6-5-2018

Musical Casual Creator for Easy Self-Expression

Louis Lin
Santa Clara University, llin@scu.edu

Ian Santillano
Santa Clara University, isantillano@scu.edu

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 5, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY

Louis Lin and Ian Santillano

ENTITLED

Musical Casual Creator for Easy Self-Expression

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THESIS ADVISOR

DEPARTMENTCHAIR

Musical Casual Creator for Easy Self-Expression

by

Louis Lin and Ian Santillano

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California

June 5, 2018

Musical Casual Creator for Easy Self-Expression

Louis Lin
Ian Santillano

Department of Computer Engineering
Santa Clara University
June 5, 2018

ABSTRACT

People with autism often struggle with the ability to communicate. As a result, many people with autism become frustrated in their attempts to express their thoughts and feelings fully. Current forms of music therapy are designed to improve the lives of patients with autism, but are inefficient at developing their communication skills. Our project is designed to help music therapy patients express themselves through the creation of music. A patient will be able to "write" a song through the power of his or her feelings. With our system, music therapy patients can actively drive the entire creative process and learn new ways to express their feelings. It is a tool that empowers people to communicate their thoughts and feelings through music.

Table of Contents

List of Figures

List of Tables

1 Introduction

People with autism often struggle with the ability to communicate. They may experience varying degrees of speech impairment, which ranges from being able to speak fluently to being unable to speak at all. Of those who can speak, they will often use language in a very limited or unusual way [1]. As a result, many people with autism become frustrated in their attempts to express their thoughts and feelings fully [2].

Currently there is no cure for autism, but there are various forms of therapy aimed to diminish the symptoms or improve the quality of life of those who are affected by it. One such therapy is improvisational music therapy, in which trained music therapists sing or play music with patients, attuned and adapted to the patient's focus of attention. Although the music helps patients feel relaxed, it does not actively engage them in a way that improves communication skills. Instead, they play a passive role by listening to the therapist singing or playing. Researchers measured the effects of this form of therapy and found that it did not make a significant difference in the symptoms of their patients [3].

Another form of music therapy gives patients the chance to slowly develop new skills by singing or playing an instrument [4]. Singing or playing an instrument can greatly assist in teaching patients to communicate because they can learn from the messages that a song is expressing [5]. However, private lessons are expensive because they require the presence of an instructor. On top of that, not all instructors in the field have the teaching skills attuned to train people with autism. While learning an instrument is highly beneficial for everyone, bringing this therapeutic form to patients with autism can be challenging.

Our solution offers a way for music therapy patients to express themselves through the creation of music without the need for a trained instructor. A patient will be able to "write" a song through the power of his or her feelings. Our system will improvise melodies, chords, and lyrics based on a series of inputs from the patient. The interface would let patients intuitively select inputs from a set of visuals, which includes emojis or other descriptive images that represent feelings. With our system, music therapy patients can actively drive the entire creative process. It is a tool that empowers patients to communicate their thoughts and feelings through music.

2 Requirements

The following are the requirements and design constraints of the project. Functional requirements specify the functions that the system must perform, while the non-functional requirements specify criteria that can be used to judge the operation of the system. Requirements are prioritized as critical, recommended, and suggested. Design constraints are technological limitations that the solution must follow. They are informed by our goals for user engagement with the system.

2.1 Functional Requirements

Critical

- Users select the feeling that they want to express
- Users choose lyrics using lines suggested by the system
- Users can listen to songs that they create
- Users can save songs in the system
- Users can view saved songs
- System creates a melody based on the feeling that the user selects

Recommended

- User interface reminds users of the feeling they have selected while creating a song
- Users have the option to create a song without having to choose lyrics

Suggested

- Users can export songs as mp3, MIDI file
- Users can export sheet music as PDF
- Users can sing along with karaoke

2.2 Non-Functional Requirements

Critical

- User interface is intuitive and easy to use
- Lyric suggestions are relevant and sensible
- Generated melodies are relevant and sensible
- Resulting songs are satisfying and easy to sing

Recommended

- System responds quickly with lyric suggestions and melody generations when prompted by user

Suggested

- User interface design is visually appealing to target audience
- System is enjoyable to use

2.3 Design Constraints

- System is a web application
- System must be compatible with Google Chrome and Mozilla Firefox

3 Use Cases

The user is our only actor. The user has nine use cases, which describe actions that the user may perform with the system. Figure 3.1 shows the use case diagram.

