# CS 242 Final Project Proposal

Title: JamATuna
Author 1: Eric Auster (eauste2), Moderator: Lucas Erb
Author 2: Thomas Warther (warther2), Moderator: Ti-Chung, Cheng

---



---

## Outline

**1. Abstract**

1.1. **Project Purpose**

The purpose of this project is to provide a fun and interactive web application for creating music with others in such a way that knowledge of music theory or other musical skills will not be required.

1.2. **Background/Motivation**

We wanted to take what we have learned in this course and combine it with our background in music to make an interactive musical application. We believe this application would let us and others experience music creation as a social group activity. There are several projects online that have motivated us to make our own such as Plink (http://dinahmoelabs.com/_plink/) and WhatsGroovy (http://www.whatsgroovy.com/), each incorporating multiplayer aspects, music creation, and visualizations.

**2. Technical Specifications**

2.1. **Platform:**

Website

2.2. **Programming Languages:**

JavaScript

2.3. **Stylistic Conventions:**

CamelCase naming convention, function signature commenting

2.4. **SDK:**

Node.js, Web Audio API, AudioContextMonkeyPatch.js, Tuna.js, web-audio-testing-api, JQuery, Bootstrap, EJS, socket.io, Firebase

2.5. **IDE:**

WebStorm or VS Code

2.6. **Tools/Interfaces:**

Google Chrome

2.7. **Target Audience:**

Everyone

## 3. Functional Specifications

### 3.1. Features

- Users can choose their own nickname
- Users can create “jam rooms” or different server instances
- Users can select instrument to play
- Users can click interface to play different notes at different times

### 3.2. Scope of Project

Application will be small scale with up to four people being able to join a jam room at a time.

## 4. Timeline

### 4.1. Week 1 - Musical and Networking Prototypes

- Eric
  - Create a Web audio prototype
  - Create a prototype of a musical instrument component

    - Make a webpage with some clickable buttons to play some sounds with the instrument component
    - Integrate musical instrument component with server
    - Write tests for musical instrument with web-audio-test-api and tests for general JavaScript with Jasmine
- Thomas
    - Create prototype with socket.io of two or more users connecting to the server each using Chrome browser and seeing each others' usernames
    - Design a profile page to show a user's username and profile picture
    - Make a chat box to allow two or more users to communicate with each other with text
    - Integrate networking with Eric's musical instrument prototype to allow users to play music simultaneously on the same webpage
    - Write unit tests with Jasmine

### 4.2. **Week 2 - Musical and Backend Development**

- Eric
    - Create multiple instrument components (at least four including the first week's), each with uniquely sampled sounds that can be triggered from the front-end of a website
    - Provide a means for audio synchronization in regards to tempo
    - Experiment and implement different audio effects and web audio that can be toggled from the front-end
    - Integrate modifications with the server
    - Further integrate testing with all newly developed code
- Thomas
    - Set up Firebase database to store and retrieve metadata about jam rooms
    - Design webpages to support creation of and joining jam rooms and update the database accordingly
    - Design webpage to login with a username and password and store passwords encrypted in the database
    - Write unit tests with Jasmine and a manual test plan

### 4.3. **Week 3 - Frontend and Server Development**

- Eric

    - Create a rough front-end interface that allows a user to select audio instruments, effects, and so on
    - Create animations and visualizations that will coincide with audio and user events
    - Users will be able to play an instrument in sync and key with the audio already being played by clicking the screen in different areas
    - Integrate modifications into server
    - Work on a manual test plan that will describe the way audio, visualizations, and audio should interact properly
- Thomas
    - For a given jam room, combine musical inputs from all users at the server into a single, synchronised audio stream and send the stream to all users.
    - Measure latencies at the server and log them
    - Implement search functionality to allow users to search for a jam room by the name of a jam room or for a specific username
    - Write unit tests with Jasmine and update manual test plan accordingly

### 4.4. **Week 4 - Additional Features, Optimizations, and Data Analysis**

- Eric
    - Improve the overall website design to make it look polished and clean, fixing any bugs, adding more user interface elements if necessary
    - Create more visualizations and animations to go with the audio
    - Further develop audio effects to include more variety
    - Make sure the front-end will synchronize with the back-end, allowing multiple users to jam together in a single instance
    - Tests should cover all major areas and manual test plan should be updated according to changes made
- Thomas
    - Allow a user to post his/her jam room code to social media to help other people find his/her jam room
    - Test and improve the stability of the website to ensure the functionality is reliable and responsive
    - Refactor backend code to make it more organized and understandable

- Collect and report statistics of the numbers of users and jam rooms by time of day with visualizations
- Write unit tests with Jasmine and update manual test plan accordingly

**5. Future Enhancements**

In the future, we can add in more parameters for creating music such as tempo adjustment and key changes. We may also integrate Discord into the website to provide more social features for users by utilizing Discord's API. We may also look into writing an AI to play with users when they have no one else to play with.

