



# Software Requirements Specification

## for Sound Recorder

Version <1.0>

Prepared by

**Group Name:** Tune Deaf Interactive

Shawn Hillstrom  
Brad Hendrickson  
James Keirnan

11544366  
11515344  
11505377

shawn.hillstrom@wsu.edu  
bradley.hendrickson@wsu.edu  
james.keirnan@wsu.edu

**Date:** October 7<sup>th</sup>, 2018

<b>REVISIONS .....</b>	<b>III</b>
<b>1 INTRODUCTION.....</b>	<b>3-4</b>
1.1 DOCUMENT PURPOSE .....	3
1.2 PRODUCT SCOPE .....	3
1.3 INTENDED AUDIENCE AND DOCUMENT OVERVIEW.....	3
1.4 DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	4
1.5 DOCUMENT CONVENTIONS .....	4
1.6 REFERENCES AND ACKNOWLEDGMENTS .....	4
<b>2 OVERALL DESCRIPTION .....</b>	<b>5-6</b>
2.1 PRODUCT PERSPECTIVE .....	5
2.2 PRODUCT FUNCTIONALITY .....	5
2.3 USERS AND CHARACTERISTICS .....	5
2.4 OPERATING ENVIRONMENT.....	6
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS.....	6
2.6 USER DOCUMENTATION.....	6
2.7 ASSUMPTIONS AND DEPENDENCIES .....	6
<b>3 SPECIFIC REQUIREMENTS .....</b>	<b>7-9</b>
3.1 EXTERNAL INTERFACE REQUIREMENTS .....	7-8
3.2 FUNCTIONAL REQUIREMENTS .....	8
3.3 BEHAVIOUR REQUIREMENTS .....	8-9
<b>4 OTHER NON-FUNCTIONAL REQUIREMENTS .....</b>	<b>10</b>
4.1 PERFORMANCE REQUIREMENTS.....	10
4.2 SAFETY AND SECURITY REQUIREMENTS.....	10
4.3 SOFTWARE QUALITY ATTRIBUTES .....	10
<b>5 OTHER REQUIREMENTS .....</b>	<b>11</b>
<b>APPENDIX A – DATA DICTIONARY .....</b>	<b>11</b>
<b>APPENDIX B - GROUP LOG.....</b>	<b>11</b>

## Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft 1.0	Shawn Hillstrom, Bradley Hendrickson, James Keirnan	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.  Initial version.	10/07/18

## 1 Introduction

Creating audio content should be easy and free. A tool which allows for audio to be recorded, edited, and mixed seamlessly on the internet is not readily available. For this reason, Tune Deaf Interactive is teaming up to develop Sound Recorder 1.0, as described ahead.

### 1.1 Document Purpose

Sound Recorder is an online audio editing and mixing tool. This document covers the goal of this project which is to introduce some rudimentary features. Said features (for version 1.0) will include the ability to record tracks directly on the Sound Recorder site via audio input and the ability to import additional audio files and export projects into a format of the user's choosing. Common audio file formats will be supported (e.g. mp3, wav, etc.).

This document will describe each of the products pieces of functionality, and the constraints under which the software is to be made.

### 1.2 Product Scope

There are several audio editing and mixing applications on the market; however, many of them are extremely complicated and expensive. Additionally, these applications carry many tools that the average user would not want nor need for his or her audio project. This project seeks to develop a free, web-based alternative to the popular applications available on the market which can serve the average audio creator.

The product described here will be available on all browsers and will implement basic multi-track editing and mixing with support for importing and exporting common audio file formats. The goal is to create a simple, lightweight, and, most importantly, free tool for the average music creator.

### 1.3 Intended Audience and Document Overview

This document is intended for the client(s) of Sound Recorder and the professor of Fundamentals of Software Engineering. It covers the basic functionality and specifications for Sound Recorder. Section 2 provides a general overview of the product including basic functionality, anticipated user demographic, constraints and dependencies, and planned documentation. Section 3 introduces more specific functional requirements and section 4 specifies non-functional requirements.

## **1.4 Definitions, Acronyms and Abbreviations**

This is not needed for the scope of Sound Recorder. No special definitions are required, as all terms used are common knowledge.