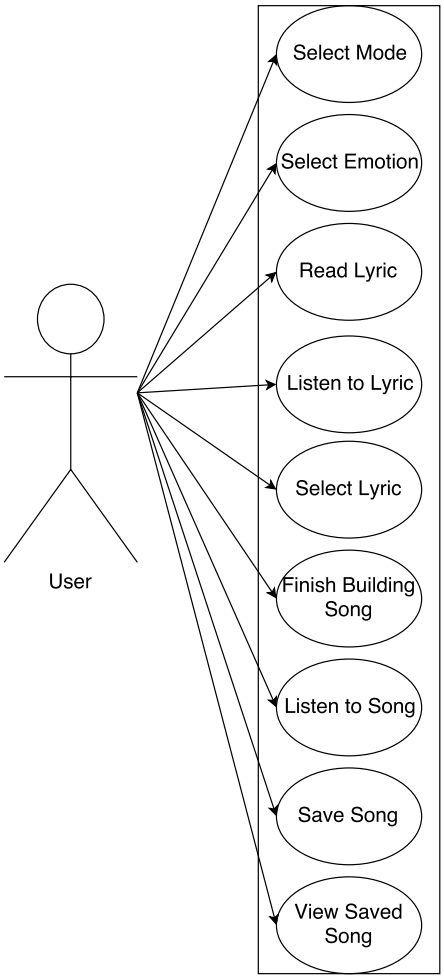

Figure 3.1: Use Case Diagram

3.1 Select Mode

- **Goal:** Select either “Create A Song” or “Quick Select” mode for the system
- **Actor:** User
- **Preconditions:** User has opened the web application
- **Steps:** User selects either “Create a Song” or “Quick Select”
- **Postconditions:** System is in either “Create A Song” or “Quick Select” mode
- **Exceptions:** N/A

3.2 Select feeling

- **Goal:** Select a feeling that represents what the user is experiencing or what the user wants to express
- **Actor:** User
- **Preconditions:** Mode is selected
- **Steps:** User selects a feeling on the chart presented
- **Postconditions:** System has a feeling to influence generation of lyrics and music
- **Exceptions:** The feeling that a user experiences or wants to express might not be implemented in the system. User may accidentally select the wrong feeling

3.3 Read Lyric

- **Goal:** Read lyrics that the system generates based on the feeling that the user selects
- **Actor:** User
- **Preconditions:** Feeling must be chosen
- **Steps:** User reads from the three lyrics presented
- **Postconditions:** User has an understanding of the presented lyrics and can make a decision to select a lyric
- **Exceptions:** The user may prefer to listen to the lyric instead

3.4 Listen to Lyric

- **Goal:** Listen to sample lyrics that system generates based on a feeling
- **Actor:** User
- **Preconditions:** Feeling must be selected
- **Steps:** The user selects the sound icon next to the lyric and hears the lyrics spoken aloud
- **Postconditions:** User has an understanding of the presented lyrics and can make a decision to select a lyric
- **Exceptions:** N/A

3.5 Select Lyric

- **Goal:** Select one lyric from three choices that will end up being in the final song
- **Actor:** User
- **Preconditions:** Feeling must be chosen
- **Steps:** User chooses the most satisfactory lyric out of the three presented lyrics, or selects “Regenerate Lyrics” to generate a new set of lyrics to choose from
- **Postconditions:** Lyric will be a part of the final created song
- **Exceptions:** User can select the wrong lyric, but can go back and correct it

3.6 Finish Building Song

- **Goal:** Finish building the song, thereby confirming the user-selected feeling and lyrics to add to the song
- **Actor:** User
- **Preconditions:** Feeling and lyrics must be selected
- **Steps:** User confirms these choices by selecting "Finish Song"
- **Postconditions:** A song (chords and melody) will be generated with the feeling and lyrics that the user selected
- **Exceptions:** N/A

3.7 Listen to Song

- **Goal:** Listen to full song (lyrics with melody and chords) that the system generated based on user-selected feeling and lyrics; includes playback and seek through
- **Actor:** User
- **Preconditions:** User must finish building song
- **Steps:** User selects the play button on the playback bar
- **Postconditions:** User listens to song and comprehends the feeling they were experiencing or trying to express
- **Exceptions:** N/A

3.8 Save Song

- **Goal:** Save user-created song in system; can also export as mp3 and/or MIDI and sheet music (with lyrics, melody and chords) as PDF
- **Actor:** User
- **Preconditions:** User must finish building song
- **Steps:** User selects "save song"
- **Postconditions:** Song is saved to system and can be exported as mp3 and/or MIDI and sheet music (lyrics, melody, and chords) as PDF
- **Exceptions:** N/A

3.9 View Saved Song

- **Goal:** User can revisit and playback a saved song in system
- **Actor:** User
- **Preconditions:** User must have saved the song
- **Steps:** User opens web application and selects "Saved Songs"
- **Postconditions:** User can playback the saved song and can be exported as mp3 and/or MIDI and sheet music (lyrics, melody, and chords) as PDF
- **Exceptions:** N/A

4 Activity Diagram

The diagram in Figure 4.1 outlines the workflow of user interaction with the system in the “Create a Song” mode. In “Quick Select” mode, the song is finished after the user selects a feeling.

