speech and playing the audio output in real-time.
Postconditions	The audio playback is paused and can be resumed later during the conversion process.
The Normal Flow	1. The user started the text to speech conversion (UC1) 2. The user presses the pause icon button. 3. The system pauses the audio playback.
Alternative Flow	1. The user presses on the generated tab in the navigation bar. 2. The user press on any of the generated audio files or press play in the generated view. 3. The system pauses the audio playback and starts playing the generated audio file in generated view.
Exception	N/A
Assumptions	The TTS conversion is running.

4.2.4 Use-Case: Play Speech (During Conversion)

ID	UC4
Name	Play Speech (During Conversion)
Description	This use-case describes how the user listens to speech that is being generated in real-time as the text is being converted into speech.
Actors	User (primary actor)
Trigger	The user presses the microphone icon to initiate text-to-speech conversion, and the speech starts playing while conversion is ongoing.
Preconditions	The app is actively converting text to speech and playing the audio output in real-time.
Postconditions	The speech is played in real-time as the text is processed.
The Normal Flow	1. The user enters text and presses the microphone icon button. 2. The system begins converting the text to speech and starts playing the audio in real-time. 3. The speech continues to play as the conversion progresses.
Alternative Flow	**Document TTS:** 1. The user clicks on the document tab in the navigation bar 2. The user long press on one of the documents 3. The user presses on "generate" button 4. The system begins converting the text to speech and starts playing the audio in real-time.
Exception	N/A
Assumptions	The TTS conversion is running.

4.2.5 Use-Case: Select and Play an Audio

ID	UC5
Name	Select and Play an Audio
Description	This use-case describes how the user navigates to the list of previously generated audio files, selects a file, and plays it.
Actors	User (primary actor)
Trigger	The user clicks on one of the generated audio files or clicks on the play button in the generated tab.
Preconditions	The app displays a list of previously generated audio files in the "Generated" tab and no audio is selected or playing.
Postconditions	The system plays the selected audio file and the audio file is highlighted.
The Normal Flow	1. The user presses the "Generated" tab in the navigation bar. 2. The system displays a list of generated audio files. 3. The user selects an audio file from the list. 4. the user taps the audio file to start playback. 5. The system plays the selected audio file.
Alternative Flow	1. The user presses the "Generated" tab in the navigation bar. 2. The system displays a list of generated audio files. 3. The user presses the play icon button. 4. The system plays the first listed audio file.
Exception	**Playback Error:** If the audio file is corrupted or unreadable, the system notifies the user and does not start playback. **No Generated Audio File:** If there are no generated audio file, the button play button will not play anything back to the user.
Assumptions	The user has previously generated audio files stored in the app.

4.2.6 Use-Case: Pause Audio

ID	UC6
Name	Pause Audio
Description	This use-case describes how the user pauses the playback of an audio file from the list of generated files.
Actors	User (primary actor)
Trigger	The user presses the pause icon button during playback.
Preconditions	An audio file is currently playing from the list of generated files in the "Generated" tab.
Postconditions	The audio playback is paused and can be resumed later.
The Normal Flow	1. The user goes to the "Generated" tab. 2. The user presses the "Pause" button. 3. The system pauses the audio playback.
Alternative Flow	N/A
Exception	N/A
Assumptions	The user is playing a previously generated audio file and is currently playing the audio file.

4.2.7 Use-Case: Skip Forward

ID	UC7
Name	Skip Forward
Description	This use case describes how the user skips forward within an audio file during playback in the "Generated" tab.
Actors	User (primary actor)
Trigger	The user presses the Skip Forward icon button during audio playback.
Preconditions	User is in the "Generated" tab and is currently playing an audio file.
Postconditions	The audio playback position advances forward by a set intervale (e.g., 15 seconds).
The Normal Flow	1. The user selects an audio file from the "Generated" tab and begins playback. 2. The user presses the skip forward icon button. 3. The system advances the playback position by a set interval (e.g., 15 seconds).
Alternative Flow	N/A
Exception	**End of Audio**: If skipping forward reaches the end of the audio file, the system stops playback or goes to the next audio file if available.
Assumptions	The user is playing an audio file from the list of generated files.

4.2.8 Use-Case: Rewind

ID	UC8
Name	Rewind
Description	This use case describes how the user rewinds the playback within an audio file in the "Generated" tab.
Actors	User (primary actor)
Trigger	The user presses the rewind icon button during audio playback.
Preconditions	User is in the "Generated" tab and is currently playing an audio file.
Postconditions	The audio playback position moves backward by a set interval (e.g., 10 seconds).
The Normal Flow	1. The user selects an audio file from the "Generated" tab and starts playback. 2. The user presses the rewind icon button. 3. The system moves the playback position backward by a set interval (e.g., 15 seconds).
Alternative Flow	N/A
Exception	**Start of Audio**: If rewinding reaches the beginning of the audio file, playback resumes from the start of the file.
Assumptions	The user is playing an audio file from the list of generated files.

4.2.9 Use-Case: Previous Track

ID	UC9
Name	Previous Track
Description	This use case describes how the user skips to the previous audio file in the playlist or list of generated audio files.
Actors	User (primary actor)
Trigger	The user presses the previous track icon button during audio playback.
Preconditions	The user is playing an audio file from a list that contains multiple generated audio files.
Postconditions	The system begins playback of the previous audio file in the list.
The Normal Flow	1. The user selects and plays an audio file from the "Generated" tab. 2. During playback, the user presses the previous track icon button. 3. The system stops the current audio and starts playing the previous audio file in the list.
Alternative Flow	N/A
Exception	**First Track in List**: If the current track is the first in the list, pressing "Previous Track" will restart the current track.
Assumptions	The user is playing an audio file from the list of generated files and has more than one audio file saved.

4.2.10 Use-Case: Next Track

ID	UC10
Name	Next Track
Description	This use case describes how the user skips to the next audio file in the playlist or list of generated audio files.
Actors	User (primary actor)
Trigger	The user presses the next track icon button during audio playback.
Preconditions	The user is playing an audio file from a list that contains multiple generated audio files.
Postconditions	The system begins playback of the next audio file in the list.
The Normal Flow	1. The user selects and plays an audio file from the "Generated" tab. 2. During playback, the user presses the next track icon button. 3. The system stops the current audio and starts playing the next audio file in the list.
Alternative Flow	N/A
Exception	**Last Track in List:** If the current track is the last in the list, pressing "Next Track" will stop playback, as there are no more tracks.
Assumptions	The user is playing an audio file from the list of generated files and has more than one audio file saved.

4.2.11 Use-Case: Adjust Playback Speed

ID	UC11
Name	Adjust Playback Speed
Description	This use case describes how the user changes the playback speed of an audio file in the "Generated" tab.
Actors	User (primary actor)
Trigger	The user presses the speed icon button during audio playback and select a new speed.
Preconditions	The user is playing an audio file from the list of generated files at normal speed.
Postconditions	The audio file continues playing at the adjusted speed.
The Normal Flow	1. The user selects an audio file in the "Generated" tab and begins playback. 2. The user presses the speed icon button which will show a playback speed menu. 3. The user selects a new speed (e.g., 0.5x, 1x, 1.5x, 2x). 4. The system adjusts the playback speed accordingly and continues playback.
Alternative Flow	N/A
Exception	N/A
Assumptions	The user is playing an audio file from the list of generated files and has at least one audio file saved in the "Generated" tab.

4.2.12 Use-Case: Rename Audio

ID	UC12
Name	Rename Audio
Description	This use-case describes how the user renames a generated audio file in the "Generated" tab.
Actors	User (primary actor)
Trigger	The user presses and holds on an audio file in the list of generated files to access the rename option.
Preconditions	There is a list of audio files in the "Generated" tab.
Postconditions	One of the audio file in the "Generated" tab name got changed base on the user input.
The Normal Flow	1. The user navigates to the "Generated" tab and locates an audio file. 2. The user presses and holds on the audio file. 3. The system displays a menu with options, including "Rename." 4. The user selects "Rename" from the menu. 5. The name of the audio file will be replaced with a textbox. 6. The user enters a new name and press enter. 7. The system saves the audio file with the updated name.
Alternative Flow	N/A
Exception	**Invalid Name:** If the new name contains unsupported characters, the system notifies the user and requests a valid name. **Cancel Rename:** If the user unselects the textbox, the audio file name remains unchanged, and the textbox will disappear.
Assumptions	The user has at least one audio file saved in the "Generated" tab.

4.2.13 Use-Case: Delete Audio

ID	UC13
Name	Delete Audio
Description	This use-case describes how the user deletes a generated audio file in the "Generated" tab.
Actors	User (primary actor)
Trigger	The user presses and holds on an audio file in the list of generated files to access the delete option.
Preconditions	The audio file exists in the list of generated files.
Postconditions	The selected audio file is deleted from the list and storage.
The Normal Flow	1. The user navigates to the "Generated" tab and locates an audio file. 2. The user presses and holds on the audio file. 3. The system displays a menu with options, including "Delete." 4. The user selects "Delete" from the menu. 5. The system prompts the user to confirm the deletion. 6. The user confirms, and the system deletes the audio file.
Alternative Flow	N/A
Exception	**Deletion Error:** If the system encounters an error while attempting to delete the file, it notifies the user. **Cancel Deletion:** The user decides to cancel the deletion action, and the audio file remains unchanged.
Assumptions	The user has at least one audio file saved in the "Generated" tab.