## **1.5 Document Conventions**

This is not needed. No special formatting is used in the document. IEEE formatting requirements are the guidelines associated with the style of this SRS.

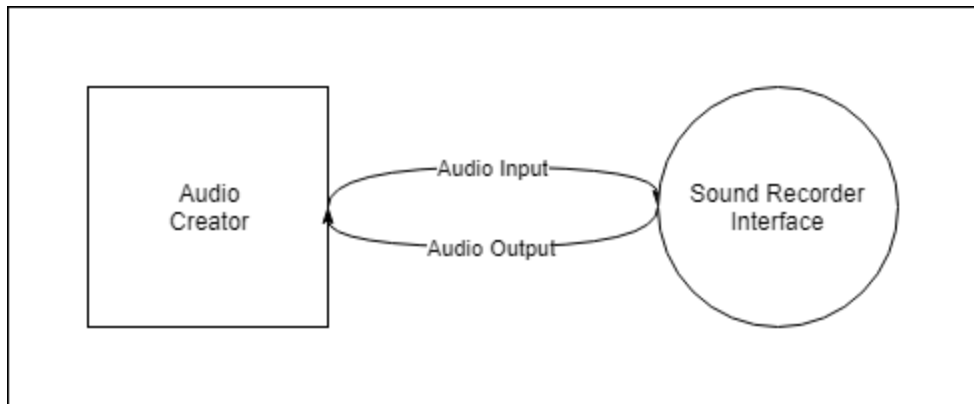
## **1.6 References and Acknowledgments**

This is not needed. No external sources are used in the document.

## 2 Overall Description

### 2.1 Product Perspective

Sound Recorder is a standalone software project designed as an alternative to more complex and costly music editing software. It is not a component of a system or a member of a product family; however, it may include expanded functionality eventually. The product will be intended to be accessible for everyone, so it will be designed to be as easy to use as possible. For this reason, the goal of this specific project is to implement simple functionality.



### 2.2 Product Functionality

- Create a new audio project with support for multiple audio files. Each project should include 0-10 of channels and a timeline.
- Create new channels within a project
- Delete existing channels.
- Import audio (if empty channel exists, a popup window appears where a file is selected and automatically imported into a new channel).
- Audio Input Device Select (button click, possibly dropdown or popup window).
- Record audio files given an audio input directly into a specified channel and display on the timeline.
- Crop and move audio files along the timeline (shown fixed above the channels, starting at 0:00:00).
- Control the relative volume of each channel via a slider per channel.
- Export audio projects (popup window, file type, location, and name selection).

Note: At this time, users will not be able to save multiple projects directly on the site. Projects must be exported to be saved locally to the user's machine. Additionally, users will not be able to import old projects.

### 2.3 Users and Characteristics

Sound Recorder is designed to be used by audio creators ranging from amateurs for basic audio manipulation, to accomplished audio creators who wish to have a sketchpad to try out new ideas. Since the platform is intended to be free and as easy to use as possible, it should be able to be used by anyone (assuming access to the software). Because of this, there are little to no assumed

user characteristics. It is anticipated that amateurs will make up a greater percentage of users than more experienced audio creators. Sound Recorder's simple and limited set of tools is more attractive to amateur users looking to try out new things without having to pay for more complicated systems. This also makes Sound Recorder less attractive to more advanced users who have access to more premium paid products, as it may lack some of the functionality they require for more detailed projects. Because of these considerations, amateur audio creators are the main user focus for this project.

## **2.4 Operating Environment**

Sound Recorder is designed to run locally on any user's browser. The software functionality will be implemented using JavaScript. The only requirement is that a user has a system which can run any one of the popular web browsers. These include Google Chrome, Firefox, and Internet Explorer.

## **2.5 Design and Implementation Constraints**

Constraints include:

- Audio file formats.
- Number of channels available in a project (likely 0-10).
- Length of project timeline (likely up to 1 hour).
- Audio file import size.
- Audio project export size.
- Size available for export storage.
- System responsiveness (mitigated delay between inputs like record/pause and their actions occurring).

