**Team 37 Product Backlog (Synquencer)**

Mitchell Bridwell, Zach Heskett, Aidan McDonnell, and Mohini Roplekar

**Problem Statement**

While many websites offer the ability to compose MIDI sequences online, and some even allow users to edit each other's sequences, these websites are not accessible, functional, or user-friendly enough to make this a simple process. Additionally, these existing solutions only allow for asynchronous editing rather than real-time collaboration that users have become accustomed to in websites like Google Docs.

**Background**

Collaboration is very common in digital music production, but there are surprisingly few tools to help musicians collaborate remotely. One essential part of music production is creating MIDI sequences to play back with virtual instruments. Musicians most commonly use piano roll sequencers within the digital audio workstation (DAW) of their choosing to write these sequences. However, since there is no convenient way to collaborate on these sequences in real time, musicians often resort to sending project files back and forth, which can be tedious.

There are a few applications that come close to solving this problem, but each falls short in one way or another. OnlineSequencer.net, for example, allows users to edit and share MIDI sequences online, but it does not allow for real-time collaboration. Users must save their sequence and send it to another user, which is just as tedious as sending a project file. Other websites, like BandLab.com, attempt to replace the user's DAW entirely, but these web-based applications inevitably fall short of professional desktop solutions. Additionally, both of these applications are lacking in accessibility to those with visual impairments.

Our solution is focused on providing a smooth, accessible, and collaborative MIDI-editing environment. If the user so chooses, they can export a sequence from the application and import into their chosen DAW. We believe that we can provide a better experience by focusing specifically on this use case.

**Functional Requirements**

1. As a user, I would like to be able to create a new, empty sequence
2. As a user, I would like to add a note to the sequence by clicking an empty spot
3. As a user, I would like to edit or remove a note from the sequence.
4. As a user, I would like to move or copy a note somewhere else.
5. As a user, I would like to undo an action.
6. As a user, I would like to redo an action after undoing twice.
7. As a user, I would like to see who is working on the same project as me
8. As a user, I would like to be able to see a note placed by another user in the same project
9. As a user, I would like to be able to see a note edited by another user in the same project
10. As a user, I would like to use multiple different instruments in my sequence
11. As a user, I can play back and listen to my project in the editor
12. As a user, I can play the sequence in time with another user
13. As a user, I can lengthen or shorten the duration of a note.
14. As a user, I can select multiple notes and move them all at once
15. As a user, I would like to change the tempo of the sequence
16. As a user, I would like to loop a specific section of the sequence (cycle) to work on
17. As a user, I would like to use my computer keyboard to play the currently selected instrument.
18. As a user, I would like to enter notes using only my keyboard.
19. As a user, I would like to be able to change instruments and other settings using only my keyboard.
20. As a user, I would like to zoom in and out of the sequence
21. As a user, I would like to scroll up, down, left, and right in the sequence.
22. As a user, I would like to choose between multiple color schemes for the application.
23. As a user, I would like to change the time signature of the pattern
24. As a user, I would like to change the length of the pattern
25. As a user, I would like to change the grid resolution of the piano roll
26. As a user, I would like to change the velocity of a note
27. As a user, I would like to store my sequences on the server and access them later
28. As a user, I would like to share my sequences with other users
29. As a user, I would like to edit a sequence alongside other users asynchronously
30. As a user, I would like to edit a sequence alongside other users synchronously
31. As a user, I would like to export my sequences to MIDI files and store them locally
32. As a user, I would like to import MIDI files stored on my computer and edit them (if time allows)
33. As a user, I would like to configure the sounds that the software uses to play back sequences (if time allows)

**Non-Functional Requirements:**

**Architecture and Performance:**

Synquencer aims to be an easy and seamless way for musicians to jot their ideas down and see how they sound. To achieve this goal, it's important that everything runs quickly and efficiently. We plan on using the React and Next.js libraries, both of which are Javascript libraries, to develop Synquencer. React allows for easy reusability of certain components of an application. Given that a sequencer requires a lot of redundant components in the form of different notes, React provides us with the perfect tools to quickly build a quality project. It also helps us use the same objects in our frontend and backend. Next.js provides easy, quick API interaction with React. On top of this, Next.js is extremely performant, which is incredibly important to our goals. The React and Next.js combination allows us to write both our backend and frontend in TypeScript, a variant of Javascript which implements static typing and transpiles to standard Javascript. Since we will only be working with TypeScript for the entire code base, and there is seamless interaction between the frontend and backend, it will be easy for all members to understand each part of our project.

**Conversion:**

The sequencer will use JSON for synchronization and storage features. All objects in a project will be in JSON on the server side. In order for our users to export and listen to their projects, however, we need to convert this data into MIDI. We will write a system that can convert our custom JSON-based notes into industry-standard MIDI files which can be used by our users for their own DAWs, or whatever purposes they see fit. We plan to use MidiWriterJS, a Javascript library for creating MIDI files, for this purpose. This will make creation of the MIDI data significantly easier.

**Synchronization:**

Synchronous editing of sequences is one of the Synquencer’s main selling points. To provide users with the best synchronous experience, we need to reduce latency between users as much as possible. Without reducing latency to an acceptable level, the experience will be too clunky for our users and it won’t be a practical solution for collaboration. We plan to use a queue system for editing notes in the sequence. New changes will be added to the queue, and the local client will add changes from that queue to the current active project. With this system, we can synchronize between users by sending as little data between users and the server as possible.