

CS 1530 - Project Requirements

Brian Choromanski ( **Choromanski**)

Eric Hunzeker ( **EricHunzeker**)

Gerard McGlone( **jmcglone75**)

 github.com/erichunzeker/SWE-Spring2019

Functional Requirements

Functional Req 1.

IDENTIFIER: **Landing/Home Page**

DESCRIPTION: User is greeted with GUS' homepage when they go the website

PURPOSE: To have a landing page

ACTORS/TRIGGERS: User enters "www.url of GUS.ext" in a browser

Functional Req 2.

IDENTIFIER: **Search Bar present**

DESCRIPTION: On the homepage, a search box for querying music is present.

PURPOSE: Easy and straightforward way to query music

ACTORS/TRIGGERS: User is on homepage

Functional Req 3.

IDENTIFIER: **Live Query results**

DESCRIPTION: As a user is typing a query in the search bar, a certain number of results are displayed in a list view below the bar in a live updating fashion.

PURPOSE: So user can potentially access a result without having to query in its entirety

ACTORS/TRIGGERS: User types the name of existent music

Functional Req 4.

IDENTIFIER: **Select Song/Artist/Album**

DESCRIPTION: Near the search bar is a selection for querying a song, artist, or album.

PURPOSE: To further refine the search to match the users wants.

ACTORS/TRIGGERS: User is on homepage.

Functional Req 5.

IDENTIFIER: **Valid Song Search Results**

DESCRIPTION: When a user chooses to query a song, and results for that song indeed exist, then they will be taken to a new page and presented with a list of songs of that name.

PURPOSE: To give the user a list of common songs matching the desired name.

ACTORS/TRIGGERS: User queries a song that is present on streaming services

Functional Req 6.

IDENTIFIER: **Invalid Song Search Results**

DESCRIPTION: When a user chooses to query a song, and no results for that name are found, then they will be taken to a new page and presented with "no songs found".

PURPOSE: To inform the user that their query did not return any results

ACTORS/TRIGGERS: User queries a string that doesn't match songs on any streaming services

Functional Req 7.

IDENTIFIER: **Valid Artist Search Results**

DESCRIPTION: When a user chooses to query an artist, and results for that artist indeed exist, then they will be taken to a new page and presented with a list of artists of that name.

PURPOSE: To give the user a list of common artists matching the desired name.

ACTORS/TRIGGERS: User queries an artist that is present on streaming services

Functional Req 8.

IDENTIFIER: **Invalid Artist Search Results**

DESCRIPTION: When a user chooses to query an artist, and no results for that name are found, then they will be taken to a new page and presented with "no artists found".

PURPOSE: To inform the user that their query did not return any results.

ACTORS/TRIGGERS: User queries a string that doesn't match artists on any streaming services.

Functional Req 9.

IDENTIFIER: **Valid Album Search Results**

DESCRIPTION: When a user chooses to query an album, and results for that album indeed exist, then they will be taken to a new page and presented with a list of albums of that name.

PURPOSE: To give the user a list of common albums matching the desired name.

ACTORS/TRIGGERS: User queries an album that is present on streaming services

Functional Req 10.

IDENTIFIER: **Invalid Album Search Results**

DESCRIPTION: When a user chooses to query an album, and no results for that name are found, then they will be taken to a new page and presented with "no albums found".

PURPOSE: To inform the user that their query did not return any results.

ACTORS/TRIGGERS: User queries a string that doesn't match albums on any streaming services.

Functional Req 11.

IDENTIFIER: **Unique Link Generation**

DESCRIPTION: When a user has a list of results from a valid query, they are able to click "generate link" for the desired result, which they can then copy and share with anyone

PURPOSE: A unique link for a song, artist, or album is a more compact, easier way to share music with a friend.

ACTORS/TRIGGERS: User queries a valid song, artist, or album and is on the results page.

Functional Req 12.

IDENTIFIER: **Create User**

DESCRIPTION: On the homepage, there are fields and a submission button for creating an account

PURPOSE: User can have an account registered with the service, which will automatically keep track of all links and the date in which they are generated

ACTORS/TRIGGERS: User is on homepage

Functional Req 13.

IDENTIFIER: **User Login**

DESCRIPTION: On the homepage, there are fields and a submission button for logging in as user

PURPOSE: User can have an account registered with the service, which will automatically keep track of all links and the date in which they are generated

ACTORS/TRIGGERS: User is on homepage

Non-Functional Requirements

Non-Functional Req 1.

IDENTIFIER: **Query timeout after 10 seconds**

DESCRIPTION: When a user types an input, the website will take 10 seconds to filter results before timing out

PURPOSE: Prevent vague queries from making expensive calls to APIs or database

ACTORS/TRIGGERS: User types input

Non-Functional Req 2.

IDENTIFIER: **Navigating the website from start to finish should take 3 clicks at most**

DESCRIPTION: The first click should choose the search result type, the second should highlight the search bar, and the third should click the desired result

PURPOSE: A maximum of 3 clicks to get to the desired result keeps the website simple and user friendly

ACTORS/TRIGGERS: User experience navigating website

Non-Functional Req 3.

IDENTIFIER: **Optimize User Interface for all browser types and screen sizes**

DESCRIPTION: The website should maintain its intended look regardless of screen size or browser type

PURPOSE: This prevents users from experiencing glitchy software

ACTORS/TRIGGERS: Browser types & Devices

Non-Functional Req 4.

IDENTIFIER: **Utilize MVC design pattern**

DESCRIPTION: Develop with model, view, controller design pattern using Flask framework

PURPOSE: Website routing will be kept simple and minimal with MVC

ACTORS/TRIGGERS: Development

Non-Functional Req 5.

IDENTIFIER: **Use continuous integration**

DESCRIPTION: Configure a continuous integration client to keep current build updated

PURPOSE: Keep each check-in verified by an automated build, allowing us to detect problems early

ACTORS/TRIGGERS: Version Control

Non-Functional Req 6.

IDENTIFIER: **Extensive testing**

DESCRIPTION: All code written should have extensive coverage

PURPOSE: More code coverage in testing leads to less bugs in production code

ACTORS/TRIGGERS: Testing

Non-Functional Req 7.

IDENTIFIER: **Reusability**

DESCRIPTION: Write code that can be used for multiple tasks

PURPOSE: Keep application lightweight and cut down on unnecessary code

ACTORS/TRIGGERS: Development

Non-Functional Req 8.

IDENTIFIER: **Scalability**

DESCRIPTION: Keep all practices within application suitable for large and small amounts of data

PURPOSE: Develop with scalability as a priority to keep application fast regardless of traffic and use

ACTORS/TRIGGERS: Development

Non-Functional Req 9.

IDENTIFIER: **Security**

DESCRIPTION: Deploy application with virtual environment so no keys are kept in plain text

PURPOSE: No keys kept in plain text to maximize security and minimize vulnerability

ACTORS/TRIGGERS: Deployment

Non-Functional Req 10.

IDENTIFIER: **Reliability**

DESCRIPTION: Application should respond the same regardless of website traffic

PURPOSE: Keep users satisfied and maintain ease of use

ACTORS/TRIGGERS: Development/Deployment