

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF TEXAS AT ARLINGTON**

**SYSTEM REQUIREMENTS SPECIFICATION
CSE 4316: SENIOR DESIGN I
FALL 2023**



**TEAM SILK SONIC
TRACK RECORDS**

**PATRICK ARZOUMANIAN
GUSTAVO CHAVEZ
ABRAHAM MOOKHOEK
AHMED ULLAH
SPENCER WHITEHEAD**

REVISION HISTORY

Revision	Date	Author(s)	Description
0.1	09.29.2023	AM	document creation
0.2	10.16.2023	PA, GC, AM, AU, SW	complete first draft

CONTENTS

1	Product Concept	8
1.1	Purpose and Use	8
1.2	Intended Audience	8
2	Product Description	9
2.1	Features & Functions	9
2.2	External Inputs & Outputs	9
2.3	Product Interfaces	9
3	Customer Requirements	14
3.1	Login/Registration Page	14
3.1.1	Description	14
3.1.2	Source	14
3.1.3	Constraints	14
3.1.4	Standards	14
3.1.5	Priority	14
3.2	Feature Menu	14
3.2.1	Description	14
3.2.2	Source	14
3.2.3	Constraints	14
3.2.4	Standards	14
3.2.5	Priority	14
3.3	Calendar Interface	15
3.3.1	Description	15
3.3.2	Source	15
3.3.3	Constraints	15
3.3.4	Standards	15
3.3.5	Priority	15
3.4	Journal Interface	15
3.4.1	Description	15
3.4.2	Source	15
3.4.3	Constraints	15
3.4.4	Standards	15
3.4.5	Priority	15
3.5	Data Visualization Interface	15
3.5.1	Description	15
3.5.2	Source	15
3.5.3	Constraints	15
3.5.4	Standards	16
3.5.5	Priority	16
3.6	Music Player Interface	16
3.6.1	Description	16
3.6.2	Source	16
3.6.3	Constraints	16
3.6.4	Standards	16
3.6.5	Priority	16

3.7	Settings Interface	16
3.7.1	Description	16
3.7.2	Source	16
3.7.3	Constraints	16
3.7.4	Standards	16
3.7.5	Priority	16
4	Performance Requirements	17
4.1	Consider the data needed	17
4.1.1	Description	17
4.1.2	Source	17
4.1.3	Constraints	17
4.1.4	Standards	17
4.1.5	Priority	17
4.2	Endpoint Reliability	17
4.2.1	Description	17
4.2.2	Source	17
4.2.3	Constraints	17
4.2.4	Standards	17
4.2.5	Priority	18
5	Security Requirements	19
5.1	OWASP Top Ten	19
5.1.1	Description	19
5.1.2	Source	19
5.1.3	Constraints	19
5.1.4	Standards	19
5.1.5	Priority	19
6	Maintenance & Support Requirements	20
6.1	Error Correction	20
6.1.1	Description	20
6.1.2	Source	20
6.1.3	Constraints	20
6.1.4	Standards	20
6.1.5	Priority	20
6.2	Documentation Availability	20
6.2.1	Description	20
6.2.2	Source	20
6.2.3	Constraints	20
6.2.4	Standards	20
6.2.5	Priority	20
6.3	Maintenance Tools	20
6.3.1	Description	20
6.3.2	Source	20
6.3.3	Constraints	21
6.3.4	Standards	21
6.3.5	Priority	21