4.2.14 Use-Case: Jump to Linked Document

ID	14
Name	Jump to Linked Document
Description	This use case describes how the user can view the document associated with a generated audio file by pressing and holding on the audio file and selecting the "Linked Document" option.
Actors	User (primary actor)
Trigger	The user presses and holds on a generated audio file and selects the "Linked Document" option.
Preconditions	The selected audio file in the "Generated" tab has an associated linked document.
Postconditions	The linked document is opened for viewing.
The Normal Flow	1. The user navigates to the "Generated" tab and locates an audio file with a linked document. 2. The user presses and holds on the audio file. 3. The system displays a menu with options, including "Linked Document." 4. The user selects "Linked Document" from the menu. 5. The system retrieves and opens the document associated with the audio file.
Alternative Flow	N/A
Exception	**No Linked Document:** If the audio file has no linked document, the system informs the user. **Retrieval Error**: If the system encounters an error when retrieving the document, it notifies the user.
Assumptions	The audio file has an associated document linked to it.

4.2.15 Use-Case: Add to Queue

ID	UC15
Name	Add to Queue
Description	This use-case describes how the user adds a generated audio file to the playback queue for sequential listening.
Actors	User (primary actor)
Trigger	The user selects an audio file and chooses the "Add to Queue" option.
Preconditions	There is an empty queue or a queue with a list of audio files.
Postconditions	The selected audio file is added to the existing playback queue.
The Normal Flow	1. The user navigates to the "Generated" tab and locates an audio file. 2. The user presses and holds on the audio file to access the options menu. 3. The user selects "Add to Queue" from the menu. 4. The system adds the audio file to the end of the playback queue.
Alternative Flow	N/A
Exception	N/A
Assumptions	There are one or more audio files saved in the "Generated" tab.

4.2.16 Use-Case: Remove from Queue

ID	UC16
Name	Remove from Queue
Description	This use case describes how the user removes an audio file from the playback queue.
Actors	User (primary actor)
Trigger	The user presses and holds on an audio file in the queue and selects the "Remove from Queue" option.
Preconditions	The playback queue contains one or more audio files, including the selected file.
Postconditions	The selected audio file is removed from the playback queue, updating the queue to reflect the change.
The Normal Flow	1. The user navigates to the playback queue. 2. The user presses and holds on an audio file to access the options menu. 3. The user selects "Remove from Queue" from the menu. 4. The system removes the audio file from the queue and updates the list.
Alternative Flow	N/A
Exception	N/A
Assumptions	There are one or more audio files currently in the playback queue.

4.2.17 Use-Case: Share/Export Generated Audio

ID	UC17
Name	Share/Export Generated Audio
Description	This use case describes how the user shares or exports a generated audio file from the app to an external location or application.
Actors	User (primary actor)
Trigger	The user presses and holds on a generated audio file and selects the "Share/Export" option.
Preconditions	The selected audio file exists in the "Generated" tab.
Postconditions	The system opens the share menu, allowing the user to select an external app or storage location for exporting the audio file.
The Normal Flow	1. The user navigates to the "Generated" tab and locates an audio file. 2. The user presses and holds on the audio file to open the options menu. 3. The user selects the "Share/Export" option. 4. The system displays the share menu with available export options (e.g., messaging apps, cloud storage). 5. The user selects a destination, and the system exports the audio file.
Alternative Flow	N/A
Exception	**Export Error:** If an error occurs during export, the system notifies the user and retains the file in the app.
Assumptions	There is one or more generated audio files available in the "Generated" tab.

4.2.18 Use-Case: Share Document

ID	UC18
Name	Share Document
Description	This use case describes how the user shares a document associated with a generated audio file to an external application or storage location.
Actors	User (primary actor)
Trigger	The user presses and holds on a document in the "Documents" tab and selects the "Share" option.
Preconditions	The selected document exists in the "Documents" tab.
Postconditions	The system opens the share menu, allowing the user to select an external app or storage location for sharing the document.
The Normal Flow	1. The user navigates to the "Documents" tab and locates a document. 2. The user presses and holds on the document to open the options menu. 3. The user selects the "Share" option. 4. The system displays the share menu with available export options (e.g., messaging apps, cloud storage). 5. The user selects a destination, and the system shares the document.
Alternative Flow	N/A
Exception	**Sharing Error:** If an error occurs during sharing, the system notifies the user and retains the document in the app.
Assumptions	There is one or more documents available in the "Documents" tab.

4.2.19 Use-Case: Import Document

ID	UC19
Name	Import Document
Description	This use-case describes how the user imports a document into the app.
Actors	User (primary actor)
Trigger	The user presses the "Import" button in the "Documents" tab.
Preconditions	In the "Document" tab, there is a list of documents that has previously been imported or there are no documents listed.
Postconditions	The selected document is added to the list of documents in the "Documents" tab and is available for viewing or conversion.
The Normal Flow	1. The user navigates to the "Documents" tab. 2. The user presses the "Import" button. 3. The system opens a file picker for the user to select a document from the device's storage. 4. The user selects a document. 5. The system imports the document and adds it to the list in the "Documents" tab.
Alternative Flow	N/A
Exception	**Unsupported Format:** If the user selects a document in an unsupported format, the system notifies the user and does not import the file. **Import Error**: If an error occurs during import, the system notifies the user that the document could not be imported.
Assumptions	The user has access to compatible documents on their device that can be imported into the app.

4.2.20 Use-Case: Rename Document

ID	UC20
Name	Rename Document
Description	This use-case describes how the user renames a document in the "Documents" tab.
Actors	User (primary actor)
Trigger	The user presses and holds on a document in the "Documents" tab and selects the "Rename" option.
Preconditions	The selected document is listed in the "Documents" tab.
Postconditions	The document is saved with the new name entered by the user.
The Normal Flow	1. The user navigates to the "Documents" tab and locates a document. 2. The user presses and holds on the document to open the options menu. 3. The user selects "Rename" from the menu. 4. The system prompts the user to enter a new name. 5. The user enters the new name and confirms. 6. The system saves the document with the updated name.
Alternative Flow	N/A
Exception	**Cancel Rename:** The user can cancel the rename action, by unselecting the name field. **Invalid Name**: If the new name contains unsupported characters, the system notifies the user and requests a valid name.
Assumptions	There are one or more documents available in the "Documents" tab for renaming.

4.2.21 Use-Case: Delete Document

ID	UC21
Name	Delete Document
Description	This use-case describes how the user deletes a document in the "Documents" tab.
Actors	User (primary actor)
Trigger	The user presses and holds on a document in the "Documents" tab and selects the "Delete" option.
Preconditions	The selected document is listed in the "Documents" tab.
Postconditions	The document is removed from the list in the "Documents" tab and deleted from storage.
The Normal Flow	1. The user navigates to the "Documents" tab and locates a document. 2. The user presses and holds on the document to open the options menu. 3. The user selects "Delete" from the menu. 4. The system prompts the user to confirm the deletion. 5. The user confirms, and the system deletes the document from the app.
Alternative Flow	N/A
Exception	**Cancel Deletion:** The user cancels the deletion action, and the document remains unchanged in the "Documents" tab. **Deletion Error**: If an error occurs during deletion, the system notifies the user, and the document remains in the "Documents" tab.
Assumptions	There is one or more documents available in the "Documents" tab for deletion.

4.2.22 Use-Case: Search Document

ID	UC22
Name	Search Document
Description	This use case describes how the user searches for a document by entering keywords in the search bar at the top of the "Documents" tab.
Actors	User (primary actor)
Trigger	The user enters keywords into the search bar at the top of the "Documents" tab.
Preconditions	The "Documents" tab displays a list of documents, and the search bar is empty at the top.
Postconditions	The list of documents is filtered to show only those that match the search keywords.
The Normal Flow	1. The user navigates to the "Documents" tab. 2. The user enters a keyword or phrase into the search bar. 3. The system filters the list of documents to display only those matching the search criteria. 4. The user views the filtered list of documents.
Alternative Flow	N/A
Exception	**No Matches:** If no documents match the search keywords, the system displays a "No results found" message. **Input Error:** If the search input contains unsupported characters, the system notifies the user to enter a valid keyword.
Assumptions	There are one or more documents available in the "Documents" tab for searching.

EchoText

4.2.23 Use-Case: Search Audio

ID	UC23
Name	Search Audio
Description	This use case describes how the user searches for a generated audio file by entering keywords in the search bar at the top of the "Generated" tab.
Actors	User (primary actor)
Trigger	The user enters keywords into the search bar at the top of the "Generated" tab.
Preconditions	The "Generated" tab displays a list of audio files, and the search bar is empty at the top.
Postconditions	The list of audio files is filtered to show only those that match the search keywords.
The Normal Flow	1. The user navigates to the "Generated" tab. 2. The user enters a keyword or phrase into the search bar. 3. The system filters the list of audio files to display only those matching the search criteria. 4. The user views the filtered list of audio files.
Alternative Flow	N/A
Exception	**No Matches:** If no audio files match the search keywords, the system displays a "No results found" message. **Input Error**: If the search input contains unsupported characters, the system notifies the user to enter a valid keyword.
Assumptions	There are one or more generated audio files available in the "Generated" tab for searching.

