trackstudio  
A recording and mixing web application

|  |  |
| --- | --- |
|  | Glen Anderson  Dominic Cabral  Jose Flores  Ramon Meza |
|  | GUI Programming II Professor Heines |
|  | Project Proposal 3.0.0 February 18, 2016 |

# Table of contents

Table of contents 2

Goal statement 3

Target audience 3

Features 3

Not-logged-in and logged-in users 3

Tutorial 4

Recorder/mixer 4

FX catalog 7

Mix menu 9

Future features 10

Components 10

Foreseeable issues 11

Development 11

Experience 11

MEAN Stack 11

Web Audio API 11

Experience 11

Browser compatibility 11

SSL certification 11

Microphone Access 12

Release functionality requirements 12

Project management 12

# Goal statement

Our goal is to create a simplified sound recording application that works in modern web browsers through the Web Audio API. We will differentiate our application by presenting the user with "creative constraints". This will help them focus on their music rather than the technicalities associated with other sound recording applications. We will achieve this by not including visual waveforms and limiting the amount of recordable tracks to four, reducing the interface and indirectly setting a goal for the artist.

# Target audience

Our target audience consists of musicians and songwriters, of both genders ages 13 and older, who like to record demos or music without having to download software. The user must have access to a modern web browser and a microphone. Our target audience is able to navigate a website and is familiar with how knobs, buttons, and sliders function. Users do not necessarily have prior sound recording experience, as a tutorial is provided.

# Features

## Not-logged-in and logged-in users

The application will be able to manage two levels of users: not-logged-in and logged-in. The application user types are described in Table 1.

Table 1: User account differences

| Table 1: User account differences (continued) | | |
| --- | --- | --- |
| User type | Description | Functionality |
| Not-logged-in | A user is by default a not-logged-in user. A not-logged-in user is unknown to the application and in turn will receive a limited set of functionality. | * Able to register * Sign in * Mixer access * Create new work |
| Logged-in | A logged-in user is known to the application. They will have the functionality of a not-logged-in user plus seven additional functionalities. To become a logged-in user, a not- logged-in user will need to register once with the application or log in if they have registered in the past. | * Functionality of an not- logged-in user * Storage space * Save new work * Clone saved work * Open saved work * Export saved work * Inspect saved work |

## Tutorial

Due to the inexperience a user might have with technology found in our application, a user will have the option of going through a tutorial. This tutorial will be accessible to all users at any time from the menu. This menu option, though initially hidden, will be presented to a logged-in user through their first logged-in session. A logged-in user will experience a greeting during this session that will give them an option to run the tutorial. If a user chooses to not participate, they will exit and they will not be prompted again during future sessions. The tutorials scope will be the interface, application usage, and how to manage work.

## Recorder/mixer

The recorder/mixer is the heart of the application, containing the four mono tracks and one stereo master track that a user will be interacting with. A proposed recorder/mixer layout is presented in Figure 1.



Figure 1: Proposed recorder/mixer layout

Each mono track, labeled as Tracks 1 - 4 in Figure 1, contain components described in Table 2.

Table 2: Mono track components

| Table 2: Mono track components (continued) | | | |
| --- | --- | --- | --- |
| Component name | Label | Quantity | Description |
| Track title | Track # | 1 | Can be modified by the user |
| Panning knob | PAN | 1 | Adjusts the stereo location of the track |
| Equalization knobs | EQ | 3 | Adjusts the amplitude of treble, mid, and bass frequency ranges |
| Mute button | M | 1 | Silences the track |
| Record button | ○ | 1 | Engages the recording sequence on the track |
| Solo button | S | 1 | Silences all other tracks |
| Fader slider |  | 1 | Adjusts the volume of the track |
| Volume indicator |  | 1 | Audio peak indicator |
| Effects Slot | FX | 1 | Container that can be occupied by an effect from the FX catalog |

The stereo master track, labeled as Master in Figure 1, contains the components described in Table 3.