---

## Rubrics

**Week 1**

- Eric
  - Requirement 1 - Create web audio prototype
    - 2.5 points: Additional functionality
    - 2 points: Web audio works with a web page
    - 1 point: Partially integrated or only researched
    - 0 points: No effort made
  - Requirement 2 - Develop templated musical instrument component
    - 2.5 points: Additional functionality
    - 2 points: Musical instrument component is functional
    - 1 point: Partially created or partially functional
    - 0 points: Not functional and little effort
  - Requirement 3 - Design web page for playing musical instrument component
    - 2.5 points: Web page has additional functionality
    - 2 points: Working web page with musical instrument component
    - 1 point: Partially created web page, no musical instrument component
    - 0 points: No web page
  - Requirement 4 - Integrate musical instrument component with server
    - 2.5 points: Additional functionality
    - 2 points: Musical instrument component integrated with server
    - 1 point: Partially integrated, buggy
    - 0 points: Not working, no integration
  - Testing
    - 2.5 points: Excellent testing
    - 2 points: Satisfactory testing

        - 1 point: A little testing
        - 0 points: No testing
- Thomas
    - Requirement 1 - Networking prototype
        - 2.5 points: At least four users can connect to the server and see each others' usernames.
        - 2 points: Two users can connect to the server and see each others' usernames.
        - 1 point: Limited functionality or buggy implementation
        - 0 points: No connectivity between users supported
    - Requirement 2 - Profile page
        - 2.5 points: Exceptional visual presentation or a user can upload his/her own profile picture
        - 2 points: A webpage can show a user's username and let a user select a profile picture.
        - 1 point: Partial functionality
        - 0 points: No profile page implemented
    - Requirement 3 - Chat box
        - 2.5 points: The chat box has exceptional visual presentation or more than two users can communicate in the chat box.
        - 2 points: Two users can communicate with text in a chat box.
        - 1 point: Chat box functionality is incomplete.
        - 0 points: No chat box implemented
    - Requirement 4 - Integrate with music
        - 2.5 points: Seamless integration of music with webpage
        - 2 points: Users can play music simultaneously.
        - 1 point: Partial functionality or buggy implementation
        - 0 points: No music integration
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing

**Week 2**

- Eric
    - Requirement 1 - Create multiple instrument components
        - 2.5 points: More than four instruments made
        - 2 points: Four instruments made
        - 1 point: Less than four instruments made
        - 0 points: One or less instruments made
    - Requirement 2 - Provide means for audio synchronization in regards to tempo

        - 2.5 points: Additional functionality
        - 2 points: Synchronization working
        - 1 point: Partially working or buggy
        - 0 points: Not working at all
    - Requirement 3 - Implement several different audio effects that can be toggled
        - 2.5 points: More than two audio effects
        - 2 points: At least two audio effects made and toggled
        - 1 point: One audio effect
        - 0 points: None
    - Requirement 4 - Integrate modifications with server
        - 2.5 points: Additional functionality
        - 2 points: Works while integrated with the server
        - 1 point: Partially working, buggy
        - 0 points: No integration
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing
- Thomas
    - Requirement 1 - Create jam rooms
        - 2.5 points: The webpage for creating a jam room has clean UI and good styling.
        - 2 points: A webpage for creating a jam room is implemented and created jam rooms get stored in the database.
        - 1 point: A webpage for creating a jam room is implemented, but jam rooms do not get stored.
        - 0 points: No creation of jam rooms is supported.
    - Requirement 2 - User login
        - 2.5 points: Other login methods, such as Facebook and Google accounts, are supported.
        - 2 points: A user can sign up for an account and login with a password. The password is stored encrypted in the database.
        - 1 point: At least one of the following is not implemented: account signup, account login, or encryption of stored passwords
        - 0 points: No account signup, account login, or encryption of stored passwords implemented
    - Requirement 3 - Joining jam rooms
        - 2.5 points: The webpage for joining a jam room has clean UI and good styling.
        - 2 points: A webpage for joining a jam room is implemented and the database is updated accordingly.