## **2.6 User Documentation**

Because of Sound Recorder's simple and mobile design, very limited user documentation is required. Planned user documentation includes a brief tutorial page on site explaining the various functionalities of Sound Recorder.

## **2.7 Assumptions and Dependencies**

- An external library for audio manipulation will be used in the implementation (this library has yet to be identified but will be specified in future versions of this document).
- An external library for graphical user interfaces will be used in the implementation (this library has yet to be identified by will be specified in future versions of this document).

## 3 Specific Requirements

### 3.1 External Interface Requirements

#### 3.1.1 User Interfaces

Two screens will be available to see:

- 1) Landing page with information about the product, how to use it, etc. Will have:
  - Button to navigate to workspace page (labelled “Go”)
- 2) Timeline (workspace) page with:
  - 0-10 rows for channels spanning the entire screen width, each about 1/5 the height of the available screen space (with a scroll bar if there are more than 5). These rows will each contain up to 1 audio track, and are able to be moved left and right along the timeline (displayed as a fixed scale at the top of the screen above the tracks).
    - Each row will have a vertical volume slider on the left, automatically set at 100%
  - A row of buttons centered at the bottom of the screen
    - Record (button click, which creates new audio track and records audio into selected channel on screen to be seen)
    - Stop (button click, stops playback/recording, setting needle back at the start of the timeline)
    - Play/pause (button click, starts playback from wherever the needle is – second press pauses, freezing the needle)
    - Trim selected track (button click, then user selects a track or channel, then drag left/right ends of track before confirming trim – trim button will read “Confirm?” once a trim is initiated, and once the user clicks away from the trim it will be reset)
    - Delete selected track (works similar to trim, but deletes)
  - Other functionality (also buttons within bottom banner):
    - Import audio (button click, followed by a popup window where a file is selected and automatically imported into a new or selected channel)
    - Export audio (button click, popup window to select file location, and file type selection – will export all existing audio on timeline into file)
    - Audio Input Device Select (button click, possibly dropdown or popup window where the user selects which device to record with)

#### 3.1.2 Hardware Interfaces

- Audio import/export will interface with hardware in the form of converting audio files on the local machine to usable/manipulatable channels in the software, and the opposite process of flattening all the channels into a single audio file and saving it to the local machine.
- Audio recording will involve a direct interface between the local machine’s audio recording devices (microphone) and the software.

#### 3.1.3 Software Interfaces

Sound Recorder will interface with the operating system by connecting an audio input device (via a JavaScript library) to the software. Similarly, the operating system will oversee the import/export of audio files to/from the software. Selecting files to import and determining file location, name, and type will all involve communication with the OS.

### **3.1.4 Communications Interfaces**

The only data communications occurring are between the software and recording devices. Since this process is completely local (doesn't require online communication or additional processing) the JavaScript libraries used to record, and import/export audio will be the only interfaces between hardware and software.

## **3.2 Functional Requirements**

Create a project.

- Initialize the timeline.
- Initialize one channel (empty).

Edit a project.

- Add channels.
- Delete channels.
- Add audio track to a selected channel.
  - Record an audio track to a channel.
  - Import an external audio track to a channel.
- Edit channels.
  - Crop audio tracks.
  - Move audio tracks along the timeline.
  - Change volume of each channel independently.

