Cumulyrics

Software Requirements Specification

Team 19

Kyle Van Landingham, Alec Schule, Alec Fong, Andrew Zolintakis, Noah Bergman

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification is to clearly and completely state the requirements, features, and functions of the system. This SRS is being written for the clients (Professor, TAs, CPs) and the developers (Team 19) to reference throughout the development of the system.

1.2 Scope

The system to be developed will be named Cumulyrics.

The system is intended to allow users to generate word clouds from selected artists and share those word clouds with their peers through Facebook. Users will be able to generate word clouds for English-speaking artists only. Available artists will also be determined by the limitations of third-party APIs. Users will have access to relevant data with regards to the artist(s)' most frequently used words and songs by the same artist(s) which contain those words.

The system shall accomplish the task of generating a word cloud of the top two hundred and fifty most frequent words in an artist(s) lyrics obtained from each song by the artist(s). The system will also allow users to see the songs in which a particular word is used where those songs will be ordered by the frequency of the word appearing within that song. In addition to that functionality, the system shall allow users to select particular songs to view the lyrics associated with the selected song with the word previously selected highlighted within the lyrics. The system will also allow users to merge multiple artists' word clouds together by taking all lyrics from both artists and picking the top two hundred and fifty words from that collective group of lyrics. Finally, the system shall allow users to share their generated word cloud to Facebook in the form of an image.

1.3 Definitions, acronyms, and abbreviations

Definition	Description
API	API is defined as an Application program interface which is a set of routines, protocols, and tools for building software applications. An API makes it easier and more convenient for a programmer to build a system because it provides useful building blocks.
SRS	SRS is defined as a Software Requirements Specification. This document is an SRS.

Contract	A legally binding document agreed upon by the customer and supplier. This includes the technical and organizational requirements, cost, and schedule for a product. A contract may also contain informal but useful information such as the commitments or expectations of the parties involved.
Client	A client is a piece of computer hardware or software that accesses a service made available by a server. The server is often (but not always) on another computer system, in which case the client accesses the service by way of a network.
Server	In computing, a server is a computer program or a device that provides functionality for other programs or devices, called "clients". This architecture is called the client–server model, and a single overall computation is distributed across multiple processes or devices. Servers can provide various functionalities, often called "services", such as sharing data or resources among multiple clients, or performing computation for a client. A single server can serve multiple clients, and a single client can use multiple servers. A client process may run on the same device or may connect over a network to a server on a different device.
User	The person, or persons, who operate or interact directly with the product. The user(s) and the customer(s) are often not the same person(s).
HTTP protocol	The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text.
TCP/IP protocol	TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet).
Stemming	Stemming a word is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form.

1.4 References

Reference	Link
830-1998 - IEEE Recommended Practice for Software Requirements Specifications.	http://ieeexplore.ieee.org.libproxy2.usc.edu/servlet/opac?punumber=5841
Lyric-api - API for fetching song lyrics from lyrics.wikia.com.	https://github.com/rhnvrm/lyric-api
Genius api used to get artists' profile image.	https://docs.genius.com/
Genius API terms of service	https://genius.com/static/terms
Professor Halfond's Lecture Slides	Accessible via Blackboard

1.5 Overview

The rest of this document is intended to completely record all requirements of the system. This includes all interfaces, functions, performance requirements, and other necessary requirements.

We start with the overall description, which will provide a high level overview of the system and will cover its most important aspects. The document will then cover all other requirements, going into specific detail about each one.

The document will end with supporting information. This will include the appendix, which clearly documents our team's process in making this SRS.

2. Overall Description

2.1 Product perspective

This product is independent and self-contained with respect to other systems and related products.

