

This is improved over your initial draft,
but I still have concerns about the project.
We'll just have to see how it goes.

I do not understand why you changed
the formatting of only some pages (the new ones).
I also once again entreat you to proofread.

Visono

Adding a visual dimension to a traditionally aural experience.

G 4

Clayton Smith

F 7

Mark McGrotty

C 5

Shaun Confrey

February 13, 2015

U 1 - see comment inside

I 3

S 4

A 4

P 3

31

Contents

1	Project Goals	2
2	Feature Description	3
3	Component Details	6
3.1	Front End	6
3.2	Back End	6
4	Possible Issues	8
4.1	Front End	8
4.2	Back End	8
5	Intended Audience	9
6	Schedule	10
7	Acceptability Criteria	12
7.1	Minimum requirements	12
7.2	Additional features	12
8	Sources	13

Project Goals

Visono aims to provide users with a visual dimension to a traditionally aural experience. This is done by presenting users with a simple and intuitive user interface, connecting them directly to our powerful visualization algorithms that will capture the essence of the user's favorite audio tracks, and ~~transform them into~~ a stunning display of light, color, and sound.

transforming those tracks into

It is good that you have included a screen mockup.
It is good that you have numbered your figure.
But you must refer to that figure by its number
in your narrative text. You need to connect the two
for the reader. You cannot assume that readers will

Feature Description

make the connection
themselves.

Visono is a single page web-app that will provide users with a simple, yet elegant interface that will facilitate the creation of beautiful visualizations from the user's favorite songs. Through the use of a carefully designed control flow and minimalistic interface, users will find Visono easy and enjoyable. Visono will have four distinct pages that users will be able to cycle through: the media selection (homepage), control panel, viewing page, and a gallery of users visualizations.

The media selection page is where the user will be brought when they visit Visono. The first step in the design process is selecting a song from either a list of music provided by Visono or from the user's personal library. Once an appropriate media file has been supplied, the file will be pushed to the server where processing will take place in order to extract key information about the user's audio track.

(see
Figure
2.1).

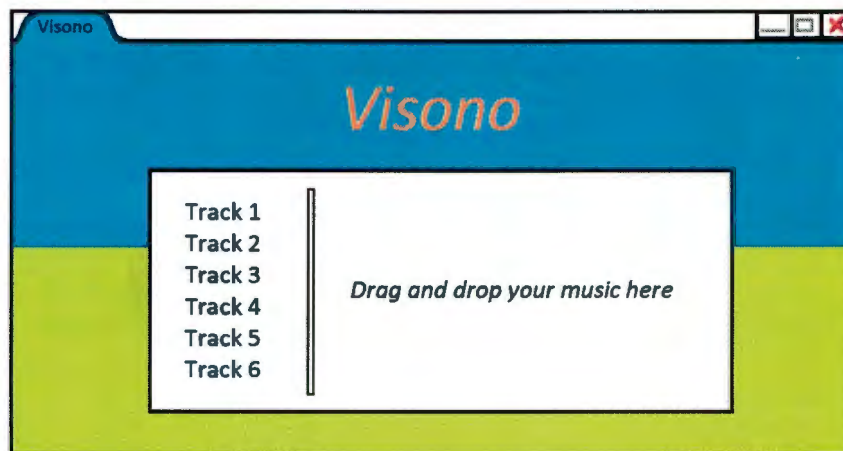


Figure 2.1: Visono's homepage and media selection page.

While the processing takes place, the user will be taken to the control panel where they can control certain aspects of the visualizer such as color, speed, and size. This page can only be displayed by fulfilling the requirements of the media selection page, which are to select or provide a media file. This is done so as to not confuse the user by presenting them with unnecessary menus and data. The default values in the

(see Figure
2.2).

control panel will be governed by the returning results from the processing performed on the user's audio track. This will allow ~~the~~⁹ users to see what settings Visono thinks would suit their music best. If the users do not want to use the values chosen for them, the value ~~may~~⁵ be easily overwritten. Once ~~the~~⁹ users are satisfied with their settings, they may ~~then~~⁹ submit ~~their~~^{those} settings to the server, where the visualizer is created.

