Ribbit

A Cost-Effective iOS Hearing Aid App

Design Document v3.1

Computer Science Department
Texas Christian University
May 2, 2016

Revision Signatures

By signing the following, the team member is stating that he has read the entire document and has verified that the information contained within this document is accurate, relevant to the project, and void of errors.

Name	Signature	Date
Duy Dang		
Robert Kern		
Esteban Kleckner		

Revision History

Version	General Description of Changes	Date Completed
V1.0	Initial Draft	12/5/15
V2.0	Fixed minor formatting	1/31/16
V3.0	Updated the title, general constraints, System Architecture, Prototypes, and use cases	5/1/16
V3.1	Updated Glossary, and grammatical, syntactical errors	5/2/16

Contents

1 Introduction

1.1 Purpose

This document is a detailed description of the design of the Ribbit application. This document is intended to give the user an overview of the design of the Ribbit application, including design constraints, system architecture, prototypes of the interface, and Unified Modeling Language (UML) class models describing the interaction of different pieces of the application.

1.2 Project Overview

The objective of this project is to create an iOS application that functions similarly to a physical hearing aid device, but at a fraction of the cost. The application will work within the federal regulations concerning hearing aid use. The aim of the application is to correct the user's hearing by changing the sound to fit their inability to hear certain frequencies.

1.3 Overview of Document

- Section 2: This section details the design constraints of the application.
- Section 3: This section details the system architecture of the application.
- Section 4: This section details the interaction of the different sections of the application.
- Section 5: This section details the interface prototypes of the application.
- Section 6: This section defines the terms used within this document.

2 Design Constraints

2.1 Assumptions and Dependencies

We assume that the user has a valid hearing deficiency, has obtained a hearing prescription from a professional audiologist, has access to an Apple iPhone running at a minimum iOS 7, and has installed the Ribbit application.

2.2 General Constraints

- All development must be finished by the end of April, 2016.
- The application, and all supporting files, must be able to be installed on an iPhone.
- The application requires that a stereo Bluetooth headset or headphones be used.
- Processing delay time is below 50 milliseconds.
- The signal amplification and filtering must be customized to each user based on his/her hearing prescription.
- The possible sound artifacts caused from sound modification need to be at a comfortable level.

2.3 Development Environment

All development was completed on:

- Apple iMac Retina Display
- Apple MacBook Pro
- Apple Xcode 7.3.0
- Apple iPhone 6/6 Plus running iOS 9

3 System Architecture

This diagram explains the architecture of the overall system, incorporating all parts and moving pieces.

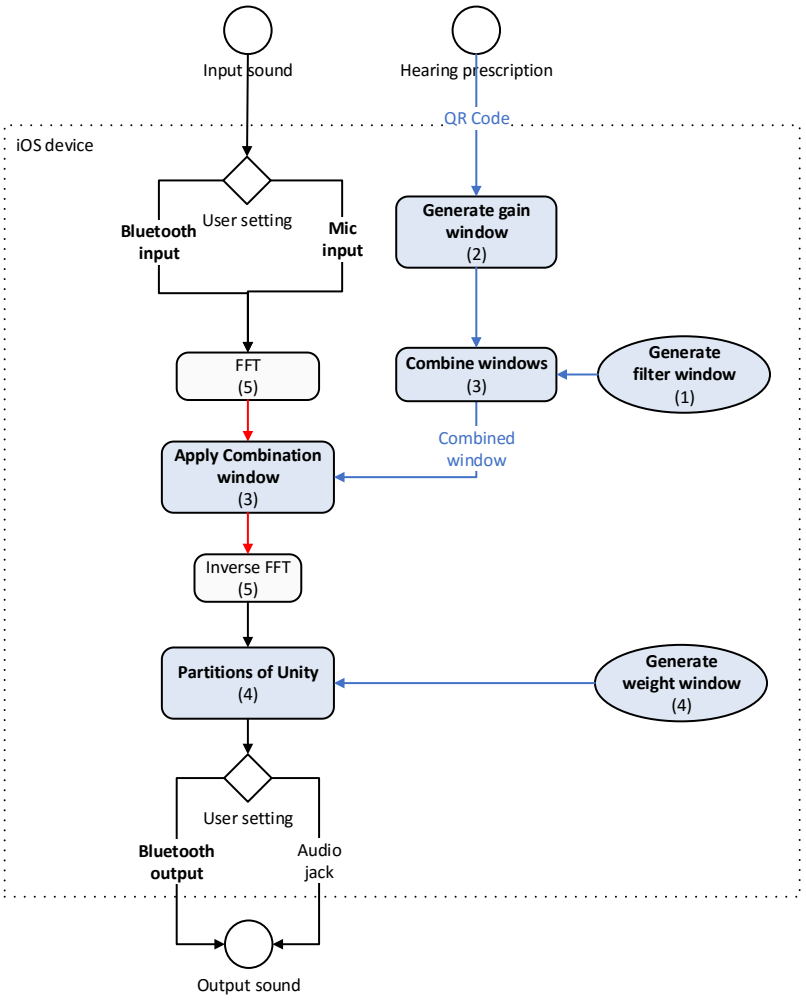


Diagram notations:

- Circles represent input and output
- Diamonds represent decision making
- Round-cornered rectangles and ovals represent processes/activities (the ones with background color are implemented in software)
- Ovals represent processes/activities that are done only once at application's start
- Bold words represent the contribution areas of this project
- Dotted line rectangle represents system boundary
- Black arrows represent sound in time domain
- Red arrows represent sound in frequency domain
- Blue arrows represent other types of input

3.1 Generate Windows

3.1.1 App Activation

Filter window creation: When the application is launched, a filter window is created. The filter window is an infinitely differentiable curve that reduces and removes sound between 4 kHz-24 kHz. The filtering window is designed and used to remove background noises, and information that is unnecessary for understanding human speech. The window can be described as a low pass filter, one that allows the lower end of the frequency to pass through. It has three characteristic parts: the low pass, the transition, and the no-pass. The transition is used to avoid any divide by zero errors, when applying the inverse Fast Fourier Transform.

