WebTracks  
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 4, 2016 |

# Table of contents

Table of contents 2

Goal statement 3

Target audience 3

Features 3

Anonymous, registered, and authenticated users 3

Tutorial 4

Recorder/mixer 4

FX catalog 6

Mix menu 8

Future features 9

Components 9

Foreseeable issues 10

Experience 10

Web Audio API 10

Release functionality requirements 10

Project management 10

# Goal statement

Our goal is to create a web application that will allow users to record and manipulate audio in a simple and accessible way.

# Target audience

Our target audience consists of people who are interested in recording music. Our primary focus will be to target more experienced users with the functionality provided by the application, without alienating beginners.

# Features

## Anonymous, registered, and authenticated users

The application will be able to manage three different kinds of users: anonymous, registered, and authenticated. The application user types are described in Table 1.

Table : User account differences

| Table 1: User account differences (continued) | | | |
| --- | --- | --- | --- |
| User type | Description | Capabilities | |
| Anonymous user | A user is by default an anonymous user, an anonymous user is unknown to the application and in turn will receive a limited set of functionality. | * Able to register * Mixer access * Create new work | |
| Registered user | A registered user is known to the application but they are not signed in, they will have the functionality of an anonymous user. To become a registered user, an anonymous user will need to register once with the application. | * Sign in * Mixer access * Storage space * Create new work | |
| Authenticated user | An authenticated user is a registered user that has logged in during their session, the application gives them the most functionality. To become an authenticated user, a registered user must sign in during their sessions. | * Sign out * Mixer access * Storage space * Create new work * 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 within the menu, will be presented to an authenticated user through their first authenticated session. An authenticated 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 : Proposed recorder/mixer layout

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

Table : 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 : 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.