4.2.24 Use-Case: Change Model

ID	UC24
Name	Change Model
Description	This use case describes how the user selects a different TTS model for speech generation through the settings menu.
Actors	User (primary actor)
Trigger	The user navigates to the "Settings" tab, selects "Speech Settings," and then "Select Model" to view a list of available models.
Preconditions	The default model is displayed in the "speech Settings" page.
Postconditions	The new selected model is applied and displayed in the "Speech settings" page and that model will be used for speech generation.
The Normal Flow	1. The user navigates to the "Settings" tab. 2. The user selects "Speech Settings" and then "Select Model." 3. The system displays a list of available TTS models. 4. The user selects a model from the list. 5. The system saves the selected model as the active TTS model.
Alternative Flow	N/A
Exception	Model Load Error: If the selected model fails to load, the system notifies the user and reverts to the previously active model.
Assumptions	There are multiple TTS models available in the app for the user to choose from.

4.2.25 Use-Case: Adjust Inference Speed

ID	UC25
Name	Adjust Inference Speed
Description	This use case describes how the user adjusts the inference speed of the Text-to-Speech (TTS) processing to optimize performance.
Actors	User (primary actor)
Trigger	The user selects the "Inference Speed" setting in the app.
Preconditions	The user has opened the "Settings" page.
Postconditions	The inference speed setting is adjusted, and the TTS model runs at the selected speed.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects the "Inference Speed" option. 3. The system displays a speed adjustment slider or options. 4. The user adjusts the slider or selects a speed option (e.g., Low, Medium, High). 5. The system saves the new inference speed setting and applies it to future TTS processing.
Alternative Flow	If the user cancels the adjustment, the system retains the previous speed setting.
Exception	Incompatible Speed: If the selected speed is not supported on the device, the system notifies the user and does not apply the change.
Assumptions	N/A

4.2.26 Use-Case: Adjust Inference Pitch

ID	UC 26
Name	Adjust Inference Pitch
Description	This use case describes how the user adjusts the pitch of the synthesized voice for customized output.
Actors	User (primary actor)
Trigger	The user selects the "Inference Pitch" setting in the app.
Preconditions	The user has opened the "Settings" page.
Postconditions	The pitch setting is adjusted, and future TTS output uses the new pitch.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects the "Inference Pitch" option. 3. The system displays a pitch adjustment slider. 4. The user adjusts the slider to increase or decrease the pitch. 5. The system saves the new pitch setting and applies it to future TTS output.
Alternative Flow	If the user cancels the adjustment, the system retains the previous pitch setting.
Exception	Invalid Pitch: If the selected pitch range is unsupported, the system notifies the user.
Assumptions	N/A

4.2.27 Use-Case: Change Application Theme

ID	UC 27
Name	Change Application Theme
Description	This use case describes how the user changes the app's visual theme, such as switching between light and dark modes.
Actors	User (primary actor)
Trigger	The user selects the "Theme" option in the app settings.
Preconditions	The user has opened the "Settings" page.
Postconditions	The app's theme is changed and applied immediately.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects the "Theme" option. 3. The system displays available theme options (e.g., Light, Dark, System Default). 4. The user selects a theme. 5. The system applies the selected theme to the app's interface and saves the preference.
Alternative Flow	If the user selects "System Default," the app theme adjusts to the device's settings.
Exception	Theme Error: If the theme fails to apply, the system reverts to the previous theme and notifies the user.
Assumptions	N/A

4.2.28 Use-Case: Create Voice Profile

ID	UC 28
Name	Create Voice Profile
Description	This use case describes how the user creates a new voice profile with custom settings for pitch, speed, and model selection.
Actors	User (primary actor)
Trigger	The user selects the "Create Voice Profile" option.
Preconditions	The user has accessed the "Settings" page.
Postconditions	A new voice profile is created and saved.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects "Create Voice Profile." 3. The system prompts the user to configure settings such as pitch, speed, and model. 4. The user customizes the settings and names the profile. 5. The user saves the profile. 6. The system creates and stores the new voice profile.
Alternative Flow	If the user cancels the creation process, no profile is created.
Exception	Profile Creation Error: If there is an issue saving the profile, the system notifies the user.
Assumptions	N/A

4.2.29 Use-Case: Delete Voice Profile

ID	UC 29
Name	Delete Voice Profile
Description	This use case describes how the user deletes an existing voice profile.
Actors	User (primary actor)
Trigger	The user selects the "Delete" option for a voice profile.
Preconditions	The user has at least one voice profile created.
Postconditions	The selected voice profile is permanently deleted.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects a voice profile from the list. 3. The user chooses the "Delete" option. 4. The system prompts the user to confirm the deletion. 5. The user confirms, and the system deletes the profile.
Alternative Flow	If the user cancels the confirmation, the profile is not deleted.
Exception	Deletion Error: If the system fails to delete the profile, it notifies the user.
Assumptions	There is already more than one voice profile in the queue.

4.2.30 Use-Case: Edit Voice Profile

ID	UC 30
Name	Edit Voice Profile
Description	This use case describes how the user modifies settings for an existing voice profile, such as changing the voice pitch, speed, or model selection.
Actors	User (Primary Actor)
Trigger	The user selects the "Edit" option for a voice profile.
Preconditions	The user has at least one voice profile saved.
Postconditions	The updated voice profile settings are saved and applied.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects a voice profile from the list. 3. The system displays the current settings of the selected profile. 4. The user modifies the desired settings (e.g., pitch, speed, model). 5. The user saves the changes. 6. The system updates and applies the modified settings to the voice profile.
Alternative Flow	If the user cancels the changes, the system reverts to the original settings.
Exception	Invalid Settings: If the settings are not supported, the system notifies the user and does not save the changes.
Assumptions	The voice profile is already in the queue.

4.2.31 Use-Case: Enter/Validate Password

ID	UC31
Name	Enter/Validate Password
Description	This use case describes how the user enters a password for secure access to the application or to specific features.
Actors	User (primary actor)
Trigger	The user attempts to access a secure section of the app.
Preconditions	The password protection feature is enabled.
Postconditions	The user gains access upon successful validation.
The Normal Flow	1. The user attempts to access a secure feature. 2. The system prompts the user to enter the password. 3. The user enters the password. 4. The system validates the password. 5. If valid, the user gains access to the feature.
Alternative Flow	If the password is incorrect, the system displays an error and prompts for re-entry.
Exception	Too Many Failed Attempts: The system locks access temporarily and informs the user.
Assumptions	The user has already created a password.

4.2.32 Use-Case: Add Password

| ID	UC 32
Name	Add Password
Description	This use case describes how the user adds a password to secure access to the app or specific features.
Actors	User (primary actor)
Trigger	The user chooses to enable password protection in the settings.
Preconditions	The user has not already set a password for the app.
Postconditions	A password is added and required for future access.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects the "Add Password" option. 3. The system prompts the user to enter and confirm a new password. 4. The user enters and confirms the password. 5. The system saves the password securely.
Alternative Flow	If the passwords do not match, the system prompts the user to re-enter them.
Exception	Weak Password: If the password does not meet security requirements, the system asks for a stronger password.
Assumptions	N/A

4.2.33 Use-Case: Change Password

| ID	UC 33
Name	Change Password
Description	This use case describes how the user changes their existing password to a new one.
Actors	User (primary actor)
Trigger	The user selects the "Change Password" option.
Preconditions	The user has set a password and knows the current one.
Postconditions	The password is updated and saved.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects "Change Password." 3. The system prompts the user to enter the current password. 4. The user enters the current password, and the system validates it. 5. The user is prompted to enter and confirm the new password. 6. The system updates and saves the new password.
Alternative Flow	If the current password is incorrect, the system does not allow proceeding and prompts the user to try again.
Exception	Weak Password: If the new password is too weak, the system asks for a stronger password.
Assumptions	The user has memorized their current password.

4.2.34 Use-Case: Remove Password

| ID	UC 34
Name	Remove Password
Description	This use case describes how the user removes the password requirement from the app or specific features.
Actors	User (primary actor)
Trigger	The user chooses to disable password protection in the settings.
Preconditions	The user has a password set for the app.
Postconditions	Password protection is removed, and access is no longer restricted.
The Normal Flow	1. The user navigates to the "Settings" page. 2. The user selects "Remove Password." 3. The system prompts the user to enter the current password for security. 4. The user enters the current password. 5. The system validates the password and removes the password protection.
Alternative Flow	If the password is incorrect, the system denies the request and prompts re-entry.
Exception	Security Confirmation Error: If the password is entered incorrectly multiple times, the system temporarily restricts access.
Assumptions	The user remembers the current password for verification.