Weight window creation: In order for sound to be played back to the user with minimal artifacts, we must apply a mathematical approach known as Partitions of Unity, as discussed later. One important part of this approach is the use of a weight window. This window is used to set the weights of two different consecutive signals so that a signal discontinuity does not occur. The window gradually decreases the weight of the first signal while gradually increasing the weight of the second signal, keeping the total weight of the signal equal to one. The process results is an average of the two signals.

3.1.2 Prescription Activation

Gain window creation: The role of gain amplification is to increase the amplitude of each frequency bin according to the specific needs of the user based on his/her hearing prescription. We effectively increase the amplitude of each frequency by multiplying the complex value (both real and imaginary) corresponding to that frequency (returned by the FFT) by a ratio greater than 1.

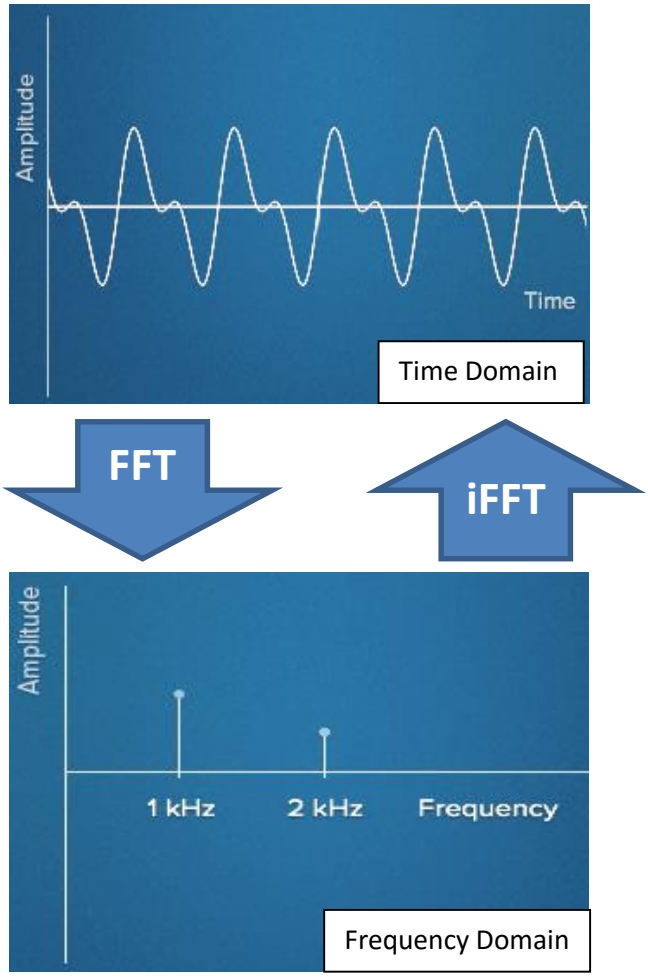
In order to make this operation more efficient, we build a gain window, an array of coefficients, which are multiplied by the input signal in the frequency domain. Most importantly, the gain window will need to be determined based on the user's hearing prescription. A hearing prescription is represented by an audiogram which contains the decibel threshold (the minimum loudness at which that person can detect a frequency) at seven bands (125 Hz, 250 Hz, 500 Hz, 1000 Hz, 2000 Hz, 4000 Hz, and 8000 Hz). Based on this data, the application will be able to determine the amount to boost a frequency bin's value. That amount, ratio, can be calculated by the formula (ratio = 10 ^ (dB to increase/20)).

Combination window creation: The combination window is the product of the Filter and Gain windows. Because of the commutative property of multiplication, we are able to apply the Filter and Gain windows in any order. Therefore, to reduce the computation time necessary to process each signal slice we create the combination window to reduce the number of operations necessary to filter and amplify the signal from two to one. Therefore, also reducing the latency between input capture to when the signal is played through the headphones.

3.2 Signal Processing

3.2.1 Fast Fourier Transform

The sound of the human voice is a combination of multiple signals of varying frequencies. The Fast Fourier Transform function can take an input signal in the time domain (upper graph) then output all of the component frequencies and present them in frequency domain (lower graph) where each component frequency is represented by a frequency bin (in the lower graph we can see 2 frequency bins: 1 kHz and 2 kHz). After our signal is in the frequency domain, we will apply sound modification techniques that will be discussed in the following sections. Once the modifications have been made, an inverse Fast Fourier Transform is applied to reframe the sound data from the frequency domain (lower graph) back to the time domain (upper graph) in an LPCM format that is accepted by the Operating System.

