



Pocket Choir

Design and Planning Document 1.1

Kevin Ballecer, Deryk Clary, Nicholas Kjorlie,
David Kunstmann, Christopher Sullivan

Revision History

| Revision | Date | Changes |
|----------|---------|---|
| 1.0 | 3/1/19 | <ul style="list-style-type: none">• Original Document Composed |
| 1.1 | 3/29/19 | <ul style="list-style-type: none">• Quantized time units for subtasks• Added consideration of design risks• Added section on design limitations |

System Architecture

- Load Existing - Switches view to a table view where a list of all previously saved in-app files will be accessible for editing. Once a file is selected and loaded, the user view will default switch to the **chord selection** screen
- View Instructions - Switches view to view of all instructions for app use will be located.

Transitions: **Start Menu** → **Recording** via Create New Project

→ **Files** via Load Existing → **Chord Selection** via file select

→ **Instructions Menu** via View Instructions

- **Instructions Menu:** Displays a list of tappable help-submenus for each main functional group in the app. Embedded in a navigation controller with the **Instructions Menu** at top level hierarchy.
 - Chord Selection - Displays information about how the chord selection page works. Details on how to use the pads in conjunction with live recording, and how to program using a pre-recorded sample will be available here.
 - Back - Returns user view to **Instructions Menu**
 - Effects - Displays information on how to use global parameters and customizable effects during live recording and with pre-recorded samples. Details on how to populate the customizable pages (1 & 2) with a library of effects and how to switch between the two while recording and editing will be available here.
 - Back - Returns user view to **Instructions Menu**
 - Recording - Displays information on how to record a sample for programming onto chord pads, how to initiate live recording, and how to edit the length and section of the recorded sample clip.
 - Back - Returns user view to **Instructions Menu**
 - Saving/Exporting Files - Details how to save files and how to export to various database platforms and applications.
 - Back - Returns user view to **Instructions Menu**
 - Back button - Returns user view to **Start Menu**

Transitions: **Instructions Menu** **← → Help Submenus** via respective menu titles

→ **Start Menu** via Back

- **Core Data Storage:** Pocket Choir will utilize the user device's core data for storage of files. The core data will be read from anytime a user wants to load an existing file for editing or export a file, and will be written to when a user creates a new file or saves the current file.

Transitions: N/A

→ **Recording Screen** via Right Swipe

- **Recording:** Provides a separate view to initiate two modes of recording: Sampled-based recording (i.e. record sample, use recorded sample and program onto **Chord Selection** pad keys) and live recording. A sample editor is also available here for basic sample manipulation and editing.
 - Record Sample - Tap to record a sample. Once tapped, the button will change to “Stop Recording”, stopping the recording session with another tap. During a recording session, the sample editor will be disabled. After “Stop Recording” has been tapped, the sample editor display will be re-enabled, and populated with the recorded sample wav.
 - Live Record - Tap to initiate live recording. Live Record allows users to swipe between **Effects/Chords Selector/Record** screens and adjust effect levels and/or change the sample harmonization via **Chord Selection**
 - Sample Editor - Displays the wav file of the last recorded sample. Once tapped, two vertical bars will appear at either end of the wav file. These bars can be moved to select the desired portion and length of the recorded sample clip. Tap and holding your finger in the middle of the selected portion of the sample wav in order to drag the selected clip to encompass different portions of the entire sample wav.

Transitions: **Recording**

→ **Chord Selection** via Left Swipe

→ **Global Parameters** via Right Swipe

- **Save:** Indicated by a floppy disk icon that is accessible/tappable from all **Record/Chord Selection/Effect** screens.
 - File Save - Tap to save the current sample file along with its chained effects. The file will be saved as a .wav. The **Core Data Storage** will be written to, adding the new file to memory. A successful save will be indicated by a custom Alert Message that can be dismissed by tapping “Ok”. An unsuccessful save due to some storage/memory error will be indicated by another custom Alert Message notifying the user that the file was not saved. In both cases, File Save is followed by returning the User View to the current **Record/Chord Selection/Effect** screen

Transitions: N/A

- **Export:** Indicated by the up arrow icon. Switches user view to Export View screen where location of export, file name, and file type can be chosen.
 - Export to - Dropdown menu that displays a list of databases and applications that Pocket Choir is able to export to.

