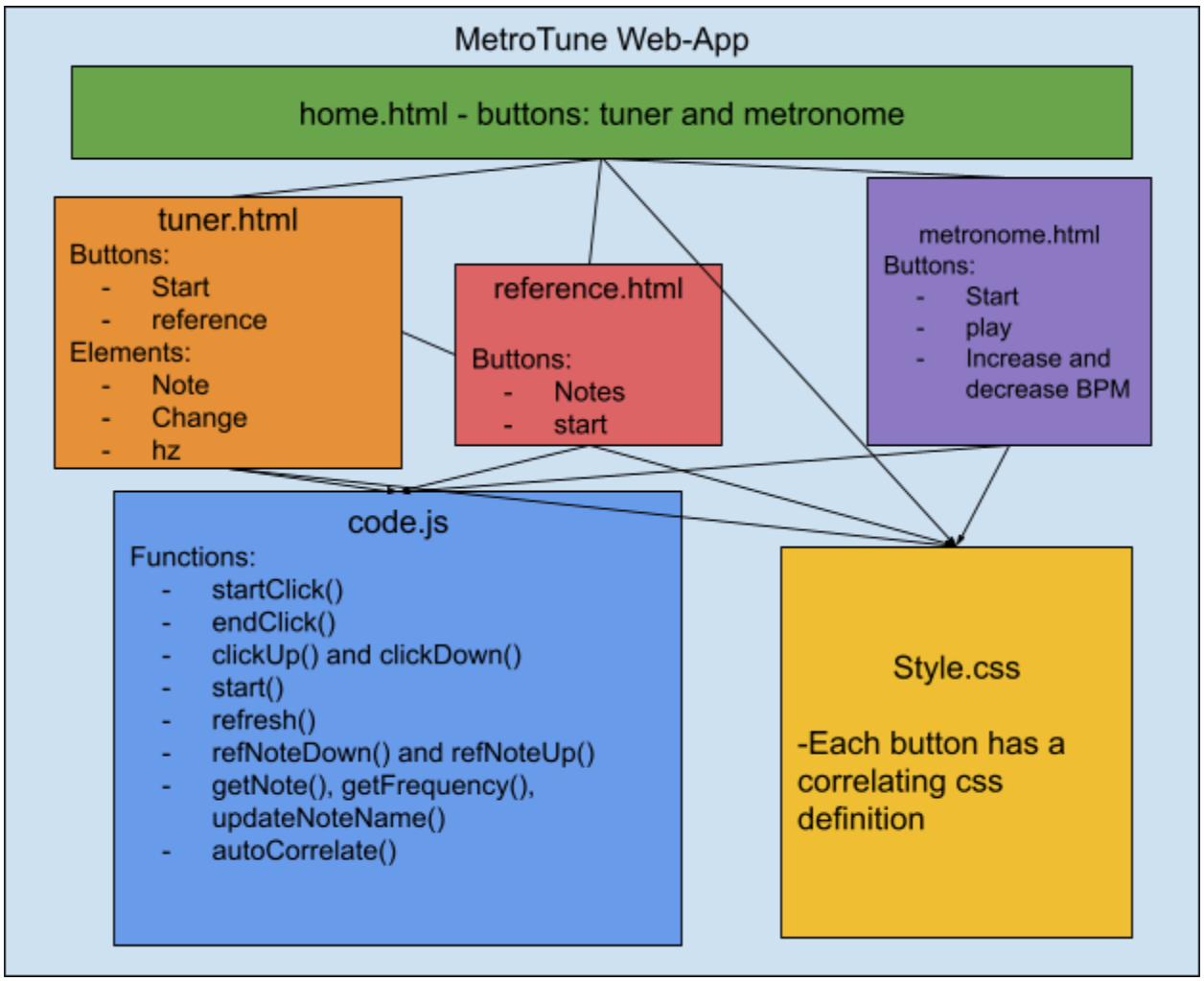
MetroTune Requirements Document

Table of Contents:

5.1 Introduction

MetroTune is a multi-page web app where the user is able to access a tuner, metronome, and reference notes. This is an all-in-one site where the user can access some of the most important tools for musicians. Each html page is controlled by a correlating javascript file as well as styled by a css file. The following sections detail the functional, performance, and environmental requirements of MetroTune.