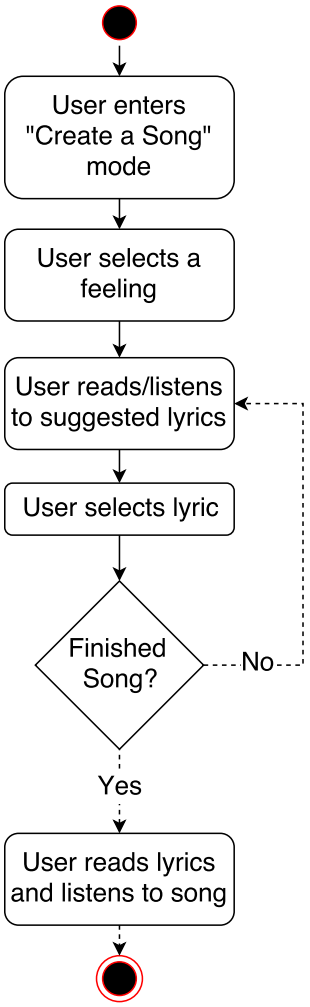

Figure 4.1: Activity Diagram

5 Conceptual Model

Users will begin navigating the application with the homepage in Figure 5.1. Upon launching the homepage, they are presented with two modes to create a song. The “Create a Song” mode walks the user through the full process of emotion and lyric selection, while the “Quick Select” mode includes emotion selection but omits lyric selection. The user is also presented with the option to view saved songs in the system.

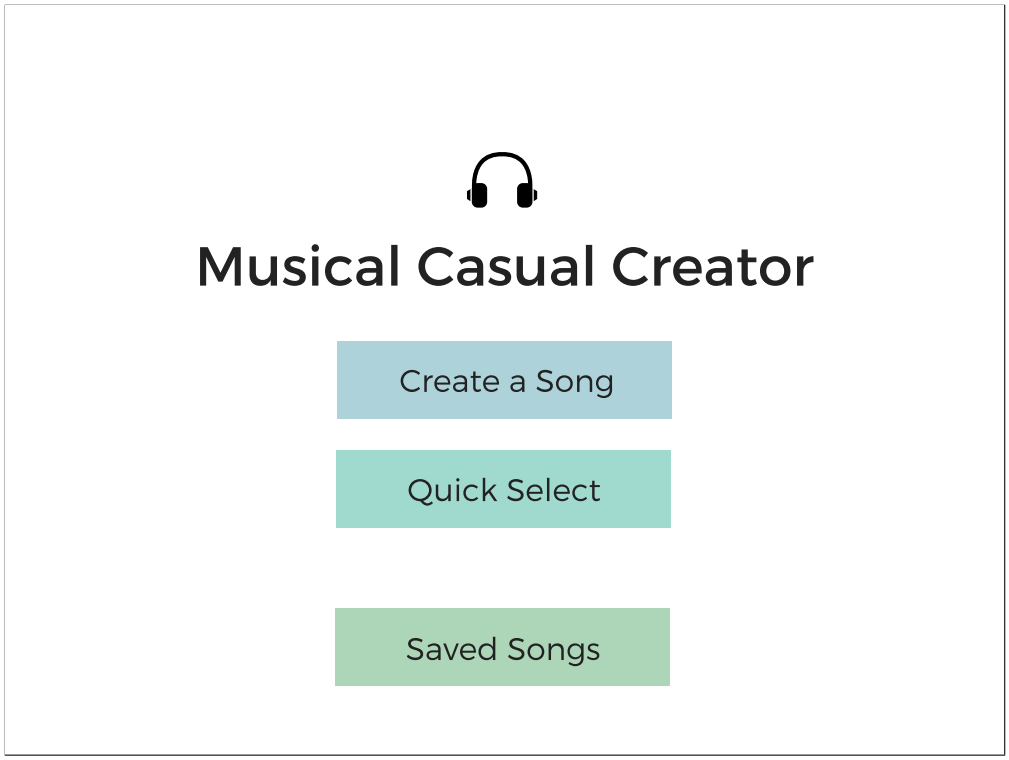

Figure 5.1: System Homepage

Upon selecting one of the modes to create a song, the user will be directed to the feeling selection page in Figure 5.2. The user is prompted to select one of the displayed feelings that best describe what the user is experiencing or what the user wants to express.

Once the user selects a feeling, the system directs the user to the lyric building page in Figure 5.3. (If the user is in “Quick Select” mode, this page is skipped) On the left side of the page, the user is reminded of his or her feeling selection. The user can select this section to go back to the feeling selection page.

