

Singing Data Labeling Tool Design Document

Group members:

Nandith Narayan nnarayan2018@my.fit.edu

Avinash Persaud apersaud2018@my.fit.edu

Carlos Cepeda ccepeda2018@my.fit.edu

Advisor:

Dr. William Shoaff wds@fit.edu

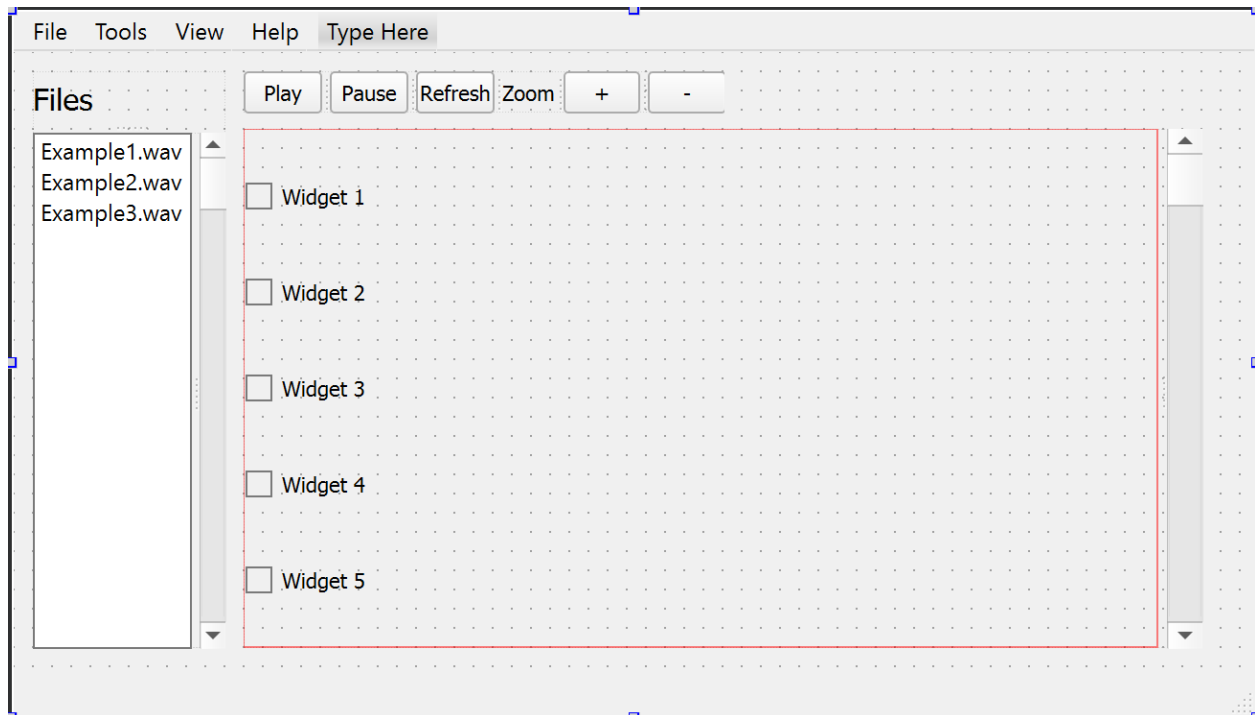
Client:

Caleb Matthew Long, Appalachian State University

Task	Completion %	Nandith Narayan	Avinash Persaud	Carlos Cepeda	To Do
1. Implement and demo GUI layout	75% As of 10/29/21		15%	60%	Have the GUI display the spectrum of the audio file provided by the user.
2. Implement intermediate representation of data	100%	80%	20%		
3. Implement and demo saving and loading of files	90%	90%			Create a button on the UI to save and load files.
4. Integrate Qt UI libraries with Antlr v4	100%	50%	50%		
5. Determine phoneme detection method	50%		50%		No method is selected yet

Task Discussion:

Task 1: In this task, we implemented a layout using Qt creator 5.0.2 for the main page of the SPELL program. This milestone was focused on setting the layout for the program because implementation of tools provided by the software will be the focus on the next milestone. The current features that can be displayed to the user are a menu bar and a list widget that will list the current directory. There is also a toolbar with the options to play, pause, and refresh the audio file.



