Requirements and Specification Document

*Digital Sampler*

Groovy Gurus: Mohnish Kalia, Daniel Ko, Chase Kozak, PJ Rubin, Aiden Tepper, Lordphone Wen

Requirements and Specification Document  
*2023-02-28*, version 1.0

2023-03-05, version 2.0

Project Abstract

Our software is a digital sampler that will function similar to a soundboard. This will allow users to upload, create, mix, and edit sounds to their specifications. There will also be a recording feature where a user can record the mix they have created. Additionally, we want to add meta information about the mixes like description and genre.

Document Revision History

*Your first version of this document is version 1.0. After that minor changes increment the minor version number (e.g., 1.1, 1.2, …) and major changes increment the major version number and set the minor number to zero (e.g., 2.0, 3.0, …). We will follow this convention with other documents as well.*

Rev. 1.0 2023-03-09: initial version

Rev. 1.1 2023-03-25: cosmetic changes

Rev 1.2 2023-04-01: changes to User requirements

Rev 1.3 2023-04-03: Changes to some places it talks about google login

Customer

The customers for our product will be music producers, audio engineers, and music enthusiasts. Producers can use our digital sampler to mix and create sounds that can be used in a song or any other production. An audio engineer can use it to finely tune sounds to make sure different sounds are properly heard and tuned. Music enthusiasts can use it to create any sort of fun or creative effects and music. There are many different physical sound boards out there, so our digital one would be beneficial for those who do not want to spend the money on a physical item.

Potential other customers who could use our app would be students going into the music field. They could use the digital sampler to help them create or edit music they are interested in or working on.

User Requirements

* Login/signup page
  + Register
    - Allows new user to enter name, username, email, and password to create account
  + Login
    - Allows existing user to sign in with username/email and password
* Sample library page
  + Displays list of uploaded samples
  + Upload button allows user to browse locally for audio files to upload
  + Allow user to preview/rename/delete/crop/get more info on individual samples
  + Allow user to organize samples into folders
  + Allow user to select one or many samples to drop into the pad
  + Allow user to record performance
* Sample pad
  + User can click on pads to trigger samples playing

Use Cases

| Name | The app has a login page that can be used for email/pass login or Google login |
| --- | --- |
| Actor | Any user of app |
| Trigger | User needs to click a link or button to create account |
| Events | * User clicks a link/button on the interface * User will be redirected to login/signup page with email/pass or Google auth |
| Exit Condition | After account is created on our end or we have auth from Google |
| Post Conditions | We have an acknowledgement message or toast show |
| Acceptance Test | * After logging in, subsequent visits will have you already signed in * User can do an info check as part of larger settings menu |

| Name | User can select from sample library to use in session |
| --- | --- |
| Actor | Any user of app |
| Trigger | User needs to click on button to “browse samples” |
| Events | * User clicks a link/button on the interface * A section of the interface will change to a sample browser * User clicks on a button next to a sample to add to list of session samples |
| Exit Condition | After close button is clicked on within interface |
| Post Conditions | The samples clicked on by the user will be available in sample list |
| Acceptance Test | * After selecting criteria, the user should be able to act in order to make a sample accessible for future playback * Should be able to take samples from this process and load them up onto pads to start playback |

| Name | User can load samples to a board |
| --- | --- |
| Actor | Any user of app |
| Trigger | User needs to click pad options and select a sample from dropdown  User needs to press options button a pad and pick the desired sample from a list |
| Events | * User clicks on options button for pad * Within menu, user selects a sample from the list |
| Exit Condition | User closes the menu or moves to another pad |
| Post Conditions | The sample chosen by the user will be able to be played |
| Acceptance Test | * After this action, playing the pad will play that sample |

| Name | User can play samples on a pad |
| --- | --- |
| Actor | Any user of app |
| Trigger | User clicks some button to start playback on the pad |
| Events | * User clicks on pad * Playback starts |
| Exit Condition | User repeats action to stop playback |
| Post Conditions | The sample for the pad starts playing |
| Acceptance Test | * After changing sample, playback should be able to start/stop via button press * User is able to see that the playback is occurring or not |

| Name | User can record performances |
| --- | --- |
| Actor | Any user of app |
| Trigger | User clicks some button on the interface to start/stop recording |
| Events | * User clicks on button * Icon changes to show that recording is in progress * User performs actions while an audio recording is occuring |
| Exit Condition | User repeats button press to stop recording |
| Post Conditions | The recording will be downloaded by the user |
| Acceptance Test | * User is able to start and stop recording * Recording contains the same audio as was played-back throughout the session * Downloaded file is able to be played by other media players |

| Name | User can save and tag recordings to the sample library |
| --- | --- |
| Actor | An actively logged in user of the app |
| Trigger | User clicks on a button to “upload sample” |
| Events | * User clicks on button * Form appears for user to fill out * User clicks on file picker * Within, a file is picked * User fills out tag (meta) information about the recording * After validating the contents, upload process can start |
| Exit Condition | User clicks on a submit button |
| Post Conditions | The recording is or isn’t uploaded to the sample library; the user gets an alert indicating resolution |
| Acceptance Test | * User provided files should only be allowed for submission if they fit certain criteria (probably audio file, and not big) * Only a logged in user can see the “upload sample” button or be allowed to upload * After confirming playback of an audio file, an upload and subsequent download/playback of the audio file is the same |

User Interface Requirements

Upon first opening the Digital Sampler, the user must be able to login with a simple login page which contains a field for a username and a field for a password. Beside these options are more fields labeled “Create Account” where a user would be able to create an account.

