Software Requirements Specification for Software Engineering: subtitle describing software

Team 8 – Rhythm Rangers

Ansel Chen
Muhammad Jawad
Mohamad-Hassan Bahsoun
Matthew Baleanu
Ahmed Al-Hayali

October 11, 2024

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project

1.1 User Business

Insert your content here.

1.2 Goals of the Project

Insert your content here.

2 Stakeholders

2.1 Client

Insert your content here.

2.2 Customer

Insert your content here.

2.3 Other Stakeholders

Insert your content here.

2.4 Hands-On Users of the Project

Insert your content here.

2.5 Personas

Insert your content here.

2.6 Priorities Assigned to Users

Insert your content here.

2.7 User Participation

Insert your content here.

2.8 Maintenance Users and Service Technicians

Insert your content here.

3 Mandated Constraints

3.1 Solution Constraints

Insert your content here.

3.2 Implementation Environment of the Current System

Insert your content here.

3.3 Partner or Collaborative Applications

Insert your content here.

3.4 Off-the-Shelf Software

There are several existing solutions that could serve as part of the music generation and recommendation system. These include:

- **Spotify API**: Provides access to a vast library of music, including song previews and metadata, which can be leveraged for generating recommendations.
- **Librosa Library**: An open-source Python package for analyzing and processing music files, suitable for extracting features from songs and facilitating generative components.
- **TensorFlow and PyTorch Pre-trained Models**: Both frameworks offer pre-trained models that could be adapted for music generation

tasks. These solutions provide a basis for deep learning models without having to build and train from scratch.

- **OpenAI Jukebox**: A generative model that is capable of producing music, which could potentially be adapted and integrated into our system.

These off-the-shelf software solutions provide a foundation upon which we can build our custom features, significantly reducing the development time and leveraging existing technologies to enhance the functionality of our platform.

Insert your content here.

3.5 Anticipated Workplace Environment

Insert your content here.

3.6 Schedule Constraints

Insert your content here.

3.7 Budget Constraints

Insert your content here.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

Insert your content here.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- Music contains the following core features
  - Tempo
  - Key signature
  - Time signature
  - Pitch
  - Timbre
- Song files have metadata that contains information such as:
  - Song title
  - Artist
  - Release date
- Most songs can be classified into a particular genre

5.2 Business Rules

- The user should be able to generate their own music
- The user should be able to figure out what musical features a song contains
- The user should be able to ask for similar songs
- the user should be able to interact with the system without any external installation

5.3 Assumptions

- Users will have at least some familiarity of music theory
- The analysis and recommendation systems will use as many well-established musical features as possible

- All API inputs will be easily accessible and reliable enough to support the recommendation and analysis systems
- The system will be written in a language that all developers are familiar with
- The system will use a local server to handle the processing of the machine learning model and large datasets
- Handling of niche features and cover art are designed to enhance the user experience, but these will not be a part of the core functionality of the system
- The generative system will be completed by the POC demo date
- The recommendation and analysis systems will be completed by the Revision 0 date

6 The Scope of the Work

6.1 The Current Situation

Insert your content here.

6.2 The Context of the Work

Insert your content here.

6.3 Work Partitioning

Insert your content here.

6.4 Specifying a Business Use Case (BUC)

Insert your content here.

7 Business Data Model and Data Dictionary

7.1 Business Data Model

Insert your content here.

7.2 Data Dictionary

Insert your content here.

8 The Scope of the Product

8.1 Product Boundary

Insert your content here.

8.2 Product Use Case Table

Insert your content here.

8.3 Individual Product Use Cases (PUC's)

Insert your content here.

9 Functional Requirements

9.1 Functional Requirements

Insert your content here.

10 Look and Feel Requirements

10.1 Appearance Requirements

Insert your content here.

10.2 Style Requirements

Insert your content here.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

Insert your content here.

11.2 Personalization and Internationalization Requirements

Insert your content here.

11.3 Learning Requirements

Insert your content here.

11.4 Understandability and Politeness Requirements

Insert your content here.

11.5 Accessibility Requirements

Insert your content here.

12 Performance Requirements

12.1 Speed and Latency Requirements

Insert your content here.

12.2 Safety-Critical Requirements

Insert your content here.

12.3 Precision or Accuracy Requirements

Insert your content here.