2.1.1 System Interfaces

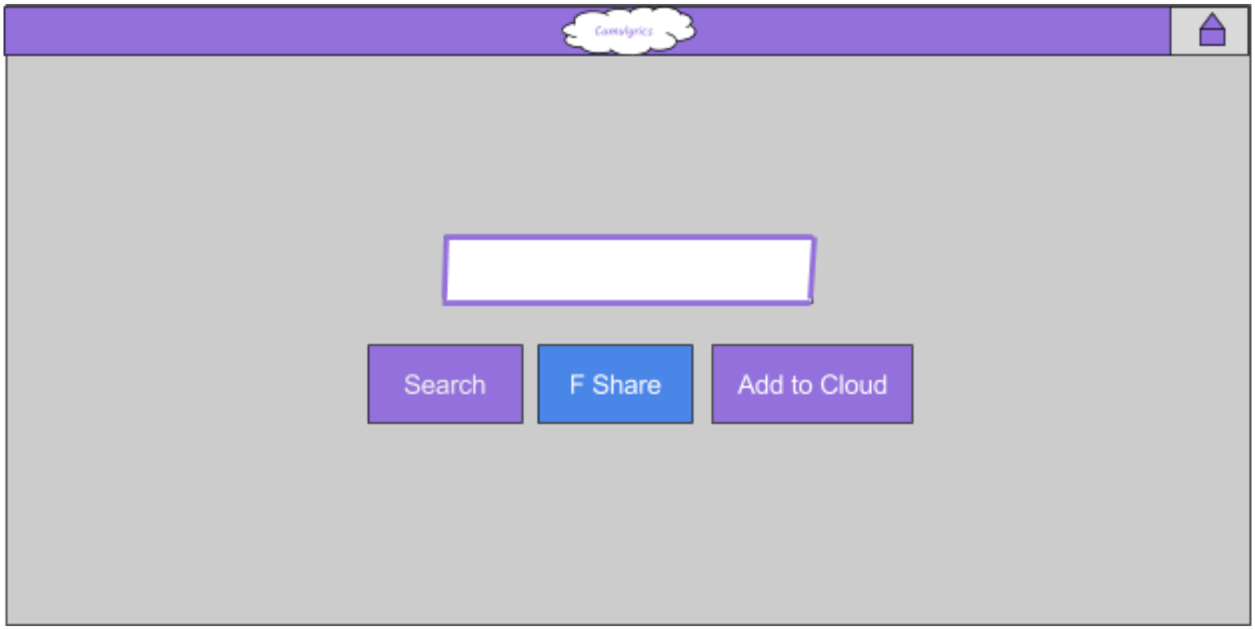
The system will not need to interface as part of a larger overall system. However, there will be server client interaction. The user will interact with the client through a web browser which will send requests to the server which will handle requests and send back the appropriate data to the user's browser. The server will also interface with third party APIs as well as a database stored on the server machine.

2.1.2 User Interfaces

The system shall be accessible to the user through a standard web browser with javascript enabled. All screens of the system will contain a title bar with the name of the system, Cumulyrics centered on the title bar, and a home button on right of the title bar. The background color of the title bar will be a shade of purple that will be identical to the purple referenced throughout the rest of the section. Any back button specified further in the document will appear within the title bar on the left side of the title bar. All screens in the system shall have a solid gray background that is uniform throughout the application.

2.1.2.1 Screen 1

The main screen displayed on the website shall contain a text box centered on the screen with three buttons directly below. Those buttons shall be in order from left to right, “Search”, “Share to Facebook”, “Add to Cloud”. The “Search” button shall have a purple background. There are no requirements for color on the other two buttons. The “Share to Facebook” and “Add to Cloud” buttons will be disabled on this screen. The only components of this screen that the user is able to interact with are the “Search” button, the textbox, and the home button. The textbox shall have a purple outline with a white background and black color text. When a user begins to type into the textbox, an autocomplete list of names will appear below the textbox aiding the user in choosing a valid artist to select. The list of names will change dynamically as the user types in the textbox. If there are multiple artists with the same name a list of selectable artists accompanied with pictures will appear in the autocomplete list for the user to choose.