Once the user has logged in, they must see a sample library interface. The user must see a list of samples and buttons to interact with. They must be able to choose which samples to use and be able to upload new samples.

On the right of the sample library interface, The user must be able to see which samples are in use on the soundboard. Upon adding a sample to this list it will be available on the soundboard immediately.

The soundboard interface must resemble a physical drum machine, with a sample dock on the side listing all samples on the pad sample list. Users need to be able to click on a pad’s options to assign a sample to a button. There will be 16 sample buttons arranged into 4 rows and columns. Left-clicking on these buttons will play the sample loaded into them, while right-clicking will bring up a menu allowing the user to add, remove, or change the sample. Somewhere on this page the user will need the ability to record their work and export it.

From both the pad interface and the sample library interface, the user needs to be able to press the edit button on a sample to edit it. Here, a wavelength will appear representing the sample. Above it, a play, pause and stop button will allow the user to control playback. Below, there will be buttons to add various effects such as distortion and reverb, along with an option to crop/shorten the sample. Upon pressing this button, two lines will appear on the wavelength, one representing the new start of the sample and one the new end. The user will be able to drag these two lines to shorten the sample.

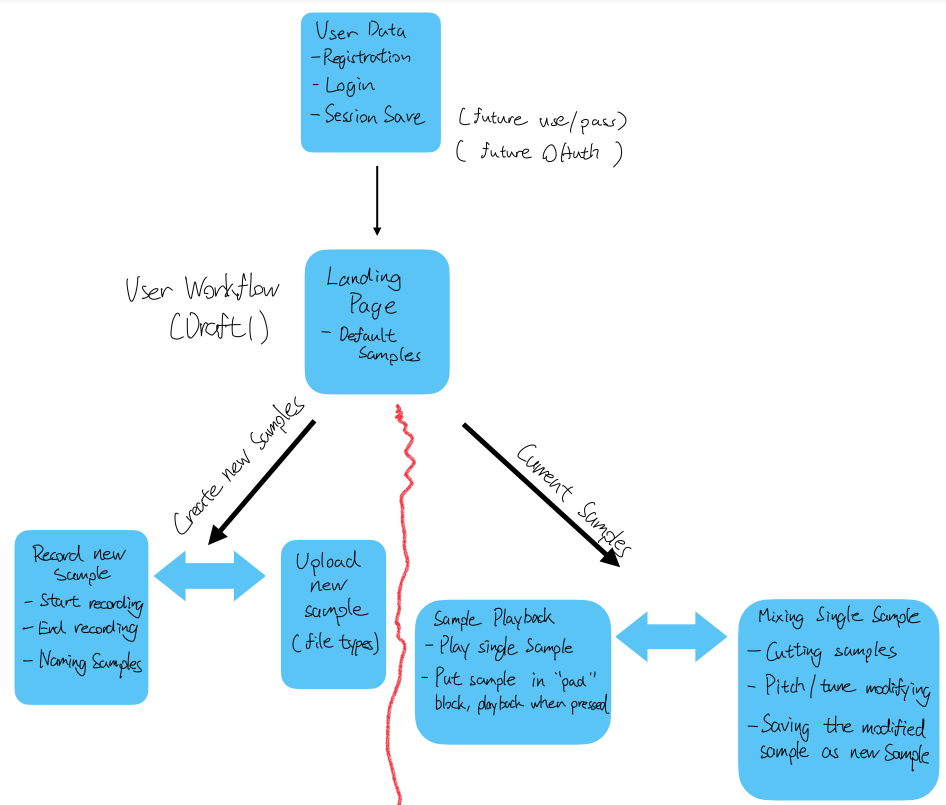
Security Requirements

* Access control mechanisms should be implemented to limit user access to the system's resources, such as sound files, database, and software functionalities. This includes user authentication, authorization, and privilege management to ensure that only authorized users can upload, download, and modify the content and that users are only given access to the resources they need.
* Our system must be able to allow users to securely upload files. To prevent malicious files from being uploaded to the system, file uploads should be checked for size, format, and content type. Additionally, user input should be sanitized and validated to prevent SQL injection, cross-site scripting, and other injection attacks.
* To protect users' privacy and confidentiality, data transmission between the user's browser and the server should be encrypted using HTTPS. Additionally, sensitive data, such as user passwords, should be stored using strong encryption algorithms.
* To prevent DoS attacks that can overwhelm the server, measures should be implemented to detect and mitigate such attacks, such as rate limiting, request throttling, and blacklisting suspicious IP addresses.
* To ensure that data is not lost in the event of a system failure or cyber attack, regular backups should be made of the system's data, and a disaster recovery plan should be in place.

System Requirements

* Cloud Services: The server for the digital sampler will be hosted on a virtual machine. Additionally, cloud storage will be used to store user projects and accounts.
* Modern Browser: Audio processing for our digital sampler will be done client side. This will require a modern browser with a recent version of JavaScript.
* Audio Library: JavaScript audio libraries such as tonejs.github.io and scribbletune will be used to manipulate user audio.

Specification

Here is a typical user workflow chart with features we currently plan to implement:

Here is a basic relationship table of the data we plan to store in our backend:

