TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Thales Santos Ribeiro 166788IVSM

ONLINE RECORDING OF A SPEECH CORPORA

Master's thesis

Supervisor: Einar Meister
Senior researcher,
PhD

Tallinn 2018

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Thales Santos Ribeiro 166788IVSM

VEEBIPÕHINE KÕNEKORPUSTE SALVESTUS

Magistritöö

Juhendaja: Einar Meister
Vanemteadur, PhD

Tallinn 2018

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Thales Santos Ribeiro

07.05.2018

Abstract

Large corpora of variable speech material are necessary for both speech research and speech technology development. This work aims to develop a web-based service for speech recordings. The software utilizes a client-server system in which the server provides both application logic and data storage, and then the client implements a browser-based user interface.

Index Terms: speech recording, web audio API, client-server system.

The thesis is in English and contains 40 pages of text, 5 chapters, 20 figure, 8 tables.

Annotatsioon

Suuremahulised kõnekorpused on vajalikud nii kõneuuringuteks kui ka kõnetehnoloogia arendamiseks. Selle töö eesmärgiks on välja töötada veebipõhine kõnesalvestuste rakendus. Tarkvara kasutab klient-server süsteemi, milles server pakub nii rakenduste loogikat kui ka andmete salvestamist, ja klient kasutab brauseripõhist kasutajaliidest.

Indeksitingimused: kõne salvestamine, veebiaudio API, kliendiserverisüsteem.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 40 leheküljel, 5 peatükki, 20 joonist, 8 tabelit.

List of abbreviations and terms

API	*Application Programming Interface*
ASR	*Automatic Speech Recognition*
CMS	*Content Management System*
DTW	*Dynamic Time Warping*
HTML	*HyperText Markup Language*
JSON	*JavaScript Object Notation*
LXC	*Linux Containers*
MTurk	*Amazon Mechanical Turk*
MVC	*Model-View-Controller*
MVP	*Minimum Viable Product*
RDMS	*Relational Database Management System*
RVM	*Ruby Version Manager*
TTS	*Text to Speech*
URL	*Uniform Resource Locator*

Table of contents

1 Introduction 11
  1.1 Problem Definition 11
  1.2 Aims 11
  1.3 Objectives 12
  1.4 Organization of the thesis 12
2 Background 13
  2.1 Literature Review 15
  2.2 Related Work 16
3 Tools 17
  3.1 Aeneas 17
  3.2 Docker 18
  3.3 Sidekiq 18
  3.4 Ruby on Rails 19
4 Application 21
  4.1 Implemented Features 21
    4.1.1 Manage User 21
    4.1.2 Manage Project 22
    4.1.3 Manage Record 23
    4.1.4 Manage Folder 24
  4.2 Usage 24
    4.2.1 Signup 24
    4.2.2 Authentication 25
    4.2.3 Projects 26
    4.2.4 Add files to a folder 28
    4.2.5 Collect audio 29
    4.2.6 Records 31
  4.3 Installation 33
    4.3.1 Requirements 33
    4.3.2 Linux 34
    4.3.3 MacOS 34
    4.3.4 Windows 35

4.3.5 Starting the application 35
4.3.6 Aeneas 36
4.4 Source-code 36
4.5 Default user 36
5 Summary 37
5.1 Future Work 37
References 38
Appendix 1 – Class Diagram 39
Appendix 2 – Admin application’s dashboard 40

List of figures

Figure 1 Example of an annotation. From top to bottom - signal waveform, spectrogram and fundamental frequency contour (blue), word-level annotation. ... 14
Figure 2 Sidekiq web UI ... 19
Figure 3 Domain Model ... 21
Figure 4 User sign-up form part 1 ... 25
Figure 5 User sign-up form part 2 ... 25
Figure 6 Admin Login UI ... 26
Figure 7 Participant Login UI ... 26
Figure 8 Project page ... 27
Figure 9 Project form part 1 ... 27
Figure 10 Project form part 2 ... 28
Figure 11 Add files to folder ... 29
Figure 12 Files in the folder ... 29
Figure 13 Pop-up window to insert an image file into a project description field ... 29
Figure 14 Example pop-up allow microphone access ... 30
Figure 15 Start/End recording ... 30
Figure 16 Audio collection sequence diagram ... 31
Figure 17 Records page ... 32
Figure 18 Record deletion confirmation ... 32
Figure 19 Record details ... 33
Figure 20 Database configuration file ... 34

List of tables

Table 1 HTML5 audio tag support 14
Table 2 Aeneas Input 17
Table 3 Default Rails directory structure 20
Table 4 Users table 22
Table 5 Projects table 23
Table 6 Records table 24
Table 7 Folder table 24
Table 8 Hardware Requirements 33

1 Introduction

As many people have noticed, we are living in the conversational era due to the possibility to address commands to automate our lives using only our voices, and to use devices to read out loud newspapers, books, or to speak our written words.

1.1 Problem Definition

During the last decades, the use of text to speech (TTS) and automatic speech recognition (ASR) has been increased, this market is expected to reach 18 billion US dollar by 2023 [1], ASR and TTS systems become more popular, and both of them require a significant amount of speech material for their training.

Collecting and managing such high volume of data requires huge effort by many people to capture the audio, process and annotate the speech. For example, a person needs to commute to a recording studio to record his/her speech samples, after concluding it a trained transcriber needs to spend hours and hours to synchronize the audio with a transcript and annotate it.

1.2 Aims

This work aims to develop a web application which enables to collect speech corpora online and thus making it possible to record speech in a home environment. In that way, it is expected to make the recruitment of volunteers easier and increase the number of subjects who want to participate in the speech recordings.

Moreover, this work aims to improve the availability of speech corpora by providing access to speech corpora through the Internet. Thus, any person who holds the rights to access may do it from anywhere around the globe. Next, it aims to reduce the amount of time spent to annotate a speech corpus using the software Aeneas[1] to automatically synchronize the audio with a transcript and annotate it by generating automatically a Praat[2] compatible TextGrid file.

1.3 Objectives

In this thesis, a web-based content management system (CMS) for speech corpora will be developed using HTML5 functionalities that are compatible with the most popular web browsers in order to capture speech without any other software dependency from the client's side. In that way, the volunteers will not need to spend time managing installation of all required software components nor facing troubles with wrong dependencies' version. Also, provided an automatic audio annotation will save hours of a professionally trained transcriber. Moreover, to increase the speech database availability to facilitate its use for training speech recognition system or phonetic studies.

The Laboratory of Language Technology, part of the Institute of Software Science at Tallinn University of Technology plans to use this application for speech collection in Estonia starting in September 2018.

1.4 Organization of the thesis

Background: this section contains the background information and the literature review

Tools: this section states and briefly explains the tools used in this project to achieve the goals.

Application: this section describes the application requirements, usage, and installation.

Summary: this section contains the thesis summary and the future work section as well.

1 https://github.com/readbeyond/aeneas

2 http://www.fon.hum.uva.nl/praat/

2 Background

Humans communicate with each other using speech [2] which is the ability to express thoughts and emotions and transfer information from one person to another in the form of a sound. Speech is a result of the articulatory movements of the mouth, lips, and tongue and it propagates through the air as waves to the listener who receives and decodes the message [3].

The term "speech corpus" refers to a collection of digital recordings of speech signal together with annotations, metadata, and documentation [4]. Speech corpora are the primary source of data for basic and applied research of spoken language communication and for technology development, e.g. in text-to-speech synthesis (TTS), automatic speech recognition (ASR), speaker recognition, etc. [5]. There are two main types of speech corpora: spontaneous and read speech. In the case of spontaneous speech, the audio captured does not follow any script, as an example, a dialog between people, and read speech is recorded following a script, for example, while reading a book or a set of sentences.

As mentioned above a speech corpus involves speech samples together with annotations, and after the signal collection process, it is needed to annotate the utterances. Nowadays, most of the annotations are automatic, yet it is still necessary to perform also manual annotation, for example, in order to fix errors of automatic annotation due to a low-quality signal. The Figure 1 illustrates an example of an annotated audio.

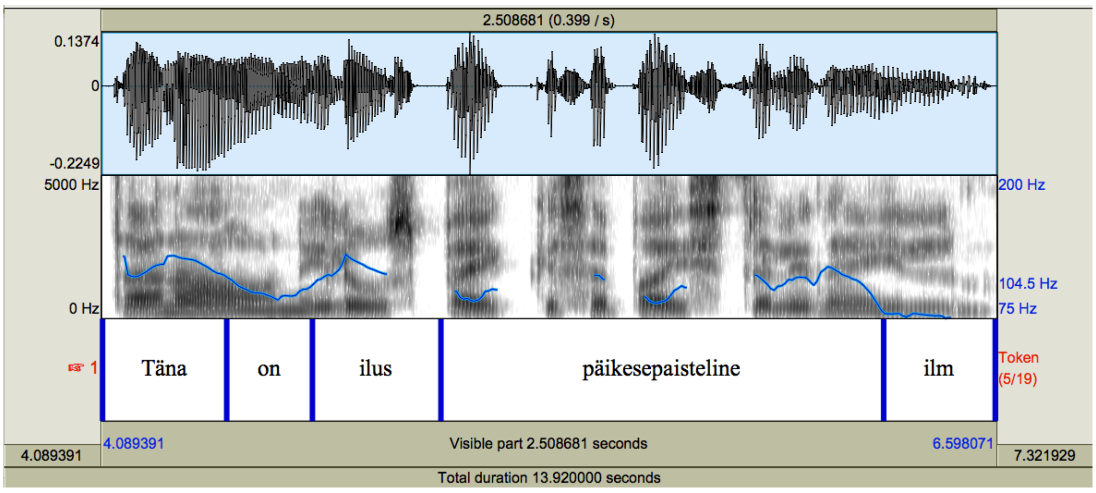


Figure 1 Example of an annotation. From top to bottom - signal waveform, spectrogram and fundamental frequency contour (blue), word-level annotation.

The web audio API [6] replaces the flash and java applet applications that were extensively used years ago for both to record and play audio and video files. In HTML5 [7] the tag <audio> provides a primary audio player and controls in most of the major web browsers according to the Table 1.

Table 1 HTML5 audio tag support

Browser	**Supported Version**
Safari	>= 11
Chrome	>= 49
Firefox	>= 58
Edge	>= 16

2.1 Literature Review

Nowadays, web-based speech corpora systems provide many benefits. The benefits include no requirement to commute to a specific place to record the corpus, no geographical limitation, and an increase in the democratization of speech and language analysis [8]. Therefore, the pitfalls of the web-based speech corpora systems involve the limitations of internet bandwidth and ambient noise.

Moreover, the costs of implementing an automatic speech recognition (ASR) has become cheaper. However, the biggest challenge to develop an ASR is to capture a large size corpus with excellent audio quality. Thus, Eyra [9] project solves the audio quality flaw providing early feedback for the field's collector to correct it during the capture process. Collecting a massive speech data to train the ASR may be stressful due lack of native speakers of the desired language or lack of volunteers. An alternative to solve this issue is to use the Amazon Mechanical Turk (MTurk)[1] which is an online workforce marketplace, so it is possible to gather speech recordings around the globe [10]. MTurk was used to collect speech data in VoxForge[2] project. Additionally, using the power of the crowd to speed up the corpus acquisition is in vogue. Using crowdsourcing for a speech corpus collection provides some benefits such as to harvest a language corpus at low cost [11].

It may be tedious to adopt and use several tools to collect, annotate and visualize a signal. To simplify it Emu web app [12] proposes all in one solution to facilitate and make more efficient to manage speech databases. Plus, to provide some statistical data, it interacts directly with R language.

The pronunciation differences between distinct gender, age, social-economic level, and dialect require a high effort while annotating a corpus. For example in 2000, annotating a spontaneous corpus in Chinese, took 60 minutes of a trained transcriber effort for every 4 minutes of a speech [13].