The lyric selection page displays three suggested lyrics based on the feeling that the user selected. The user can either read the lyrics or listen to them by selecting the sound icon next to each lyric. If the user does not like any of the suggested lyrics, he or she can choose from a new set of lyrics with the option “Regenerate Lyrics.” Once users are satisfied with their selected lyrics, they can select “Finish song” to conclude the song.

After the user concludes song lyric selection, the system presents the final result in the finished song page in Figure 5.4. This page displays the full song lyrics and a music player to play the song. The user can also select “Save Song” to save the song in the system.

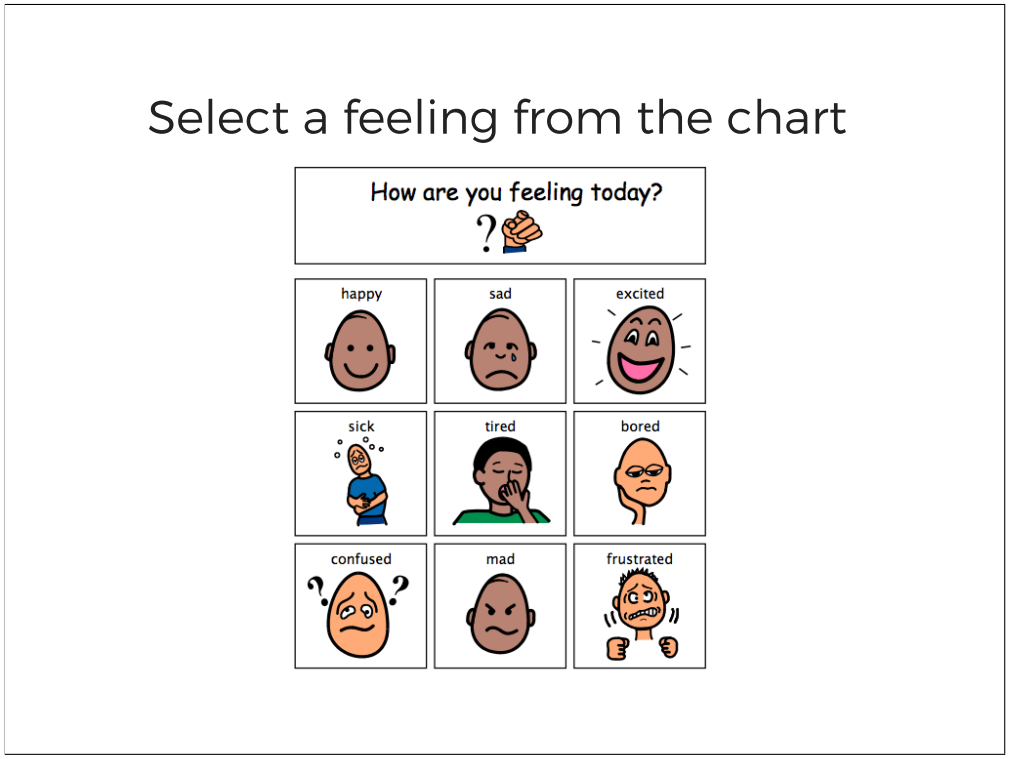

Figure 5.2: Feeling Selection Page

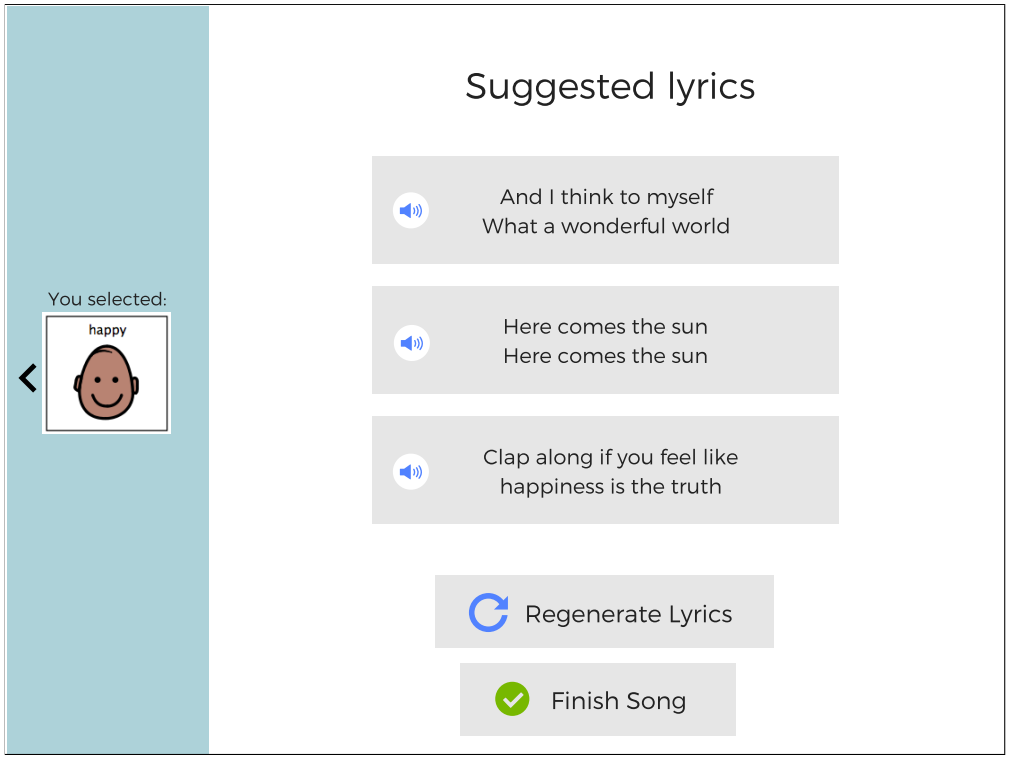

Figure 5.3: Lyric Building Page

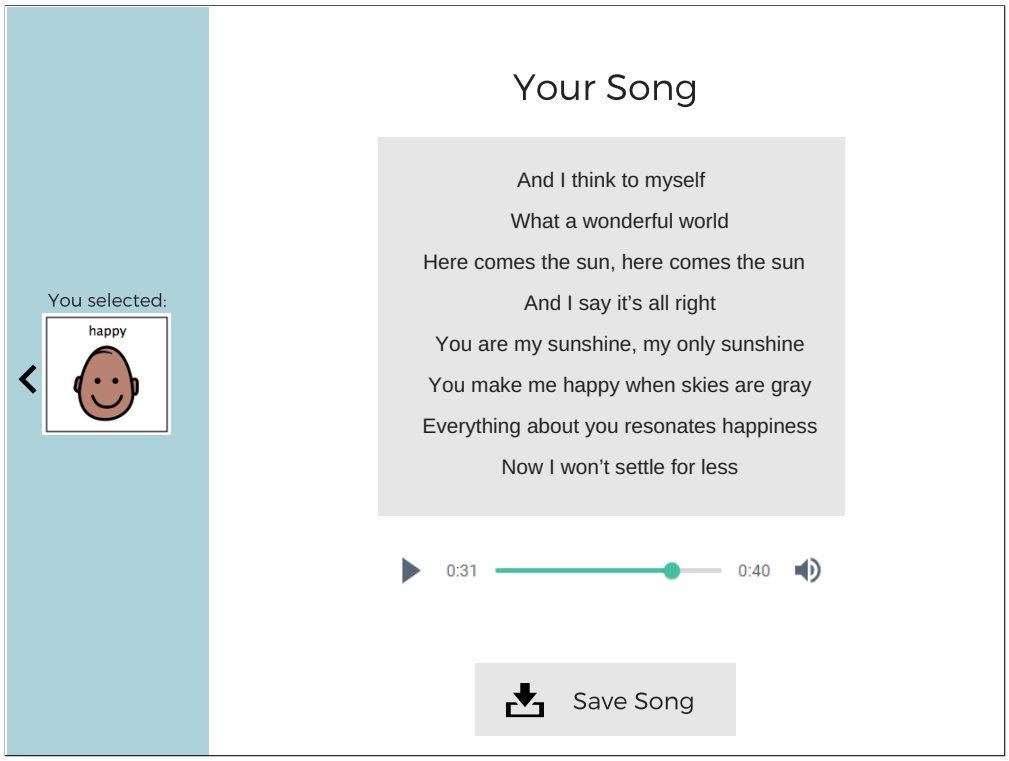

Figure 5.4: Finished Song Page

6 Architectural Diagram

We plan to utilize a client-server architecture for this application, as shown in Figure 6.1. Clients (users) will be able to interact with the server through a web application. The server is responsible for handling user input, processing data, and delivering results back to clients.

The system will use PHP as the main server-side language. The server will also incorporate Python scripts for data processing and a MySQL database to store and retrieve data.