6.4	Software Environment Upkeep	21
6.4.1	Description	21
6.4.2	Source	21
6.4.3	Constraints	21
6.4.4	Standards	21
6.4.5	Priority	21
6.5	Customer Support	21
6.5.1	Description	21
6.5.2	Source	21
6.5.3	Constraints	21
6.5.4	Standards	21
6.5.5	Priority	21
7	Other Requirements	22
7.1	Database	22
7.1.1	Description	22
7.1.2	Source	22
7.1.3	Constraints	22
7.1.4	Standards	22
7.1.5	Priority	22
7.2	Customer Setup/Config	22
7.2.1	Description	22
7.2.2	Source	22
7.2.3	Constraints	22
7.2.4	Standards	22
7.2.5	Priority	22
7.3	Extensibility	22
7.3.1	Description	23
7.3.2	Source	23
7.3.3	Constraints	23
7.3.4	Standards	23
7.3.5	Priority	23
7.4	Portability	23
7.4.1	Description	23
7.4.2	Source	23
7.4.3	Constraints	23
7.4.4	Standards	23
7.4.5	Priority	23
8	Future Items	24
8.1	Music Player Interface	24
8.1.1	Description	24
8.1.2	Source	24
8.1.3	Constraints	24
8.1.4	Standards	24
8.1.5	Priority	24
8.2	Journal Entries Tone Analysis	24
8.2.1	Description	24

8.2.2	Source	24
8.2.3	Constraints	24
8.2.4	Standards	24
8.2.5	Priority	24

LIST OF FIGURES

1	Rudimentary Login/Registration Mock-up	10
2	Rudimentary Feature Menu Mock-up	11
3	Rudimentary Calendar Mock-up	11
4	Rudimentary Journal Mock-up	12
5	Rudimentary Visualization Mock-up	12
6	Rudimentary Settings Mock-up	13

1 PRODUCT CONCEPT

This section describes the purpose, use and intended user audience for Track Records, a music analytics web application. Track Records will offer a variety of data analytics tools in an easy and intuitive form. Users will be able to utilize these tools in order to look back on their Spotify or Apple listening habits.

1.1 PURPOSE AND USE

Track Records functions as an online data analytics tool-set. The data in question relates to each user's provided music streaming history. Users will be able to understand their listening habits by utilizing our application's data visualization as well as keep a detailed journal of their musical history.

1.2 INTENDED AUDIENCE

This application is being designed for anyone that uses music streaming services, particularly music enthusiasts. It serves as a complementary tool-set to products such as Spotify or Apple Music. Customers of music streaming services have no way to easily review their listening habits. These are the people Track Records will cater to. By filling the void left by Spotify and Apple, our product will be used by an already sizeable set of customers.

2 PRODUCT DESCRIPTION

This section provides the reader with an overview of Track Records. The primary operational aspects of the product, from the perspective of end users, maintainers and administrators, are defined here. The key features and functions found in the product, as well as critical user interactions and user interfaces are described in detail.

2.1 FEATURES & FUNCTIONS

Upon initially opening the web-page, users will be prompted to log-in or perform an account set-up. Once done, the user will be able to select between a variety of features. By default, users will see a calendar view showing when songs were added to playlists and more. If a user desires to utilize different functionality, they can open a slide-out menu to select the desired feature. Once such feature is the user journal. Each user will have access to a digital journal where they can create personal entries. These entries can be tagged with specific songs, artists, and more. Another feature is the modular data visualization tool. This will be provided to users for the purpose of viewing and comparing a variety of listening metrics. Different graph types and data metrics will be selectable in a modular fashion.

2.2 EXTERNAL INPUTS & OUTPUTS

Before anything can be done, user's must create an account to use the product. This is essential to tracking a user's data and storing it in a server. In order for this product to offer solid analysis, users will be required to provide their Spotify extended listening history. Without the user manually entering this information into our system, there is no way to gather information on listening history further than 1 year in the past. Once provided, the system will store the extended listening data in a server. Once the extended listening history has been provided, subsequent activity on the user's streaming accounts can be queried using API calls. Our product relies on API calls to maintain up-to-date and accurate information. If the information provided by Spotify or Apple API calls were to change, this would drastically impact development.