4.3 sequence diagram

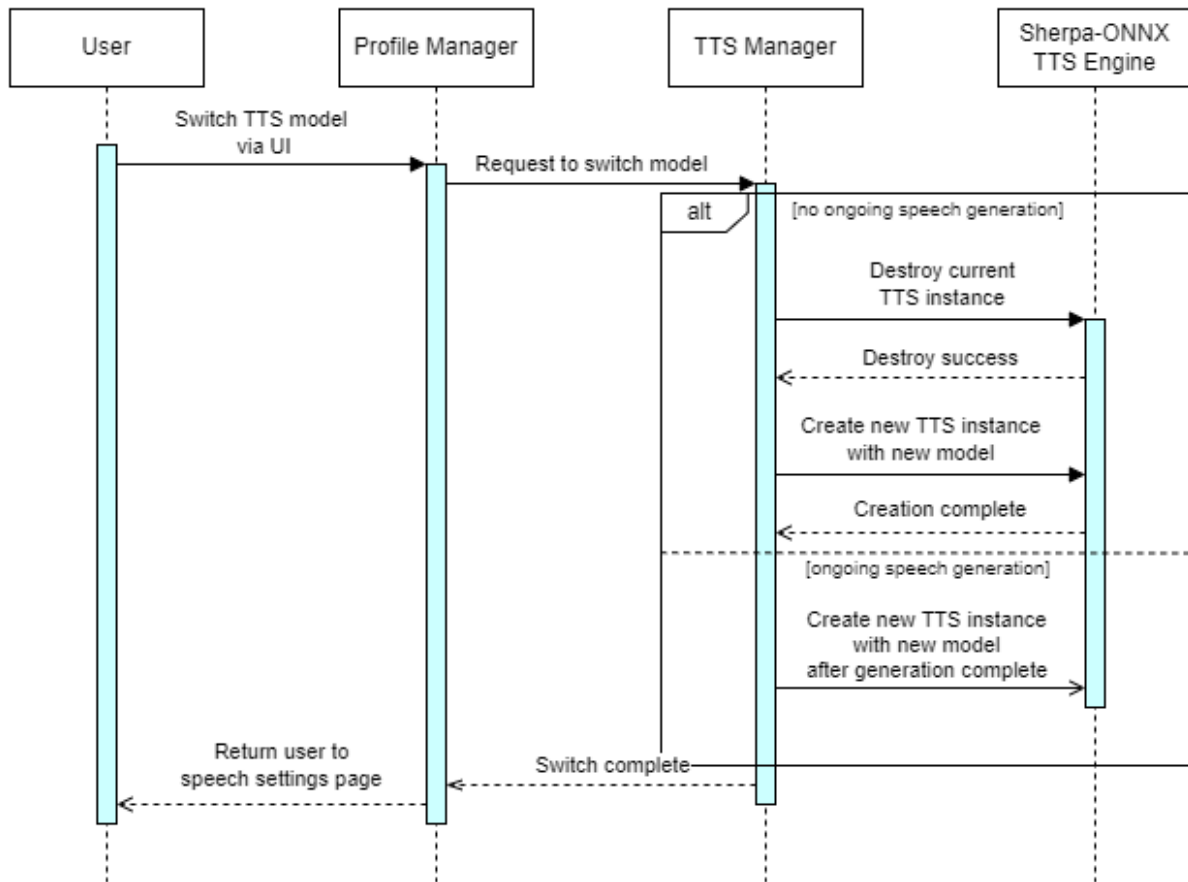### 4.3.1 Convert text into speech
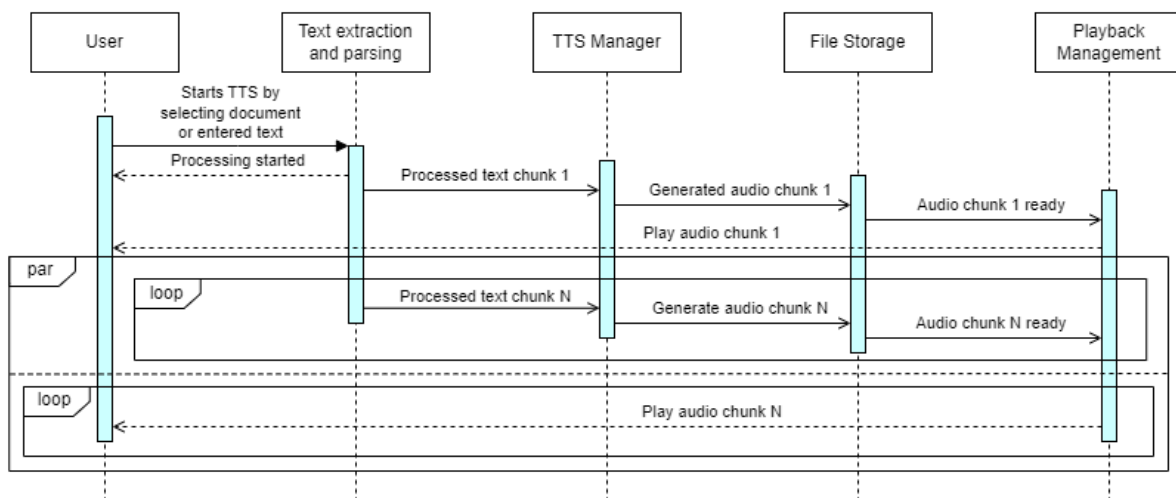
4.3.2 Batch Inference
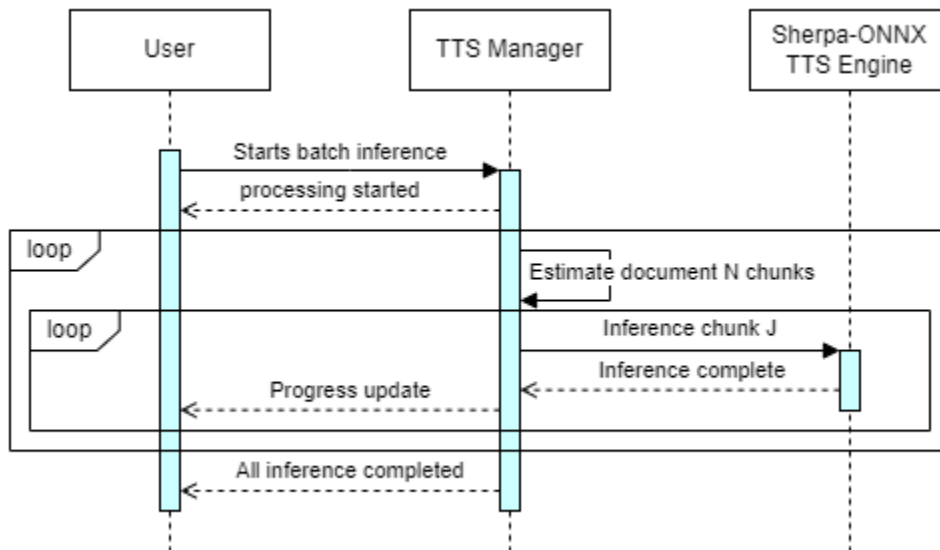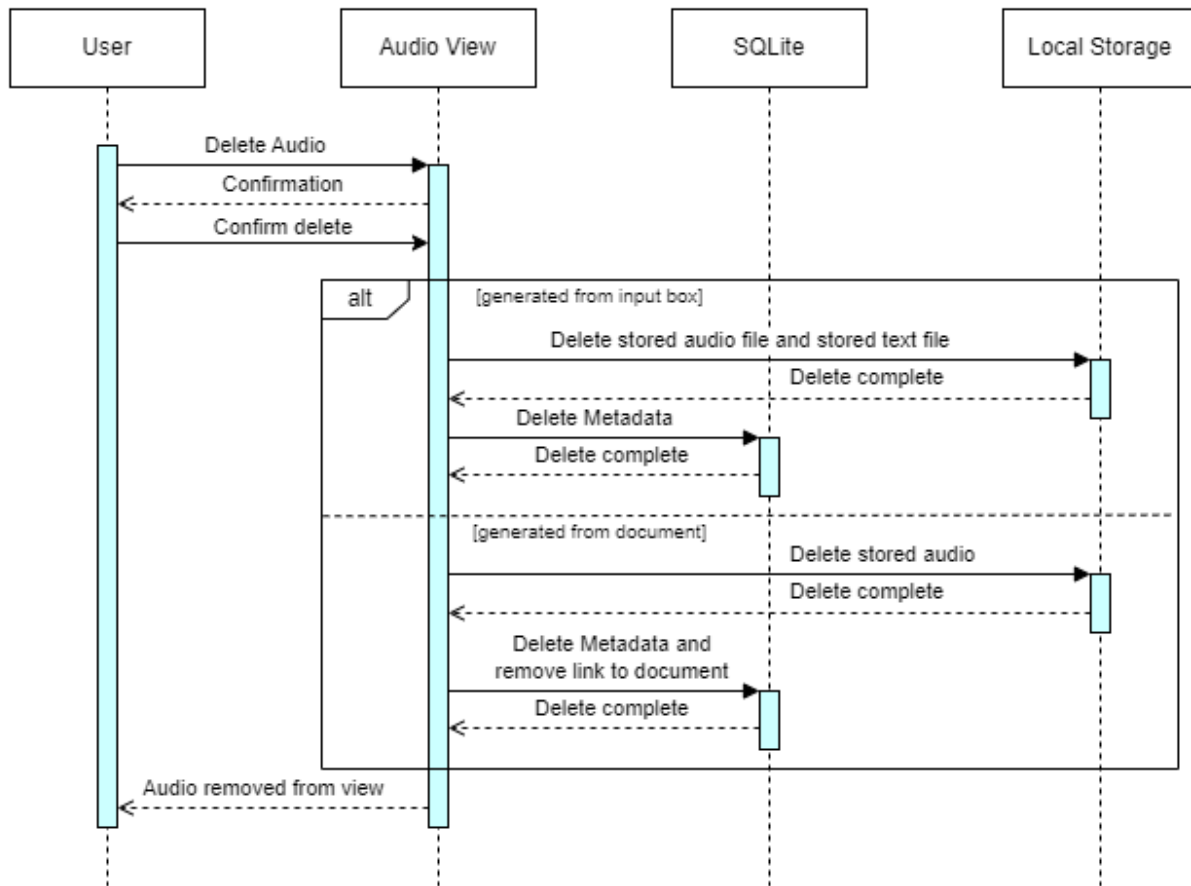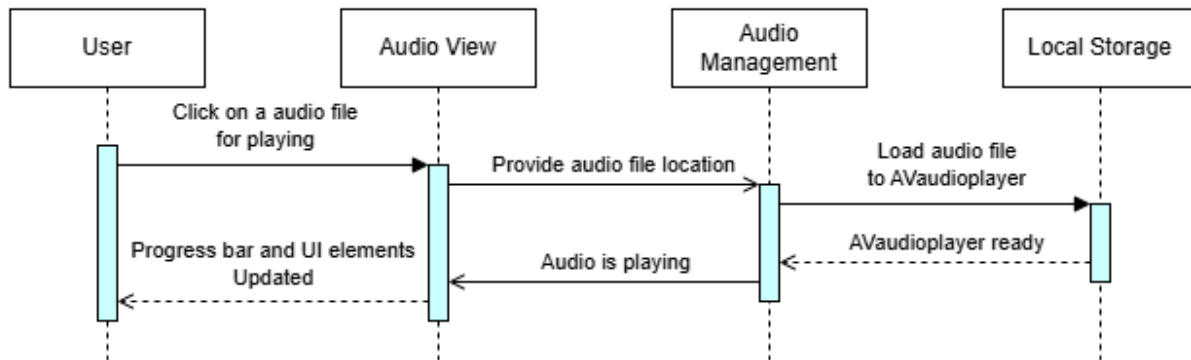