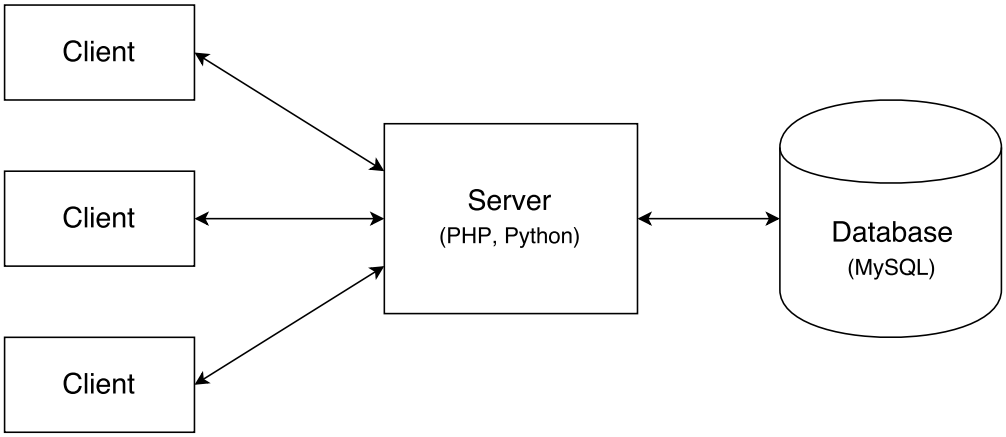

Figure 6.1: Architectural Diagram

7 Technologies Used

Listed are the technologies we plan to utilize to implement our solution. These include programming languages, frameworks, and any other tools for development.

- PHP
  - Primary back-end language to connect web application with the MySQL database, process requests, and execute Python scripts
- Python
  - Scripting language to utilize machine learning libraries for generating lyrics and melodies
- MySQL
  - Database used to store and retrieve application data
- HTML
  - Standard markup language to create web content
- CSS
  - Language for format and styling of the web content
- JavaScript
  - Front-end language for web content functionalities
- jQuery
  - JavaScript library for simplifying client-side scripting
- Github
  - Web-based version control repository hosting service
- FileZilla
  - FTP application for transferring files over the internet

8 Design Rationale

8.1 Back-end Decisions

We chose PHP as our back-end language because of its easy deployment with the SCU Engineering Computing Center linux machines. The Computing Center already has a PHP server installed and ready to use in the student webpages folder. In addition, PHP is optimized for making web applications quickly. It is designed to work well with the web, especially with the built-in HTTP request capabilities that we will use for our application.

We chose to use Python primarily because it is the most popular language to use for machine learning applications. Python has a large collection of machine learning and music libraries that we plan to leverage for lyric and melody generation. This way, we can avoid reinventing the wheel and focus on building a useful product instead. We plan to integrate Python with PHP by calling Python scripts in the background.

The reason we chose to use MySQL is that it has well-documented support for integration with PHP and Python. The advantage of using MySQL for our database is that it is a powerful database system that is widely supported and used. It is also simple to get started with while offering a complex set of tools that is more than sufficient for the basic needs of this application.

We decided to build a client-server architecture to give users centralized access from anywhere, as long as they have an internet connection. Another advantage of this architecture is the ability to delegate all of the heavy computing to be done in server-side instead of client-side. The motivation behind this is to ensure that the application runs consistently fast for each user. Although this architecture creates a single point of failure in the server, we decided that the advantages outweighs the disadvantages for this application.

8.2 User Experience Decisions

Before designing our user experience, we consulted with a specialist who is certified in teaching children with special needs. Our decisions were shaped by her recommendations in order to make the application useful for our target audience.

We decided to offer two modes for creating songs in order to add a degree of creative freedom in the creative process. Having the “Create a Song” and “Quick Select” mode serves two purposes; it assists users who have difficulty selecting lyrics and gives the user more control over their creations.

We want to implement a feature to save songs in the system so that users can revisit songs that they created. This adds significant usefulness because it allows them to share their creative expressions with others. This also gives them the opportunity to revisit how they have previously expressed a given emotion and learn from it.

We chose to use the feelings chart as depicted in Figure 5.2 because it is the standard chart used in helping children with special needs identify how they are feeling. By using an interface that is more familiar to our target audience, the interface will help provide a more intuitive experience.

In the lyrics building page, as depicted in Figure 5.3, we intend to present three lyric options because having multiple options helps users feel more in control. However, we recognize that adding more options can also overwhelm users, so we settled for having three lyric options and the option to generate a new set of three lyrics to choose from.

9 Test Plan

Our test plan consists of two types of testing. For white box testing, we would like to test individual features through unit testing. After we have successfully unit tested a component, we would integrate it into our system. We would then use regression testing here to make sure any new features we add to the system do not break any of the other features already in the system. For black box testing, we would like to test the features without any knowledge of the code. We would also try to identify bugs in our system. Our last form of black box testing is verification testing, which will be done after our system is working.

9.1 White Box Testing

- Unit Testing
  - Test each individual feature module in the system by trying each of them out
    - Read a lyric
      - Is lyric displayed correctly?
      - Do the lyrics that evoke the selected emotion?
    - Listen to a lyric
      - Is the lyric correctly read back?
    - Select a lyric
      - Does selecting a certain lyric build a song that makes sense and connects to the lyric selected?
      - Does the individual lyric make sense?
    - Listen to a song
      - Does the system correctly playback the song?
      - Does the music (chords and melody) evoke the selected emotion?
    - Saving/Revisiting a song
      - Does the system correctly save a song?
      - Does the system allow for proper revisiting of saved songs?
- Regression Testing
  - Test that the system works after adding each unit to the system

9.2 Black Box Testing

- Try to create a song with only the knowledge of the system requirements
- Test every possible action on each of the webpages to make sure nothing breaks the system
- Verification Testing
  - Test the system with our target audience in the field to see if it correctly helps them express or feel and emotion

10 Risk Analysis

Our risk analysis deals with risks we can see ourselves encountering while working on this project. We would not like to only limit our analysis to technical risks; school and other life risks can also arise and affect our project. Table 10.1 shows our risk analysis table.

