CS 386 Assignment 1

Requirements Elicitation and Specification

2/18/15

Cameron Gaskin

Herbie Duah

John Dance

Nico Bedford

**Task 1: Project Scope**

Justification

Our system will help people integrate their music listening on Youtube with their music listening on Spotify along with added functionality that will be beneficial to all the parties involved. This is the reason I believe your business needs this product.

First this product will address the problem of people having to jump back and forth between Spotify and Youtube for their music listening needs. Second this product will allow additional functionality to listening to music on Youtube, such as links to lyrics or artist wikipedia pages. Lastly this system will be beneficial for all the products involved, Youtube will become that much more functional, Spotify will receive more plays and customers, and the artist will have their information that much more accessible.

In conclusion our product will allow for two of the biggest music listening platforms, Spotify and Youtube, to be more integrated along with added functionality, making certain information pertaining to the song and artist more accessible, all while being beneficial to Spotify, Youtube and the artist involved.

Scope Description

The key characteristic of the product is that we want to insure is that our product is able to scan an inputted Youtube video link, with audio data similar to that of the commercial mp3 we want to look up, and pull certain information from the video. The main information that we are looking for is the artist and song title, so long as it exists within the music fingerprinting database. That information will be displayed to the user, so long as the mp3 is within the database we are querying for information.

The next characteristic is that we will take that information that we have obtained, song title and artist, and query Spotify’s API to provide us with a link to the song on Spotify. Another characteristic is that our system will also be able to provide the user with a link to the artist wikipedia page and a link to the song lyrics, given that they both exist. This will be the key characteristics of our system.

What our system will not be able to do is: Scan any youtube video for music and return the information. The video has to be formatted similarly to a mp3 file for a scan to work properly on it. Our system will also not be able to guarantee that all the information mentioned above will exist. It is possible that none of the information exist, in which we will return a message to the user letting them know that no information on the song exists, within the databases that we will be querying.

