Problem Statement and Goals
ProgName

Team #, Team Name
Student 1 name
Student 2 name
Student 3 name
Student 4 name

Table 1: Revision History

Date	**Developer(s)**	**Change**
Date1	Name(s)	Description of changes
Date2	Name(s)	Description of changes
...	...	...

1 Problem Statement

[You should check your problem statement with the problem statement checklist. —SS]

[You can change the section headings, as long as you include the required information. —SS]

1.1 Problem

1.2 Inputs and Outputs

[Characterize the problem in terms of "high level" inputs and outputs. Use abstraction so that you can avoid details. —SS]

1.3 Stakeholders

1.4 Environment

[Hardware and software environment —SS]

2 Goals

Song Analysis

- **Explanation:** GenreGurus will analyze a song or snippet and extract important musical features. The data gathered from this analysis will then be fed into the recommendation and generation systems, ensuring they work with accurate data about each song's musical features.
- **Reasoning:** By providing a detailed analysis, users can better understand the components of a given song. The analysis also ensures that the recommendations and generated music are based on actual musical data, which in turn improves the system's accuracy and output quality.

Music Generation

- **Explanation:** GenreGurus will allow users to input one or more reference songs or snippets and generate new music based on the features of these inputs. Users can adjust certain musical characteristics, and the system will produce an original track reflecting these changes.
- **Reasoning:** Users will be able to create and customize music through AI without needing extensive musical knowledge, making the platform more accessible while still appealing to expert musicians.

User Customizable Recommendations

- **Explanation:** GenreGurus will allow users to adjust the musical features of the input clip/song. Based on these adjustments, the system will update its recommendations in real-time.
- **Reasoning:** Customizable recommendations give users more control over the output, increasing user engagement and user satisfaction.

User Centric Design and Interface

- **Explanation:** GenreGurus will include a clean, intuitive interface where users can easily access the music recommendation, generation, and analysis features. The UI will be designed to require minimal understanding of how music adjustments work for a better user experience.
- **Reasoning:** An accessible and simple interface will appeal to a broader audience, from casual listeners to professionals.

Machine Learning Cover Art Generation

- **Explanation:** GenreGurus will use AI models to generate custom art based on the features of a generated song or user preferences. The art will visually reflect the song's mood, genre, style, etc.

- **Reasoning:** Music encompasses more than just using one's auditory senses; it is also a visual and emotional experience. By generating art that matches the music, the system offers more connection for creators, appealing to both their auditory and visual senses.

Supportive of Many Music Genres

- **Explanation:** GenreGurus will include a variety of both popular and niche genres.
- **Reasoning:** This will allow users to not only customize and explore their favorite genres of music but will allow them to explore genres they've never heard or experienced before.

3 Stretch Goals

4 Challenge Level and Extras

[State your expected challenge level (advanced, general or basic). The challenge can come through the required domain knowledge, the implementation or something else. Usually the greater the novelty of a project the greater its challenge level. You should include your rationale for the selected level. Approval of the level will be part of the discussion with the instructor for approving the project. The challenge level, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

[Teams may wish to include extras as either potential bonus grades, or to make up for a less advanced challenge level. Potential extras include usability testing, code walkthroughs, user documentation, formal proof, GenderMag personas, Design Thinking, etc. Normally the maximum number of extras will be two. Approval of the extras will be part of the discussion with the instructor for approving the project. The extras, with the approval (or request) of the instructor, can be modified over the course of the term. —SS]

Appendix — Reflection

[Not required for CAS 741 —SS]

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

1. What went well while writing this deliverable?
2. What pain points did you experience during this deliverable, and how did you resolve them?
3. How did you and your team adjust the scope of your goals to ensure they are suitable for a Capstone project (not overly ambitious but also of appropriate complexity for a senior design project)?