4.3.3 Terminate Speech Generation
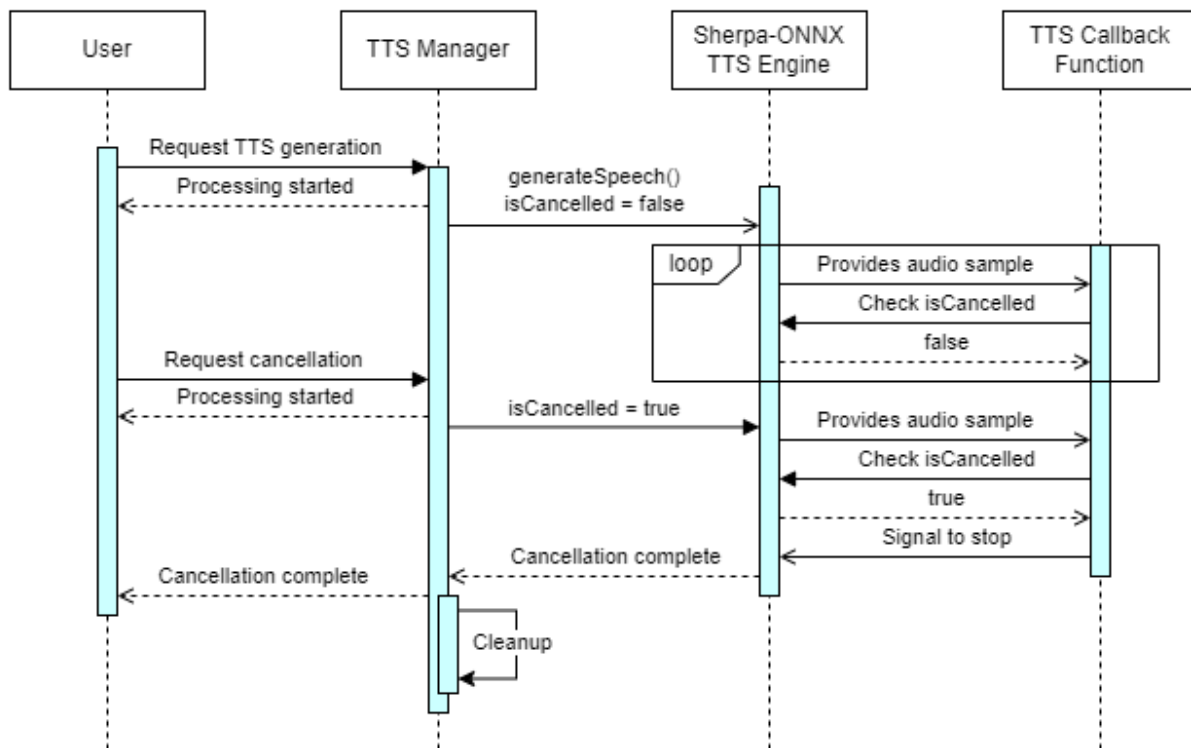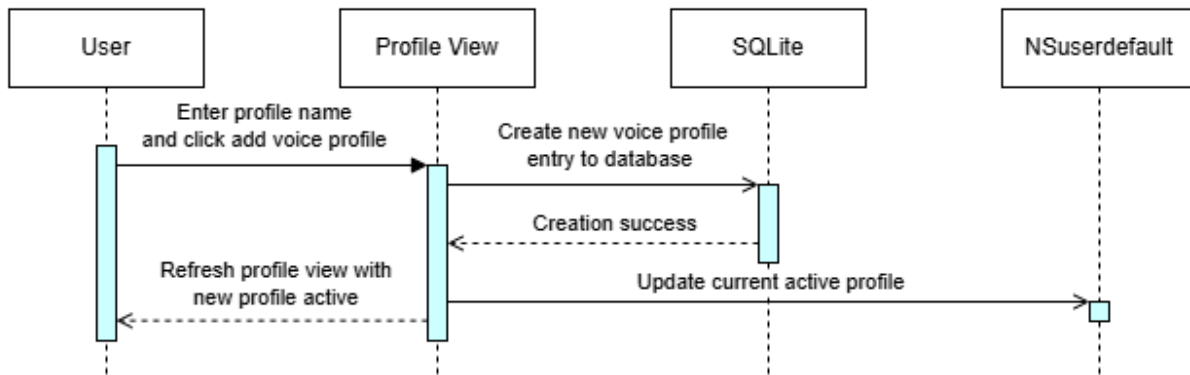
4.3.4 Change Model

4.3.5 Delete Audio

4.3.6 Select and Play an Audio

4.3.7 Create Voice Profile

4.3.8 Add Password
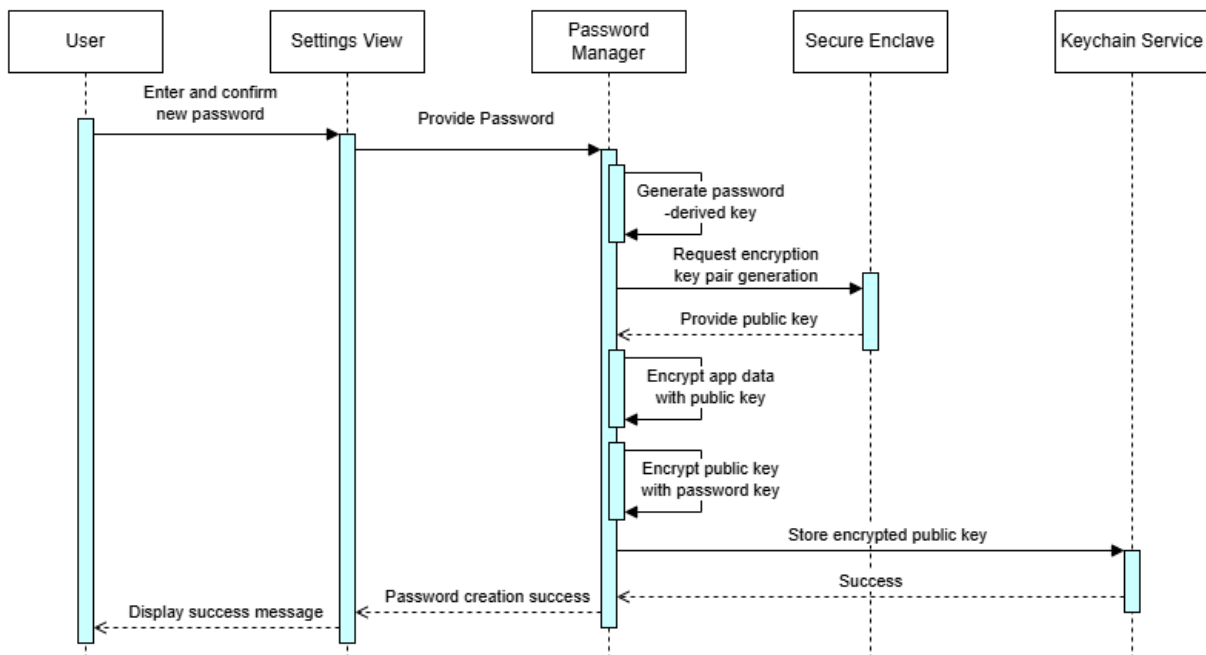
4.4 data flow diagram

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.

