Tabulator Testing Manual

CA400

Authors:
Andrew Cullen - 17378471
Eoin McKeever - 17436202

Supervisor:
Alan Smeaton

29th March 2021

0. Table of contents

1. User Testing

2. Use Case Testing

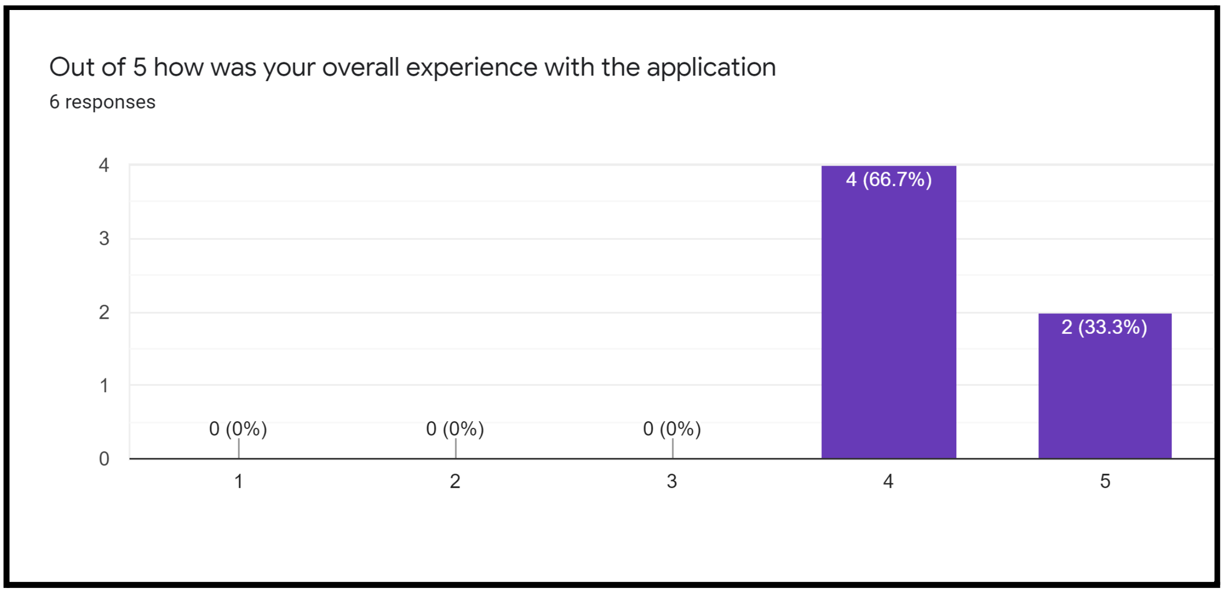
3. Component Testing

3.1 Visual Component Testing

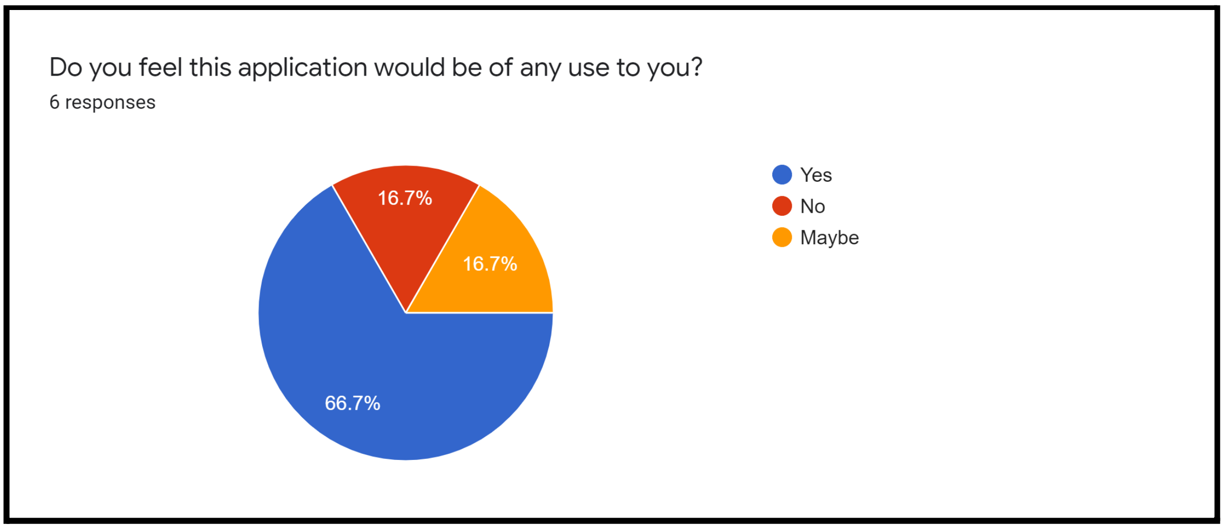
3.2 Audio Component Testing

4. System Testing

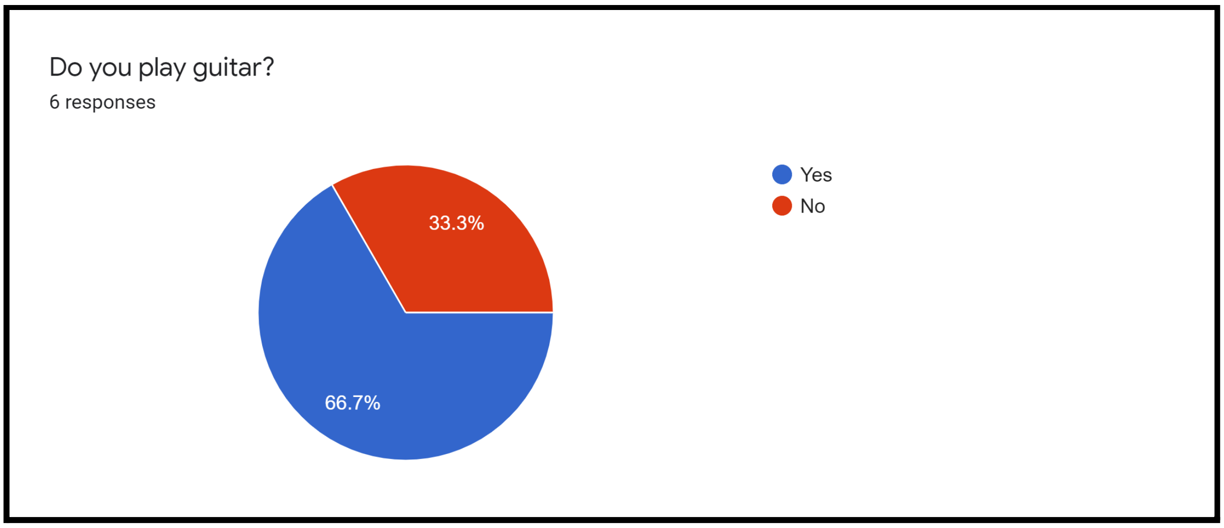
1. User Testing