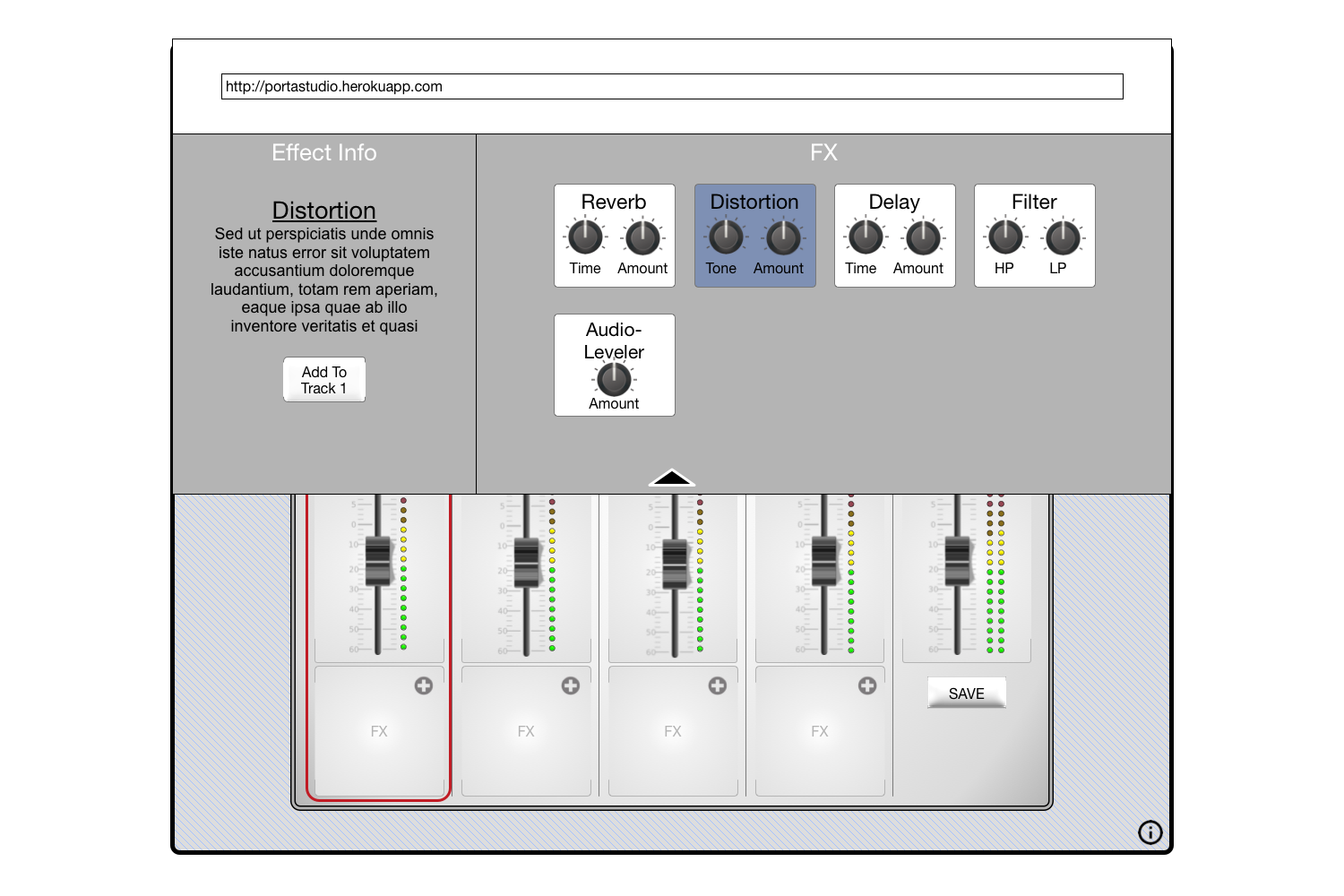


Figure : FX catalog

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 : 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

An authenticated user that 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 authenticated 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.