---

1 https://www.mturk.com

2 http://www.voxforge.org

2.2 Related Work

Content management system (CMS) for managing a speech corpora and using the Internet to capture signals via a web browser is in use in projects such as Wikispeech [14] and Meyda [15].

WikiSpeech is a CMS and wiki systems developed by the Bavarian Archive for Speech Signals in Munich, Germany. It provides translations to German, English, Romanian, and Russian. In this system both server and client are implemented in Java using XML for the data exchange between a server and a client.

The WikiSpeech application provides features such as annotation, session management and speech recording. To achieve some of these functionalities the application uses additional tools such as:

- WebTranscribe – which is a web-based annotation tool and so it is used for doing annotations.
- SpeechRecorder – it is used for recording speech and it requires Java runtime installed.

The project Meyda (which means "information" in Hebrew) is a JavaScript library built on top of the web audio API. It was built to facilitate the capture, visualization, processing of sounds through the web in real-time.

3 Tools

This section contains the description of specialized tools that are in use in this project, without them it would not be possible to accomplish the goals.

3.1 Aeneas

Aeneas[1] is an open-source library, written in Python and C programming languages, which synchronizes text and audio using dynamic time warping (DTW) signal processing algorithm to force the alignment. It supports 38 distinct languages including English, Estonian, Finnish and Russian among others. Also, Aeneas provides multiple file formats for time-aligned text output. One of these formats is TextGrid for further analysis and research using the Praat software.

It is necessary to provide a text file as in Table 2 and audio in the WAV format and the following parameters to synchronize audio with text, and output is a file in TextGrid format.

```
python -m aeneas.tools.execute_task audio.wav input.txt
 \"task_language=est|is_text_type=plain|os_task_file_format=textgrid\"
/home/rb/text.TextGrid
```

Table 2 Aeneas Input

Text of the first prompt
Text of the second prompt
Text of the third prompt
Text of the fourth prompt

1 https://github.com/readbeyond/aeneas

3.2 Docker

Docker[1] is an open-source tool that uses Linux Containers[2](LXC) to create an abstraction level of an Operational System(OS) and encapsulate application in the different section that is called containers. In that way, Docker provides a secure software distribution and deployment since there is no OS's dependency level, in other words, the same container can run on top of different OS such as Linux, Windows, and MacOS. Besides the benefits stated above, Docker also provides a simple way to build a container using a description file called Dockerfile which contains information related to the container's version, maintainer, and steps to build. Moreover, the docker container's image is easily shared through Docker hub[3] that is a cloud-based repository to store, build, and share Docker images.

Docker was used during the application development to test the application portability to Linux environment since the application was developed in a macOS environment. Yet, it was not used for production.

3.3 Sidekiq

Sidekiq is a simple and efficient background message processing framework for Ruby, and it integrates with Rails with ease providing an asynchronous job processing. The messages are persistent, and it is in JSON[4] format. Also, it provides a web User Interface (UI) similar as the one in the Figure 2, with localization for 25 different languages, for monitoring the jobs and their errors. As an example, the task which synchronizes the transcript with an audio file using Aeneas runs asynchronously and its basic structure is presented below.

1 https://www.docker.com

2 https://linuxcontainers.org

3 https://hub.docker.com

4 https://www.json.org

```
class AeneasWorker
  include Sidekiq::Worker
  def perform(record_id)
    # add code here
  end
end
```