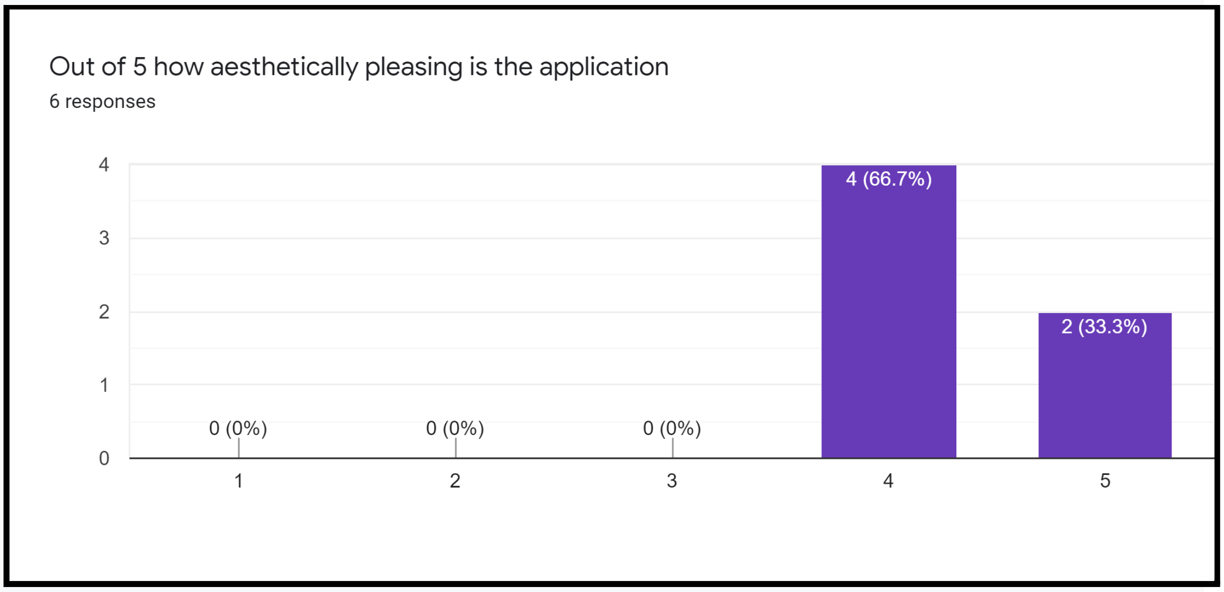

Out of 5 how was your overall experience with the application

6 responses


Do you feel this application would be of any use to you?

6 responses





In your personal opinion what can be improved?6 responses

The application is very simple and easy to use which is good. The simplicity of the application makes it somewhat bare which is my only observation. But it is better simply then overcomplicated .