Table 3: Stereo master track components

| Table 3: Stereo master track components (continued) | | | |
| --- | --- | --- | --- |
| Component name | Label | Quantity | Description |
| Master title | Master | 1 | Cannot be modified by the user |
| Location indicator |  | 1 | Indicates position in song using the format:  Minutes: Seconds. Milliseconds |
| Stop button | ██ | 1 | Stops the selected track |
| Play/pause button | ►║ | 1 | Plays the selected track |
| Rewind button | ◄◄ | 1 | Rewinds the selected track |
| Fast forward button | ►► | 1 | Fast forwards the selected track |
| Fader slider |  | 1 | Controls the sum of all four mono tracks |
| Volume indicator |  | 2 | Audio peak indicator. There is one each for the left and right audio channels. | |
| Save button | SAVE | 1 | Allows a user to save their mix |

## FX catalog

The user will be able to use an effects catalog, named FX catalog, to choose effects for each of their tracks. These effects will be self-contained modules, as all controls that are needed to modify the effect will be present on the module’s graphical interface. The proposed FX catalog is presented in Figure 2.

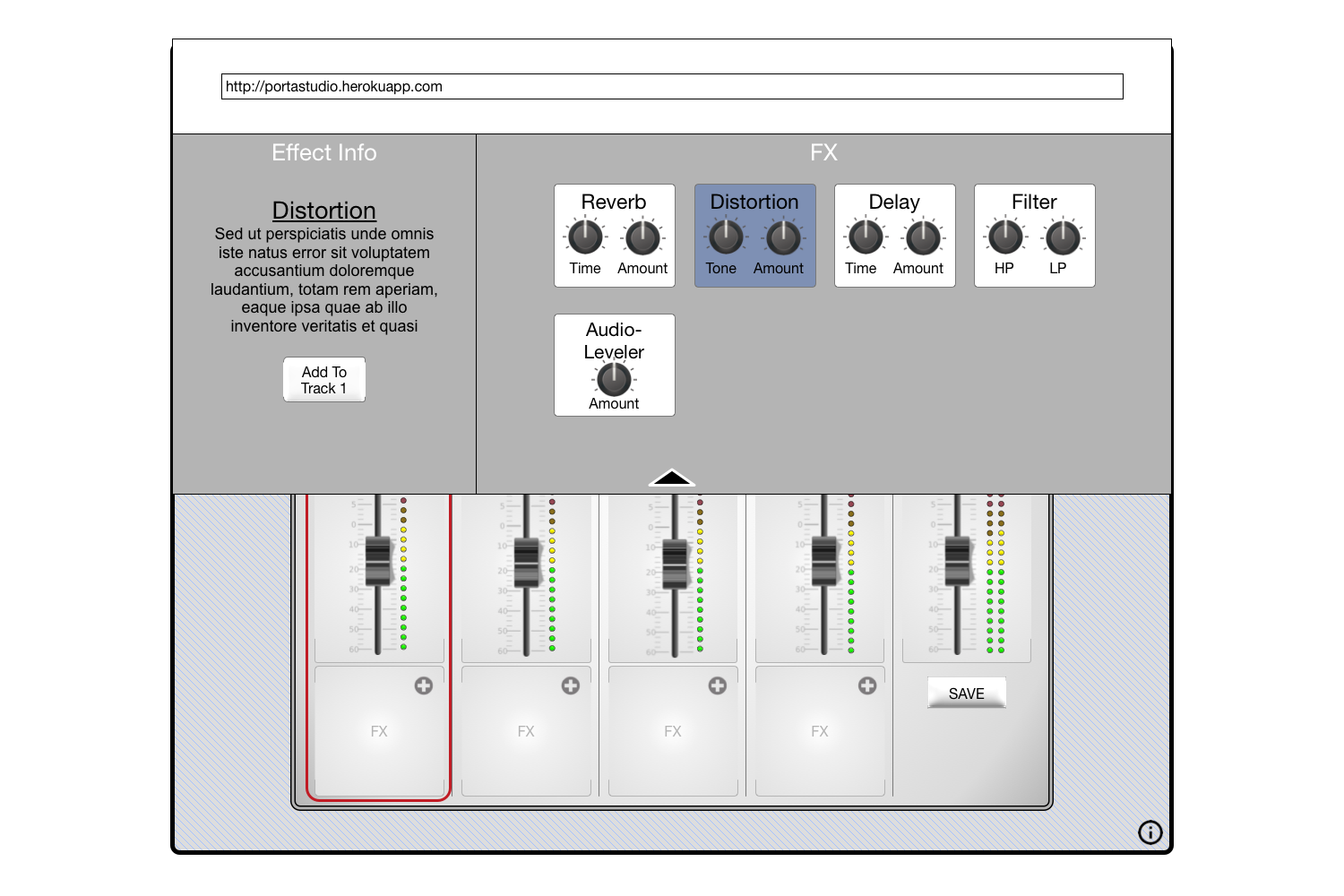


Figure 2: FX catalog

The user interface for the effects catalog is split into two panes. The left pane will hold the description of that effect as well as a selection button. The right pane will hold a list of effects, described in Table 4.

Table 4: List of effects

| Table 4: List of effects (continued) | | |
| --- | --- | --- |
| Effect name | Description | Example usage |
| Reverb | Reflects the input signal until it decays | To simulate space or room |
| Distortion | Alters the input signal in the harmonic (tone, timbre) domain | Simulate the sound of a rock guitar |
| Delay | Holds an input signal to an audio storage medium, and then plays it back after a period of time | To create the sound of a repeating, decaying echo |
| Filter | Remove sections of the audio frequency spectrum | To muffle sounds |
| Audio-Leveler | Reduces the volume level of an input signal if it exceeds a certain value | Automatic volume control |

## Mix menu

A logged-in-user user who would like to manage their work will do so in the Mix menu. This menu will be their storage space interface and will allow an logged-in user to inspect, create, delete, clone, open, and export work. The layout of the Mix menu as depicted in Figure 3 will be similar to that of the FX catalog in Figure 2. The Mix menu will be split into two panes, the left pane will display information about the selected work and have buttons for the actions that can be performed on them. The right pane will hold a list of all saved work, as well as a create button for new work.