Table 10.1: Risk Analysis Table

Risk	**Consequence**	**Probability**	**Severity**	**Impact**	**Mitigation**
Busy School Schedule	A busy school schedule could get in the way of working on our system in implementation or testing. This can lead to unfinished features or requirements because of limited time.	4.4	0.5	2.2	Do our work as soon as we can so that our schedules, however busy they may be, can be as free as possible in order to work on our system.
Out of Touch with Target Audience	Our system would not suit the target audience correctly and even if it works technically, does not successfully help its users to feel or express a certain emotion.	4.8	0.3	1.44	Meet with a sample of target users and those who work with people in our target audience to fully understand what they would want from our system.
Sickness	Sickness can come up and affect either or both of us at any time. This can be a setback and can lead to unfinished features or requirements because of limited time.	3.6	0.5	1.28	Try our best to exercise regularly and live healthy lives. This would help minimize the chance of getting sick.
Technical Mishaps	Our system could be malfunctioning. Fixing bugs would lessen our time in implementing other requirements.	5.6	0.2	1.12	Test our system extensively on a unit-by-unit basis and work on mock ups and small features to ensure our system works correctly.

11 Development-Timeline

The Gantt chart in Figure 11.1 represents the development timeline for our project. Time is broken up by each academic quarter, and each quarter is broken up into three or four week periods. Each assigned owner oversees the task at hand but execution will be a collective effort. Each task is located on the left-most column of the figure. Furthermore, each task is broken up into three different types: documentation, implementation, and testing. Blue represents documentation, red represents implementation, and green represents testing.

		Weeks of Fall 2017			Break	Weeks of Winter 2018			Weeks of Spring 2018		
	Owner	1-3	3-6	6-10		1-3	3-6	6-10	1-3	3-6	6-10
Documentation											
Problem Stament	Ian										
Design Report	Louis										
Design Review	Ian										
Final Report	Louis										
Final Presentation	Ian										
Implementation											
Lyric Generation	Louis										
Music Generation	Ian										
Front End	Ian										
Back End	Louis										
Testing											
Unit Testing	Louis										
Regression Testing	Ian										
Verification Testing	Louis										

Figure 11.1: Development Timeline Gantt Chart

12 Social Implications

As part of our motivation for creating and maintaining this project, we considered multiple areas of social implications related to our work. As in any engineering project, there are not only problems that we aim to solve, but also potential problems that can arise along with our work.

12.1 Ethical

The ethics of machine learning solutions have been historically controversial [6]. As our project heavily relies on machine learning, we acknowledge that our work has to address the issue of ethics. On top of that, our project domain crosses between machine learning and music therapy for a use case that has never been explored. The practice of music therapy is a field that requires clinical training and advanced education. So by starting a project that has the potential to fill the role of a musical therapist, we carry the ethical responsibility of making a system that can either do a lot of good or fail disastrously. Acknowledging that we do not have the qualifications to practice music therapy, we have been grateful to collaborate with a special needs teacher who provided us with direction on the project. We took careful steps to mitigate the risks of machine learning mistakes such as screening data before it entered our system and thoroughly testing the results of our system.

12.2 Societal

Societal implications describe the impact that our project would have on improving the lives of people. While our project is specifically designed to benefit our users directly, there can be indirect effects that arise from its usage. One of the main advantages and motivations for our project is that it can be used to provide music therapy benefits for people with autism. By providing this tool to be freely accessible to anyone with an internet connection, we empower users to play and learn with this tool. In addition, it can be leveraged by teachers or others who do not necessarily have the proper training in music therapy to provide a fun and engaging musical activity for people with autism.

12.3 Economic

With our commitment to making our project freely accessible to anyone with an internet connection, we help reduce an economic burden. We provide this alternative solution for people seeking music therapy, who would otherwise have to pay high costs for hiring a trained music therapist. We realize that the current state of our implementation alone would not be sufficient to replace the job of a music therapist, but the concept can potentially be borrowed to create a system that eliminates the need for music therapists one day.

12.4 Usability

A main concern of our songwriting tool was that the user interface would not be easily used by our target audience. To address this issue, we collaborated with a teacher for children with special needs to design the user interface and experience. As addressed in our design rationale, we took active measures to create a useful platform for our audience.