-
- File Type - Drop-down menu displaying available file types to export
 - File Name - User can input desired file name for the exported file
 - Export - Tap to perform and confirm Export (via Alert Message dialogue window)
 - Back - Return view to current **Record/Chord Selection/Effect** screen

Transitions: **Export** → **Record/Chord Selection/Effect** via Back

Design Details

Model Classes

Harmonizer

This will be the core class doing the polyphonic pitch shifting. This class will have 4 pitch shifter objects from the AudioKit library. This object will receive chord selection data from the view and output the pitch shifted signals to the iPhone speaker. Inside this class there will be numerical structure for based on the MIDI protocol to specify musical notes and chords. This structure will be set up relative to a root note which can be adjusted by the user at any time.

AutoTuner

This class is very similar to the above class, the only difference being that instead of producing chordal outputs it will only produce monophonic outputs based on the current key set by the user. The only reason it is separate from the Harmonizer is for cleanliness and organization.

Sample

This class will contain an audio file which can be recorded in the app or loaded from a file on the user's phone. Once loaded, the sample may be played back either manually by tapping the play button or on a loop. The sample can be routed through either of the pitch shifting units described above as well as the effects rack

Effects Rack

This class will consist of 3-5 audio effects from the AudioKit library. The effects rack will have the ability to be ordered in any combination, meaning that the user will be able to decide the order in which the effects process the signal. Changing this order can have drastic effects on the sound of the output and allows the user flexibility in creating an original sound. Additional it should be noted that the harmonizer and autotuner cannot be placed after these effects though they may be bypassed.

View Classes

StartMenuView

This is the start page for the app and will allow the user to navigate to any of the other views via a table of icons. In order to do this the class will require a number of segues to other pages. A small icon will persist across other pages which will allow the user to return to this page.

InstructionView

This view will simply give a detailed description of how the app works to the user. It will essentially just be a page the user can scroll through to read about functionality, no real user interactions happen on this page.

LoadPresetView

This is where presets saved by the user will be listed in a table view. The user will be able to save the current settings from any page and name the preset. Then the user can navigate to this page and see all of the saved presets listed and select one.

ParameterView

This view will display the core settings of the apps main feature, the harmonizer. In this window the user will be able to modify core aspects of the pitch shifting algorithm which can have an effect on the sound of the output. These parameters will include things like Input Volume, Output Volume, Sample Volume, Root Note, Minor/Major key selection, Window Size, and Crossfade Length. The volume parameters are pretty self explanatory. The Root Note parameter will control the root of the current scale while the Minor/Major button will determine the scale's quality. Window Size and Crossfade are internal aspects of the pitch shifting algorithm which play a role in the coarseness of the pitch shifted output. They can also play a role in the delay time of the overall system and thus can be viewed as a sort of trade off between quality and speed which is left up to the user.

ChordView/KeyboardView

The ChordViewController is where the primary music manipulation happens and is where the user goes to change the current chord. This page will consist of a grid of large buttons which the user can tap to change the chord from one to the next in real time as they are singing or playing into the phone. This page will also have the option to switch to a keyboard view where the user can then select specific notes to shift the input to rather than full chords.

SequenceView/LoopView

This view will allow the user to create a sequence of notes or chords and automate the process of switching from one to the next. This way the user can decide on a chord progression they like and then allow it to loop while they adjust other settings within the app or play another instrument.

SampleView

This view is where the sample is recorded or loaded. The user will be able to see the incoming signal as a waveform in real time and decide when to start and stop recording.

EffectsMenuView

This view is where the user will be able to select which effects are present in the signal chain and in which order they occur. Additionally, from this page the user will be able to enter a separate page for each effect where the effect can then be adjusted.