4.5 Database Design

audios	♫
Id	integer pk
Name	text
FilePath	text
DateGenerated	datetime
Model	text
Pitch	text
Speed	text
DocumentId	integer fk

documents	📄
Id	integer pk
DocumentName	text
UploadDate	datetime
FilePath	text
Flag	boolean

modelProfiles	⚙
Id	integer pk
ProfileName	text
Pitch	real
Speed	real
Model	text
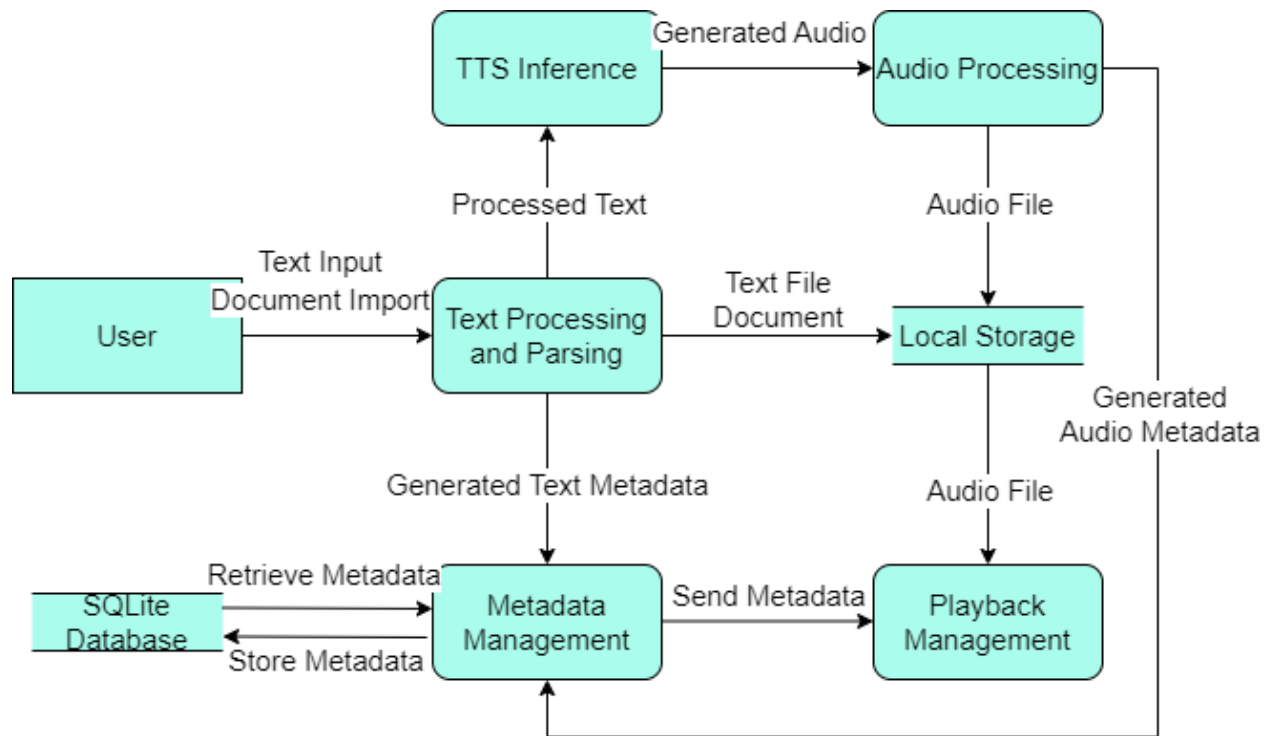
4.6 Class diagram
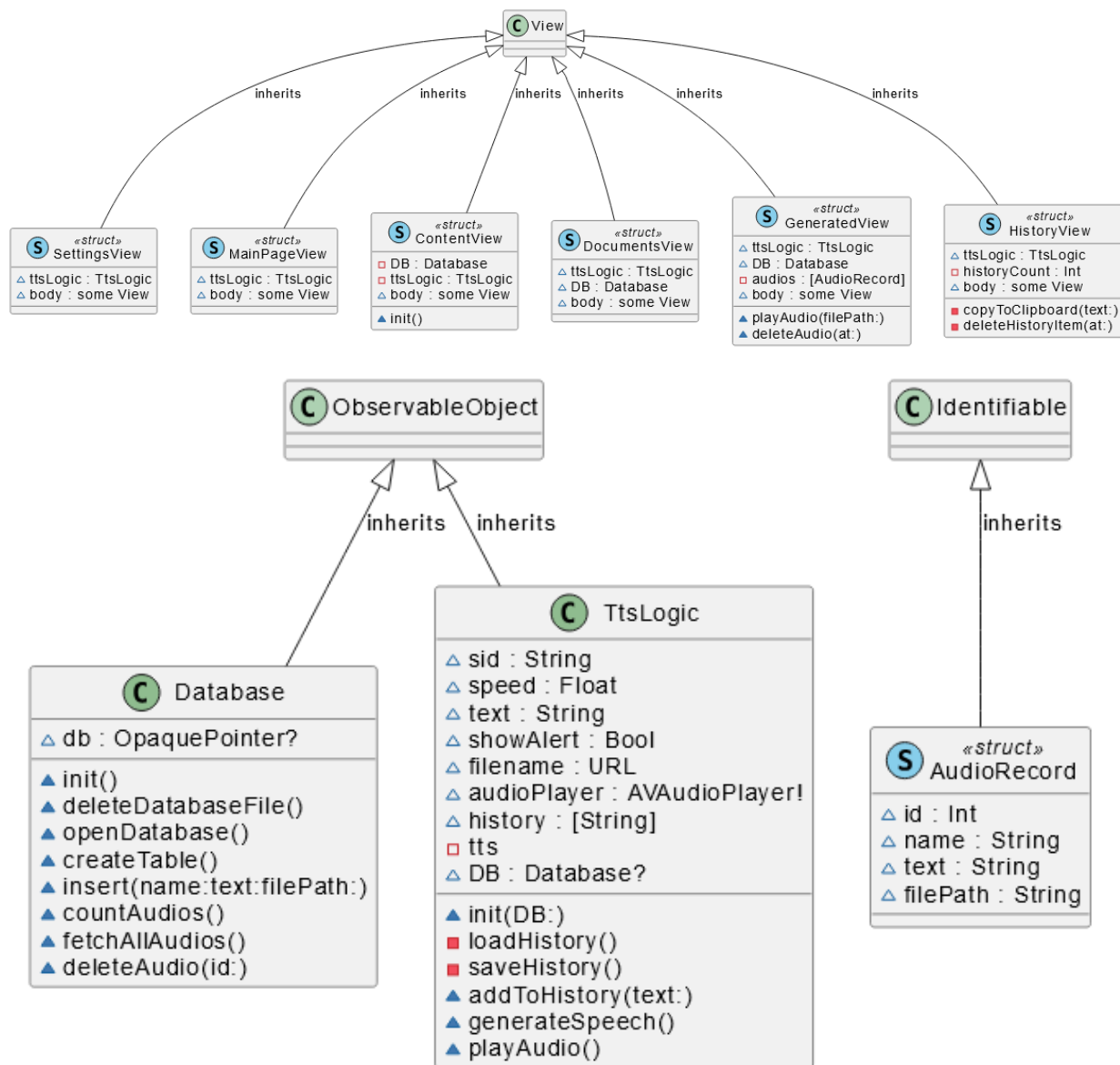
4.7 Application Program Interfaces

This section describes the key Application Program Interfaces (APIs) utilized in the EchoText application, focusing on the SQLite API and the Sherpa-ONNX API. These APIs are integral to the application's functionality, enabling efficient data management and on-device Text-to-Speech (TTS) processing, all while operating entirely offline.

4.7.1 SQLite Overview

EchoText leverages SQLite, a lightweight, serverless, self-contained relational database engine, to manage and store user data locally on the device. The SQLite API provides a set of C-language functions that allow the application to execute SQL commands for creating, querying, updating, and deleting data within a local database file.

Usage in EchoText/Data Storage Requirements
- sqlite3_open(): Opens a connection to the SQLite database file.
- sqlite3_prepare_v2(): Prepares an SQL statement for execution.
- sqlite3_step(): Executes a prepared statement.
- sqlite3_finalize(): Deletes a prepared statement object, releasing its resources.
- sqlite3_bind_text(): Binds a text value to a parameter in a prepared statement.
- sqlite3_bind_int(): Binds an integer value to a parameter in a prepared statement.
- sqlite3_column_int(): Retrieves integer data from a result column in a query result.
- sqlite3_column_text(): Retrieves text data from a result column in a query result.
- sqlite3_errmsg(): Returns a human-readable string describing the most recent error.
- sqlite3_close(): Closes the database connection (Note: Not explicitly called in your code but recommended to include in practice to close the database when done).

4.7.2 Sherpa-ONNX API Overview

Sherpa-ONNX is a real-time neural network-based TTS engine optimized for on-device inference. It provides an API for processing text inputs and generating corresponding audio outputs. The API is designed to be lightweight and efficient, suitable for mobile devices with limited computational resources.

Usage in EchoText
SherpaOnnxGeneratedAudioCallback(): function used for TTS audio generation. The call back argument lets you decide whether to keep generating more audio or stop.

4.8 User Interface Design

4.8.1 Main Page

This is the primary interface where you can input text and generate speech.
The Main Page is the initial screen you see when you launch EchoText, acting as the central hub for starting text-to-speech generation. You can easily return to the Main Page at any time by selecting the Main Tab (Home Icon) in the menu bar.