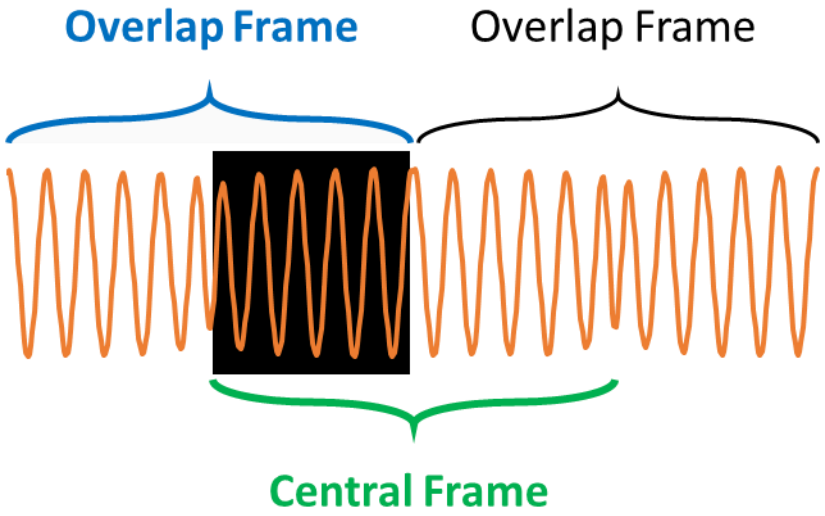


3.2.2 Apply Combination Window

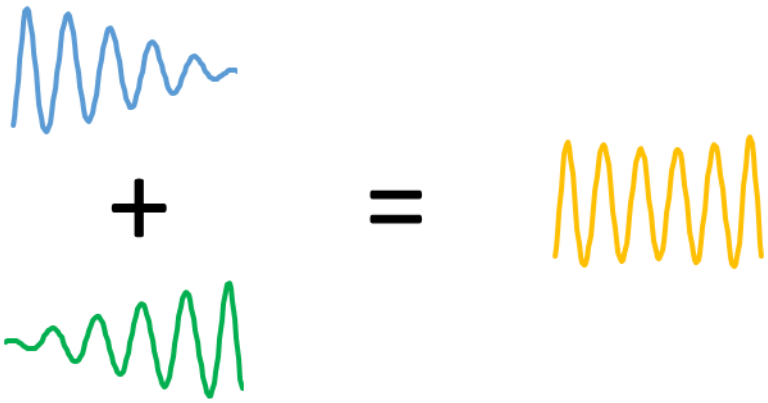
The Combination window, which is the product of the Filter and Gain window, is applied when the signal is in the frequency domain. The signal is applied using a SIMD, single instruction multiple data, call to vDSP_vmul. This operation multiplies the Combination window by the input signal in the frequency domain, and stores the result in the data object used to store the input signal. Once applied, the input signal is then transformed using the inverse FFT, returning the signal to the time domain. The signal is then used by the Transition Smoothing section of the algorithm.

3.2.3 Partitions of Unity

An issue that may arise when modifying signals in frequency domain is the Gibbs Effect, which is caused by the discretization of the Fast Fourier Transform. The Gibbs effects is the cause of the discontinuity in time domain between the end of one frame and the beginning of the next frame, which is the cause of the clicking sound in the output signal, which is an annoyance and a distraction. The Gibbs Effect can be overcome by applying the concept of Partitions of Unity. In layman terms, we will apply a frequency domain modification to not only adjacent signal frames but also overlapping frames.



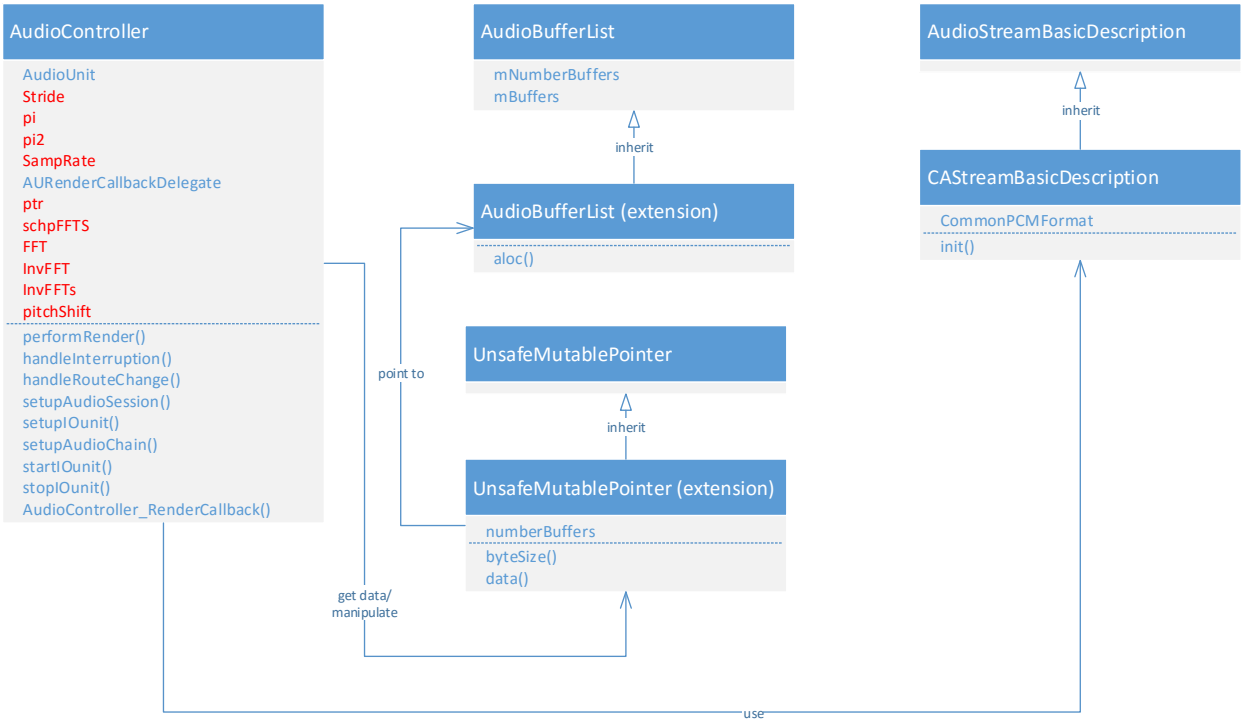
Next, the modified and partially overlapping signals will be blended into each other by gradually reducing the weight of the frame (frame 1) and increasing the weight of the frame (frame 2) that overlaps it (frame 1). The weights will be stored in the Weight window. We will apply the weights to the signal in the time domain in a way that guarantee that the sum of two corresponding weights will be one, in order to preserve the original properties of the signal. The overall effect is an average of the signals, which create a continuous signal across multiple input buffers.