Acceptance Criteria

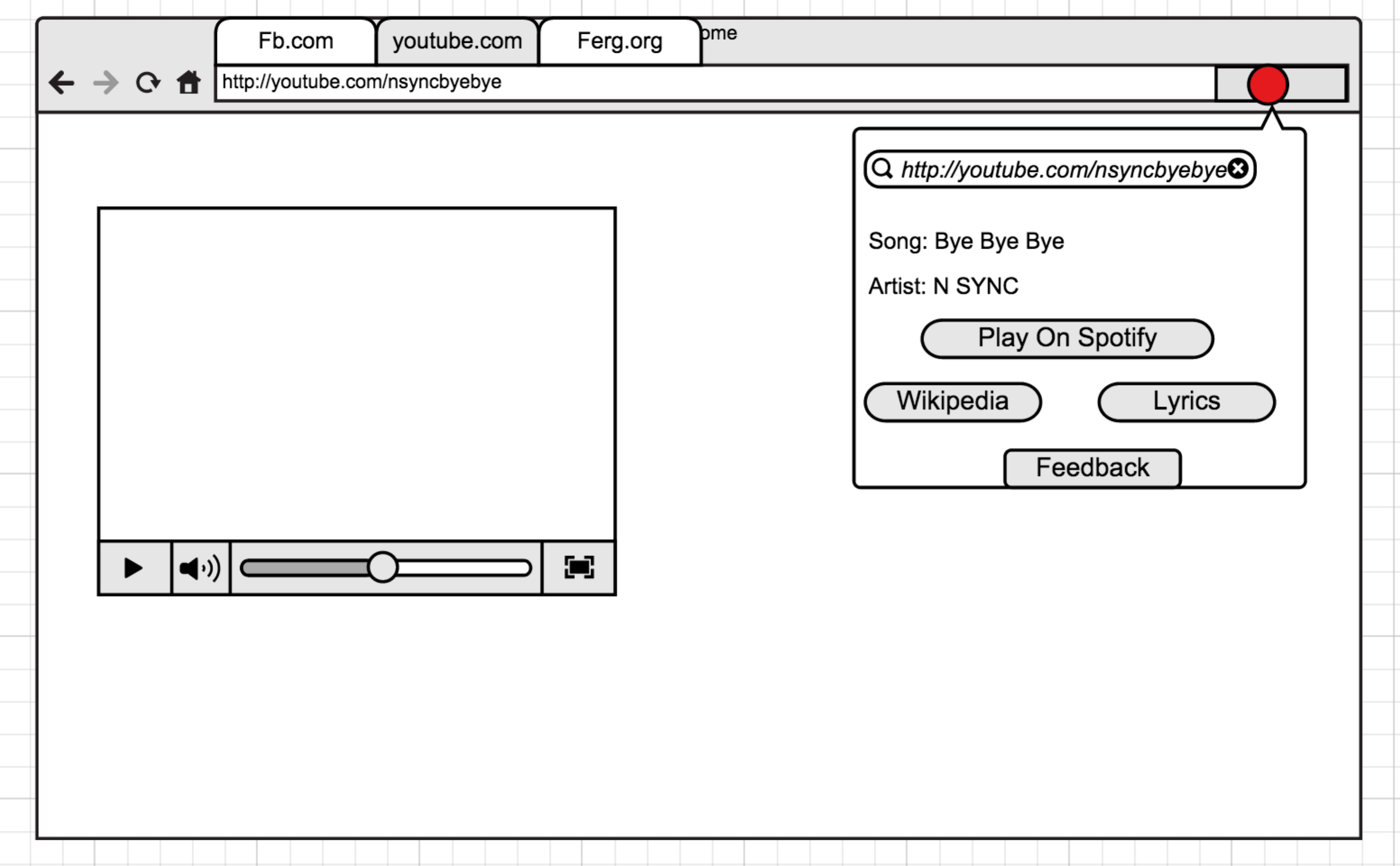
For our potential client to accept the product a few criteria must be met. The first criteria is that our product must be able to scan a Youtube video, whose mp3 data does not exist in our database, and return a message to the user letting them know that were unable to find the song in our database. This may mean that the song may not exist in the database or that the video passed into our program was not a song at all.

Second our product must be able to scan a Youtube video, whose mp3 data does exist in our database, and return all the aforementioned data regarding that song that we are able to find. If the only information we are able to find is the song and artist name, then that may be the only thing displayed to the user. If we are able to find certain information then they will be displayed. The user will know that information beyond the artist and song title was not findable by the absence of a link to that information.

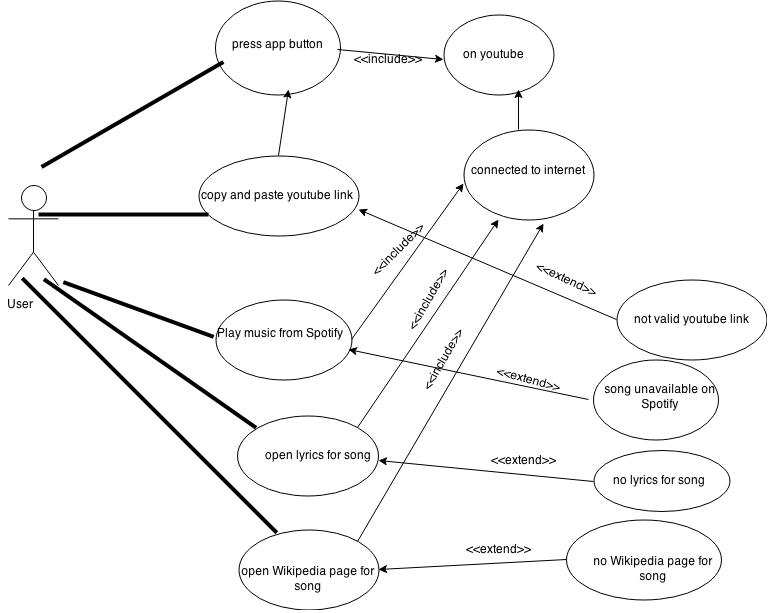
Lastly, our product must have a feedback functionality to let the user send a message to us and let us know if certain functionality is not working correctly, or to just let us know how great the product is. An example of expected functionality that may not work correctly is that the song and information returned may not match the actual song inputted. This will allow us to adjust after the fact to better our product in the future.