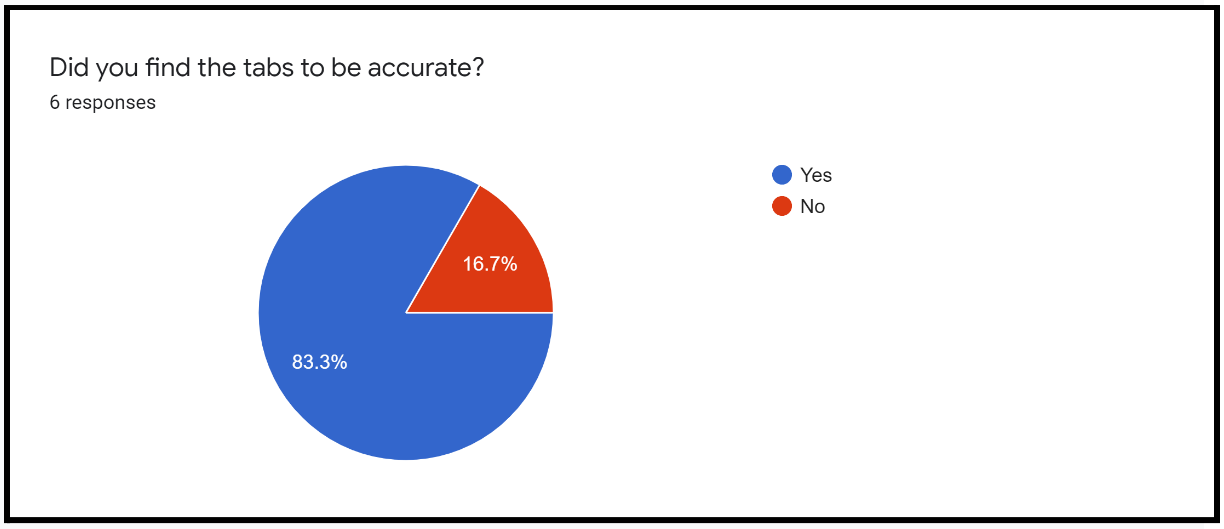
I have no complaints involving the UI

The application had a nice background

The time it takes to load

You could name the tab file

How fast it is

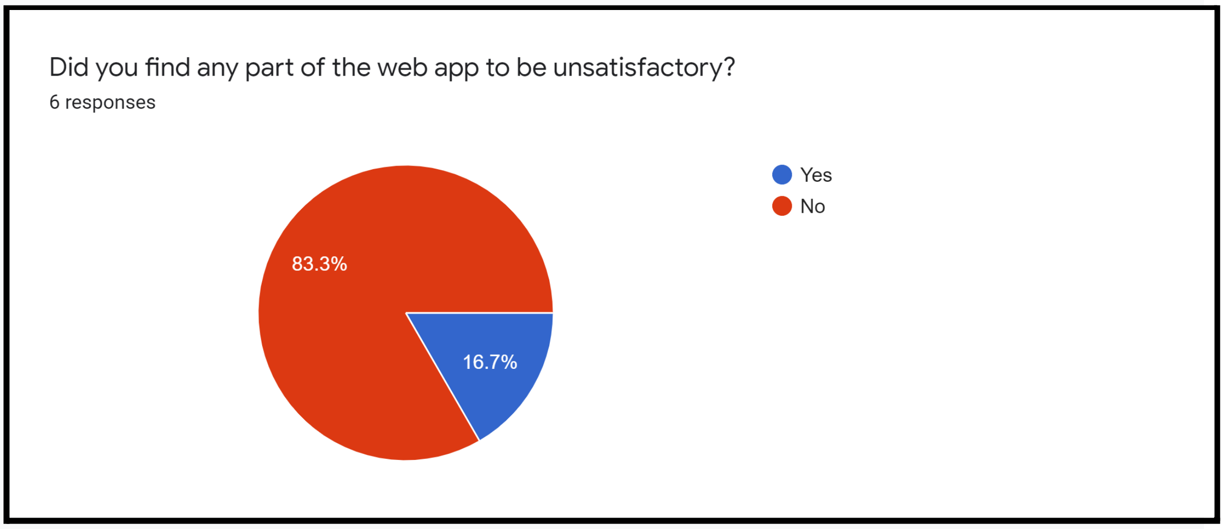


If inaccurate could you describe the error?3 responses

No they were perfectly accurate.

The tabs were mostly accurate with a few small errors. The application includes a feature which allows the user to edit the outputted tabs. With only a few small changes the tabs were ready to be downloaded.

The timing was a bit off



If so could you describe the issue?1 response

Speed

2. Use Case Testing

No.	Job	Steps	Status
1	Choosing video file from local machine	1. Press “Choose file” button 2. Choose file from file explorer 3. Confirm choice	PASS
2	Upload file	1. After a file has been chosen 2. Click “Upload file” button	PASS
3	Download Tabs	1. After upload has been completed 2. Click “Download” button 3. Enjoy your tabs	PASS
4	Editing Tabs	1. After upload has been completed and tabs are displayed 2. Click on tabs and edit freely 3. Click “Download” button once editing is completed 4. Enjoy your edited tabs	PASS

3. Component Testing

3.1 Visual Component Testing