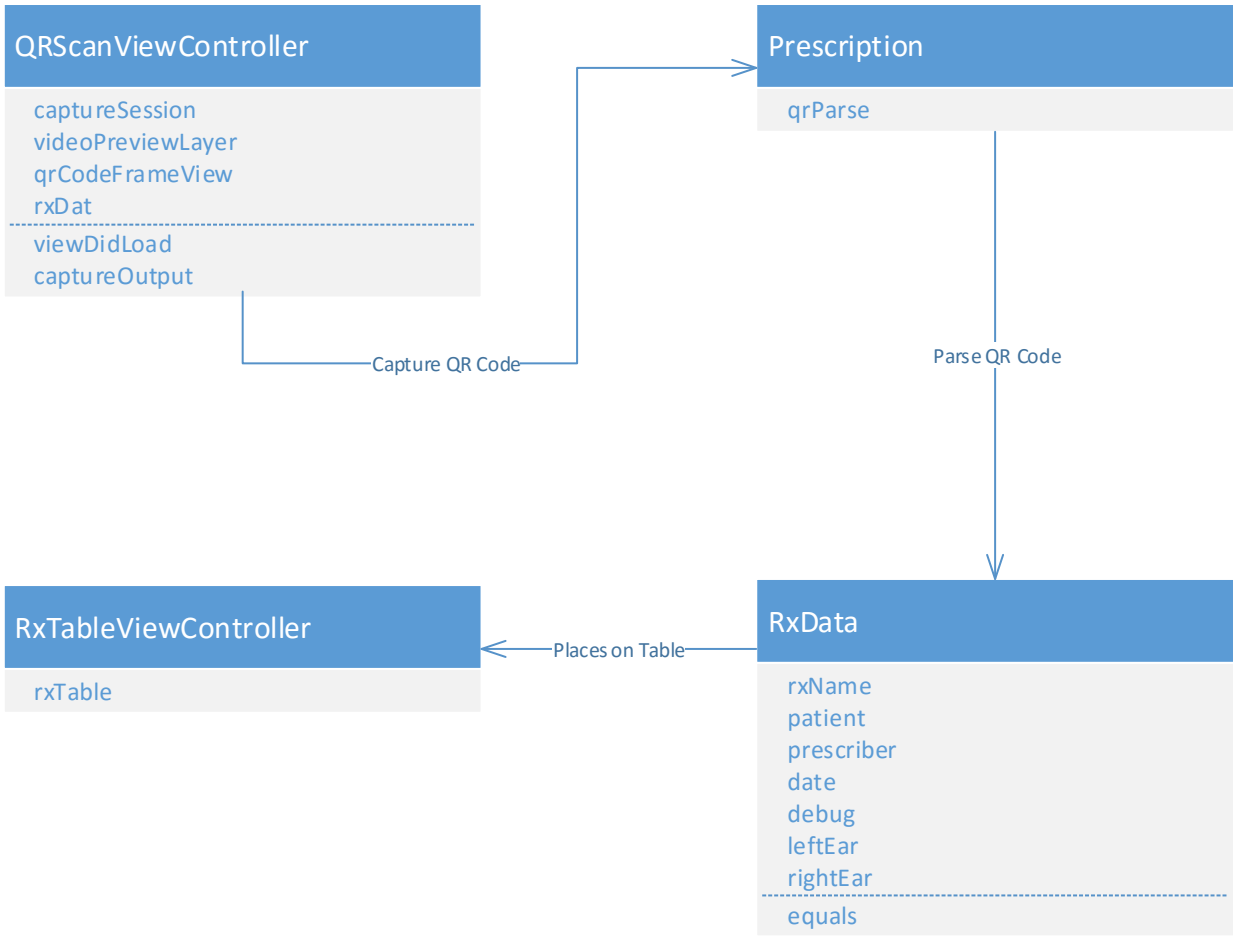
The weight-based blending of signal based on the concept of partition of unity will generate a new time domain signal that will have a very similar shape to the old one (containing the same frequency components at the same magnitudes) but will be continuous (no mismatch at the edge of the signal frame).

4 Class Diagrams

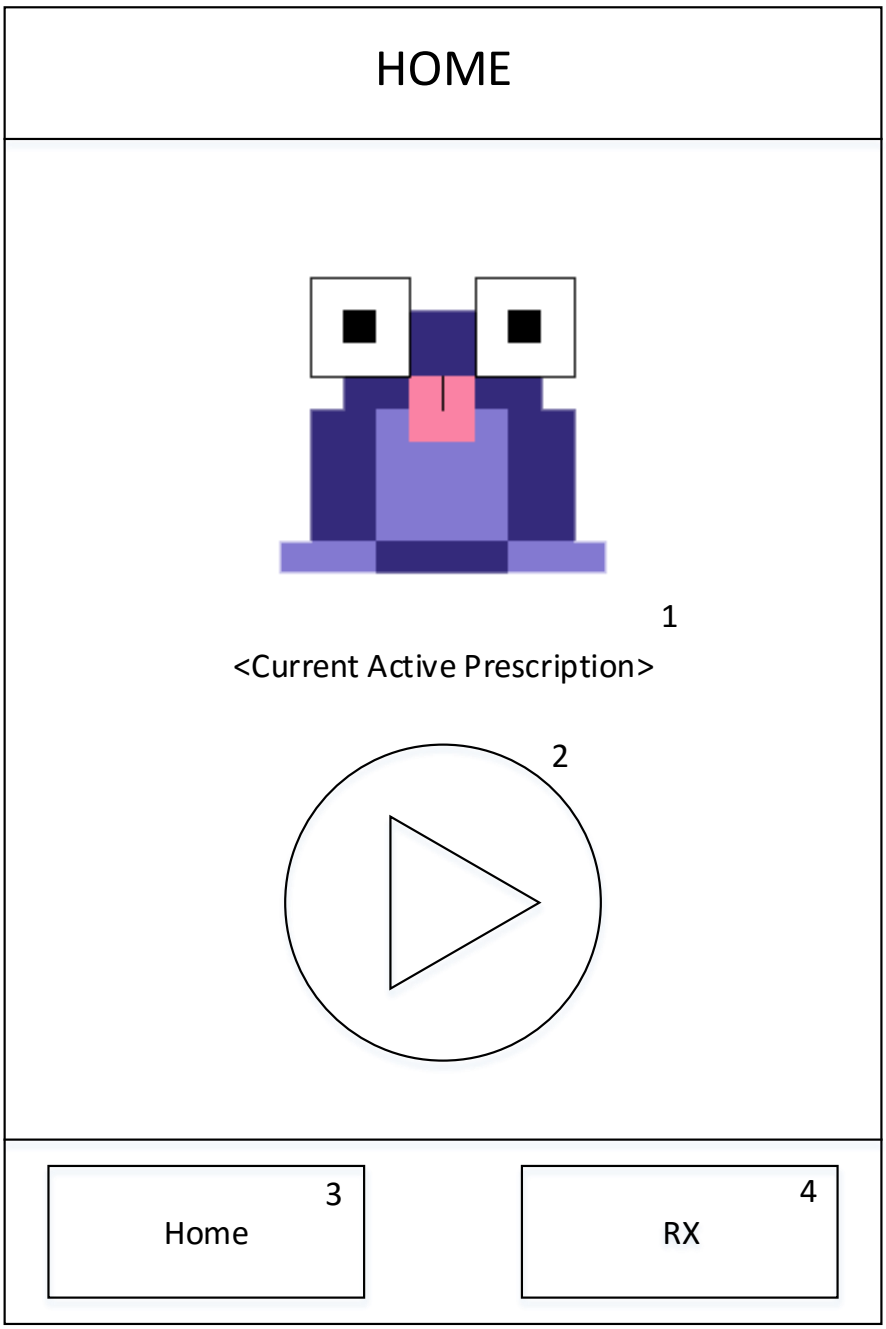
The following class diagram defines the interaction of the classes pertaining to the AudioController class, which is class where sound is taken from the microphone and sent in to the sound engine.