5.2 Functional Requirements

The functional requirements of the MetroTune web app include that the user must be able to activate the audio context on each page. This will allow each tool to be able to run properly. After clicking start to start the tool on each page, the user must then be able to interact with the tools. The MetroTune web app is divided into two subsystems. The first being the webpage subsystem that includes the HTML and CSS files. The second is the application subsystem that includes the Javascript file that controls the web app.

5.2.1 Webpage Subsystem

5.2.1.1 Each page shall have a start button to activate the audiocontext
5.2.1.2 Each page (other than the home page) shall have a linked button to go back to the homepage
5.2.1.3 The homepage shall have linked buttons to access the tool pages
5.2.1.4 The homepage shall have a brief description of the app itself
5.2.1.5 The homepage shall have the MetroTune logo displayed at the top of the screen
5.2.1.6 The tuner page shall display the current note name
5.2.1.7 The tuner page shall display the cents away from the note being in tune
5.2.1.8 The tuner page shall display the hz of the note being played
5.2.1.9 The tuner page shall have a linked button to access reference notes
5.2.1.10 The metronome page shall have a play button to start the metronome
5.2.1.11 The metronome page shall have a pause button to stop the metronome
5.2.1.12 The metronome page shall display the current beats per minute (BPM)
5.2.1.13 The metronome page shall have a plus button to add a beat to the BPM
5.2.1.14 The metronome page shall have a minus button to subtract a beat to the BPM
5.2.1.15 The reference page shall have a series of playable notes
5.2.1.16 The reference page shall have the note names displayed on their buttons
5.2.1.17 The reference page shall display the hz of the current note being played
5.2.1.18 The pitch tests page shall display the link to the interval testing page
5.2.1.19 The pitch tests page shall display the link to the note testing page
5.2.1.20 The interval testing page shall display the list of clickable notes
5.2.1.21 The interval testing page shall display if the user clicks the right or wrong interval
5.2.1.22 The interval testing page shall display a next button to refresh the interval if correct
5.2.1.23 The note testing page shall display the list of clickable notes
5.2.1.24 The note testing page shall display if the user clicks the right or wrong note
5.2.1.25 The note testing page shall display a next button to refresh the interval if correct

5.2.2 Application Subsystem

5.2.2.1 The start function shall activate the audio context from the Web Audio API
5.2.2.2 The start function shall be called when any start button is pressed
5.2.2.3 The start function shall call the refresh function
5.2.2.4 The refresh function shall update the window
5.2.2.5 The refresh function shall call the autocorrelate function
5.2.2.6 The refresh function shall call the changeHz function
5.2.2.7 The refresh function shall call the updateNoteName function
5.2.2.8 The refresh function shall call the updateChange function
5.2.2.9 The autocorrelate function shall detect pitch using the ACF2+ algorithm
5.2.2.10 The changeHz function shall update the text context of the Hz element
5.2.2.11 The updateNoteName function shall update the text context of the note with the current note