12.4 Robustness or Fault-Tolerance Requirements

Insert your content here.

12.5 Capacity Requirements

Insert your content here.

12.6 Scalability or Extensibility Requirements

Insert your content here.

12.7 Longevity Requirements

Insert your content here.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

Insert your content here.

13.2 Wider Environment Requirements

Insert your content here.

13.3 Requirements for Interfacing with Adjacent Systems

Insert your content here.

13.4 Productization Requirements

Insert your content here.

13.5 Release Requirements

Insert your content here.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

Insert your content here.

14.2 Supportability Requirements

Insert your content here.

14.3 Adaptability Requirements

Insert your content here.

15 Security Requirements

15.1 Access Requirements

Insert your content here.

15.2 Integrity Requirements

Insert your content here.

15.3 Privacy Requirements

Insert your content here.

15.4 Audit Requirements

Insert your content here.

15.5 Immunity Requirements

Insert your content here.

16 Cultural Requirements

16.1 Cultural Requirements

Insert your content here.

17 Compliance Requirements

17.1 Legal Requirements

Insert your content here.

17.2 Standards Compliance Requirements

Insert your content here.

18 Open Issues

Insert your content here.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

Insert your content here.

19.2 Reusable Components

Insert your content here.

19.3 Products That Can Be Copied

Insert your content here.

20 New Problems

20.1 Effects on the Current Environment

Insert your content here.

20.2 Effects on the Installed Systems

Insert your content here.

20.3 Potential User Problems

Insert your content here.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Insert your content here.

20.5 Follow-Up Problems

Insert your content here.

21 Tasks

21.1 Project Planning

Insert your content here.

21.2 Planning of the Development Phases

Insert your content here.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

There are no migration requirements as this project is not a replacement or upgrade of a previous project

22.2 Data That Has to be Modified or Translated for the New System

Similarly, there currently is no data that needs to be modified

23 Costs

The monetary cost estimate of the project is $0 CAD. All of the necessary equipment is owned by at least one group member.

The total time cost estimate of the project is 8 months (September 2024 - April 2025).

The function point cost estimate is 12. This is derived from the business rules, which list out all the high level function points of the project.

24 User Documentation and Training

24.1 User Documentation Requirements

The music featurization feature is heavily API-driven, and as such, detailed documentation will primarily be covered within the API reference section. This approach ensures that developers and advanced users can understand the feature's capabilities without needing additional user guides.

To ensure that users can effectively interact with the music generation and recommendation platform, the following user documentation will be provided:

- **Quick Start Guide**: A concise guide aimed at helping new users get started with the basics of generating and recommending music.

- **API Reference and Technical Specifications**: Detailed documentation of the platform's API, including available endpoints, request/response formats, and example queries. This reference is crucial for developers and advanced users who want to integrate the platform with other applications or automate tasks.
- **Installation Guide**: A step-by-step guide for installing the platform on local servers, including system requirements, installation commands, and troubleshooting common setup issues.
- **FAQs and Troubleshooting**: A list of frequently asked questions and troubleshooting tips to help users solve common issues independently.
- **Video Tutorials**: Step-by-step video guides that visually demonstrate key features and workflows, including setting up the platform, using the API, and generating music.

These documents will be designed for users of varying technical backgrounds to ensure they can fully utilize the platform's capabilities. The documentation will be created and maintained primarily by the development team, ensuring accuracy and alignment with the latest platform features. However, feedback from user groups will be actively sought to improve clarity and address any documentation gaps. Updates to the API reference and technical specifications will be managed as part of the regular software update cycle.

24.2 Training Requirements

To provide users with sufficient knowledge to operate the platform effectively, the following training resources will be developed:

- **Video Tutorials**: Developed by the development team, these tutorials will cover various aspects of the platform, including API usage, generating music, and using advanced features.
- **Online Training Modules**: If additional resources become available, online training modules could be developed to provide users with a structured learning path. However, due to current resource constraints, we do not plan to offer live training sessions.

These training requirements aim to encourage users to explore the full potential of the platform, regardless of their prior experience in music production or technology.

