Software Configuration Management Plan (SCMP)
Group 7

MuseAI

Lyrics and Chord Generation System

Software Engineering Project

METCS 673

BOSTON UNIVERSITY

Tridev Rapeti
Shreni Singh
Lingjie Yuan
Aayush Raghuvanshi

September 2024

1 Purpose

This documentation aims to manage all the documents and code that will be created in the Lyrics and Chord Generation System - MuseAI project.

2 Management

2.1 Organization and responsibility

- Project Managers :
  - Responsible for overseeing the management of all the configuration items of this project.
  - The only person who is permitted to create and delete any file in this project.

2.2 Code management

- Codes should be managed by Git and stored in GitHub consistently.
- Codes should follow code management techniques including version control and branching strategies.
- Codes should be properly indented, information regarding every part of the code should be kept in the comments.
- Developers shouldn't try to access other branches of the repository without proper allowance.

2.3 Documentation management

- Documents should be in Latex format and edited on Overleaf.
- Documents should be pushed into the git repository in a separate folder for backup purposes.
- All documents must be written by Technical Writers and should be processed through all technical writers' approval before finalizing.

2.4 Environment and software management

- The version of the software should be consistent among various computers.
- All the dependencies and libraries installed must be mentioned in a requirements.txt file
- The environment should be managed by Conda to ensure consistency.

3 Configuration Identification

3.1 Naming conventions

- Document names should be [name]-[month-date]-[version].[file type extension].
- Code file names should remain consistent across different computers to ensure version control and collaboration when using GitHub.
- Variables' and Functions' names should be relevant to their purpose

4 Configuration Control

4.1 Tools and Platforms

- **Overleaf:** All official documents are written in LATEX on Overleaf.
- **Jupiter:** All software engineers of the LLM sub-team should use Jupiter Notebook to ensure the consistency of the environment. (subject to change)
- **Github:** All the software engineers should have access to Github to update work.
- **Pythonanywhere:** To host the web app online in case of need (subject to change)
- **Whatsapp:** Whatsapp shall be used for communication between the teammates
- **Blackboard:** Blackboard tasks in my groups could be used to update individual tasks.
- **Google Meet**: For conducting scrum meetings online.

4.2 Formal Languages

- **Python:** To be used in developing the LLM and Backend of the Web App
- **Latex:** To be used for drafting official documents related to the project
- **HTML:** To be used for basic frontend architecture.
- **Javascript:** To be used for animations and effects in frontend.
- **SQL:** To be used for database calls.
- **Jinja:** To be used for making the pages dynamic by connecting front-end to the back-end

5 Status accounting

- All documents have a document change record in which all changes concerning the previous versions are recorded.
- All codes are recorded in GitHub.

6 Security Measures

- Only the Team Managers are allowed to delete documents from any library respectively related to their team after discussing it in a scrum meeting.
- Create a new branch in the git repository for each task allocation and merge the branches after discussing in the scrum meeting with the scrum masters to maintain consistency.
- Usage of illicit or copywritten code parts and material is not allowed.

7 Probable problems and precautions

- **Algorithm Feasibility Issue:** The AI model might produce irrelevant lyrics and chords or the system might experience delays in generating lyrics and chords, leading to poor user experience.
  - Train different models with diverse datasets to improve accuracy and efficiency and incorporate user feedback for fine-tuning.
  - Consider using dimension reduction or approximate methods to manage computational complexity.
- **Effectiveness Issue:** There may be a risk of not meeting goals within the set timeframe due to scope expansion, unforeseen challenges, or force majeure.
  - Focus on delivering a minimum viable product first, ensuring core functionality is completed on time, and then gradually adding additional features.
  - Monitor project progress regularly to identify potential delays and take corrective actions.
  - Adjust the project plan and resource allocation based on actual progress to ensure timely completion.
- **Data Security Issue:** Data may be tampered with or corrupted, affecting the reliability of generated content.
  - Schedule regular backups of all critical data to ensure that up-to-date copies are available in case of data loss or corruption.
  - Implement robust key management practices to protect encryption keys and ensure they are securely stored and used.
- Copyright issues: The music generated, the code used, or the media used could be challenged for a copyright claim
  - Try to avoid using copyrighted content in the first place and if wanted to use it, give the author or the owner their credits.
  - AI-generated music can be protected from copyright laws by limiting human intervention to a limit as AI-generated content with no sufficient human intervention doesn't come under the Copyright Act.