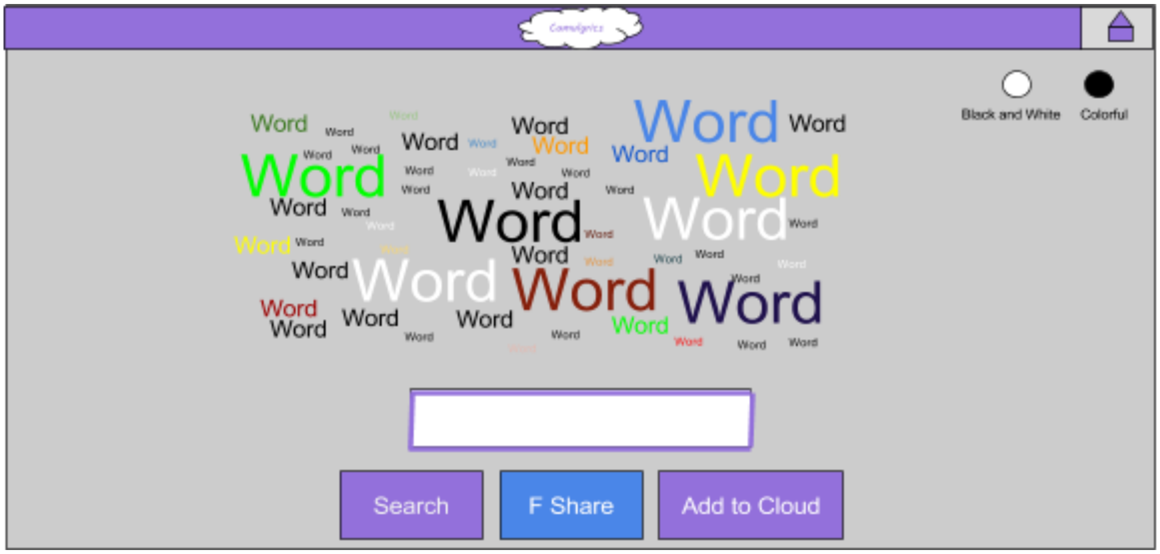
2.1.2.2 Screen 2

After a user specifies an artist and selects the "Search" button, a rectangular word cloud shall be displayed centered on the screen while the textbox and three buttons below the textbox are shifted toward the bottom of the user's screen. If the artist does not exist or does not return results from our system then a descriptive error message will appear where the word cloud would otherwise be generated. If there is an already existing word cloud generated and a specified artist does not exist or does not return results from our system then a smaller descriptive error message will appear within a dialog box that can be dismissed by the user by pressing the dialog box "Ok" button. If there is an already existing word cloud generated and a specified artist is valid then the previous word cloud will be replaced with a new word cloud. Two radio buttons shall be displayed on the top right of the screen titled "Black and White" and "Colorful" that determines the color of the word cloud.

If a user specifies another artist and clicks on the "Add to Cloud" button the word cloud shall be generated in such a way that the two artists' two hundred and fifty most frequent words are displayed in the cloud. If the specified artist does not exist or does not return results from our system then a smaller descriptive error message will appear within a dialog box that can be dismissed by the user by pressing the dialog box "Ok" button.

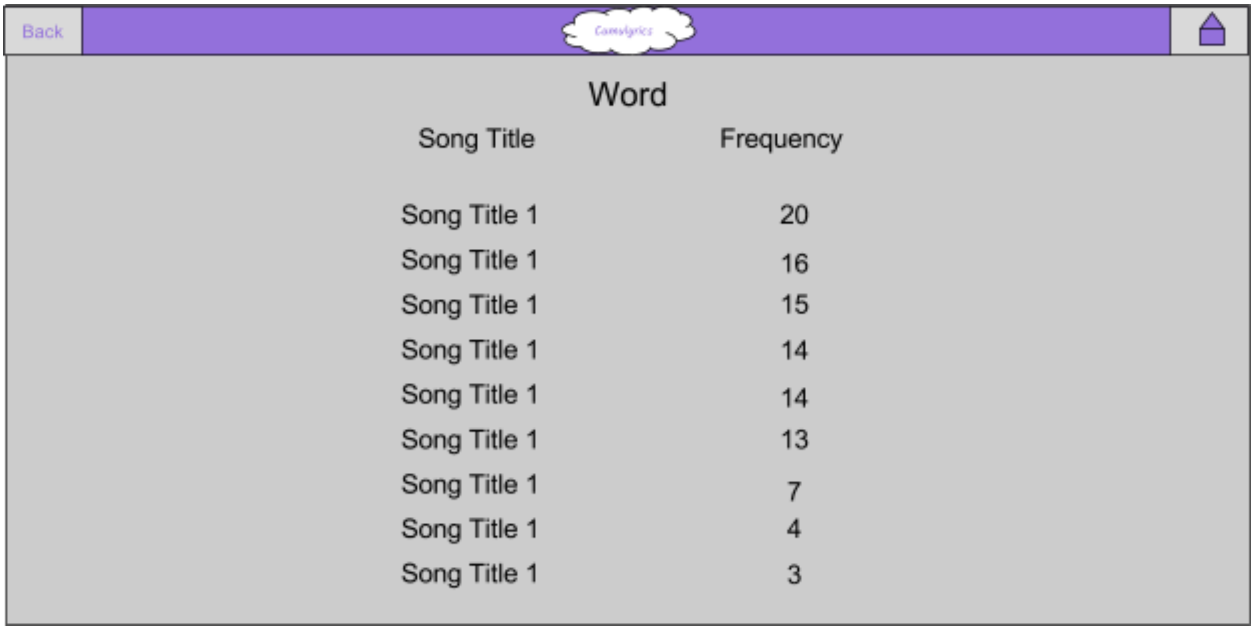
When a user selects the "Share to Facebook" button a pop up window will be generated and allow the user to login to Facebook and post an image of the word cloud using their account.