The following class diagram defines the class interaction to read in the QR code and parse it into useable information.

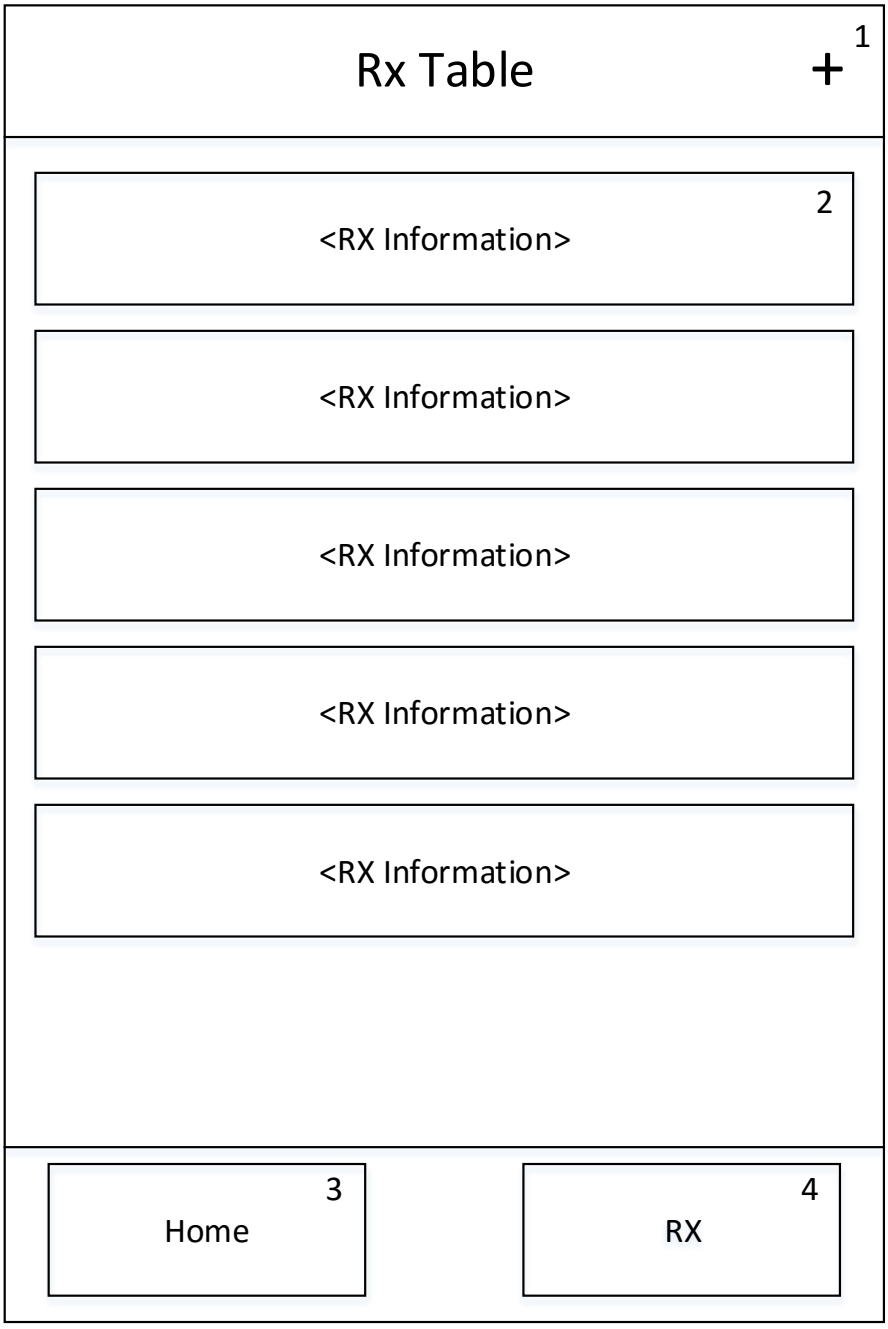


5 Interface Prototypes



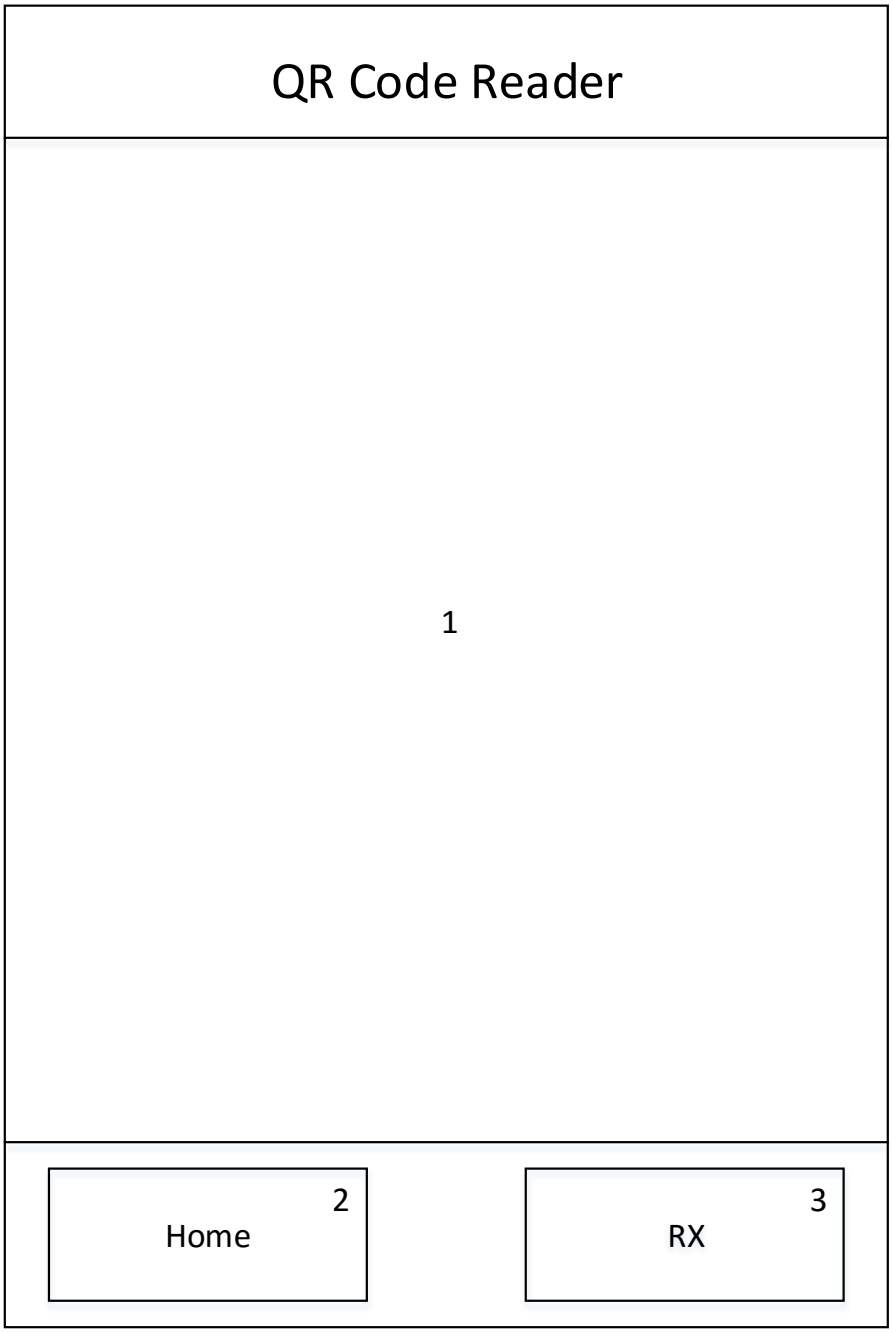
This is the home page of the application. It contains the play/pause button that will allow the user to play/pause sound playback.

1) This label will show the user what prescription is currently active.
2) The play/pause button will allow the user to pause and resume sound playback.