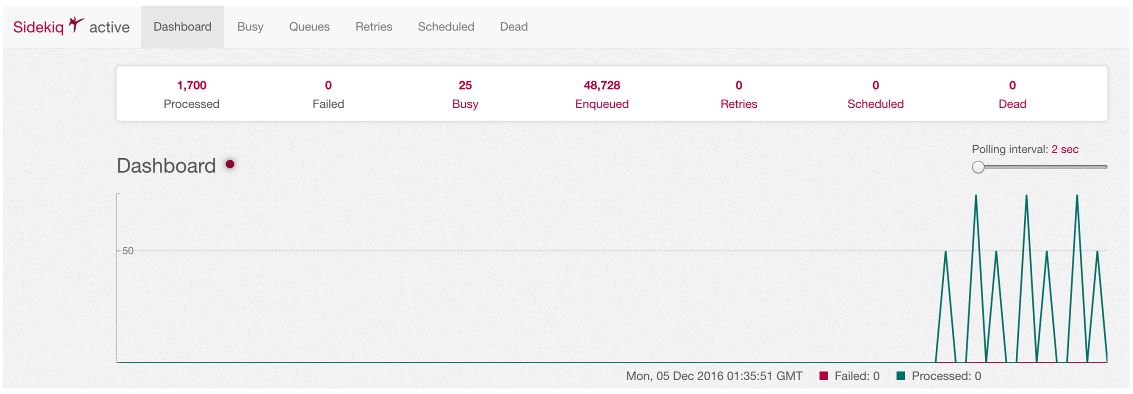


Figure 2 Sidekiq web UI

3.4 Ruby on Rails

Ruby is a very high level, dynamically typed, a flexible and open-source programming language created by Yukihiro Matsumoto and it went public 1995. Its most featured use is for web development especially along with the Rails framework. Also, Ruby is an object-oriented language inspired by Smalltalk and provides a syntax similar to plain English. The following code is the simple "Hello, World!" and the other is its object-oriented version, both are meant to illustrate the Ruby syntax and simplicity.

```
# hello_world.rb
puts "Hello, World!"

# hello_world_OO.rb
class HelloWorld
 def greet
  puts "Hello, World!"
 end
end

hello_world = HelloWorld.new
hello_world.greet # prints Hello, World!
```

Rails [16] is a broad popular web framework that relies on model-view-controller MVC architecture that separates the code in different layers according to their role and in principles such as do not repeat yourself (DRY) and convention over configuration because the framework components do not need any additional configuration to work together. To create a new Rails application, it is necessary to run the following command and it creates a directory app_name with subdirectories as show in Table 3

```
rails new app_name
```

Table 3 Default Rails directory structure

Folder	**Purpose**
app	Contains the application components
app/controllers	Contains the application controller
app/models	Contains the application models
app/views	Contains the application layout and other interfaces.
app/assets	Contains the application assets such as images, JavaScript and cascading style sheets files.
app/config	Contains the application configuration files
db/	Contains the database structure and configuration file.
log/	Contains the application log files

By default, Rails provides three different environments to run the application according to the desired use. These three environments are:

- Development – this environment is used during the application development phase.
- Test - this environment is to execute test, as an example, unit tests.
- Production – this environment is used to run the application’s stable version that can be used by the final user.

4 Application

The Figure 3 illustrates the relation between the classes and it contains only the primary and foreigner key as attributes, the full version of the class diagram is shown in the appendix section. The class DataStorage has no interaction with the user because it only interacts with another class. This class keeps a file metadata.

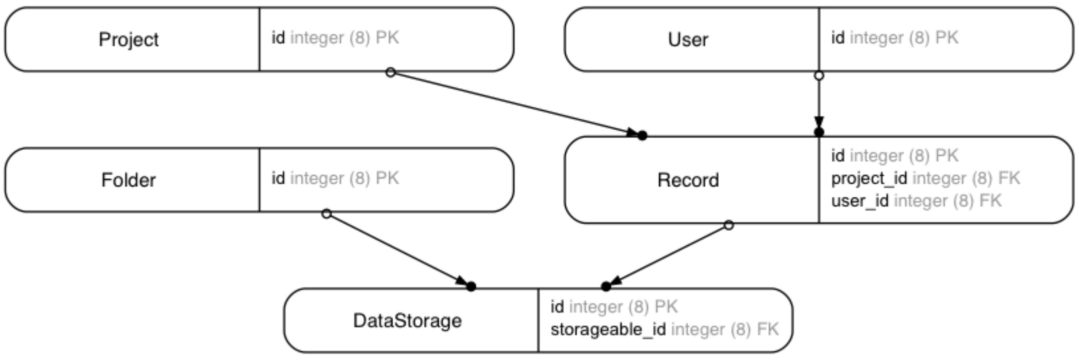


Figure 3 Domain Model

4.1 Implemented Features

The requirements selected to build a minimum viable product (MVP) related to a CMS for managing speech databases. The chosen functionalities are: manage users, manage projects, manage records and manage folder.