The word cloud will consist of horizontally orientated words in a rectangular shape with the size of specific words determined by their frequency in songs by the specified artist(s). The more frequent a word is used by an artist the larger that word appears within the overall word cloud. The two hundred and fifty most frequent words used by the artist(s) will be displayed in the word cloud. If a user clicks on a word within the word cloud they will be taken to another screen within the same tab of the browser.



2.1.2.3 Screen 3

This other screen will be titled with the word the user clicked and below a list of songs containing the word from the specified artist(s) will appear in descending order based on the frequency of the word in each song. The list will consist of two columns titled “Song Title” and “Frequency”. The “Song Title” column will contain the song titles for the specified artist(s) and the “Frequency” column would contain the number of times the word appears within the corresponding song on the left column. Songs that do not contain the selected word will not show up in the list. A back button will appear in the top left of the screen. The user is able to select a song title from the first column of the list to take them to the next screen described below.

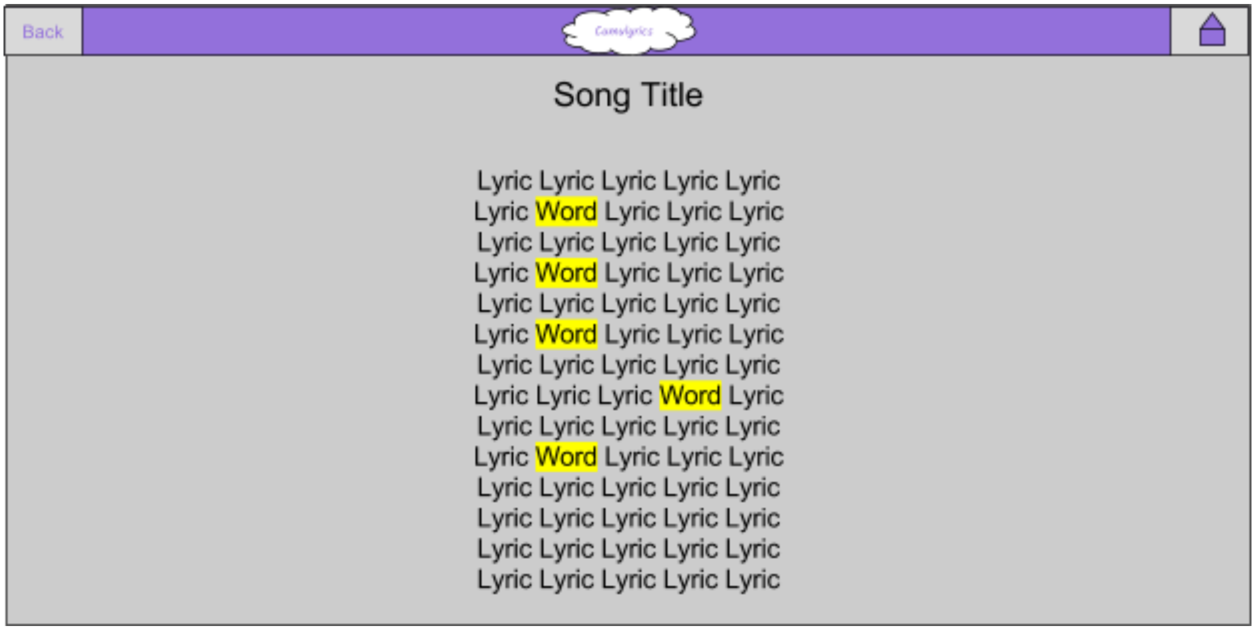


Word

Song Title	Frequency
Song Title 1	20
Song Title 1	16
Song Title 1	15
Song Title 1	14
Song Title 1	14
Song Title 1	13
Song Title 1	7
Song Title 1	4
Song Title 1	3