Controller Classes

AppController

There will only be one controller class and it will retain instances of each of the above classes. From this class almost all of the interaction between the view and the model will be conducted. This class will be listening for buttons pressed and passing the event along to the model or audio processor in order to update its features. This is also the class which will control the writing of presets to the database.

Design Risks

One of the main concerns of implementing our application, is the potential delay in output from all the audio processing being performed using the Audiokit package. We will have to wait until approximately iteration 2 until we can start producing tests to check how significant the output delay is in response to the input signal, as well as how the quality of the output sound is when light and heavy processing is performed on the recording.

Another concern we have is when we start incorporating more effects for effect chaining and have a fully functional sample recorder programmed ready to take in user recordings for processing. Because tests for this section will have to wait until iteration 2, we cannot determine right how much power will be consumed once multiple effects have been chained and are included in the overall audio processing.

Implementation Plan

Iteration 1: Complete User Interface and View Transitions

Iteration 1 will consist primarily of compiling an aesthetically pleasing user interface for all screens accessible to the user in the Pocket Choir app. The goal will be to find the easiest, and cleanest way to present the scrollable screens for recording and editing (i.e. the keyboard screen, effect pages, global parameters, and recording screen). Transitions between screens will also all be implemented at this point. This stage will also act as a “set-up” for iterations 2 and 3 as we will make decisions on what end effects we will end up implementing in the next iteration, as well as create algorithms for these effects. These chosen effects will be partially to fully implemented in iteration 2, with finalizations in iteration 3. Effect chaining is a possibility we as a group are hoping to be able to implement, however due to the potentially large amount of data processing this requires in a small amount of time, this may not be practical for our project.

Iteration 1 General Objectives

- Build views for each view controller/screen (Start Menu, Instructions Menu, Effects/Recording/Chord Selection, etc.)
- Implement all button transitions: make sure we can get to each desired screen without issues
- Implement all button animations (knob turns, sliders, dropdown menus)
- Decide on what effects we want to implement in our effect library
- Create algorithms for signal processing of samples
 - Effect chaining
 - Algorithms for each effect
 - Dynamic/Static Sample Harmonization

Iteration 2: More Advanced Features

Iteration 2 will focus on implementing most signal processing functions/use cases. This includes creating a keyboard with potential to include multiple-voices (ie you can press more than 1 key and two different tones will be played), building the harmonization feature, achieving low-delay recording for dynamic and static purposes (live recording vs. programmed sample), and developing functions for effects chosen in the previous iteration. This stage will also establish connectivity to and from external databases and applications that users may want to export their files to for further editing and sharing. The core data memory will also be set up for local saving and loading of files.

Iteration 2 General Objectives

- Achieve fully functioning recording feature with no processing and minimal output delay
- Check that data is being properly transferred between view controllers
- Implement Dynamic (Live Recording) and Static (Prerecorded Sample) Harmonization
- Create functions for each chosen effect
 - Test each single effect and run unprocessed sample through effect filter
 - Minimize output delay and refine signal quality
- Set up internal database/memory for saving and loading files locally
- Set up connectivity to databases chosen to export to

Iteration 3: Final Product

Iteration 3 will involve putting the finishing touches on the Pocket Choir app to ensure it is user friendly, fully functional, and also works efficiently. In the case that we are able to develop a substantial effect library, we will begin to develop the user-customizable effect pages described in the View Controllers section. We want to avoid dealing with this if we have less than 10 effects created as there would be limited customization available for the user (mainly just moving the location of the effect knobs/sliders which may end up being useful). Additionally, if testing for single effects on the audio input samples results in minimal output delay and low loss in audio data, we will also try to implement effect chaining. Due to the large amount of processing the audio samples must undergo with this function, effect chaining may prove to be impractical for this version of Pocket Choir to prevent handling of potentially significant output signal delays.