4.1.1 Manage User

A user refers to the registration of a person who will interact with the application in some way. The required field to insert a new user into the database are email, encrypted password, created at, and update at according to the table 1. Email field requires a valid email address. In Rails, encrypted password becomes two intransient fields they are *password* and *password_confirmation*. The password is a non-encrypted string, before saving it to the database it is encrypted to provide more security and inserted in the field *encrypted_password*. Also, it uses *password_confirmation* to verify and validate the password during the creation and updating events.

A user can sign in one of the two roles: a participant or an admin. The admin is the only type of users able to manage other users' account and to create another system administrator. And the user with a participant role can only create new records. The Table 4 describes the user fields with their types, and if mandatory to fill it out while inserting a user in the database.

Table 4 Users table

Field	**Mandatory**	**Type**
age	No	Integer
created_at	Yes	Datetime
encrypted_password	Yes	String
education	No	JSON
email	Yes	String
first_name	No	String
last_name	No	String
native_language	No	JSON
foreign_languages	No	JSON
childhood_city	No	JSON
current_city	No	JSON
gender	No	JSON
updated_at	Yes	Datetime

Field with JSON type are persisted in the column name data and its type is JSONB. Moreover, by default JSON fields are string.

4.1.2 Manage Project

A project refers to a speech corpus collection's initiative, it must contain a title, description, agreement message and active status information as shown in Table 5. The title refers to the project name and how it is going to be publicly known. The description's field includes the prompts that are going to work as a guide during a recording section. There are four different kinds of prompts: text, image, audio or video. The field category defines the type of prompt that is in use. The agreement message is the message about the project's purpose and use of the speech corpus and the participant must agree with the conditions before starting a recording session. Active status informs whether the project is active or not because participants can only start a new recording session for an active project.

To create a new project, it requires an authenticated user with admin role and with the current authorization mechanism all administrators can see all existing projects.

Table 5 Projects table

Field	**Mandatory**	**Type**
title	Yes	String
description	Yes	String
agreement_message	Yes	String
active	Yes	Boolean
category	Yes	String

The parameters for audio recording are currently fixed (that means they cannot be changed via web interface) and have the following values:

- Number of channels: 1
- Wav bit resolution: 16 Bit
- Sampling rate: 44100 Hz

The only way to change these parameters is done by change the interface code which is located at app/views/records/new.html.slim, line 165 and looks similar as the following

```
recorder = new Recorder({
    monitorGain: parseInt("0", 10),
    recordingGain: parseInt("1", 10),
    numberOfChannels: parseInt("1", 10),
    wavBitDepth: parseInt("16", 10),
    sampleRate: parseInt("44100", 10),
    encoderPath: "#{asset_path('waveWorker.min.js')}"
});
```

4.1.3 Manage Record

A record is stored when a participant contributes to an active project and captures an audio file. Its fields are the foreign key (FK) of a user and project, user_id and project_id, respectively. Therefore, it is mandatory the existence of a project and user. Also, the user with participant's role must be authenticated to join an ongoing project and create a new record section into the database.

Moreover, there are two extra fields to add relevant information regarding the audio collected. These fields are audio_duration which states the length of sound in seconds, and original_script it is a copy of the project's description field during the recording session. So, in case the project associated with the record is updated, it does not affect the records transcripts.

Table 6 Records table

Field	**Mandatory**	**Type**
project_id	Yes	Bigint
user_id	Yes	Bigint
audio_duration	No	String
original_script	No	String

4.1.4 Manage Folder

Folder means a place that keeps files according to Table 7. It is needed and used for the case when creating a new speech corpus collection's project, it is necessary to add media files such as images, audios or videos from a local computer and make them available in the project's description or agreement message fields.