Component	Required Input	Intended Output
Registration	Name, email, password	New user account
Login	email, password	Opens account and gives access to application
Current Streaming Activity Data	API call response from relevant music streaming service	Creation of logs for immediate listening history
Extended Spotify Listening History Data	User-provided file containing extended listening history	Creation of logs for extended listening history, reaching back to music streaming account creation

2.3 PRODUCT INTERFACES

- User Login/Registration Page
 - First screen users see. Required to store user data.
- Application Feature menu
 - Slide-out menu for selecting feature interface. Required to access all of the product's features.
- Calendar Interface

- Default feature interface on login. Designed for intuitive viewing of streaming history
- Journal Interface
 - Allows user to view existing entries or create new ones. Upon opening an entry, user can make edits.
- Data Visualization Interface
 - An interface for modular data visualization. Designed to maximize user control over their own statistical analysis.
- Settings Menu
 - A simple screen meant to let users tweak account and data options.

Logo

Username

Password

Submit

Don't have an account? Register here!

Figure 1: Rudimentary Login/Registration Mock-up

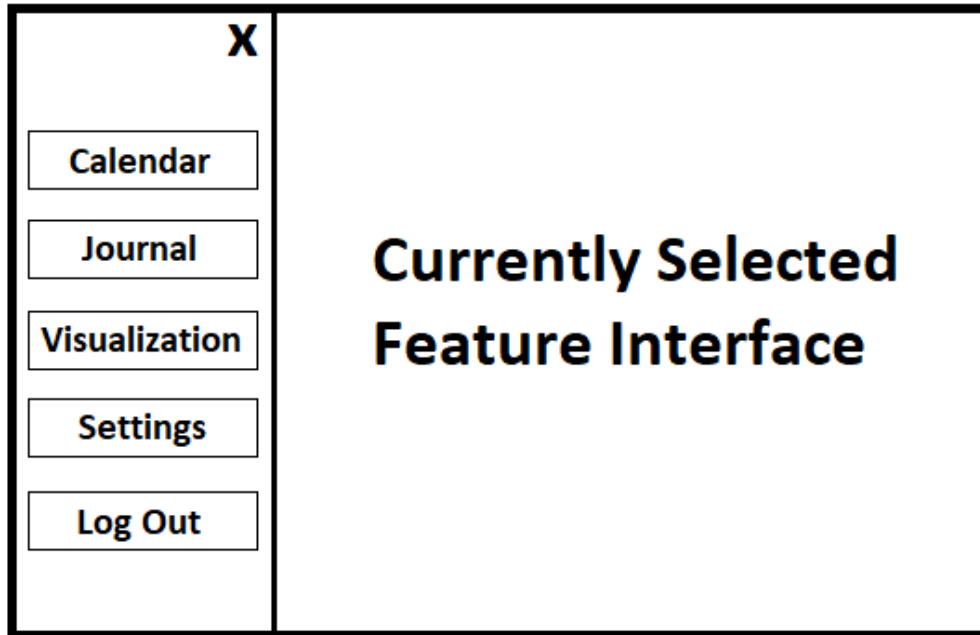


Figure 2: Rudimentary Feature Menu Mock-up

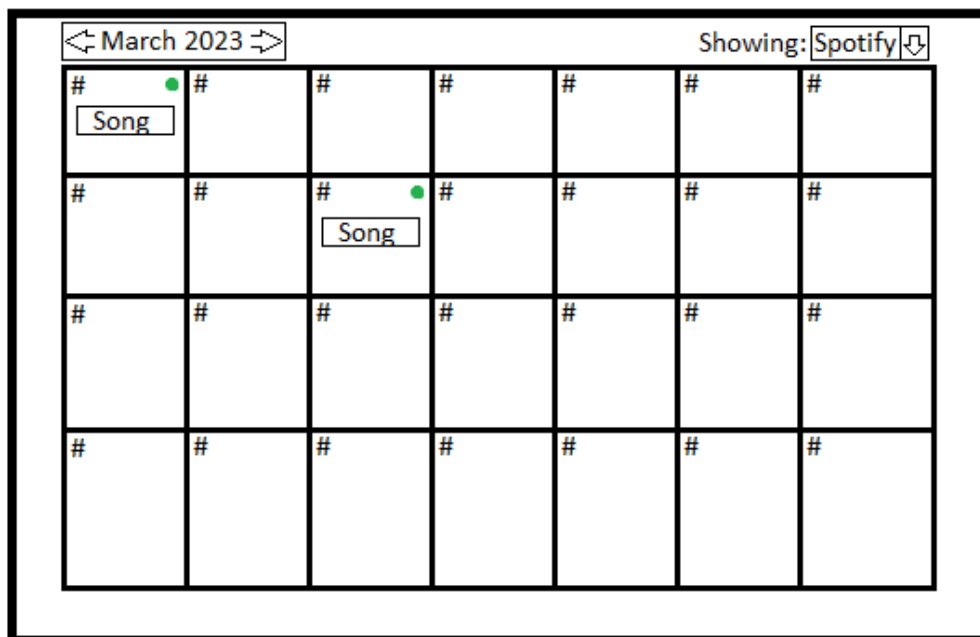


Figure 3: Rudimentary Calendar Mock-up

Search

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

Entry Name

+ Add Entry

Entry Name

Edit Tags

↓

User entered text

Figure 4: Rudimentary Journal Mock-up

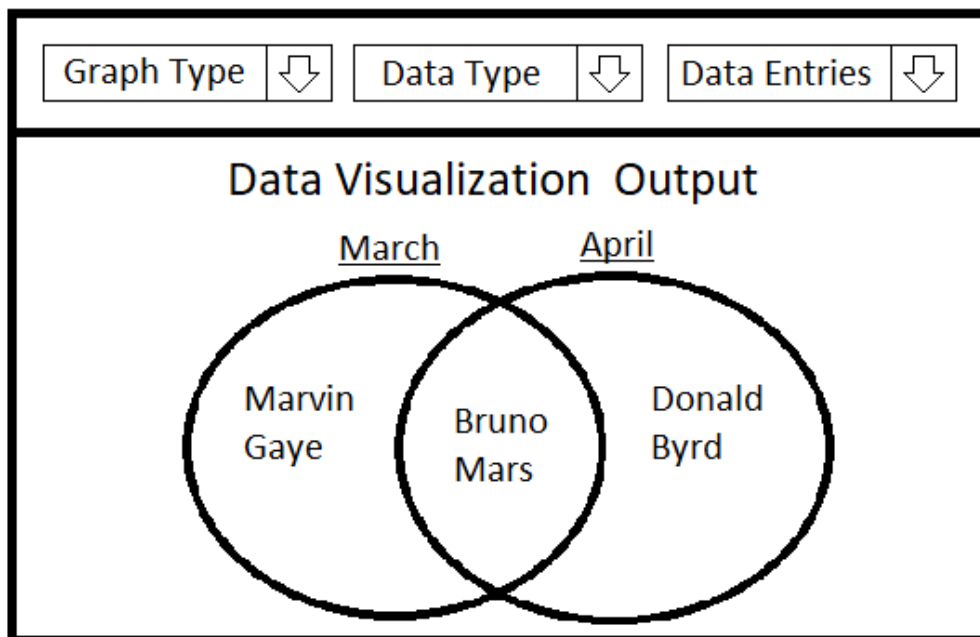


Figure 5: Rudimentary Visualization Mock-up

<u>Settings</u>	<u>Subsection Settings</u>
Subsection	
Subsection	Settings option
Subsection	Settings option
Subsection	Settings option
Subsection	Settings option
Subsection	Settings option
Subsection	Settings option
Subsection	
Subsection	

Figure 6: Rudimentary Settings Mock-up

3 CUSTOMER REQUIREMENTS

This section will provide an overview of the product's functional and design requirements. The purpose of listing these requirements is to clearly communicate the "look and feel" of the product as well as what end-users can expect the product to do. These requirements have been formed with the input of the product sponsor and cannot be changed without additional sponsor input.