2.1.2.4 Screen 4

If a user chooses to select a song within the list they will be taken to another screen with the title of the screen being the song title selected. Below the title will be the lyrics for the selected song with the specified word from the previous screen highlighted in yellow wherever it appears in the song lyrics. In the top left of the screen a back button will appear taking the user back to the previous screen with the list of songs containing the specified word in order of frequency.



Song Title

Lyric Lyric Lyric Lyric Lyric
Lyric Word Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Word Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Word Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Lyric Lyric Word Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Word Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric
Lyric Lyric Lyric Lyric Lyric

2.1.3 Hardware Interfaces

The system is intended to be used through a standard web browser with javascript enabled on a laptop or desktop computer. Mobile optimization for cellular or tablet devices is not a requirement for the system. The server will take care of all computations with respect to the word cloud and send data to the user's browser through TCP/IP and HTTP protocol.

2.1.4 Software Interfaces

The front-end will consist of javascript running in a user's browser. This front-end will request data from a back-end server as needed. The back-end server will be written in PHP. It will rely on a database as well as third-party API's for it's information on artists and song lyrics.

2.1.4.1 Incoming and Outgoing Data

- Outgoing data consists of an artist’s name (when the user requests data from the server for generating a word cloud or types in the search bar), and an artist and track name (when the user requests data from the server for song lyrics)
- Incoming data consists of an artist’s top 250 words, the songs those words appear in, as well as the word’s frequencies (from the server when the user requests data for generating a word cloud), song lyrics (from the server when the user requests the lyrics of a specific track), and a list of three suggested artists (from the server to autocomplete the user’s searches)

2.1.4.2 Services and Communications

As mentioned above, the application will rely on third-party APIs for information. Some information will also be stored in a database to accommodate performance requests. The communication between the front-end and the server will occur in short bursts after the user takes one of the following actions:

- Types into the search bar
- Clicks the search button to generate a word cloud
- Clicks to “add to cloud” button to add an artist into an existing cloud
- Clicks a song from the song list on Screen 3 to view the song lyrics
- Clicks the “Share to Facebook” button to share an image of the word cloud

2.1.5 Comm Interfaces

The system is design as around a client-server model where a web based networked server created using PHP will communicate with a user’s javascript enabled browser. The server handles requests prompted by user interaction with the system and collects and manipulates data to send in the form of html to the user’s browser. The system will also have a database to store relevant information regarding valid artists that shall be accessible to the server.

2.1.6 Memory Constraints

No memory constraints required.

2.1.7 Operations

For a list of operations see section *2.1.2 User Interfaces*.

2.1.8 Site Adaptation Requirements

Before the system is completely functional, the database must be filled with the necessary data including all valid artist names. This data will be collected through scraping the Genius API and the Lyric-API cited in the reference section of the SRS.

2.2 Product functions

The main function of this system will be to receive a musical artist's name and produce a word cloud of the two hundred and fifty most popular words said by that particular artist in his or her songs, excluding stopwords. The generated word cloud shall be sharable as an image through Facebook. Word clouds for more than one different artists shall be able to be merged together. Words in the word cloud shall be clickable to show a list of all songs by the artist(s) that contain that specific word. Then the user shall be able to view the lyrics of any of the songs.

2.3 User characteristics

The intended users include the Professor, TAs, and CPs. These users will have completed high school and have a high level of technical expertise with regards to the system.

2.4 Constraints

The system will only provide word clouds for artists that are provided by free API calls offered by music websites. Additionally, the song lyrics the system takes into account for each artist will be limited to songs available from free API calls offered by music websites. Song lyrics will also be limited to English.

2.5 Assumptions and dependencies

The system will need internet connection and will need the API's we use from the websites specified in the references section to work, in order for the system to work correctly.