Below is a detailed explanation of the components and functionality of this page:

1. Header

- **Logo and App Name:**
  - o Displays the app name ("EchoText") alongside the app logo for easy identification.
- **PDF counter:**
  - o 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":**
  - o Instructional text that prompts you to type or paste your desired text for text-to-speech generation.
- **Word Counter:**
  - o 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:**
  - o 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.
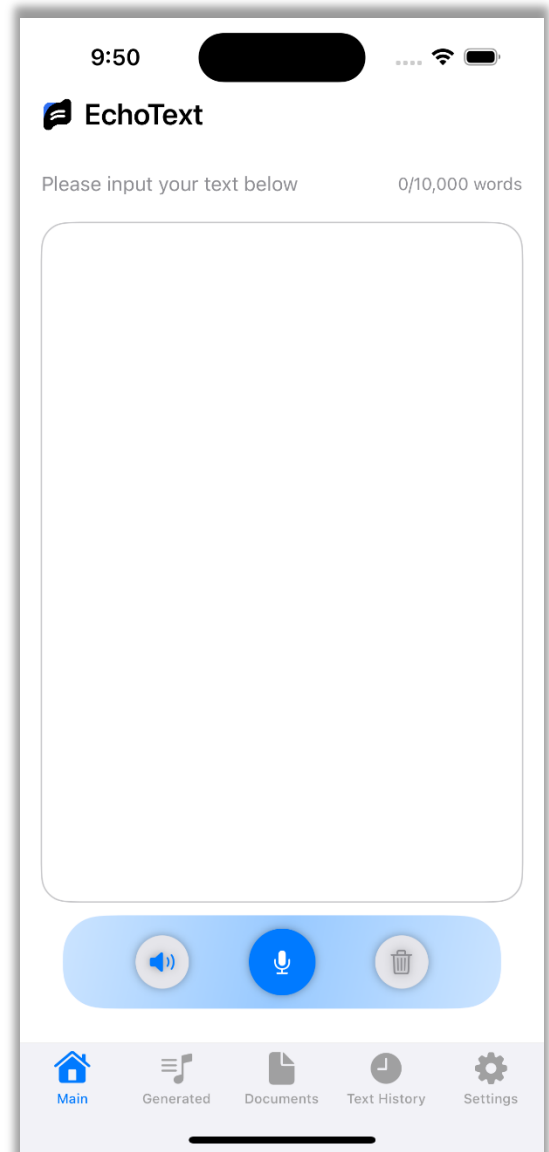
3. Action Buttons and Progress Bar (Bottom Section)
- **Speaker Icon (Mute/Unmute):**
  - o Toggles the audio output of the current text-to-speech generation on or off.
  - o Use this to quickly mute or unmute the playback during or after generation.
- **Microphone Icon (Generate/Terminate):**
  - o Starts the text-to-speech generation process when pressed.
  - o 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:**
  - o Clears all text from the input area when not generating.
  - o 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:**
  - o Appears below the action buttons during text-to-speech generation.
  - o 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.

4.8.2 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.
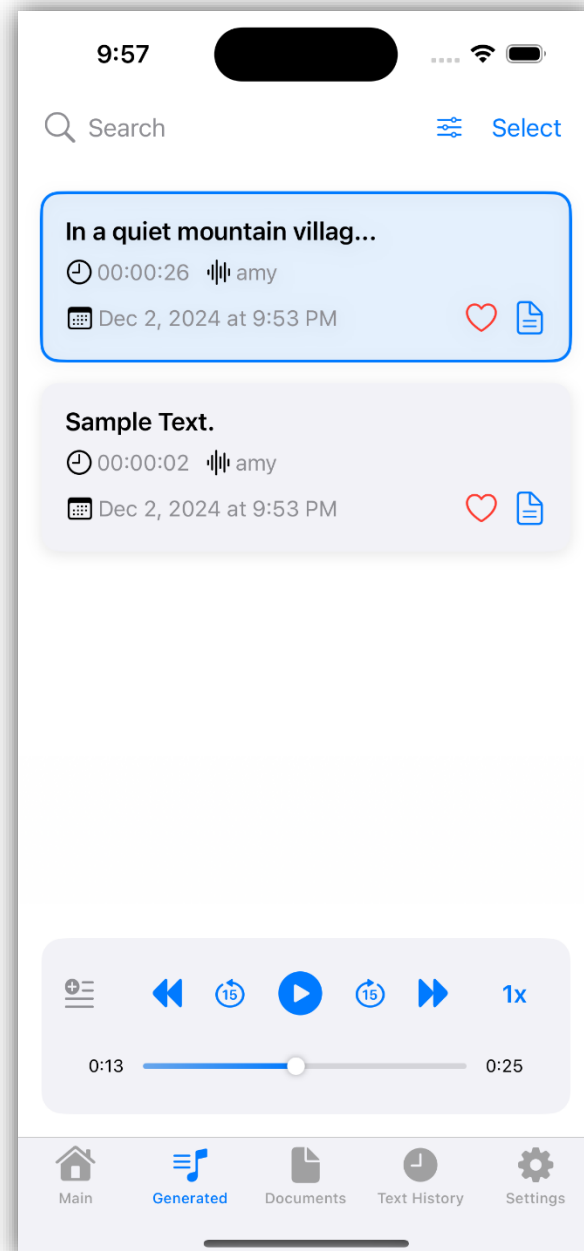
3. Playback Controls
Located below the audio list, the playback controls allow for seamless navigation and playback of your audio files. The controls include:
1. **Queue Icon**: Opens the queue view to manage upcoming audio playback.
2. **Previous Audio**: Skips to the previous audio file in the list.
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**:
    o **Rename**: Edit the name of the audio file.
    o **View Document**: Opens the text used for generating the audio.
    o **Add to Queue**: Adds the audio file to the playback queue.
    o **Share**: Allows you to share the audio file with others.
    o **More Info**: Provides detailed metadata about the audio file.
    o **Delete**: Permanently removes the audio file.

4.8.3 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.
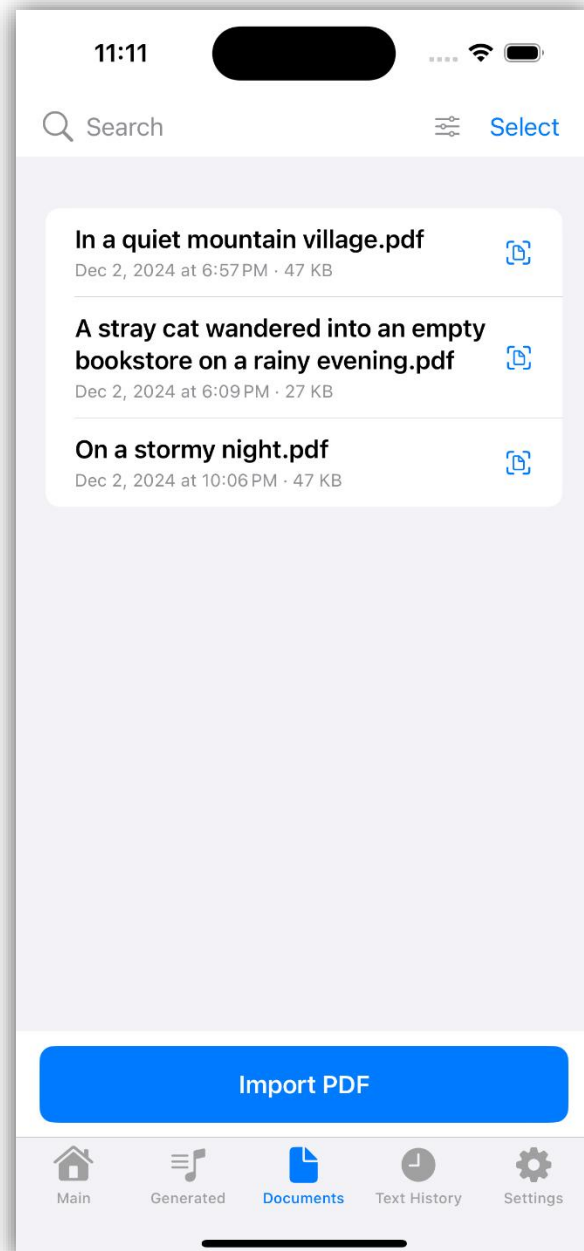
3.1.4 Document selection
- **Select all**: Select all documents that are displayed to the user.
- **Done:** to exit out of selected mode.
- **Select document**: the user can press on the individual document to select. A blue icon with a checkmark will be displayed to let the user know that the document 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 I 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.

4.8.4 Text History Page

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.
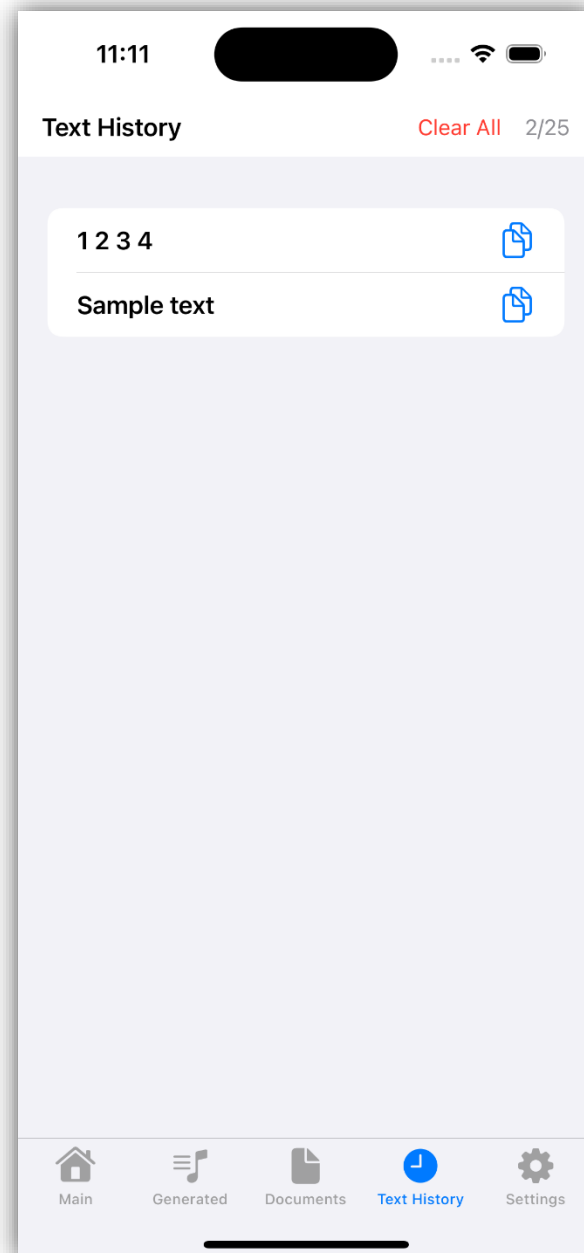
3.1.3 Copy and Delete