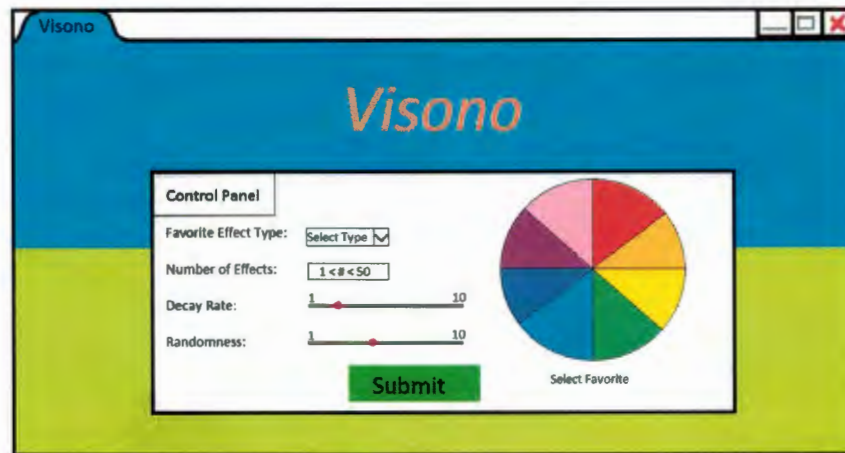


Figure 2.2: Visono's control panel

As soon as the visualizer is complete, the control panel will be replaced by a viewing element where the user's visualizations will be displayed. This page will also give users the option of downloading or saving their visualizations to Visono's database. If they decide to save a visualization to the server, their visualization will be displayed in a public viewing gallery.



Figure 2.3: Visono's control panel

I certainly like the inclusion of diagrams, but this one has two problems. First, why is there a red arrow that is crossed out? Why not just omit that red arrow? There are no red arrows for other paths (Gallery to Control Panel and Gallery to Viewing Page) that are not allowed, so why the red arrow from

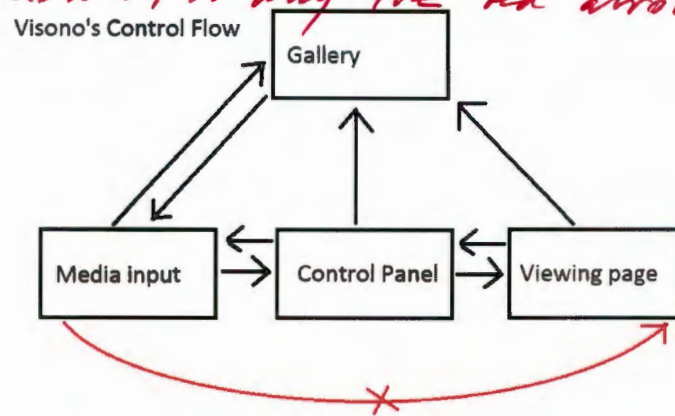


Figure 2.4: Visono's control flow showing the order that pages can be viewed and accessed.

Media Input to Viewing Page? And now that I look at this more closely, you must be able to go from the Gallery directly to the viewing page, right? Thus, I think an arrow is missing. Second, you can't just plop this or any other figure into your document without discussing it in your narrative text.

↑ referencing and

Component Details

3.1 Front End

The user interface (UI) will be constructed using modern tools *such as* ~~like~~ HTML5, CSS3, and various JavaScript libraries such as AngularJS to create a reactive and responsive experience for the users. We will be using [Foundation](#), a CSS framework, to aid in the construction of the UI and to maintain an even look throughout the project. Our use of CSS frameworks will guarantee compatibility across browsers and devices. Visono will be a single page web application built using AngularJS's and Foundation's templating engines. We have decided to use a single page application because it will allow us to easily guide users between the different stages in the visualization development process.

Once a file has been fed into Visono's media input page, it will be checked to make sure the file has the proper media extensions. This is done using basic string manipulation in JavaScripts⁹. Files also need to be checked to make sure that they do not exceed 7 minutes in length or 150 megabytes in size, as we do not want to generate excessively large video files and clog the network. The Web Audio JavaScript library will allow us to easily validate the user's files. When the media file has been validated it will be passed to the server for processing using AngularJS's "post" requests. Until the server is ready, the UI will display a loading or processing window to indicate that Visono is working on their~~x~~ song. As soon as the server is done processing it will push a newly created MP4 to the user ~~which it is then~~ *that can be* played synchronously with the original audio track.

3.2 Back End

Visono's back end will be run on Microsoft's Azure platform, a high grade cloud computing and hosting platform currently free for all college students. Users will post

their media files to the server where they will be caught and dealt with appropriately. The first step in the processing phase is to perform an analysis of the audio file to extract key information such as tempo, downbeats, and patterns in music. This will be done in either C# or C++. A data file will then be created that reflects the characteristics of the user's song which is then sent to an OpenGL application that will generate the visualization. The last step in the processing phase is saving the resulting output from the OpenGL application as an MP4. The conversion to MP4 will be done using OpenCV. Once the visualization has been created and saved, it will be sent directly back to the user.

