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.Compare technical Tools- GUI, Parsers, Phoneme alignments	90%	30%	30%	30%	Compare bison and yacc Test integration and portability of SHIRO
2. Simple demos of tools	80%	30%	15%	35%	Demo wxWidgets Demo SHIRO, demo CMU Sphinx for singing
3. Resolve Technical Challenges	76%	33%	10%	33%	Can CMU Sphinx align singing and/or existing transcription?
4. Compare and select tools for	100%	34%	33%	33%	Nothing

document creations, and version control					
5. Requirement s Document	100%	20%	80%		
6. Design Document	100%	80%	20%		
7. Test plan	100%			100%	
8. Preliminary UI design	80%	10%	35%	35%	

Task Discussion:

1.Compare technical Tools- GUI, Parsers, Phoneme alignments: For parsers, we tested antlr v4 and Bison. We found antlr v4 to be more suitable to our application. The major hurdle we faced with this task was compiling the antlr c++ runtime.

SHIRO was not yet tested, while CMU Sphinx has. Testing indicates that if there are no existing transcriptions, CMU Sphinx is a useful tool for phoneme detection, for speech. However, there does not appear to be an easily accessible toolkit for alignment from existing phonemes with it. Further investigation into the toolkit may reveal a missed function, but current investigation is not promising. For singing, it appears to be less capable, however tuning the analysis parameters may result in better performance. Investigation of SHIRO will be needed to see if it can fill the gap and still allow us to have portable code. Investigation into the phoneme models indicate that if needed, a phoneme language model can be constructed for a language easily. If further investigation proves fruitless further techniques will need to be investigated.

2. Simple demos of tools: On the GUI side of the project, comparisons were made between Qt, GTK+, and wxWidgets by implementing a simple window that can be dragged around the screen and output "Hello, World!" when a button is clicked. So far GTK+ and Qt have been tested, and wxWidgets is next on the list.

3. Resolve technical challenges: We were able to understand the HTS singing label output format and create a method to generate it. We gained more experience with designing and implementing User Interfaces. We were able to generate a list of phonemes.

4. Compare and select tools for document creations, and version control: We selected google docs for creating documents, and we selected git with github for version control.

5. Requirements Document: We used google docs to write the requirements document. We discussed with the client over discord to formulate the requirements.

6. Design Document: Using the requirements document, we were able to design a system architecture diagram and write the design document.

7. Test Document: We were able to write a test document that outlined the ways in which we plan to test the tool. We discussed with the client over discord to help out gain an idea of how to go about testing.

8. Preliminary UI design: We designed the layout of the User Interfaces for saving and loading files, the phoneme editor, the phoneme timeline, the spectrogram, and the graphical representation of the audio data.

Contribution Discussion:

Nandith Narayan: Tested parsers and wrote the sample grammar for the parser. Wrote the design document and helped with the requirements, progress evaluation, and test documents. Designed the intermediate data structure and system architecture diagrams. Helped with the presentation.

Avinash Persaud: Wrote most of the requirements document and helped partially with the others. Investigated the phoneme alignment/detection tools.

Carlos Cepeda: Tested GUI development environments, and helped decide which tool to use. Wrote most of the test plan.

Milestone 2 task matrix:

Task	Nandith Narayan	Avinash Persaud	Carlos Cepeda	
1.Implement and demo GUI layout	Assist in understanding libraries for efficient implementation	Providing basic UI layout for demo	Begin GUI development in either Qt or wxWidgets	
2.Implement and demo automatic phoneme alignment	Assist when needed	Continue research and testing Sphinx and SHIRO	Assist when needed	
3.Implement and demo intermediate representation and structure of data	Begin implementing method for converting results from phoneme alignment tool into output	Assist with converting results of phoneme alignment tool	Assist when needed	
4.Implement and demo saving and loading of files	Begin implementing data structure for saving and loading files	Assist with implementing data structure and GUI if needed	Implement GUI options for saving and loading files	

Milestone 2 Task Discussion:

Task 1: After choosing which GUI development tool to use, begin implementing basic outline for UI using it. The goal is to have the layout running, and implementing basic functions such as accepting input

Task 2: Due to the unexpected results from early investigation, further investigation into the tools is necessary before a proper implementation can be made.

Task 3: Implementing the intermediate data structure will allow us to connect the different components of the system together.

Task 4: Saving and loading of files is a crucial feature of this tool.

Date of meeting with representative Client: 9/30/21

Client Feedback

Everything so far looks like what they would want from a labeling tool.

Date of meeting with Faculty Advisor: 10/1/21

Faculty Advisor feedback on milestone tasks:

1.Compare technical Tools- GUI, Parsers, Phoneme alignments:

- 2. Simple demos of tools:
- 3. Resolve technical challenges:
- 4. Compare and select tools for document creations, and version control:
- 5. Requirements Document:
- 6. Design Document:
- 7. Test Document:
- 8. Preliminary UI design:

- 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: _____