Iteration 3 General Objectives

- Attempt to implement effect chaining unprocessed samples
- If effect library is large enough, will try to implement the customizable effect pages, otherwise they will be defaulted to pre-selected pages designed by the team
- Build functionality for customizable effect pages assuming effect chaining is possible with minimal output delay and good output signal quality
- Final testing stages to fix any lingering errors
- Polish user interface

Use Cases and Associated Programming Tasks

** Difficulty (1-5) | Iteration (1-3) | Team Member (Kevin, David, Nick, Chris, Deryk)

| Use Case Categories | Programming Tasks |
|--------------------------------|-------------------|
| Core Data Storage Implement | |

functions that deal with writing to, or reading from, the user's internal device memory

| Subtask | Difficulty | Iteration | ETA (hrs) | Team Member |
|--|------------|-----------|-----------|-------------|
| Create new project <ol style="list-style-type: none"> User input file name text field Check for duplicate file names Save file if check is passed and proceed to the Recording screen <ol style="list-style-type: none"> If check is not passed, user will be prompted to input another file name | 3 | 2 | 2 | Chris |
| Load existing file <ol style="list-style-type: none"> Instantiate/segue to a new table view keeping an updated list of all previously saved files <ol style="list-style-type: none"> Table should include: <ol style="list-style-type: none"> File Name File Size Date created Description File Tapped → read all data necessary to complete project load from core data memory <ol style="list-style-type: none"> Input sample wav file Effect levels File name Open project and load the keyboard screen | 3 | 2 | 2 | Deryk |
| Save current file <ol style="list-style-type: none"> Tap button to open dialog window showing save success Write to core data memory and update current file Check if save was successful and output Success/Unsuccessful event in dialog window Return to last used (or current) Record/Effect/Keyboard screen | 3 | 2 | 2 | Kevin |
| Export current file <ol style="list-style-type: none"> Switch view to new view controller Create dropdown menu to display list of available databases to export to | 3 | 2 | 2 | Nick |

| | |
|--|--|
| | <div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><</div></div></div></div> |
|--|--|

| | | | | | |
|---|--|------------|-----------|-----------|-------------|
| | d. Route output signal post-autotune through speakers/headphones e. Test: import onto an iPhone to assess lag time and audio quality | | | | |
| | Set up <i>Record Sample</i> button to initiate recording, and to change to <i>Stop Recording</i> after tapped once | 2 | 2 | 2 | Chris |
| | Set up <i>Live Recording</i> button | 2 | 2 | 2 | Kevin |
| | Implement Sample Editor a. Populate with a visual of the resultant sample wav b. Variable selection bars to “crop” and select specific portions of recorded audio sample - audio data within the bars should be highlighted a certain color or made brighter to indicate selection c. Enable only after recording is finished | 3 | 2 | 3 | David |
| | | | | | |
| Build User Screens Complete the user interface for all viewable screens: Finish all transitions between view controllers via buttons and swiping, create alert dialog windows for saving/loading/exporting, and provide a clean overall aesthetic | | | | | |
| | Subtask | Difficulty | Iteration | ETA (hrs) | Team Member |
| | Start Menu a. Create new project button b. Load existing button → Load File Browser screen c. View Instructions button → Instructions menu screen | 1 | 1 | 1 | Nick |
| | Instructions Menu & Submenus a. List of 4 submenus that can be tapped i. Chord Selection/Keyboard button ii. Recording button iii. Effects button iv. Saving/Exporting button b. Create submenu views (no text yet) | 1 | 1 | 1 | Deryk |
| | Keyboard/Chord Selection a. Single octave keyboard b. Swipe Left = Customizable Effects Page 1 | 1 | 1 | 1 | Nick |