3.1 LOGIN/REGISTRATION PAGE

3.1.1 DESCRIPTION

If not logged-in, when users open the web-page, they will be met with a login/registration screen. This will simply prompt the user to either create a new account or to login with an existing one. An account is essential since our servers will need to store the listening history of each individual user.

3.1.2 SOURCE

The entire team specified this requirement.

3.1.3 CONSTRAINTS

The only constraint to this requirement is the potential of having too many users, thus increasing the cost of maintaining a server to store user information.

3.1.4 STANDARDS

There are no specific standards that this requirement must abide by.

3.1.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. Without a way for user's to create and log into an individual account, there is no feasible way to provide statistical analysis of a user's listening history.

3.2 FEATURE MENU

3.2.1 DESCRIPTION

When users wish to switch between application features, they will utilize a slide-out menu. This menu will simply list the main features of the application, allowing users to click on the menu option the users wish to be taken to.

3.2.2 SOURCE

The entire team specified this requirement.

3.2.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.2.4 STANDARDS

This requirement has no applicable standards.

3.2.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. It is necessary for users to be able to access all of the product's features.

3.3 CALENDAR INTERFACE

3.3.1 DESCRIPTION

Upon login, the user will be met with a calendar view interface. This view will show users information about when songs were added to playlists, which dates have journal entries, and more. The calendar will show the current month by default.

3.3.2 SOURCE

The entire team specified this requirement.

3.3.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.3.4 STANDARDS

There are no applicable standards for this requirement.

3.3.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. This is a core feature of the product and is one of the main draws of using this product instead of competitor applications.

3.4 JOURNAL INTERFACE

3.4.1 DESCRIPTION

The journal interface will present users with a list of entries already added to the journal as well as a button that lets users create a new entry. When viewing an individual entry, the user may make edits or return to the entry list.

3.4.2 SOURCE

The entire team specified this requirement.

3.4.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.4.4 STANDARDS

There are no applicable standards for this requirement.

3.4.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. This is a core feature of the product and is one of the main draws of using this product instead of competitor applications.

3.5 DATA VISUALIZATION INTERFACE

3.5.1 DESCRIPTION

The data visualization interface will present users with a set of drop-down menus used to determine graph-type, data metric, as well as individual data entries. Once all of the parameters have been selected, a visual representation of the selected data will be presented to users.

3.5.2 SOURCE

The entire team specified this requirement.

3.5.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.5.4 STANDARDS

There are no applicable standards for this requirement.

3.5.5 PRIORITY

The priority of this requirement relative to other specified requirements is high. This is a core feature of the product and is one of the main draws of using this product. However, since this feature is similar to what is offered by competitors, there are other features with a higher priority than this.

3.6 MUSIC PLAYER INTERFACE

3.6.1 DESCRIPTION

This feature will simply play the song specified by the user. If a song is currently played, then a banner at the bottom of the screen will appear, much like in music streaming apps. This banner will display the song being played, the ability to pause and play the song, etc.

3.6.2 SOURCE

Spencer Whitehead and Ahmed Ullah specified this requirement.

3.6.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.6.4 STANDARDS

There are no applicable standards for this requirement.

3.6.5 PRIORITY

The priority of this requirement relative to other specified requirements is low. Although it would be a nice feature to have, it is not the primary focus of the application. The feature will be included if there is enough time to effectively implement it.

3.7 SETTINGS INTERFACE

3.7.1 DESCRIPTION

The settings menu will show users a series of editable options in relation to the user's account, linked streaming services, the data stored on the product's servers, and more. The view will consist of a vertical list on the left of the screen accompanied by a main panel. Each settings category will be selectable on the vertical list, with the corresponding options being visible on the main panel of the screen.