3) The “Home” button on the bottom of the application will bring the user back to the Home screen of the application.
4) The “RX” button on the bottom of the application will take the user to prescription table of the application.



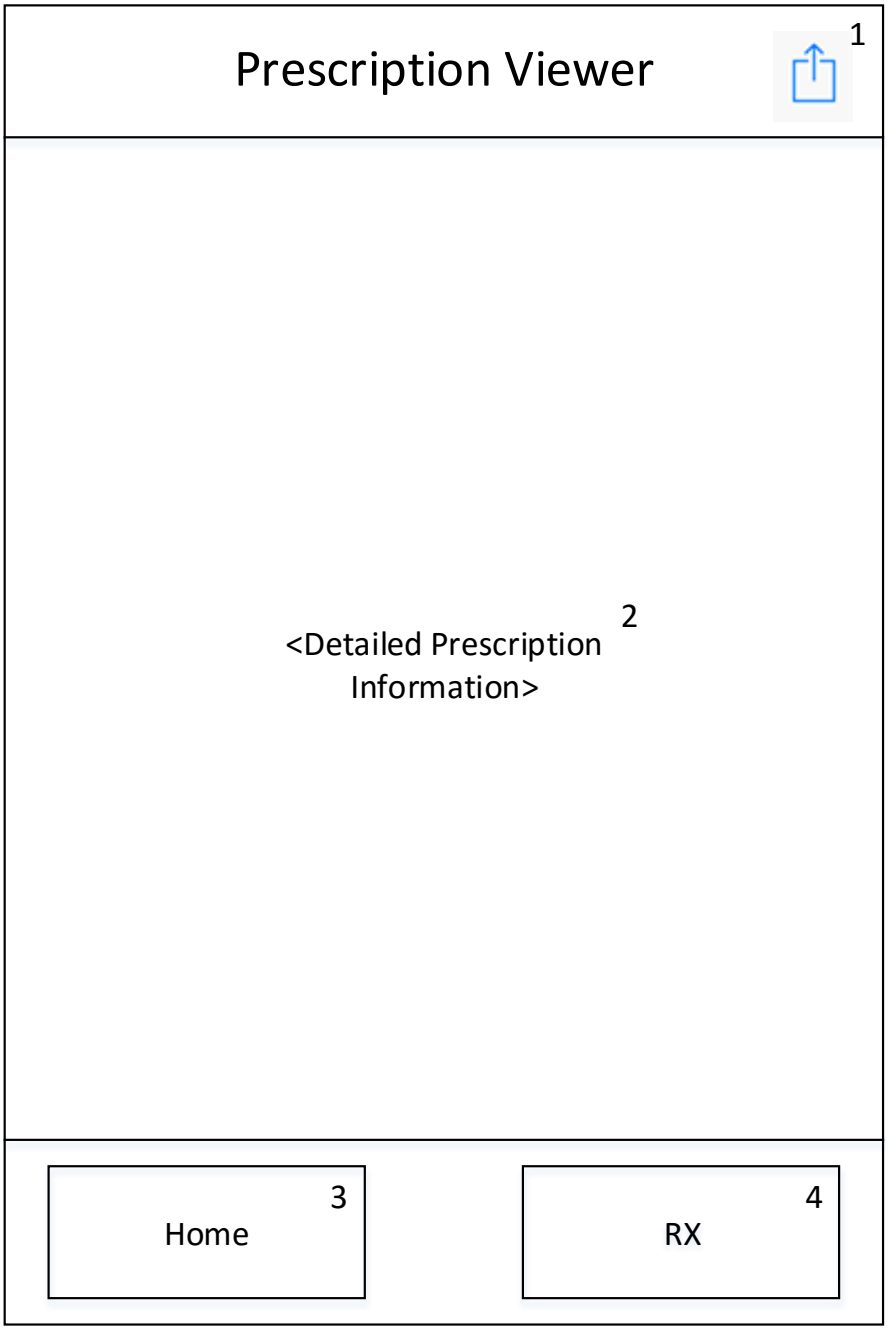
This is the Prescription Table, which will allow the user to change their prescription based on the environment around them, be it loud, like a sporting event, or quiet, like a coffee shop.