        - 1 point: A webpage for joining a jam room is implemented, but the database is not updated accordingly.
        - 0 points: No joining jam rooms supported
    - Requirement 4 - Jam Room Metadata Storage
        - 2.5 points: Efficient or clever implementation of database storage and retrieval
        - 2 points: Metadata about jam rooms and users can be stored in a database and retrieved
        - 1 point: Either jam room or user metadata is not stored or metadata is not retrievable
        - 0 points: No storage implemented
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing

**Week 3**

- Eric
    - Requirement 1 - Create better front-end interface
        - 2.5 points: Additional functionality
        - 2 points: More user interface elements, polished looking
        - 1 point:
        - 0 points:
    - Requirement 2 - Create animations and visualizations
        - 2.5 points: Additional functionality
        - 2 points: Animations and visualizations present
        - 1 point: Little animation and visualization work
        - 0 points: No animations or visualizations
    - Requirement 3 - Instrument note and beat quantization
        - 2.5 points: Additional functionality
        - 2 points: Note and beats quantized
        - 1 point: Little quantization, buggy
        - 0 points: No quantization
    - Requirement 4 - Integrate modifications with server
        - 2.5 points: Additional functionality
        - 2 points: Integrated changes with the server
        - 1 point: Partially integrated
        - 0 points: Not integrated
    - Testing
        - 2.5 points: Excellent testing with test plan
        - 2 points: Create manual test plan along with more testing

        - 1 point: No test plan, little testing
        - 0 points: No testing, no test plan
- Thomas
    - Requirement 1 - Audio sync
        - 2.5 points: User inputs are combined efficiently.
        - 2 points: Combine user inputs into a single, synchronised audio stream
        - 1 point: Partial synchronization, or buggy implementation
        - 0 points: No functionality
    - Requirement 2 - Output synced audio
        - 2.5 points: Output is reliable and efficient.
        - 2 points: Combined audio stream is sent to all users.
        - 1 point: Partial output or buggy output
        - 0 points: No output
    - Requirement 3 - Measure and log latency performance metrics
        - 2.5 points: Latency metrics demonstrate that the latency is imperceptible.
        - 2 points: Latency metrics are accurate and informative.
        - 1 point: Latency metrics are inaccurate or uninformative.
        - 0 points: No latency metrics implemented
    - Requirement 4 - Jam Room Search
        - 2.5 points: Extra ways to search and filter results are supported.
        - 2 points: A user can search for a jam room by name of jam room or username.
        - 1 point: Partial implementation or buggy implementation
        - 0 points: No search implemented
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing

**Week 4**

- Eric
    - Requirement 1 - Polish web page design
        - 2.5 points: Additional functionality and design
        - 2 points: Web page should look finalized
        - 1 point: Little changes from previous week
        - 0 points: No improvements
    - Requirement 2 - More animations and visualizations
        - 2.5 points: Exceptional work
        - 2 points: More animations and visualizations, better looking
        - 1 point: Little additions made
        - 0 points: No improvement

    - Requirement 3 - Develop and integrate more audio effects
        - 2.5 points: Exceptional work
        - 2 points: More audio effects and features added
        - 1 point: Little additions made
        - 0 points: No additions made
    - Requirement 4 - Synchronize front-end with back-end
        - 2.5 points: Additional functionality
        - 2 points: Front-end is fully synchronized with back-end
        - 1 point: Not fully synchronized, buggy
        - 0 points: No synchronization
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing
- Thomas
    - Requirement 1 - Social Media Integration
        - 2.5 points: Multiple social media websites supported
        - 2 points: A user can post his/her jam room code to Facebook to help other people find his/her jam room.
        - 1 point: Partial or buggy implementation
        - 0 points: No functionality
    - Requirement 2 - Test and improve platform stability
        - 2.5 points: Platform functions flawlessly.
        - 2 points: Platform functions reliably.
        - 1 point: Platform somewhat unstable
        - 0 points: Platform unusable because of instability
    - Requirement 3 - Refactor backend code
        - 2.5 points: Code is easily understood by another programmer.
        - 2 points: Code quality improved
        - 1 point: Code disorganized or confusing
        - 0 points: Code unreadable to another programmer
    - Requirement 4 - Collect and report user statistics
        - 2.5 points: Creative or multiple visualizations
        - 2 points: Visualizations of the number of users and jam rooms by time of day
        - 1 point: Poor visualizations
        - 0 points: No visualizations
    - Testing
        - 2.5 points: Excellent testing
        - 2 points: Satisfactory testing
        - 1 point: A little testing
        - 0 points: No testing