The object detector and hand detection components were tested using a dataset consisting of 100 test images, results were analysed and placed in one of three categories. Correct, Correct post error handling and unacceptable. With regards to the object detector it should be noted that post evaluation a small amount of padding is added to each bounding box and with regards to the Hand detection error handling in the form of outlier removal is applied. The three categories that are applied to the edge detection model consist of Acceptable , Off Slightly and unacceptable. The category off slightly applies to situations in which the edge detection does not output a 100% accurate result but due to the characteristics of the situation a correct result is still returned assuming other components perform to an acceptable standard.

	Acceptable	Acceptable post Error handling	Unacceptable	Overall Accepted
Object Detector	59	33	8	92
Hand Detection	45	52	3	97
	Acceptable	Off slightly	Unacceptable	
Edge Detection	87	9	4	87

Examples of test categories

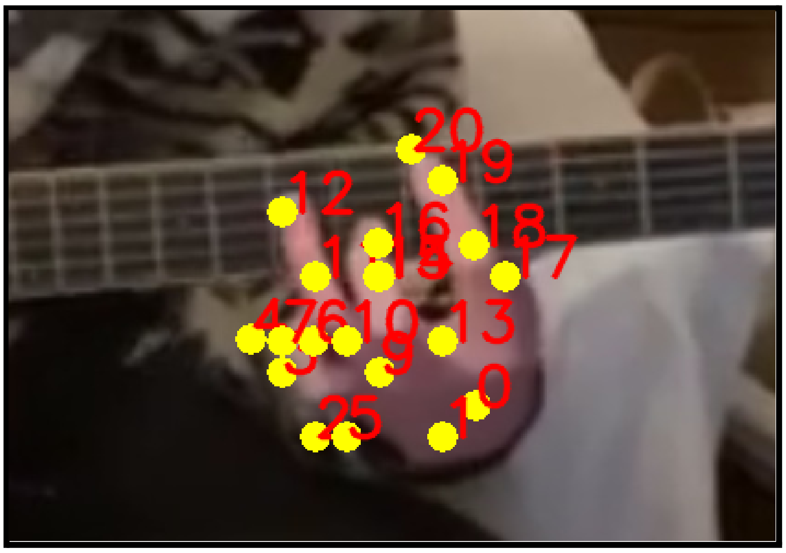
The three categories chosen for the object detector and the hand detection test results to be assigned to were Acceptable, Acceptable post error handling and unacceptable. Examples of these situations can be seen on the following pages.