3. Specific Requirements

3.1 External Interface

Inputs:

- Mouse: This is used to select an item on our website, such as buttons or search bars
- Keyboard: This is used to type in the names of artists
- Touch Screen: This can be used in both the same ways as the keyboard and mouse

Outputs:

- Computer Screen: Displays the website to the user, this includes user options and the word cloud
- Smart Phone/Tablet Screen: This has the same purpose as the computer screen

3.2 System Features

3.2.1 Creating a Word Cloud For a Single Artist

- Purpose: Generate and present a word cloud from the lyrics of an artist to the user
- Stimulus/Response: On the main page, if the user types an artist into the search bar and then clicks the Search button a word cloud will appear

- Word Cloud Requirement 1: The word cloud shall contain the 250 most commonly used words in the lyrics of all of the songs of the artist. The cloud will stem words to their base form and exclude stop words
- Word Cloud Requirement 2: The lyrics processed will be stemmed to their base form and stop words will be excluded before determining the words that will make up the word cloud
- Word Cloud Requirement 3: The more prevalent a word is found in the lyrics the larger the word shall be in the word cloud
- Search Requirement 1: If the user clicks the Search button without entering in any letters the system shall not do anything
- Validity Requirement 1: If a user searches an artist that does not exist an error message shall be shown. If there is a word cloud present it should remain visible when the error message is shown

3.2.2 Sharing the Word Cloud to Facebook

- Purpose: Allow the user to share an image of a generated word cloud to Facebook
- Stimulus/Response: On the main page, if the user clicks the share button he or she shall be prompted with a Facebook pop up window. The user shall then be able to sign in to their Facebook and share the word cloud image
- Share Button Requirement 1: If there is no word cloud present the share button shall be disabled

3.2.3 Adding an artist to the current Word Cloud

- Purpose: Generate and present a word cloud with the lyrics from more than one artist
- Stimulus/Response: If there is an already created word cloud the user shall be able to type in an artist name into the search bar and press the add to cloud button. The system shall then generate and present a new word cloud
- Multiple Artist Word Cloud Requirement 1: The artists included in this word cloud shall be the newly searched artist as well as all of the artists that were included in the previously generated word cloud
- Multiple Artist Word Cloud Requirement 2: The word cloud shall contain the 250 most commonly used words in the lyrics of all of the songs of all of the artists. It can be thought of as putting every lyric from every song of each artist in the cloud into a bucket and then creating a word cloud with the 250 words that show up most in that bucket
- Multiple Artist Word Cloud Requirement 3: The lyrics processed will be stemmed to their base form and stop words will be excluded before determining the words that will make up the word cloud
- Multiple Artist Word Cloud Requirement 4: The words that show up more often in the lyrics shall be larger in size than words that show up less often

- Multiple Artist Word Cloud Requirement 5: No artists shall be cleared when the add to cloud button is pressed, instead the new artist shall be added to a cloud that contains all previously added artists
- Multiple Artist Word Cloud Requirement 6: The user shall have the ability to combine at least 2 artists together. If the system has a max artist restriction, the add to cloud button will be disabled if the current word cloud contains the max amount of artists
- Multiple Artist Word Cloud Requirement 7: The user shall be allowed to add an artist to the cloud even if the artist is already included in the crowd
- Search Requirement 1: If the user clicks the add to cloud button without entering in any letters the system shall not do anything
- Clearing the Added Artists Requirement 1: If the user types a valid name into the search bar and presses Search, all previous artists shall be cleared and the new word cloud shall only contain words from the lyrics of the currently searched artist

3.2.4 Search bar Autocomplete

- Purpose: Help the user find a valid artist names
- Stimulus/Response: If the user types into the search bar the system shall show a dropdown box showing the autocomplete suggestions
- Autocomplete Requirement 1: If the user is typing in the search bar, the system shall show a dropdown box that has up to 3 names of artists found in the system's database that start with the letters currently typed into the search bar
- Autocomplete Requirement 2: If there is no artist name found in the system's database that starts with the letters typed in the search bar, no dropdown box shall be visible
- Autocomplete Requirement 3: If the user clicks on one of the suggested names in the dropdown box, the selected name shall appear in the search bar and and the drop down box shall disappear. The user shall then have to click to Search button or add to cloud button in order to generate a new cloud
- Search Requirement 1: If multiple artists exist for the query a icon with a picture of the artist will be displayed allowing the user to distinguish which artist

3.2.5 Presenting all songs that contain a selected word

- Purpose: Show songs in which the selected word is contained by the artist(s).
- Stimulus/Response: A list of song and word frequencies are presented to the user.
- Requirement 1: All songs in which the word is contained are shown.
- Requirement 2: Frequencies are listed beside song name.