3.7.2 SOURCE

Abraham Mookhoek specified this requirement.

3.7.3 CONSTRAINTS

There are no constraints this requirement must consider.

3.7.4 STANDARDS

There are no applicable standards for this requirement.

3.7.5 PRIORITY

The priority of this requirement relative to other specified requirements is moderate. This is not a core feature of the product and the application can still largely operate without a settings menu. Despite this, for the product to function exactly as intended, this requirement must be implemented.

4 PERFORMANCE REQUIREMENTS

When taking into consideration the performance of the system we need to ensure that actions will be completed promptly. As we will have multiple parts that are not in our control such as API response times, database response time, and other factors, we must ensure that additional processing time on our end is minimal. Another thing to note is that since we will be processing vast amounts of data we expect the initial startup to take some time. Consequently, we must utilize the efficient data structures that will benefit our use case.

4.1 CONSIDER THE DATA NEEDED

4.1.1 DESCRIPTION

When the user begins using the application, what data is necessary for the action the user selected? This is a question that we as developers need to take into consideration. If we plan to minimize the memory footprint and efficiently use the processing power on the user's side, then we must only use data that is critical to the action at hand. We can cut back the data that is not only queried, thus reducing any delay, but also the data that is in memory on the user's side.

4.1.2 SOURCE

This constraint will become apparent as more users begin accessing the application.

4.1.3 CONSTRAINTS

There would be no constraints on how we access and request data. Although, there would be processing involved to structure the data efficiently.

4.1.4 STANDARDS

The use of classes and objects will allow us to represent the data concisely.

4.1.5 PRIORITY

The priority of this requirement may not seem that big at first, but as the application grows this will become apparent. If we were to ignore this requirement a lot of memory and processing power would be wasted on a feature that, in reality, did not require much data in the first place.

4.2 ENDPOINT RELIABILITY

4.2.1 DESCRIPTION

The processing server will contain multiple endpoints to query data depending on the user's action. The client side of the application is highly dependent on the data that it receives through the use of internal endpoints, as well as external applications.

4.2.2 SOURCE

This requirement comes from the critical component endpoints that will serve for fetching data.

4.2.3 CONSTRAINTS

Any constraints will occur during the implementation phase of the endpoint.

4.2.4 STANDARDS

Microsoft has published common practices and standards that a robust endpoint application should consider.

4.2.5 PRIORITY

This will be a critical priority as the client, server, and outside sources will be able to communicate with each other.

5 SECURITY REQUIREMENTS

Track Records, being a web application, must take cyber-security standards into account. Although there are no mandatory standards that our product must adhere to, it is good practice to build a secure product. Choosing to ignore security would be detrimental were there to be a breach and our product's end users had sensitive information stolen. As such, it is vital to practice good information security and ensure user privacy.

5.1 OWASP TOP TEN

5.1.1 DESCRIPTION

The OWASP Top Ten is a document meant for developers to use when making web applications. This document covers the top ten critical security risks facing web applications in recent times. This ranges from injection attacks to improper data handling. Although it is not necessary that our application takes this document into account, it is good practice and will ensure that our end-users are protected against cyber attacks.

5.1.2 SOURCE

This requirement was proposed by Abraham Mookhoek.

5.1.3 CONSTRAINTS

Following the OWASP Top Ten document will force development to follow secure programming practices and building a secure design for the product's operation.

5.1.4 STANDARDS

Open Worldwide Application Security Project (OWASP)

5.1.5 PRIORITY

The priority of this requirement relative to other specified requirements is high. If this product is designed in an insecure fashion, then the product will have to be taken down and fixed post-attack. This is not a concern for the duration of development as the product will not be available to the public. However, if this product is to be available for use on the internet after development concludes, it is vital to be prepared for malicious activity.