Assumptions

The first assumption that will be made is that the Youtube video passed through our product will be a video whose mp3 file is a song within the database we will use to look up songs. Second, we will also assume that there will be no specific user accounts, therefore no need for logins (Maybe in a later version). Third, we are assuming that the people using this program already have Spotify accounts, as the Spotify link relies on that.  Fourth, we are assuming that the user will not want to see a history of their scanned songs, as they can just add the song to a Spotify or Youtube playlist. Fifth, we are assuming that the user has time to wait for the mp3 to be extracted from the video and scanned for the proper information (which may take a bit of time). The sixth assumption being made with this product is that the user uses both Youtube and Spotify to fulfill their music needs.

**Task 2: Use Case Model**

**Task 3: Interface Prototype**

****

**Task 4: Critique the development process**

When using the waterfall process model we have to consider what the situations that would cause this process to have optimal implementation. With our system we aim to create a seamless interaction between Spotify and Youtube using a Google Chrome Extension that will also serve as a means to provide the user with Lyrics and a Wikipedia page if available. The biggest downfall of this process we see right away is the ability for flexibility within the development process. Since no one in our group is very familiar with Chrome Extensions or the Spotify API this could be a very difficult challenge since the waterfall with prototyping does not allow for much flexibility when dealing with changes in the system or with changes in the direction of the system at all.

When we are first looking at implementing our system using the Waterfall Model (with prototyping), we took a look at the strengths that the Waterfall Model has to offer. One of the main strengths and reasons to use the Waterfall Method (with prototyping) is that it works well for projects or problems that are very well understood and not likely to have any particular changes to the design or the requirements. When applying this to our project it looks like a very poor model to be using. First of all, as the teacher, you have hinted at possibly changing the requirements and not being very decisive with what you are going to require so this is the first problem. Secondly, out of the four people in our group, we have never implemented a Google Chrome Extension before so we are not going to know 100% what to expect in what code we will be using, what programs we will be using, or how we will put it all together, so this is a big downside for us when using the Waterfall Method (with prototyping). Also, if we are approached with a problem that is unsolvable or impossible to get around, we may need to change some of the requirements or ways we approach the project, which is not something that the Waterfall Method (with prototyping) really allows for. All this aside, I think the strictness of the Waterfall Method (with prototyping) will be beneficial to making sure that certain tasks and deadlines are met because of the lack of flexibility, which does not allow for much procrastination. All in all, I think completing our project with the idea that we have today will be quite the challenge especially considering the fact that we are using the Waterfall Model. So no, the Waterfall Model does not seem to be the most optimal for the project we would like to do.

Lastly, after discussing with our team, we came to the conclusion that the model that would probably be more suitable with what we have chosen to work on would be Agile. Agile is a very flexible and easy to work with process. Since Agile focuses more on producing working software rather than documentation we would be able to have much more time to figure out what would be a more realistic approach to completing the project. Also, Agile allows many opportunities to respond to changes in the system or project that the Waterfall Method (with prototyping) does not. An example of this could be if we could not get a working Spotify API in our current project then we would be in a very rough place with the Waterfall Model, but with the Agile Model we would be able to decide that maybe we could focus on changing the requirements to help match what we have to work with and then push to get out a working product for our customer.

To conclude our critique, I think it will be a difficult project due to the lack of flexibility of Waterfall in addition to the lack of knowledge of what is required from our project. Because of this, we will be very thorough on our requirements and every other step of the process in order to prevent any future struggling in the process we are working with.