3.2.6 Presenting the lyrics of a selected song

- Purpose: To show the lyrics of an individual song.
- Stimulus/Response: User is brought to window displaying the lyrics.

- Requirement 1: Original word in which the user clicked on is highlighted in yellow within the lyrics.

3.2.7 Going back a page

- Purpose: Allow navigation of the software by returning user to the previous page
- Stimulus/Response: User is brought back to previous window
- Requirement 1: Button exists on all windows except for main window.

3.2.8 Returning to the main page

- Purpose: Allow navigation of the software by returning user to the main page
- Stimulus/Response: User is brought back to main page
- Requirement 1: Button exists on all windows.

3.2.9 Toggling the Word Cloud color scheme

- Purpose: Allow the user to change color of word cloud from black and grey to colorful.
- Stimulus/Response: On the main page, if the user clicks the radio button denoting a black and white color scheme the word cloud will become black and white. If the user clicks on the radio button denoting a colorful color scheme the word cloud will become colorful.
- Requirement 1: The buttons are disabled when no cloud exists.

3.3 Performance Requirements

The system must generate the word cloud within 5 secs given feasible internet connection and feasible user hardware.

3.4 Logical Database Requirements

The system will store all artists names that have valid search results within our API calls.

3.5 Design Restraints

The system will need to rely on third party APIs to obtain the songs for each artist and the lyrics for each song.

3.5.1 Standards Compliance

The system shall adhere to the Terms of Service associated with the Genius API.

3.6 Software System Attributes

3.6.1 Reliability

No reliability requirements requested.

3.6.2 Availability

No availability requirements requested.

3.6.3 Security

No security requirements requested.

3.6.4 Maintainability

No maintainability requirements requested.

3.6.5 Portability

No portability requirements requested.

4. Supporting Information

4.1 Index

4.2 Appendix

4.2.1 Appendix A: Document Creation Process

Elicitation:

Initially, Team 19 met and compiled a list of questions based on what information was needed to begin the project. (see Appendix B) Next, the team proceeded to carry out the elicitation phase using the interview technique. The first stakeholder interviewed was Professor Halfond, with questions being asked from the previously compiled list. Next, the team interviewed CP Glarence to understand the requirements of the other stakeholders. The same list of questions was used as with Professor Halfond, to understand the CPs' requirements to an equal degree.

Specification:

Following each interview, the team translated the answers given into requirements specifications. After both interviews were completed, the team compiled the two separate specifications list into one final list.

Negotiation:

In compiling the separate requirements of the stakeholders into one final document, there were conflicts in which the stakeholders desired different things. In these cases, the team attempted to come up with a solution that satisfied both stakeholders. For example, the stakeholders desired different color schemes, but both agreed to have the option to switch between them. Once the final list of specifications was obtained, the team was forced to negotiate over requirements that seemed unfeasible. Specifically, a one second requirement for loading web pages did not seem feasible, so the team contacted stakeholders with a proposed substitute. Stakeholder Matthew Lam agreed to the team's proposed 5 second timeline for word cloud generation.

Validation:

After fully understanding the requirements of the project, the team proceeded to create this software requirement specification, including mock-ups of the application for stakeholder validation.

4.2.2 Appendix B: List of Interview Questions

Open-Ended Questions:

What is your intended use for this product?
How do you want the word cloud to be presented?
How does the navigation of artists work? (search details)
What user customizations should we have for the word cloud?

Close-Ended Questions:

What color theme do you have in mind?
Is there an option to shape the word cloud?
Should all of the songs of an artists be included?
Can the user choose the color and font of the word cloud?
How many queries/users should the system be able to handle at once/a day?
What set of artists should be included?
How should the website react to window resizing? (absolute position or alter the size and position according to the window)
What should happen if an artist doesn't exist or no lyrics are found?
Should a specific database of songs be used?
Is there a speed requirement?
How big should the words be?
What user options should there be to export the word cloud? (jpg, png, pdf, etc.)
How many words/minimum times said to include?
Should we exclude explicit lyrics from the word cloud?