6 MAINTENANCE & SUPPORT REQUIREMENTS

It is crucial to maintain Track Records' functionality, security, and integrity after launch. Continually improving the system based on user feedback and ever-evolving technical breakthroughs is essential for ensuring flawless user experiences. The following explains the specifications for system upkeep and support, which will ensure the product's durability and preserve its standing in the market.

6.1 ERROR CORRECTION

6.1.1 DESCRIPTION

To find, address, and fix any system flaws that may emerge during use, proactive methods must be put in place. This covers unexpected software abnormalities such system crashes, bad data processing, and others.

6.1.2 SOURCE

Ahmed Ullah specified this requirement.

6.1.3 CONSTRAINTS

Must be resolved within a set timeframe depending on the severity of the error.

6.1.4 STANDARDS

ISO/IEC 25010:2011 - System and software Quality Requirements and Evaluation (SQuaRE)

6.1.5 PRIORITY

Addressing and rectifying errors is of critical importance to ensure uninterrupted and safe usage by the end users.

6.2 DOCUMENTATION AVAILABILITY

6.2.1 DESCRIPTION

Access to thorough documentation that describes the system architecture, codebase, and troubleshooting procedures should be available to maintainers.

6.2.2 SOURCE

The entire team specified this requirement

6.2.3 CONSTRAINTS

Documentation should be consistently updated with each software release.

6.2.4 STANDARDS

IEEE 829 - Software Test Documentation

6.2.5 PRIORITY

Providing updated and comprehensive documentation is a high priority to enable efficient maintenance.

6.3 MAINTENANCE TOOLS

6.3.1 DESCRIPTION

All specialized tools required for maintenance should be documented and available.

6.3.2 SOURCE

Ahmed Ullah specified this requirement

6.3.3 CONSTRAINTS

Potential licensing issues regarding third-party tools.

6.3.4 STANDARDS

ISO/IEC 14764:2006 - Software Engineering â Software Life Cycle Processes â Maintenance

6.3.5 PRIORITY

Ensuring that the maintenance team has access to all essential tools is a high priority to avoid potential disruptions.

6.4 SOFTWARE ENVIRONMENT UPKEEP

6.4.1 DESCRIPTION

The software environment hosting Track Records should receive periodic updates.

6.4.2 SOURCE

Ahmed Ullah specified this requirement

6.4.3 CONSTRAINTS

Potential downtimes during updates and compatibility issues with new software versions.

6.4.4 STANDARDS

NIST SP 800-40 - Guide to Enterprise Patch Management Technologies.

6.4.5 PRIORITY

Maintaining a secure and updated environment is critically important for both functionality and security.

6.5 CUSTOMER SUPPORT

6.5.1 DESCRIPTION

A support team should assist end users through various channels.

6.5.2 SOURCE

Ahmed Ullah specified this requirement

6.5.3 CONSTRAINTS

The challenges of providing 24/7 support and overcoming language barriers for global users.

6.5.4 STANDARDS

ISO 10002:2018 - Quality management - Customer satisfaction - Guidelines for complaints handling in organizations.

6.5.5 PRIORITY

Offering timely and effective customer support is of high importance to sustain user trust and satisfaction.

7 OTHER REQUIREMENTS

There are a few extra aspects of the product to consider outside of the previously listed requirements. This includes the method of keeping data, the setup/configuration process on the customer side, the portability of the product for different devices, as well as extensibility.

7.1 DATABASE

7.1.1 DESCRIPTION

This product will use MongoDB as its database of choice. This will be where user data, such as listening history and account info, will be stored.

7.1.2 SOURCE

The entire team specified this requirement.

7.1.3 CONSTRAINTS

As the number of users increase, so will the data that needs to be stored, especially with the plan to track each user's listening history from the point they connect their account onward. Issues may arise if the data needed to be stored exceeds the amount which can be accommodated.

7.1.4 STANDARDS

There are no specific standards that this requirement must abide by.