1) This “+” button will allow the user to load in another prescription in the form of a QR code.
2) The patient’s prescription history, along with different environments, are listed here to be easily switched to.
3) The “Home” button on the bottom of the application will bring the user back to the Home screen of the application.
4) The “RX” button on the bottom of the application will take the user to prescription of the application.



This is the QR reader, which will allow both the user and the audiologist to load in new prescriptions. This is also where the developer loads in a debug capable QR code.

1. The information within the QR code will be displayed here.
2. The “Home” button on the bottom of the application will bring the user back to the Home screen of the application.
3. The “RX” button on the bottom of the application will take the user to prescription of the application.



This is the view for seeing more detailed information about a particular prescription. The user can get here by selecting a prescription on the Prescription menu.

1. The “Load” button will allow the user or audiologist to set the currently viewed prescription as the active working prescription.
2. Detailed prescription information will be displayed here, including the prescription itself, the name of the prescribing audiologist, and which environment the prescription is intended for.
3. The “Home” button on the bottom of the application will bring the user back to the Home screen of the application.
4. The “RX” button on the bottom of the application will take the user to prescription of the application.

6 Glossary of Terms

Amplitude - the power of the air vibration that can be perceived as volume

Audiogram - graphical hearing diagnosis produced by audiologist

Audiologist: a health care professional specializing in identifying, diagnosing, treating and monitoring disorders of the ear

Clipping - a situation in which the peaks and/or troughs of the sound wave are cut off

DSP - Digital Signal Processor; a microprocessor specially designed to handle continuous analog signal

Decibels - a unit of measurement used to represent a signal's amplitude

FFT - Fast Fourier Transform; an algorithm used to transform a signal from the time domain into the frequency domain

Frequency - equivalent to the inverse of the period, frequency is a rate. Frequency is generally measured in Hz (hertz)

iOS - the operating system for Apple's mobile device product line

LPCM – linear pulse code modulation, where quantization is uniformly quantized; the codec used for the sound

QR code - a data encoding mechanism in which bits are transformed into camera-recognizable (camera friendly) visual representation

Sound Profile (prescription profile, user profile) - a system that manages user's preferences and hearing prescriptions

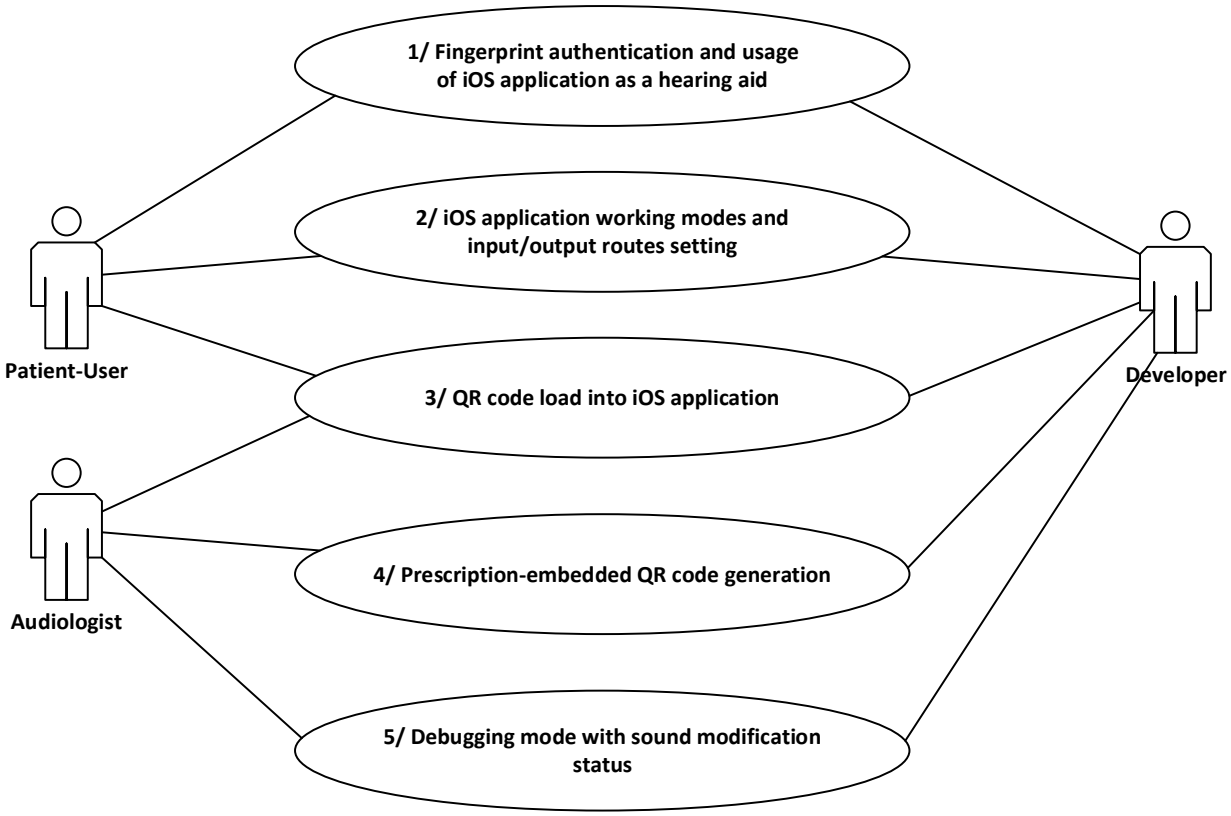
Touch ID - a security mechanism on iPhones to maintain exclusive access based on a user's fingerprint