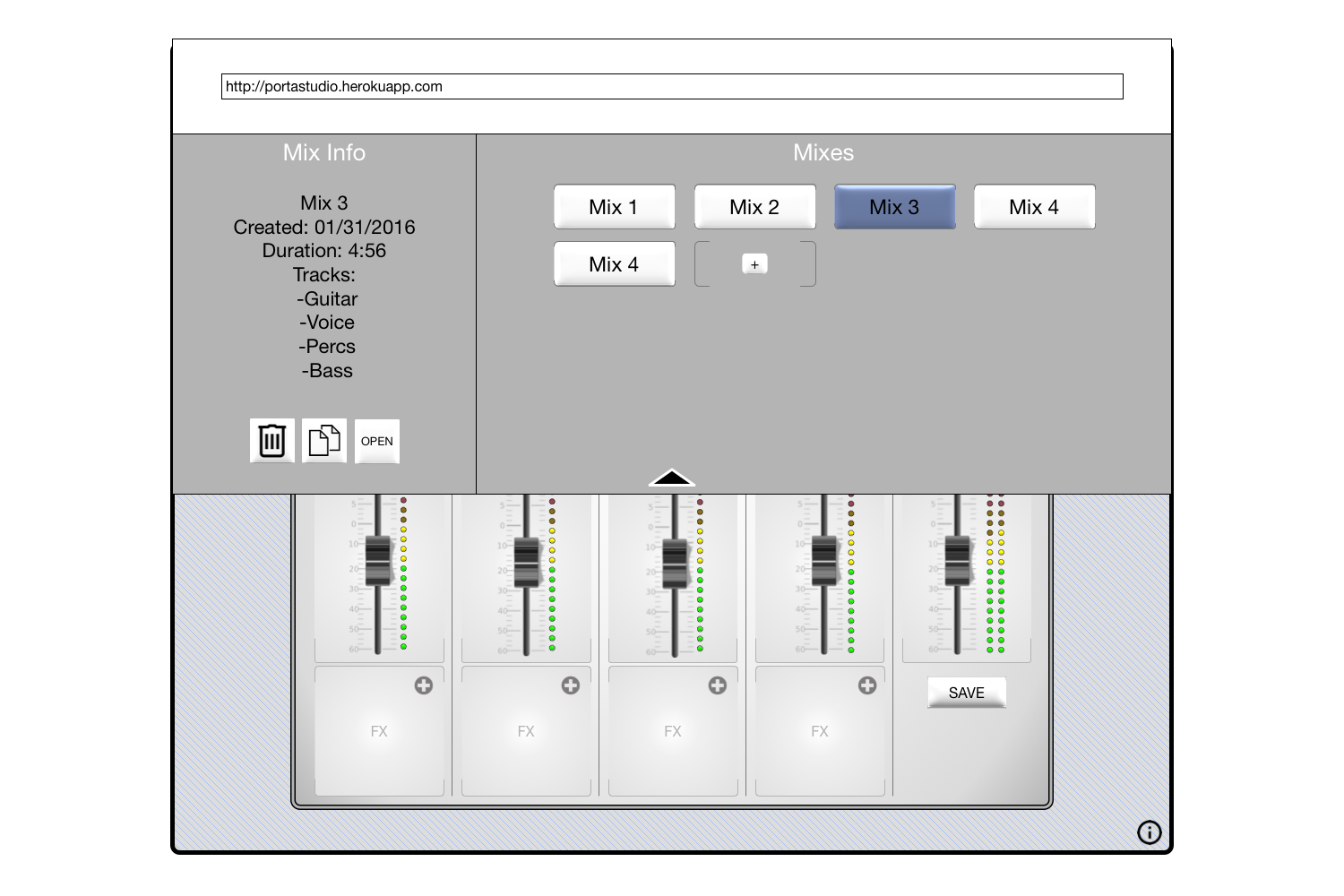


Figure 3: Mix menu

## Future features

If the application is complete and time allows, we will implement the additional features listed in Table 5.

Table 5: Future features list

|  |  |
| --- | --- |
| Table 5: Future features list (continued) | |
| Feature title | Description |
| Export | Will allow logged-in users to export saved or ongoing work directly to music sharing applications, such as [SoundCloud](http://www.SoundCloud.com) |
| Sign up expansion | Will allow not-logged-in and logged-in users to register and sign in respectively with external website OAuth methods, such as those provided by Twitter, Facebook, and Google |

# Components

Most of the components that we will implement exist in some form and will be developed using the external component dependency list found in Table 6 that will be integrated into our project.

Table 6: Dependency list

| Table 6: Dependency list (continued) | |
| --- | --- |
| Name | Usage |
| AngularJS | JavaScript client side framework |
| Bootstrap | CSS framework, to keep our project responsive across devices |
| Express | Server framework |
| jQuery | Keeps our work clean and readable |
| MongoDB | User account data storage |
| NodeJS | Server side JavaScript engine |
| Web Audio API | Handles sound processing |

# Foreseeable issues

## Development

### Experience

Not everyone has experience with music recording equipment and this became apparent during our first scrum meeting. Inexperienced developers will therefore need to put more effort in to learn about the application concepts, such as signal flow, before actually integrating components.