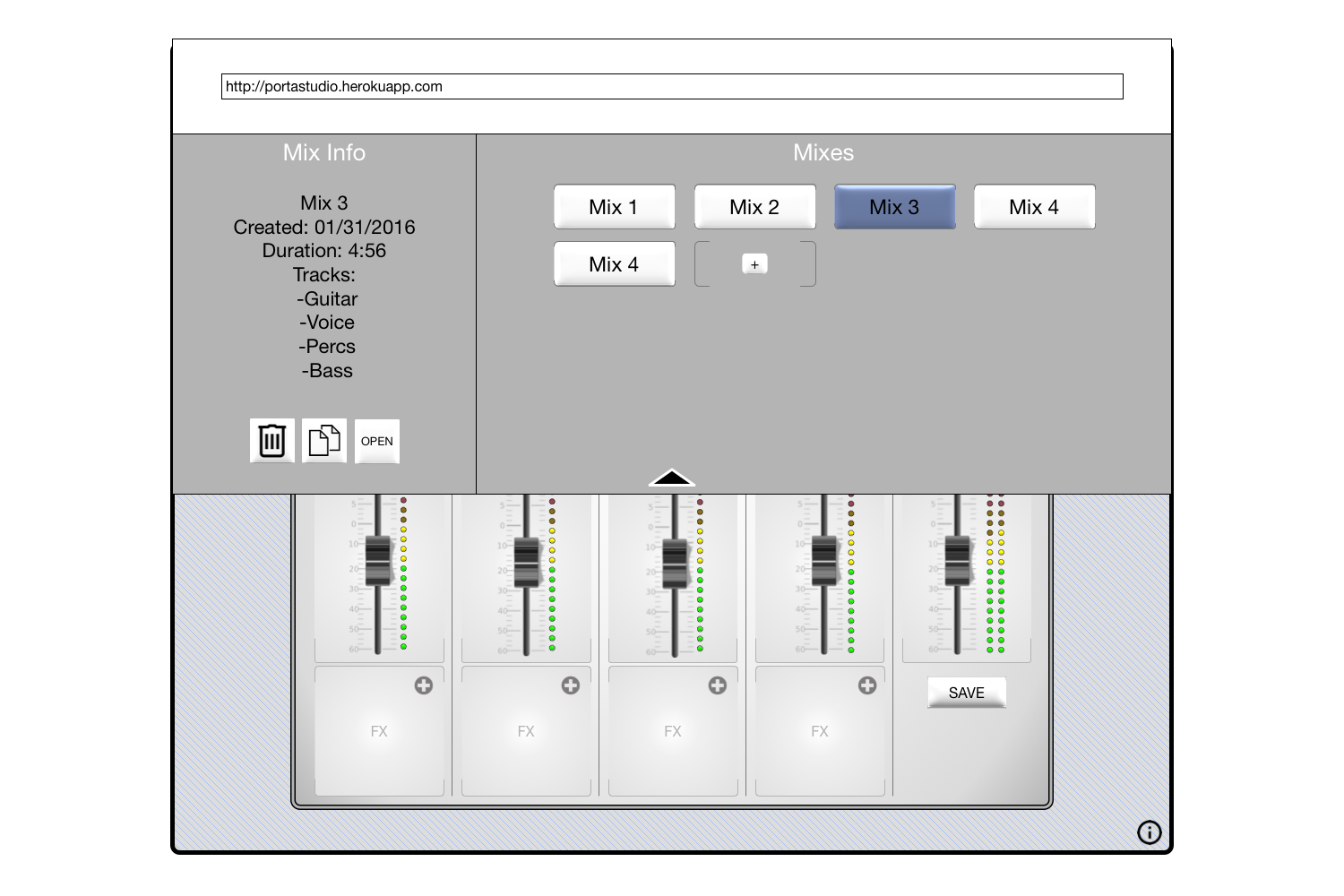


Figure : Mix menu

## Future features

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

Table : Future features list

|  |  |
| --- | --- |
| Table 5: Future features list (continued) | |
| Feature title | Description |
| Export | Will allow authenticated users to export saved or ongoing work directly to music sharing applications, such as [SoundCloud](http://www.SoundCloud.com) |
| Sign up expansion | Will allow anonymous and registered 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 : 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

## 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 before actually integrating components.

## Web Audio API

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).

# Release functionality requirements

Our product will require a decent amount of functionality to be implemented for us to consider it a successful release. The one feature we absolutely need to implement is the recorder/mixer. The recorder/mixer needs to allow the user to record audio from their microphone, manipulate it in some way, and to export the audio so the user can download it to their machine. If we can accomplish this, then we have met our requirements of a functional release.

# 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 7 and our schedule in Table 8.

Table : 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 : 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 |
| GitHub | 01/31 | Setup organization |
| 01/31 | Make *master* branch |
| 01/31 | Make *heroku* branch |
| Trello | 01/31 | Setup deadlines board |
| Alpha | Recorder/mixer | 03/08 | Editable track labels | Meza |
| 03/08 | Knobs |
| 03/08 | Buttons |
| 03/08 | Fader slider |
| 03/08 | Volume indicator |
| 03/08 | Effects slot |
| 03/08 | Location indicator |
| 03/08 | Component frame |
| Web Audio API | 03/08 | Pan | Anderson |
| 03/08 | EQ |
| 03/08 | Mute |
| 03/08 | Solo |
| 03/08 | Record | Flores |
| 03/08 | Volume, mono and stereo |
| 03/08 | Location indicator |
| 03/08 | Peak visualizer |
| 03/08 | Playback | Cabral |
| 03/08 | Input and output processing |
| 03/08 | Effects |
| 03/08 | Exporting |
| Graphics | 03/08 | Recorder/mixer | All |
| 03/08 | Dropdown partials |
| Beta | Tutorial | 04/05 | Tooltip | Meza |
| 04/05 | Overlay |
| 04/05 | Script |
| Dropdown | 04/05 | Sign in/ sign up | Anderson |
| 04/05 | About |
| 04/05 | Account info |
| 04/05 | FX catalog | Cabral |
| 04/05 | FX modules |
| 04/05 | FX info |
| 04/05 | Mixes | Flores |
| 04/05 | Mix info |
| 04/05 | 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 |

.