7.1.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. It is necessary for users to be able to access all of the product's features.

7.2 CUSTOMER SETUP/CONFIG

7.2.1 DESCRIPTION

Since the product is a website, little setup is needed on the customer side. However, to be able to access the majority of the site's features, the user needs to have a premium subscription to Apple Music or Spotify and log in with their account.

7.2.2 SOURCE

The entire team specified this requirement.

7.2.3 CONSTRAINTS

The only constraint pertaining to this requirement would be if we can not get access to the Apple developer token and have to drop Apple as an option for users to connect with.

7.2.4 STANDARDS

There are no specific standards that this requirement must abide by.

7.2.5 PRIORITY

The priority of this requirement relative to other specified requirements is critical. It is necessary for users to be able to access all of the product's features.

7.3 EXTENSIBILITY

7.3.1 DESCRIPTION

The product will be designed with continued development in mind. This could include adding extra statistics, more information available to user's for their history, new views, etc. Because of this, the product should be able to be worked on even after a finished product can be delivered.

7.3.2 SOURCE

The entire team specified this requirement.

7.3.3 CONSTRAINTS

It is possible for some issue to arise in future development that prevents some features from being added on as planned, and lowering the possible extensibility of the product.

7.3.4 STANDARDS

There are no specific standards that this requirement must abide by.

7.3.5 PRIORITY

The priority of this requirement relative to other specified requirements is moderate. This is not a core feature of the product and the application can still largely operate without extensibility in mind. Despite this, for the product to function exactly as intended, this requirement must be implemented.

7.4 PORTABILITY

7.4.1 DESCRIPTION

Since the product is a website, compatibility with different devices does not need to be considered as seriously as it would with a software that needs to be installed. However, there are still types of views to be considered, specifically pertaining to mobile devices. To be able to accommodate these, the site must be able to recognize mobile devices and provide a mobile view.

7.4.2 SOURCE

The entire team specified this requirement.

7.4.3 CONSTRAINTS

There may be some issues that arise if accessing the site from a less popular device that was not thought of, the view for which may not work as well as it would on other devices.

7.4.4 STANDARDS

There are no specific standards that this requirement must abide by.

7.4.5 PRIORITY

The priority of this requirement relative to other specified requirements is moderate. This is not a core feature of the product and the application can still largely operate without a mobile view. Despite this, for the product to function exactly as intended, this requirement must be implemented.

8 FUTURE ITEMS

8.1 MUSIC PLAYER INTERFACE

8.1.1 DESCRIPTION

This feature will simply play the song specified by the user. If a song is currently played, then a banner at the bottom of the screen will appear, much like in music streaming apps. This banner will display the song being played, the ability to pause and play the song, etc.

8.1.2 SOURCE

Spencer Whitehead and Ahmed Ullah specified this requirement.

8.1.3 CONSTRAINTS

There are no constraints this requirement must consider.

8.1.4 STANDARDS

There are no standards this requirement must consider.

8.1.5 PRIORITY

The priority of this requirement relative to other specified requirements is low. Although it would be a nice feature to have, it is not the primary focus of the application. The feature will be included if there is enough time to effectively implement it.

8.2 JOURNAL ENTRIES TONE ANALYSIS

8.2.1 DESCRIPTION

This feature will use a text model to classify the tone of the journal entry. With this, we can make a pseudo mood chart based on the user's "opinion" of the track they're listening to on that day. This can be extended further to provide potential music recommendations based on their perceived mood.

8.2.2 SOURCE

Professor Gieser specified this requirement in our team meeting.

8.2.3 CONSTRAINTS

There are no constraints this requirement must consider.

8.2.4 STANDARDS

There are no standards this requirement must consider.

8.2.5 PRIORITY

The priority of this requirement relative to other specified requirements is low. Although it would be a nice feature to have, it is not the primary focus of the application. The feature will be included if there is enough time to effectively implement it.

REFERENCES