record-log
Software Requirements Specification

1. SRS Revision History

Date	Author	Description
4-2-2024	JHT	Updated SRS for final submission.
5-10-2024	JHT	Created the initial document using the template.
5-13-2024	JHT	Complete initial version of SRS document.
5-19-2024	JHT	Updated SRS to align with project rescope.
6-01-2024	EH	Modifying sections for final submission.
6-02-2024	EH	Specific Requirements Adjustment

2. The Concept of Operations (ConOps)

This document is the Software Requirements Specification (SRS) for the Record-Log (RL) Software. Record-log is a browser based application designed to help music enthusiasts organize and keep track of their thoughts and opinions of their favorite tunes. The software will help music enthusiasts and Deejays curate cohesive playlists or DJ sets through the organization and visualization of their track entries. The user can organize the entries into lists, helping them mockup potential playlists and sets. All the users track entries will be displayed in the homepage upon launching RL, where the user can add, edit, or delete tracks.

Record-log provides a space for the user to keep track of what songs they have listened to, the title of the track, important notes or features of the track, and how they felt about it. It will also allow tracks with similar features or themes to be linked together so that the user can reflect on their entries and construct a cohesive playlist or set based on the links. RL does this through the use of relationships between track entries that the user defines. Record-log also allows for the user to create lists, and add their entries to lists in order to further categorize their entries. This could be useful to keep an account of different tracks to be used in a DJ set based on their thematic links, or simply categorizing them to a particular genre.

The functionality for the track entry component consists of an area for track title text entry, review text entry, thoughts text entry (for key musical features), and an area to rate the track out of five stars. Users can link together tracks with similar features or thematic connections through the related track functionality, creating a detailed and personalized musical archive. These connections are visualized in the graph view component of the software, where each node is an entry and each edge is a connection to a related entry. Each user's notes will be stored on a MongoDB server, allowing for easy retrieval of past entries, so that they can be viewed, edited, or deleted. They can access the Record-Log Database from any location provided they have installed the software on the system they are attempting to run it on.

2.1. Current System or Situation

There are a multitude of note taking apps, designed to allow the user to record their thoughts on general topics, but from a quick internet search there is hardly any software that is designed specifically for taking detailed notes on music. However, some of the software that we looked towards for inspiration was Notion, Obsidian, and Discogs. By combining the modularity of Notiona and Obsidian with the music oriented intention with Discogs, we were able to create something that was unique and curated towards a niche audience.

2.2. Operational Features of the Proposed System

The key operational features of record-log (RL) include: (a) facilitating tracking of songs through what we call “track entries”, (b) enabling users to record thoughts and musical features of songs in the track entries, (c) allowing personalization of grouping of track entries in the form of lists, (d) enabling linking of track entries with related features and themes, (e) storing track entries on a server for retrieval, (f) providing Node Graph visualization of links between track entries.

2.3. User Classes

There is one User class:

1. A “User” who is attempting to use RL to keep an account of songs they have listened to too, and record notes for them. The User could be anyone, from a music enthusiast, to professional Deejay.

2.4. Modes of Operation

The system has one primary mode of operation, where the user launches the system, which starts the server that connects to the database and launches the web browser Record-Log user interface. Upon initial startup, the user will be taken to a home screen, prompting them to create track entries, and allowing them to view the details of their previously created entries. A Navigation bar will also be displayed at the top allowing the user to access the accompanying Record-Log menus.

3. Specific Requirements

The basic functionality for RL is to provide a system to the user for recording their thoughts on songs they've listened to, along with the important musical features of that song. In addition the software should allow for retrieval of and visualization of entries so that the user can reflect and organize their thoughts. To facilitate this, the software must include the following functionality:

3.1 *Externally-Visible User Interactions*

1. A home page displaying the track entries the user has created.
2. The user can expand track entries to view details, and update or delete entries.
3. A Navigation bar at the top of the home page allows the user to access the home page, view list page, and graph view page.
4. A create new track entry component in the home page with the following functionality:
    a. Track Title (text box).
    b. Track Review (text box).
    c. Track Thoughts (text box).
    d. Track Ratting (5 star system).
    e. Set Related Entry (pop up menu).
5. A list menu, allowing the user to view and interact with their created list.
    a. A create new list component, allowing the user to enter the title for the new list they want to create.
    b. A view current list component, displays the current lists that the user has created.
    c. Clicking on a current list will display a popup of that list detailing what track entries are contained in it.
    d. The list popup will allow the user to view entry details, add or delete entries from the list, or delete the whole list itself.
    e. The add entry button will display a popup of all the entries in the database, accompanied by a text box which lets the user refine their track entry search.
6. Graph view that provides a visualization of thematic links between entries.
    a. Nodes representing track entries in the graph view can be clicked on to display the entry details.

3.2 *Target Platform*

- The software should run on a laptop or desktop machine. The program should be designed for use with a real keyboard, not a smartphone. The system should run on macOS 12.

3.3 *Data Storage*

- Data will be stored on a MongoDB server on an Atlas cluster (official MongoDB cluster).
- The system will save the entry data to the Atlas cluster, allowing for retrieval from any system.

3.4 Build-Related Constraints

3.4.1 System Document File Formats

All system-related and system-development-related documents that are intended for human reading must be in either plain text or PDF. For example, Microsoft Word, Microsoft Excel, or markdown language documents must be converted into plain text or PDF.

3.4.2 Programming Constraints

The following programming languages and technologies will be used:

- Python 3 along with the Python Standard Library https://docs.python.org/3/library/index.html, and pymongo library.
- Python code will run in 3.12
- Node.js for runtime environment.
- React.js for interactive UI components.
- Express.js for API request/response handling.
- MongoDB for storing user data.

3.4.5 Installation Constraints

- Instructions will be provided for how to compile/run the code and install the system. There will be at most 20 commands required to set up the server, and run the program.
- An experienced computer programmer will not require more than 20 minutes working alone with the submitted materials to set up the server, and run the code.

4. References

Hornof, A. (n.d.). *CIS 422/522 Document Templates*. CIS 422/522. https://classes.cs.uoregon.edu/24S/cs422/Templates.html. Accessed 5-10-2024.

5. Acknowledgements

This SRS builds on the template from https://classes.cs.uoregon.edu/24S/cs422/Templates.html