25 Waiting Room

1. The recommendation system will be able to recommend songs from less popular music genres (jazz, blues, etc.)
2. The analysis system will be able to extract musical features from less popular music genres (jazz, blues, etc.)
3. The generative system will be able to generate songs from less popular music genres (jazz, blues, etc.)
4. The recommendation system will be able to recommend songs from unpopular music genres (gnawa, libyan funk, etc.)
5. The analysis system will be able to extract musical features from less popular music genres (gnawa, libyan funk, etc.)
6. The generative system will be able to generate songs from less popular music genres (gnawa, libyan funk, etc.)
7. The recommendation system will be able to search for songs with cover art similar to an input song
8. The generative system will be able to generate new cover art for a newly generated song, based on the user's input criteria
9. The generative system will be able to generate new covert art for an existing song

26 Ideas for Solution

- **Hybrid Recommendation System**: A hybrid recommendation system combines content-based filtering and collaborative filtering techniques to provide a more personalized experience for users. Content-based filtering analyzes song features, such as genre, key, and rhythm,

to suggest similar tracks. Collaborative filtering uses user preferences and historical listening patterns to suggest music. By combining these approaches, the system can offer users personalized suggestions while also helping them discover new genres and music styles.

- **Generative Music Model**: To enable the creation of new music, a generative model will be used. This model could be based on techniques such as a Generative Adversarial Network (GAN) or Recurrent Neural Network (RNN). A GAN would allow for the generation of realistic music by having the generator and discriminator work together to produce convincing compositions. An RNN, on the other hand, would be well-suited for learning the sequential nature of music, generating new melodies based on learned patterns. This solution provides users with an innovative way to create new music based on their inputs and preferences.

- **Feature Manipulation Interface**: This interface will allow users to interact directly with song features, such as tempo, key, and rhythm, enabling them to create customized versions of existing tracks or generate entirely new compositions. By adjusting different musical parameters, users can personalize their musical experience and experiment with creative variations, providing a high level of control over the output.

- **Integration with Existing Platforms**: Integrating the system with existing music platforms, such as Spotify, will allow users to easily access and analyze a large library of songs. Users will be able to input their favorite tracks from these platforms and generate variations or receive recommendations. This integration ensures a smooth user experience, allowing seamless interaction between existing music libraries and the platform's generative capabilities.

Appendix — Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Lifelong Learning. Please answer the following questions:

1. What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.

   As a team, the Rhythm Rangers, we need to acquire a diverse range of knowledge and skills from various domains, including software development, music generation, and collaborative teamwork, to successfully complete our capstone project. Given the scope of this task, it is essential for each team member to focus on specific areas of expertise that align with their skills, passions, roles, and responsibilities, as well as learn new skills and gain new knowledge. Outlined below is the knowledge and skills the team will collectively need to acquire to successfully complete this capstone project:

   Music Analysis and Signal Processing: This capstone project involves developing expertise in audio signal processing to analyze sound data and extract valuable insights for music recommendation and generation systems. The team will learn to implement machine learning models for tasks such as genre classification and feature extraction. Proficiency in Python libraries for audio analysis and model training is essential. This will deepen the teams understanding of music theory and the connections between song features and genres.

   Frontend or Backend Development: The team will need to understand backend frameworks for building and managing the recommendation system's infrastructure. The will be integrating external APIs to access song previews and features. They will also gain knowledge in database management for storing and organizing song data and

user preferences. Furthermore, this involves learning how to scale and efficiently handle data for a local server-based system.

UI/UX and Design: The team will need to design user-friendly interfaces that ensure smooth interaction with the music recommendation and generation systems. UI/UX design skills will need refinement and utilizization of frontend development frameworks will be needed to craft the systems user interface. They will also learn to connect frontend components with backend APIs for real-time updates, such as delivering song recommendations.

Music Generation: An understanding of generative models to create music snippets from input tracks or references will be a huge component for this project. The team will delve into music feature engineering, transforming audio data into usable features for machine learning applications. Familiarity with music data will assist in generating new content and making recommendations.

Team Management and Infrastructure: For this project to be a success improving project management and team coordination skills will foster effective communication, sprint planning, and task assignment. We will need to learn how to establish and maintain local server infrastructure for efficient hosting and operation of the platform. Understanding security best practices to safeguard user data and ensure the system's resilience against vulnerabilities is critical. Mastery of version control and Git management will promote seamless collaboration and code review among the team.

2. For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?