| | | | | | |
|--|--|---|---|---|-------|
| | c. Swipe Right = Recording Screen | | | | |
| | Effects a. Decide what effects to include in library b. Create icons for each effect | 1 | 1 | 2 | Chris |
| | Customizable Pages <i>Starting on Customizable Page 1</i> a. Large square area to place effects b. Swipe Left = Customizable Page 2 c. Swipe Right = Keyboard/Chord selector <i>Starting on Customizable Page 2</i> a. Large square area to place effects b. Swipe Left = Global Parameters c. Swipe Right = Customizable Page 1 | 1 | 1 | 2 | Kevin |
| | Global Parameters (no association with processing sample yet) a. Master Volume knob b. Cutoff knob c. Resonance knob d. Rotation animation for all knobs e. Swipe Left = Recording f. Swipe Right = Customizable Page 2 | 1 | 1 | 2 | David |
| | Load File Browser a. Empty scrollable table view b. Cancel button → Start Menu c. Load button → Keyboard/Chord Selector | 1 | 1 | 1 | Chris |
| | Export Screen a. Export Icon button (up arrow) → Export Screen b. Back button → return to current Record/Effects/Keyboard screen c. “Export to” dropdown menu i. Sample database/apps first before respective connectivity is achieved d. File Name user input text field e. File Type dropdown menu | 1 | 1 | 2 | David |

| | | | | | |
|---|--|------------|-----------|-----------|-------------|
| | <ul style="list-style-type: none"> i. Sample audio file types until output file types are decided on f. Export button → export file to database/app and return to current Recording/Effects/Keyboard screen <ul style="list-style-type: none"> i. Initiate alert dialog window for export confirmation | | | | |
| | <p>Alert Dialog Windows</p> <ul style="list-style-type: none"> a. File Save confirmation alert → will tell user whether the file save was successful or not or if there was an error with the memory write <ul style="list-style-type: none"> i. Tapping “Ok” will return user to last used (or current) Record/Effect/Keyboard screen b. File Export confirmation alert → will tell user whether file export was successful <ul style="list-style-type: none"> i. Tapping “Ok” will return user to last used (or current) Record/Effect/Keyboard screen | 1 | 1 | 1 | Kevin |
| | Test all possible buttons/transitions between user screens | 1 | 1 | 2 | Deryk |
| Keyboard Playing Create functions necessary to produce sound by playing keys on the keyboard, single notes and multiple notes | | | | | |
| | Subtask | Difficulty | Iteration | ETA (hrs) | Team Member |
| | Send key identifier to audio processor | 3 | 2 | 2 | Nick |
| | Play pitch of key when pressed, stop when lifted | 3 | 2 | 2 | Deryk |
| | Sum frequencies if multiple notes are played <ul style="list-style-type: none"> a. Detect when multiple notes played simultaneously b. Send key identifier of each individual note to audio processor | 3 | 2 | 2 | Kevin |

| | | | | | |
|--|--|------------|-----------|-----------|-------------|
| | c. Obtain frequency of each pressed key, sum, output | | | | |
| | Establish refresh rate to check if a different key has been pressed | 3 | 2 | 1 | David |
| | Output correct tones/pitches through speakers/headphones | 3 | 2 | 1 | Chris |
| | Implement sample harmonization using 2-4 additional voices a. Harmonization algorithm implemented for chords selected | 5 | 2 | 3 | Chris |
| | Expand keyboard model to handle multiple keys pressed at once a. Implement the ability to press two keys at once b. Create and send array of note indicators to audio process → empty array = silence c. Generate new instances of the frequency multiplier for each additional note being played d. Sum all altered streams back into one output stream, route to speakers/headphones | 4 | 2 | 2 | Deryk |
| | | | | | |
| Effects Implementation Design an effect library that can be added to customizable effect pages 1 2 to modulate the audio sample signal Note: If effect library does not end up being extensive, we will implement pre-selected effect pages based on the library we | | | | | |
| | Subtask | Difficulty | Iteration | ETA (hrs) | Team Member |
| | Implement Global Parameter knobs a. Master Volume - the volume of the final output signal (ie sample + harmonization + effects) b. Cutoff Frequency - create a lowpass filter with a variable cutoff frequency c. Resonance - implement adjustability of resonant cutoff frequency of the LP filter | 3 | 2 | 3 | David |
| | Chorus a. Create algorithm b. Dry/Wet parameter | 3 | 2 | 3 | Nick |