### MEAN Stack

From the default Jade examples given in the textbook, we have learned how to extend templates with partial templates, but we have not learned how to include partials as sub-components of larger templates. This is crucial in developing our larger components without having to include redundant files and code.

## Web Audio API

### Experience

Web Audio API is new to all of the developers and getting started with the API is going to be a challenge. Thankfully there are tutorials (http://code.tutsplus.com/tutorials/the-web-audio-api-what-is-it--cms-23735) and documentation available (https://developer.mozilla.org/en-US/docs/Web/API/Web\_Audio\_API).

### Browser compatibility

The API will also give us difficulty due to its browser compatibility, due specifically to microphone access. Users will only be able to use one of the browsers listed in Table 7.

Table 7: Compatible browsers

| Table 7: Compatible browsers (continued) | |
| --- | --- |
| Browser | Compatible version |
| Edge | All |
| Chrome | 48+ |
| Opera | 34+ |
| Firefox | 44+ |

### SSL certification

To use a microphone on a non-localhost such as Heroku, we will need SSL certification. This can be done through Heroku (https://devcenter.heroku.com/articles/ssl-endpoint#acquire-ssl-certificate). Another option is to go through the Computer Science department specifically Ken Kleiner, who has requested SSL certificates for other projects in the past such as Dr. Yu Cao’s CSR project that was hosted on Amazon Web Services through https://csr.cs.uml.edu.