7 References

Jovanovic, J. (n.d.). Shazam It! Music Recognition Algorithms, Fingerprinting, and Processing. Retrieved April 23, 2016, from https://www.toptal.com/algorithms/shazam-it-music-processing-fingerprinting-and-recognition

Appendix A: Use Case Models

Use case model



1/ Fingerprint authentication and usage of iOS application as a hearing aid	
Actors	Patient-user, Developer
Description	The patient-user swipes his/her fingerprint on the phone's fingerprint reader to gain access to the application. After getting access, the patient-user is authorized to perform use case 2 and 3 as well as using the iOS application as a hearing aid. Developer will have access to all use cases to do application testing and enhancement.
Pre-condition	The patient-user has provided the phone's operating system platform with his/her fingerprint. The iOS application has been loaded with user's personal hearing prescription provided by an audiologist (use case 3).
Post-condition	None.
Data	Patient-user's fingerprint is confirmed through the phone's fingerprint reader. Input sound is recorded into the iOS application, which is then return the processed output sound.
Stimulus	Because the iOS application acts as a personal hearing aid for patient-user, it needs to be safe from external interference. In other words, unauthorized people must not have access to the iOS application installed on the patient-user's phone. The patient-user's hearing condition requires a hearing aid for effective communication.
Response	A form of authentication is needed to prevent unauthorized access. The patient-user uses the iOS application as a personal hearing aid.
Comments	None.

2/ iOS application working modes and input/output routes setting	
Actors	Patient-user, Developer
Description	Patient-user can specify the input routes (built-in microphone or Bluetooth connection) and output routes (earphone or Bluetooth headset) in settings. Patient-user can specify the currently surrounding environment to let the iOS application choose the most appropriate processing modes. Developer will have access to all use cases to do application testing and enhancement.
Pre-condition	The iOS application usage authorization has been confirmed through fingerprint reading (use case 1). The iOS application has been loaded with user's personal hearing prescription provided by an audiologist (use case 3).
Post-condition	None.
Data	Plugging in headphone/earphone or connecting to Bluetooth device(s) Working mode selection in settings.
Stimulus	Different users may prefer different input and output routes. Different environments may require different types of processing to make the output sound most suitable for the patient-user.
Response	The user needs to be able to change the input/output routes as well as processing modes in order to get the most effective output sound.
Comments	None.

3/ QR code load into iOS application	
Actors	Audiologist, Patient-User, Developer
Description	The audiologist or the patient-user uses the iOS application to scan the prescription-embedded QR code, which will automatically update the prescription data in the iOS application. From then on, the new prescription data will be used in sound modification process. Importantly, the audiologist may choose to load temporary prescription data into the iOS application, let the patient-user try hearing through the application, and implement final change on the prescription based on the patient-user's feedback. Developer will have access to all use cases to do application testing and enhancement.
Pre-condition	The prescription-embedded QR code has been generated.
Post-condition	If a new prescription is released, this use case should be repeated as soon as possible.
Data	Hearing prescription data includes, but is not limited to, different frequency ranges and the corresponding decibel levels required to effective hearing.
Stimulus	This use case is needed when: -The iOS application is used for the first time -A new prescription is released -The audiologist wants to test a temporary prescription with the patient-user
Response	The audiologist or the patient-user loads the prescription data in the form of QR code into the iOS application.
Comments	None.

4/ Prescription-embedded QR code generation	
Actors	Audiologist, Developer
Description	The audiologist input prescription data into a special application, which will automatically generate QR code containing the input data. Developer will have access to all use cases to do application testing and enhancement.
Pre-condition	The audiologist has already diagnosed the patient-user and generated a traditional hearing prescription.
Post-condition	None.
Data	Prescription data from a traditional hearing prescription.
Stimulus	QR code adaptation aims at increasing convenience and preventing human error in the process of loading prescription data into the iOS application.
Response	Traditional hearing prescription data is transformed into QR code before being loaded into the iOS application
Comments	None.

5/ Debugging mode with sound modification status	
Actors	Audiologist, Developer
Description	Audiologist can take a look at the real sound modification done by the iOS application to get a hint at the properties of the sound that will get to the patient-user's ears. Developer will have access to all use cases to do application testing and enhancement.
Pre-condition	None.
Post-condition	None.
Data	Visual (graph) and numeric representations of input and output sound as well as the modification parameters.
Stimulus	Audiologists and developers need a convenient way to evaluate the performance of the prescription and the iOS application.
Response	The debugging mode is an easy and accurate way of demonstrating the influence of the application on the input sound as well as the quality of the output sound.
Comments	None.