| | | | | | |
|--------|---|---|---|----|----------------------------------|
| create | <ul style="list-style-type: none"> c. Feedback parameter d. On/off switch e. Implement & test individually with a played sample | | | | |
| | Overdrive <ul style="list-style-type: none"> a. Create algorithm b. Dry/Wet parameter c. Filter frequency parameter (BP filter) d. On/off switch e. Implement & test individually with a played sample | 3 | 2 | 3 | Kevin |
| | Delay <ul style="list-style-type: none"> a. Create algorithm b. Dry/Wet parameter c. Feedback parameter d. On/off switch e. Implement & test individually with a played sample | 3 | 2 | 3 | David |
| | Reverb <ul style="list-style-type: none"> a. Create algorithm b. Dry/Wet parameter c. Decay time parameter d. On/off switch e. Filter frequency parameter (BP filter) f. Implement & test individually with a played sample | 3 | 2 | 3 | Nick |
| | ADSR - These effects will also act as global parameters (ie affecting the final output signal, not just the initial input audio sample) <ul style="list-style-type: none"> a. Attack knob/slider b. Decay knob/slider c. Sustain knob/slider d. Release knob/slider e. Implement & test each individual slider separately with a played sample f. Test altering multiple parameters at once and see if the expected output signal is achieved | 3 | 2 | 4 | Kevin |
| | Effect Chaining <ul style="list-style-type: none"> a. Implement signal processing for 2 effects simultaneously being used | 5 | 3 | 10 | Deryk, Kevin, Nick, David, Chris |

| | | | | |
|---|--|--|--|--|
| <ul style="list-style-type: none"> i. Run unprocessed sample through first effect ii. Run processed sample through 2nd effect iii. Check for significant output lag or drop in audio quality b. Implement signal processing for 3 effects simultaneously being used <ul style="list-style-type: none"> i. Part a ii. Run processed sample through 3rd effect iii. Check for significant output lag or drop in audio quality c. Implement SP for multiple global parameters being set <ul style="list-style-type: none"> i. Test with 1,2,3+ effects chained as well d. Perform tests with all combinations of effects | | | | |
|---|--|--|--|--|

● Tentative Effect Selections....more to come in future revisions

- Tentative Effect Selections....more to come in future revisions

Design Limitations

Our current design for Pocket Choir implementation is limited by the processing power of the user's iPhone/iPad. Older models may incur more significant output delays and produce poorer sound quality in comparison to newer iPhones/iPads with better processors. In order for a user to have a good experience using Pocket Choir, the application must output a good quality sound in line with what the product offers (live-harmonization), and perform this in a timely fashion and without audible lag. We plan to test our application on older models via XCode's Build feature, however we believe that older models may suffer more from these issues than newer iPhones/iPads.

Testing Plan

Timing Tests

One of the core features of pocket choir is live harmonization with a user recorded sample. If there is significant delay between the time the user inputs an audio sample, and when it is finally outputted to the device's speakers/headphones, the program will not be desirable. We need to ensure that minimal to zero delay is achieved with both

regular recording, as well as when effects, autotune, or harmonization features are added to the original audio recording. We will also test how long it takes to make transitions between screens, and the time lapsed for saving, loading, and exporting project files. Below is a list of functions/features that will need to be tested for their delay time as a performance metric:

- No autotune
 - With regular live recording
 - With recorded sample played on keyboard key
- Autotune applied
 - to regular live recording
 - To recorded sample played on keyboard key
- Regular live recordings with all possible combinations of effects
- Harmonization
 - Pre-recorded samples
 - Live recorded samples
- Loading projects of various sizes
- Creating new projects
- Saving new projects
- Exporting files
 - Test the time it takes to export to each database or application

Transition and Program Flow Tests

- Test all possible transition combinations between screens to ensure connectivity throughout application
 - Ensure that data is being properly transferred between view controllers
 - Perform unit tests on visual output in all possible scenarios to ensure that UI do not ever interfere.

Data Storage and Export Tests

- Data/File tests
 - Create file then load it
 - Check for data persistency and quality in presets and sample recording
 - Perform unit testing on any functions we decide to implement
 - Load nonexistent file
 - Export to: try to export to all platforms we decide to include