### Microphone Access

For a browser to capture audio from the user, access must be granted. In most cases when the user will first reach the site they will be prompted with the option to either “Allow” or “Block”. It is important that the user selects “Allow” to access the full functionality of our application. Our first attempt to ensure this will be to have a modal appear that will explain to the user why to allow access to their microphone, when they are prompted to do so. Our second attempt, in case the user has blocked access, will be to re-direct the user to a different page with step-by-step instructions on how to allow microphone access to our site.

# Release functionality requirements

The following list specifies the minimum required functionality for a success release of trackstudio.

* Mixer
  + Recording
  + Volume control (each track and master)
  + Transport functionality
    - Play/Pause
    - Stop
    - Forward
    - Reverse
    - Time Indicator
  + Track Panning
  + Track Equalization
  + Track Muting and Soloing
  + FX Slots for each track
* FX Catalog
  + Effect Information
  + Effects
    - Reverb
    - Delay

# Project management

We’ve decided to take an Agile approach to developing our application. We believe daily communication, weekly meetings, and an integrated work management system will allow us to work together efficiently. Our management tools are listed in Table 8 and our schedule in Table 9.

Table 8: Project management tools

| Table 7: Project management tools (continued) | |
| --- | --- |
| Name | Description |
| Heroku | Our deployment server, set to deploy when our *heroku* GitHub branch is pushed to |
| GitHub | Our source control |
| Slack | Used for daily team communication and application status updates |
| Trello | Our task board, all tasks and milestones are being tracked here |

Table 9: Development schedule

| Table 8: Development schedule (continued) | | | | |
| --- | --- | --- | --- | --- |
| Milestone | Story | Date | Task description | Owner |
| Project setup | Slack | 01/31 | Setup Account | Cabral |
| 01/31 | Integrate Heroku | Flores |
| 01/31 | Integrate GitHub |
| 01/31 | Integrate Trello |
| Heroku | 01/31 | Setup domain |
| 03/08 | Process CSR for SSL certification |
| GitHub | 01/31 | Setup organization |
| 01/31 | Make *master* branch |
| 01/31 | Make *heroku* branch |
| Trello | 01/31 | Setup deadlines board |
| Alpha | Recorder/mixer | 02/19 | Editable track labels | Meza |
| 02/19 | Knobs |
| 02/19 | Buttons |
| 02/26 | Fader slider |
| 02/26 | Volume indicator |
| 02/26 | Effects slot |
| 03/04 | Location indicator |
| 03/04 | Component frame |
| Web Audio API | 02/19 | Mute | Anderson |
| 02/19 | Solo |
| 02/26 | Pan |
| 03/04 | EQ |
| 02/19 | Record | Flores |
| 02/19 | Volume, mono and stereo |
| 02/26 | Location indicator |
| 03/04 | Peak visualizer |
| 02/19 | Input and output processing | Cabral |
| 02/19 | Effects |
| 02/26 | Playback |
| 03/04 | Exporting |
| Graphics | 02/19 | Recorder/mixer | All |
| 03/04 | Dropdown partials |
| Beta | Tutorial | 03/18 | Tooltip | Meza |
| 03/25 | Overlay |
| 04/01 | Script |
| Dropdown | 03/18 | Sign in/ sign up | Anderson |
| 03/25 | About |
| 04/01 | Account info |
| 03/18 | FX catalog | Cabral |
| 03/25 | FX modules |
| 04/01 | FX info |
| 03/18 | Mixes | Flores |
| 03/25 | Mix info |
| 04/01 | Mix modules |
| Usability Test and Feedback Review | Test | 04/05 | Questionnaire | All |
| 04/05 | Task list |
| 04/05 | Feedback survey |
| Feedback | 04/14 | Bug fixes | All |
| 04/14 | Usability fixes |
| Class presentation | Presentation | 04/19 | Presentation slides | All |
| 04/19 | Live demo |
| Final submission | Submission | 04/28 | Bug fixes | All |
| 04/28 | Future feature implementations |