- **Images table of content**

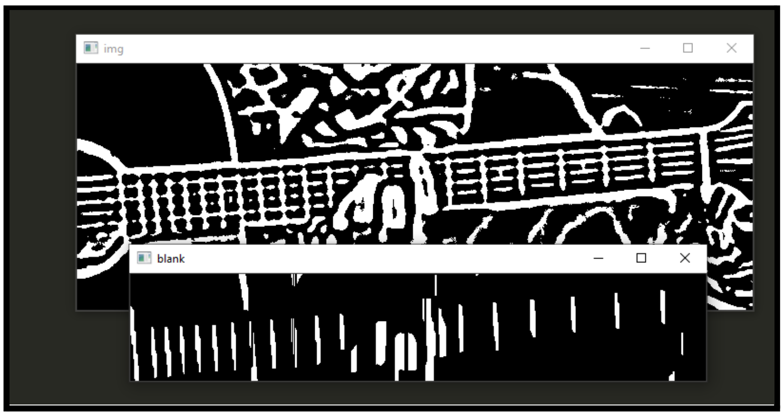
1. **Acceptable**
   1.1 Object Detector
   1.2 Hand detection
   1.3 Edge Detection
2. **Acceptable post error handling**
   2.1 Object Detector
   2.2 Hand detection
   2.3 Edge Detection
3. **Unacceptable**
   3.1 Object Detector
   3.2 Hand detection
   3.3 Edge Detection