It would be optimal to be able to merge both the MP3 and MP4 files into one unified MP4 file, as it would be easier to play back for the user and would allow them to download the video if they so choose. However, due to the overly complicated nature of most video and audio libraries we may have ^{to} settle with simply playing both video and audio tracks simultaneously. We are currently looking into integrating FFmpeg into Visono, as it is the most commonly used free media library available.

⊛ I asked about (or flagged with ??) the word "caught" in your initial draft, yet you've used it again with no explanation. I don't think this word is appropriate here.

Possible Issues

4.1 Front End

One of two issues that we must be aware of is video playback becoming out of sync with the music. There is no easy way to merge video and audio, and until such a method is found, the UI will have to try to play both the MP4 and MP3 synchronously. We can check to make sure the two streams are playing synchronously by comparing the time-codes of each stream. If the two come out of sync then we simply adjust the audio streams' time-code to match that of the video.

4.2 Back End

Problems that the back end may experience range anywhere from unexpected server downtime to the inability to correctly produce a video for the user. Due to the nature of our project we feel that memory allocation errors may be our undoing. The videos that are produced will not be small and generating a large number of visualizations simultaneously may consume all of the server's resources.

A possible solution would be to implement a load balancing mechanism to ensure the server is not overwhelmed with processing requests. This load balancing mechanism would be a queue that holds processing requests until enough resources are available. There is also the case where the user leaves the web-app while their video is still being processed. We would need to devise a way to terminate any abandoned processing requests as to increase server performance.

*this phrase makes no sense.
As I said about your draft, you need to proofread much more carefully.*

Intended Audience

There are countless people across the globe who thoroughly enjoy listening music but rarely ever get the opportunity to see what they're listening to. Visono is geared towards anyone who wishes to experience a deep and immersive representation of their music. With its simple UI and straightforward controls, Visono is a product that can be used by people of all ages.

This is not much better than what you originally submitted, which I said was unacceptable. "Anyone" and "of all ages" are red flags in user descriptions. I don't think that my 3-year-old granddaughter could use Visono, much less my 3-month-old granddaughter. You have to think much more deeply about who your user is.

Schedule

ALPHA RELEASE: The first release (02/26/2015) will consist of the first draft of a working UI that is able to push media files up to the server. The server will **NOT** return any of the files back to the user but will create a data file that represents the original file provided by the user. Visualization will not be implemented at this point.

BETA RELEASE: The beta release (04/10/2015) will have a working back end that will accept media files from the UI, create the visualization, and push the resulting MP4 back down to the user. The video and the audio should be able to be played synchronously for the user's enjoyment. If we are ahead of schedule we may begin to implement additional features and general improvements and optimizations.

Responsibility	Goal	Due Date
Clayton/ Mark/ Shaun	Create usable work flow, Acquire Microsoft Azure server, and register a domain name.	02/10/2015
Mark/ Shaun	Begin construction on UI and back end server	02/17/2015
Clayton	Able to open and process audio data	02/24/2015
Shaun	Finish first draft of UI	02/24/2015
Mark	Create a server that accepts pushes from UI	03/10/2015
Shaun	UI must be able to push files to server and wait for a file in response.	03/15/2015
Clayton	Able to create a basic visualization	03/17/2015
Clayton/ Mark/ Shaun	Design visual effects	03/24/2015
Mark/ Shaun	Have UI able to play media pushed back from the server. In time for user testing	04/07/2015
Clayton/ Mark/ Shaun	Optimization and debugging / Final revisions	04/14/2015
Clayton/ Mark/ Shaun	IFF ON TIME: Implement extra features	04/21/2015
Clayton/ Mark/ Shaun	Presentations	04/23 - 28/2015

OK

Acceptability Criteria

7.1 Minimum requirements

There are several requirements that must be met before Visono can be considered a finished project. The items listed below are key components of Visono. If any component were to fail, Visono would be rendered the project useless.

1. UI must be able ^{to} accept MP3 files *proofread!*
2. Media files need to be validated and pushed to the Azure server
3. Able to open and process media files and create meaningful data representation
4. Able to generate an MP4 file from the data collected
5. Server must push newly created video back down to the user
6. UI needs to be able to play the MP4 f and MP3 files synchronously

7.2 Additional features

When **all** of the minimum criteria are met, the group may begin implementing additional features. One of the features that we would like to see implemented is rating feature for the visualizations the users of Visono thought were good.

Surely there are others you could include here.

Sources

- Foundation Framework: <http://foundation.zurb.com/>
- Templating with AngularJS: <https://scotch.io/tutorials/single-page-apps-with-angularjs-routing-and-templating>
- Web Audio: <http://webaudio.github.io/web-audio-api/#introduction>
- OpenGL: <https://www.opengl.org/>
- Microsoft Azure: <http://azure.microsoft.com/en-us/>
- Video.js <http://www.videojs.com/>

good