Task 2: Part of this task was designing the build system pattern. CMake is the selected build system. We also implemented an abstract label class to handle our different label types (such as phonemes, notes, and syllables). Each label type will have a class of its own and inherit from the label class. We implemented the loading of audio file paths into the project. This part of the project was packaged into its own library. Doing so allows us to keep the project structure organized. This also makes integration with the user interface simpler.

Task 3: We were able to write the loaders for inputting singing audio files into the intermediate data structure. We check if the path is valid as well as if it has the right file extension. Since these files are expected to be large, we don't copy the files over. Instead we simply store the path to the file and load it when needed.

```
D:\Senior_Design_Project\SPELL\build>spell.exe
Initializing Project: TestProject at C:/test/project
ERROR: Failed to add C:/test/songs/babyShark.wav to TestProject (Invalid Path)
ERROR: Failed to add C:/test/yolo.txt to TestProject (Invalid Path)
Added C:/test/crab.wav to TestProject
```

Task 4: The library structure on how to communicate with Antlr has been chosen. It will act as a submodule to the main library of the tool. This way the scope of Antlr does not affect the scope of our project. We originally encountered an issue where Antlr and ur GUI library Qt had conflicting reserved keywords and functions. Adding Antlr as a sub library solved this problem.

Task 5: Tests with building a specialized singing detector in CMU Sphinx failed due to complications in the training scripts. The input format for Shiro is not understood well. Research indicates that phoneme detection and timing for singing is considered a much more difficult task than speech recognition. Some methods report as low as 20% success rate. Further methodology and research is necessary.

Contribution Discussion:

Milestone 3 task matrix:

Task	Nandith Narayan	Avinash Persaud	Carlos Cepeda
1. Find, test, and determine phoneme detection algorithm	Aid in searching	Search and test models against NIT JP 80 public database	
2. Get basic UI functionality	Integrate and direct usage of the intermediate data structure	Aid in understanding requirements and goals	Implement waveform display and marker placement
3. Start on language processing tools	Aid in integration with project data structure. Write a visitor for parsing the grammar.	Establish a dictionary format and read. Get basic operability for phonemes to syllable	Design and add language definition window
4. Implement the template creation window.	Integrate the parser with the window's user interface.		Create the template creation window.

Milestone 3 Task Discussion:

Task 1:

Being a critical component to our goal, continued testing will occur to ensure one of the better and usable options are selected and implemented.

Task 2:

This task refers to the ability to place label markers onto a visualized waveform. This is the bare minimum functionality for a labeling tool. This task also includes being able to display the waveform to the user by allowing the user to upload the .wav file onto the program.

Task 3:

Part of the goal of this project is to allow conversion between phonemes to syllables, and words to syllables and phonemes. An algorithm will be designed based on the idea of phonotactics. This will be a programmable model to adapt for other languages.

Task 4:

The template creation window allows the user to specify a template for labels. This allows the user to decide what data needs to be included in the labeled output.

Client Feedback

Everything so far looks like what they would want from a labeling tool. They suggested a name change from “Singing Data Labeling Tool” to “Singing Phoneme Ergonomic Labeler” or “SPELL” for short.

Date of meeting with Faculty Advisor: **10/29/21**

Faculty Advisor feedback on milestone tasks:

- Evaluation by Faculty Advisor
- Faculty Advisor: detach and return this page to Dr. Chan (HC 214) or email the scores to pkc@cs.fit.edu
- Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Carlos Cepeda	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Avinash Persaud	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Nandith Narayan	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

■ Faculty Advisor Signature: _____ Date: _____