1.1

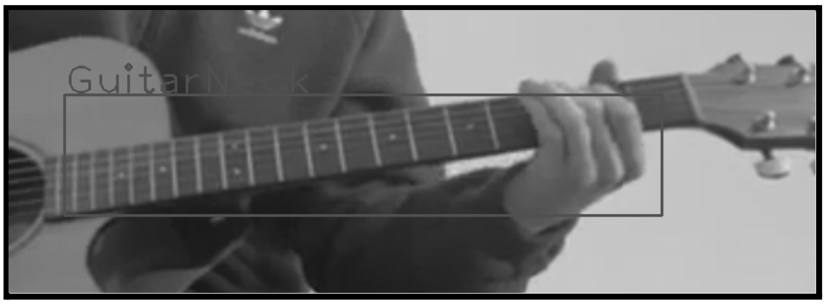
1.2



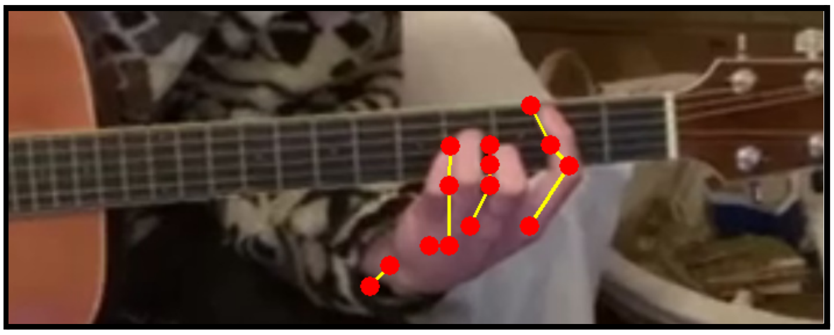
1.3



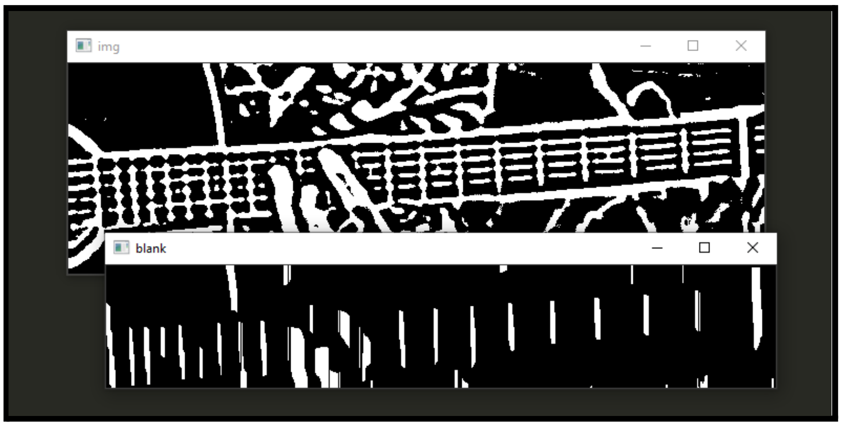
2.1



2.2

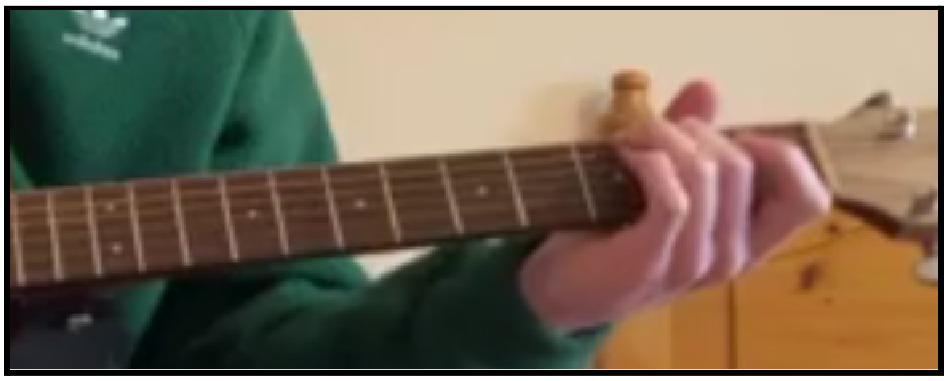


2.3

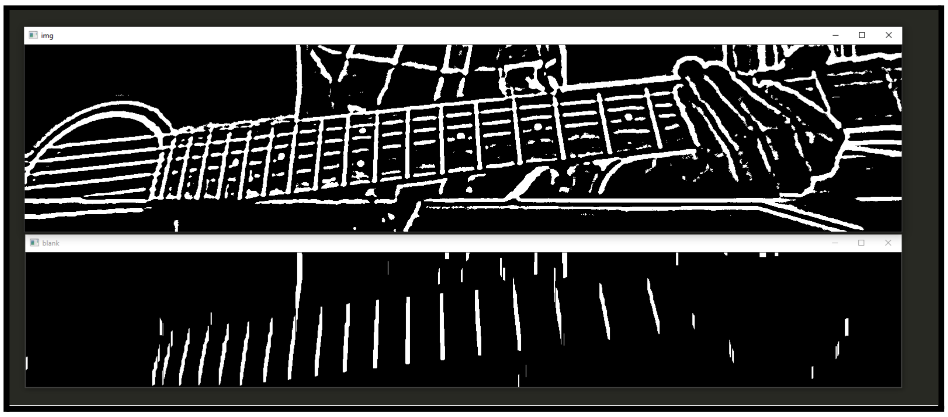


3.1

3.2



3.3



3.2 Audio Component Testing

The audio component consisting of the DoReMir API and it's XML parser script were tested with a suite of 30 mp3 files. These consisted of audio of guitar playing,2 with background noise to test the capabilities of and limitations of the API's output however these returned null from the API. The audio files are split into three categories, fast and slow monotonic and noisy. The Audio API is limited to monotonic audio; single notes at a time. The speed of play of these notes or the time between the notes is the main factor of difficulty in analysing the audio. When the exception is caught as the audio returns null we quickly display an error message and instruction as to improve the audio for analysis. Acceptable post handling means the output of the audio API had to be altered in order to achieve correct tabs. (Ghost note removal or removing tied notes)
Results are below:

	Acceptable	Acceptable post Error handling	Exceptions Caught but Unreadable	Overall Accepted	Amount Tested
Slow Monotonic	6	7	1	13	14
Fast Monotonic	1	5	6	6	12
Noisy Audio	0	0	2	0	2

6.4 System Testing

System testing was done through a suite of 40 videos. Some of which were edge cases; black video with guitar audio, video without any guitar playing at all as well as noisy audio. Of these edge cases none were able to produce correct tabs as expected but they were caught and handled appropriately (edge cases made up 12.5% of the suite). The other videos contained video and audio of sufficient quality but of differing degrees of difficulty for audio and image analysis.

	No Editing Required	Acceptable Post But requiring editing	Exceptions Caught but No Tabs Returned (Visual)	Exceptions Caught but No Tabs Returned (Audio)	Tabs Created	Amount Tested
Correct video	12	9	3	11	21	35
Edge Cases	0	0	2	3	0	5

As seen in these results the audio API used (DoReMir) has its limitations with nearly a third of the "acceptable" videos returned from the API without acceptable audio analysis. This is the greatest shortcoming of the app. When a player plays the song too quickly or lets notes ring over one another the API runs into problems and returns null. However when we receive some output from the API we can normally interpret the output to give acceptable results of which only half require any editing at all, with the majority of the ones which require editing only requiring minor adjustments. As stated in results and future work we would like to improve the app by doing our own audio analysis in future to improve this.