- **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.
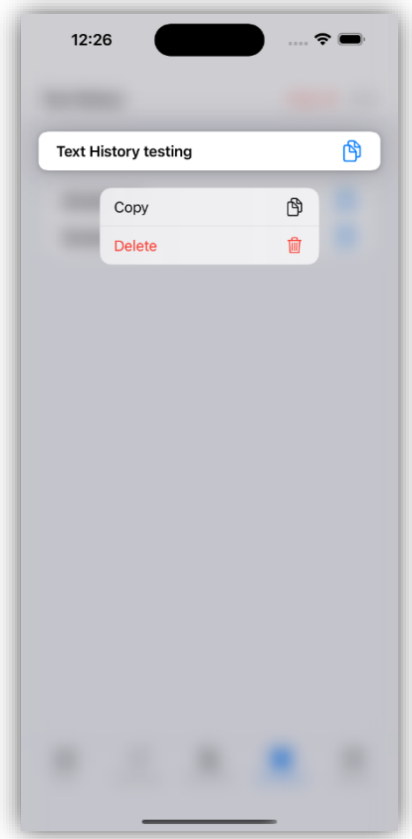
4.8.5 Settings Page

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:
    o 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:
   o A button that brings up a dropdown menu listing available inference models.
3. Speed Slider:
   o A slider to adjust the speaking speed of the TTS engine.
   o Range: 0.5x (slower) to 1.5x (faster), with 1.0x as the default value.
4. Pitch Slider:
   o A button that brings up a dropdown menu listing available inference models.
   o 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:
   o A toggle switch to enable or disable manual theme selection.
2. Light/Dark Theme (Visible only when Override System Theme is enabled):
   o Allows manual switching between light and dark themes.
   o Light Theme: Bright background with dark text.
   o 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.
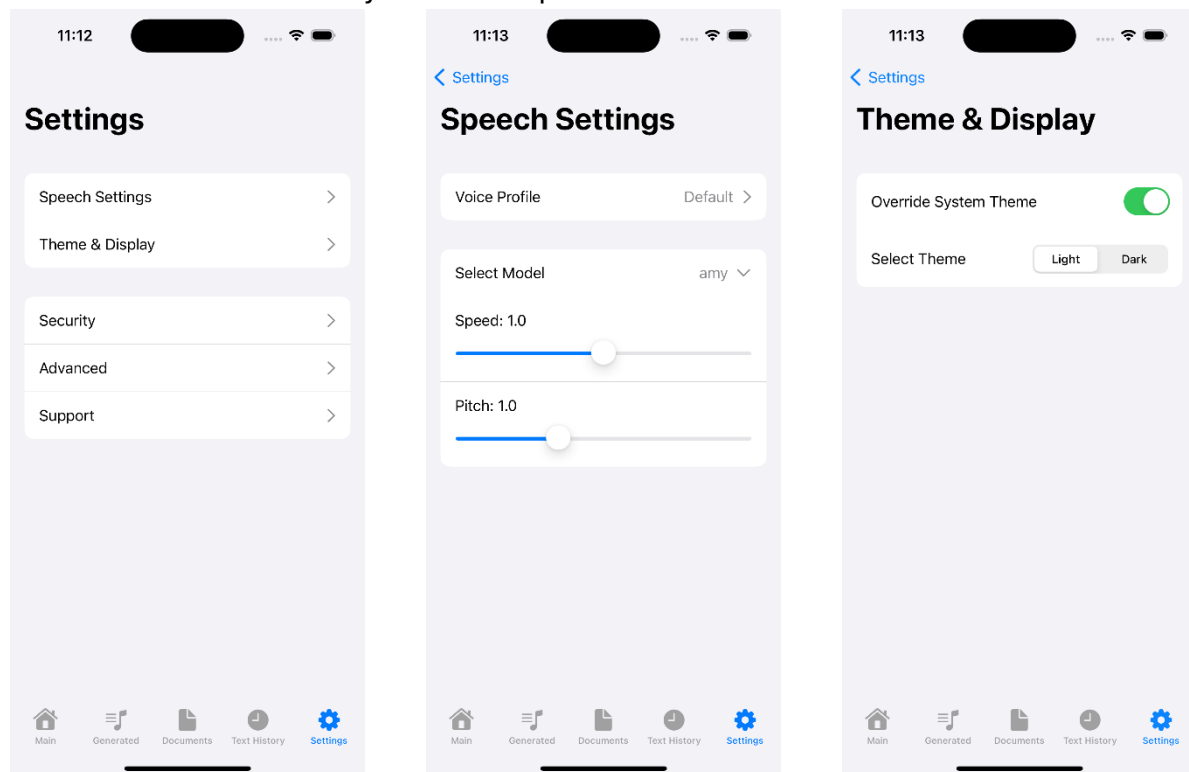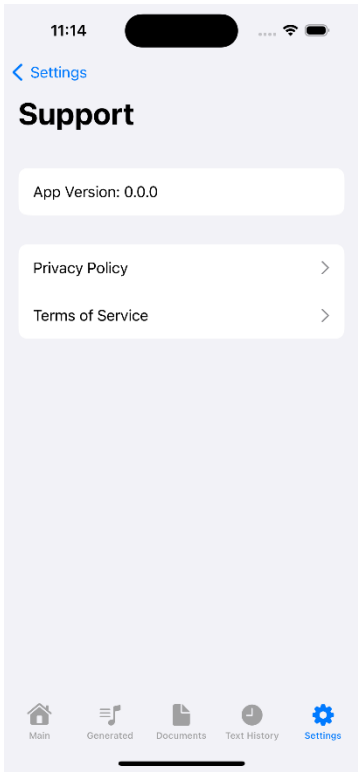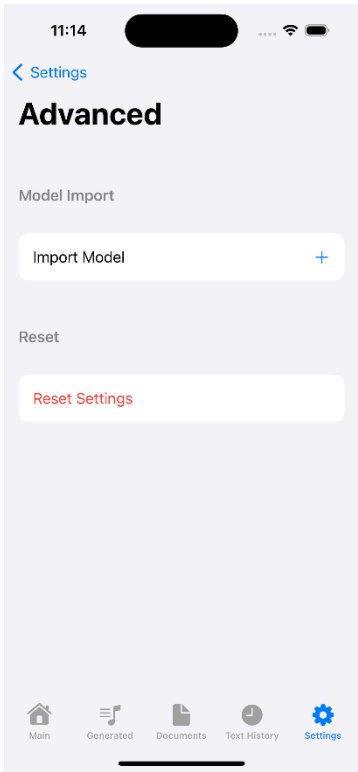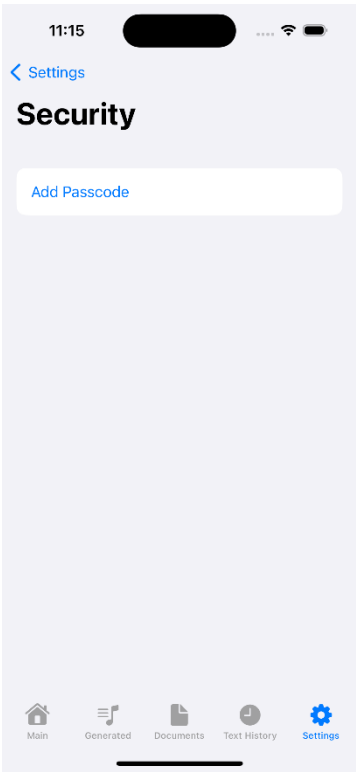
To access the Theme & Display section:
1. Open the application.
2. Tap the gear icon located in the bottom-right corner of the home screen.
3. Select Security from the list of options.

3.5.6 Features in the Security Section

1. Add Passcode:

- o Visible Only When No Passcode is Set.
- o 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:
   - o Visible Only When a Passcode is Set.
   - o A Button to navigate to the Add Passcode screen.
     - Enter your current passcode and new passcode twice to confirm the change.

3. Remove Passcode:
   - o A Button to navigate to the Remove Passcode screen.
     - Enter your current passcode to confirm the action.

Security

11:15

< Settings

Add Passcode

Main | Generated | Documents | Text History | Settings

Advanced

11:14

< Settings

Model Import

Import Model +

Reset

Reset Settings

Main | Generated | Documents | Text History | Settings

Support

11:14

< Settings

App Version: 0.0.0

Privacy Policy >

Terms of Service >

Main | Generated | Documents | Text History | Settings

A. Appendices

A.1 References

NSFileProtectionCompleteUntilFirstUserAuthentication:
https://developer.apple.com/documentation/foundation/nsfileprotectioncompleteuntilfirstuserauthentication
kSecAttrAccessibleWhenUnlocked:
https://developer.apple.com/documentation/security/ksecattraccessiblewhenunlocked
Keychain:
https://developer.apple.com/documentation/security/keychain-services
NSFileProtectionComplete:
https://developer.apple.com/documentation/foundation/nsfileprotectioncomplete
Secure Enclave:
https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web

A.2 Key Terms

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 int he 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.