13 Conclusion

Over the course of the academic year we followed the development plan to achieve many of our goals for the project. We made some adjustments along the way in order ensure that the implementation would be completed in time to allow for thorough testing and results analysis.

13.1 Results

We are satisfied with our main accomplishment of building a free and accessible tool that people can use to create songs with the help of machine learning. Our project is publicly accessible through our website hosted by the SCU Engineering Computing Center Student Webpages for anyone to use and learn from:

`http://students.engr.scu.edu/~isantill/mcc/select_emotion.php`

We succeeded in dividing the work between two team members before combining all of the components together in one easy-to-use web application. We implemented an emotion-based lyrics generator to create new lyrics that rhyme. We also implemented a chord progression generator that is able to musically express the feelings that users would select.

However, the success of our project is not without notable failures. Along the way, we made design changes to trim parts of the original plan that would have otherwise prevented us from finishing the project on time. For example, we abandoned work on the ability for users to save songs and view saved songs because it was unnecessary for demonstration purposes. We also cut out the ”Select Mode” feature at the beginning of the application. These changes allowed us to free up time for thorough testing, an area that we prioritized.

In our efforts to test our project, we conducted field testing and survey testing. For field testing, we were fortunate to have been able to work with MAGIC Dance Arts, a dance class for children with special needs, to evaluate how our project works with a sample audience. For survey testing, we recruited family and friends to try creating a song with our application and take a Likert scale survey to analyze areas of strength and weakness in our project. The feedback that we received is valuable for future improvements and maintenance of the project.

13.2 Lessons Learned

We learned a lot from designing and building this project in its entirety. In essence, we learned that complexity is a huge factor in the ability for engineers to accomplish a project in a limited amount of time. While we initially gave a lot of consideration to the project ideas and possibilities, we should have been more careful to narrow down the scope of our project. It would have saved us a lot of time if we did not have to improvise design changes along the way.

With the feedback we received through testing, we learned of several points of weakness about our project. The first is the problem of incoherence in the lyrics generator. Because we utilized using machine learning tools to generate seemingly random lyrics, we acknowledge that this is also a common limitation with machine learning applications in the industry. In addition, the generated music sometimes does not match our user's idea of a specific emotion. We also acknowledge this limitation because music is subjective; one person's feelings can be represented in many ways with music.

13.3 Future Work

We identified potential areas of improvement that can be made in the future. We would like to improve the coherence of the song lyrics by generating new lyrics using the context of previously selected lyrics. We would also like to

automate the process of adding different themes of song lyrics for different audiences. In addition, we would like to add even more emotions that can be represented using the tool. Beyond the scope of our project, the concept of our songwriting tool can be repurposed for many other applications, including entertainment and education. We truly believe in the positive impact that this project will have for our users and for future development.

14 References

[1] "Characteristics", *Autism Spectrum*, 2017. [Online]. Available: https://www.autismspectrum.org.au/content/characteristics. [Accessed: 06- Oct- 2017].

[2] "Autism Spectrum Disorder: Communication Problems in Children", *National Institute on Deafness and Other Communication Disorders*, 2017. [Online]. Available: https://www.nidcd.nih.gov/health/autism-spectrum-disorder-communication-problems-children. [Accessed: 06- Oct- 2017].

[3] L. Bieleninik, M. Geretsegger, K. Mössler, J. Assmus, G. Thompson, G. Gattino, C. Elefant, T. Gottfried, R. Igliozzi, F. Muratori, F. Suvini, J. Kim, M. Crawford, H. Odell-Miller, A. Oldfield, Ó. Casey, J. Finnemann, J. Carpente, A. Park, E. Grossi and C. Gold, "Effects of Improvisational Music Therapy vs Enhanced Standard Care on Symptom Severity Among Children With Autism Spectrum Disorder", *The JAMA Network*, 2017. [Online]. Available: http://jamanetwork.com/journals/jama/article-abstract/2647867. [Accessed: 06- Oct- 2017].

[4] M. De Vivo, "Music Therapy May Help Children with Autism", *ASF Blog*, 2017. [Online]. Available: https://autismsciencefoundation.wordpress.com/2013/08/30/music-therapy-may-help-children-with-autism/. [Accessed: 06- Oct- 2017].

[5] "The Benefits Of Music Therapy For Autistic Children", *Nurse Journal*, 2017. [Online]. Available: https://nursejournal.org/community/the-benefits-of-music-therapy-for-autistic-children/. [Accessed: 06- Oct- 2017].

[6] N. Bostrom and E. Yudkowsky, "The Ethics of Artificial Intelligence", *Cambridge Handbook of Artificial Intelligence*. [Online]. Available: https://intelligence.org/files/EthicsofAI.pdf. [Accessed: 03- Jun- 2018].