5.2.2.12 The updateNoteName function shall call the getNote function
5.2.2.13 The getNote function shall get the note name from the note pitch
5.2.2.14 The updateChange function shall call the getChangeNeeded function
5.2.2.15 The updateChange function shall update the cents text content
5.2.2.16 The getChangeNeeded function shall return the cents value
5.2.2.17 The refNoteDown function shall play the note being clicked
5.2.2.18 The refNoteUP function shall stop playing the note that was clicked
5.2.2.19 The startMetronome function shall start the metronome
5.2.2.20 The startMetronome function shall call the play function
5.2.2.21 The startMetronome function shall call the stop function
5.2.2.22 The startMetronome function shall initiate the audio context
5.2.2.23 The play function shall set the interval for the metronome
5.2.2.24 The play function shall call the playSound function
5.2.2.25 The play function shall control the beats according to tempo
5.2.2.26 The playSound function shall create an oscillator
5.2.2.27 The playSound function shall play a frequency according to the accent
5.2.2.28 The playSound function shall link the audio context to the destination
5.2.2.29 The playSound function shall play a frequency for a specified amount of time
5.2.2.30 The stop function shall clear the interval
5.2.2.31 The stop function shall reset the currentBeat
5.2.2.32 The add function shall add one to the tempo variable
5.2.2.33 The add function shall call the updateBPM function
5.2.2.34 The add function shall call the stopPlay function
5.2.2.35 The sub function shall subtract one to the tempo variable
5.2.2.35 The sub function shall call the updateBPM function
5.2.2.37 The sub function shall call the stopPlay function
5.2.2.38 The updateBPM function shall set the BPM text content to the current BPM value
5.2.2.39 The updateSlider function shall set the tempo to the slider value
5.2.2.40 The updateSlider function shall call the updateBPM function
5.2.2.41 The updateSlider function shall call the stopPlay function
5.2.2.42 The addBeat function shall add a beat up to 14 beat(s)
5.2.2.43 The addBeat function shall call the updateBeatPB function
5.2.2.44 The addBeat function shall call the stopPlay function
5.2.2.45 The subBeat function shall subtract a beat up to 1 beat(s)
5.2.2.46 The subBeat function shall call the updateBeatPB function
5.2.2.47 The subBeat function shall call the stopPlay function
5.2.2.48 The updateBeatPB function shall update the beatPB text content with the current beats value
5.2.2.49 The stopPlay function shall check if the metronome is playing
5.2.2.50 The stopPlay function shall call the stop function
5.2.2.51 The stopPlay function shall call the play function
5.2.2.51 The accentedClick function shall check if the beats are accented
5.2.2.52 The accentedClick function shall assign the beats accented class
5.2.2.53 The refNoteDown function shall create the audio context
5.2.2.54 The refNoteDown function shall call the playRef function
5.2.2.55 The refNoteUp function shall disconnect the audio context
5.2.2.56 The playRef function shall connect the audio context to the destination
5.2.2.57 The playRef function shall play a defined frequency
5.2.2.58 The getRandFreq function shall call the getRandomInt function
5.2.2.59 The getRandFreq function shall randomly select one of eight frequency values

5.2.2.60 The getRandomInt function shall return a random number between a defined min and max
5.2.2.61 The intCheckClick function shall check if the interval clicked is correct
5.2.2.62 The intCheckClick function shall call the intCheckAnswer function
5.2.2.63 The intCheckAnswer function shall set the text class to true or false
5.2.2.64 The checkNext function shall check if answered is true
5.2.2.65 The checkNext function shall clear the answerChecker text content if answered is true
5.2.2.66 The checkNext function shall call the getRandFreq function

5.3 Performance Requirements

5.3.1 Time to load windows
5.3.1.1 When moving between windows, the web app shall load the next window in under 1 second(s)

5.3.2 Time to start tool
5.3.2.1 When starting a tool (tuner or metronome), the tool shall be usable in under 1 second(s)

5.3.3 Time to play note
5.3.3.1 The time to play a reference note shall be under 1 second(s) for it to play a sound

5.3.4 Time to control metronome
5.3.4.1 The metronome shall begin playing within 1 second(s) after starting
5.3.4.2 The metronome shall stop playing within 1 second(s) after stopping
5.3.4.3 The BPM shall update within 1 second(s) after adding a beat
5.3.4.4 The BPM shall update within 1 second(s) after subtracting a beat
5.3.4.5 The accented note shall play a different frequency under 1 second(s) after selecting option
5.3.4.6 The accented note shall stop playing a unique sound after deselecting option under 1 second(s)
5.3.4.7 The slider shall update the BPM under 1 second(s) after clicking

5.3.5 Time to display tuner information
5.3.5.1 The note shall appear in under 1 second(s) after note recognition
5.3.5.2 The hz shall appear within 1 second(s) that the note appears on screen
5.3.5.3 The cents shall appear within 1 second(s) that the note appears on screen
5.3.5.4 The note’s color shall coordinate with cents within 1 second(s) that the note appears on screen

5.3.6 Time to use pitch tests
5.3.6.1 The time to display if the clicked note/interval is right or wrong shall be under 1 second(s)
5.3.6.2 The time to check if answered shall be under 1 second(s)
5.3.6.3 The time to reset the note/interval after clicking next shall be under 1 second(s)

5.3. Time to return
5.3..1 Each function listed within 5.2.2 shall return its desired output in under 1 second(s)

5.4 Environment Requirements

5.4.1 Metrotune shall have the following requirements in each category:

Category	Requirement
Hard Drive Space	5MB
Operating System	Microsoft Windows OS or Mac OS
Browser	Chrome, Safari, Bing, etc.

5.4.2 MetroTune shall be able to run using any standard web browser
5.4.3 MetroTune shall run locally in any standard web browser
5.4.4 MetroTune shall not require any special computing hardware to operate
5.4.5 MetroTune shall require 5 MB of hard drive space to download the web app