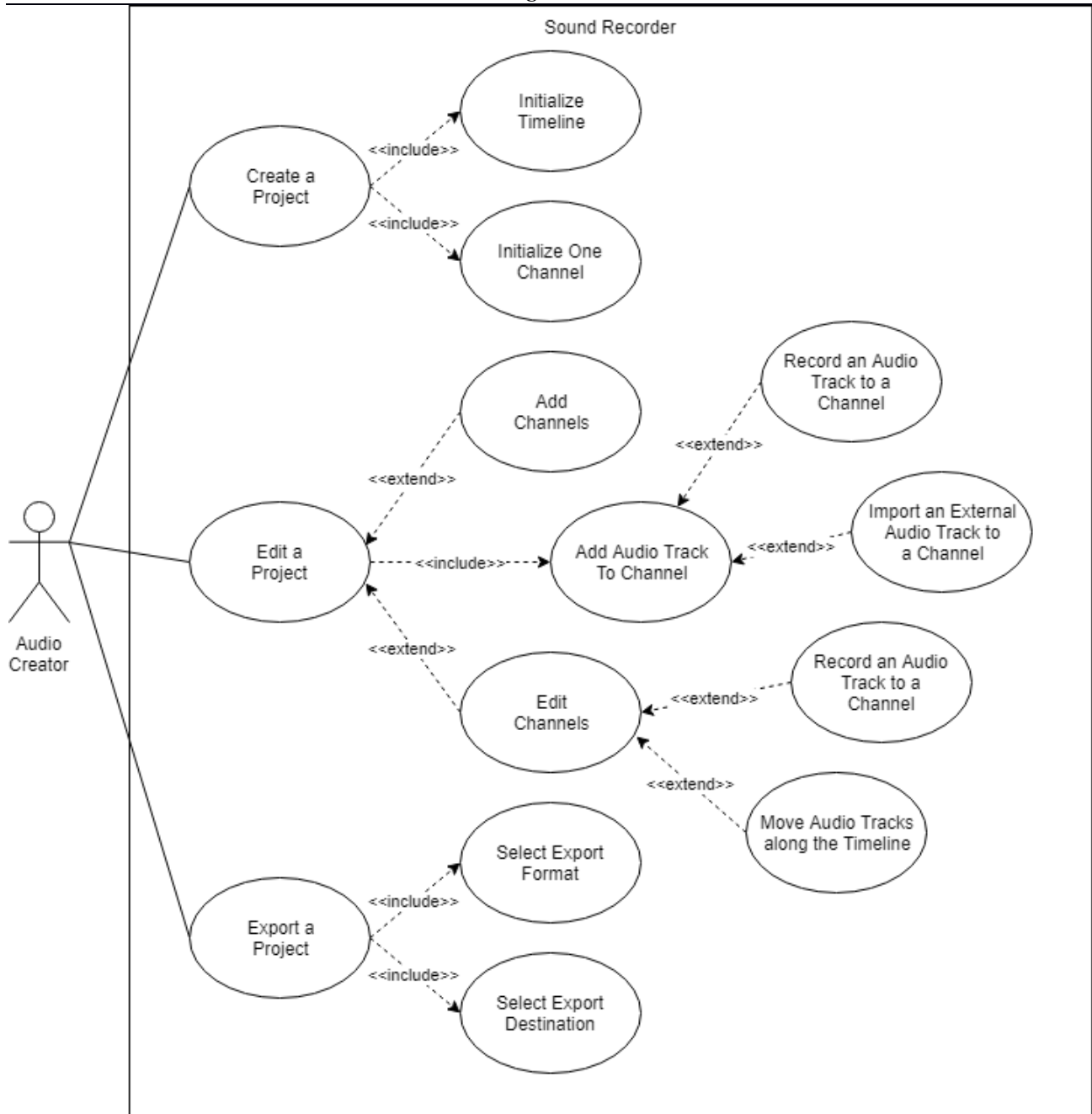
Export a project.

- Select export format.
- Select export destination.

## **3.3 Behaviour Requirements**

### **3.3.1 Use Case View**





## **4 Other Non-functional Requirements**

### **4.1 Performance Requirements**

The software should not take over a minute to render a finished track. There should not be excessive latency in track adjustment (movement along timeline, trim, delete, record, play/pause).

### **4.2 Safety and Security Requirements**

This Project does not require a great level of security, as no user information is saved on servers. As data is recorded/added it is processed locally In JavaScript and there are no user accounts, so login information, etc. is not used or gathered.

### **4.3 Software Quality Attributes**

Software should be easy to use and understand, and quickly responsive. Its web implementation should make it portable by the nature of web applications. All functionality of the program should be testable via unit tests.

## 5 Other Requirements

*Not required for this project.*

## Appendix A – Data Dictionary

*Not required for this project.*

## Appendix B - Group Log

Date	Activity	Name
10/3	Sections 1-2	Shawn
10/3	Section 2	James
10/4	Section 4	Brad
10/4	Section 3	James
10/7	Sections 2-3 Diagrams	Shawn
10/7	Revisions on Sections 1-4	James
10/7	Section 4, revisions	Brad