Table 7 Folder table

Field	**Mandatory**	**Type**
file	yes	String

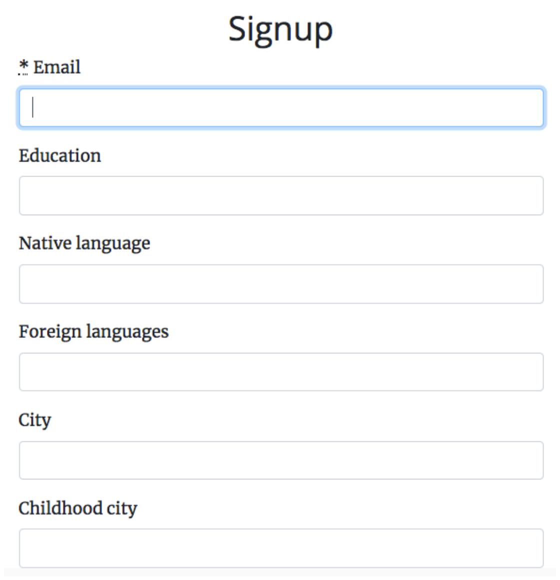
4.2 Usage

In this section it is described how to use and access the application functionalities.

4.2.1 Signup

To sign up a user needs to reach the application web domain and press the option named “access” located at the right top corner and press the sign-up link located at the bottom left side as displayed in Figure 7.

The user will face a sign-up form according to Figure 4 and Figure 5. All required fields are maker with “*”. After adding information to at least the required field, the user needs to press the button "Sign up" to complete the process and creates a new user. Then, the new user is able to authenticate and join a project.


Signup

* Email

Education

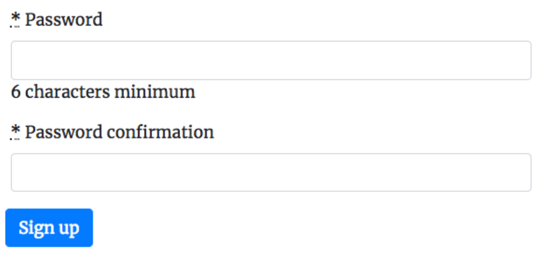
Native language

Foreign languages

City

Childhood city

Figure 4 User sign-up form part 1


* Password

6 characters minimum

* Password confirmation

Figure 5 User sign-up form part 2

4.2.2 Authentication

Authentication is the act of confirming the user identity and accessing the application by providing an email and password. In that way, it requires the user is present in the system's database. There exist two different ways to authenticate into the application as an admin or a participant. The admin UI is located at /admin path, for example, if the application is running the URL http://localhost:3000 the admin login UI can be reached at http://localhost:3000/admin and it displays an interface similar to Figure 6

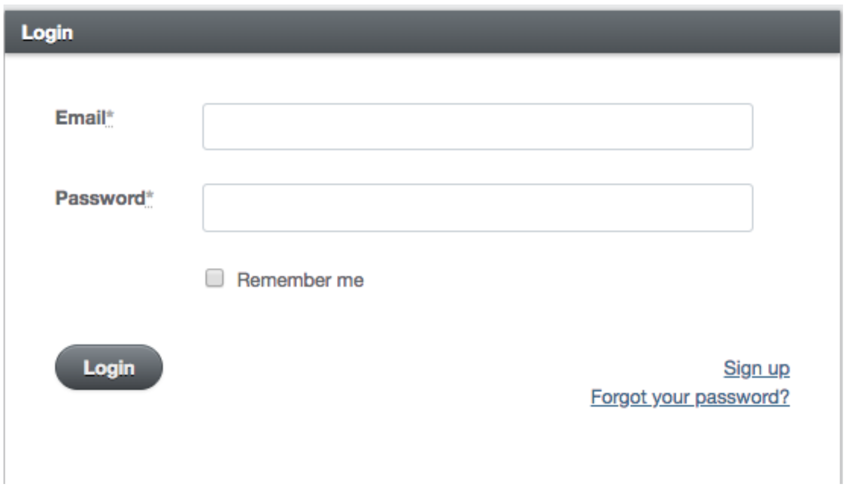


Figure 6 Admin Login UI

The participant authentication page is accessible by clicking the link *Access* present in the home page on the top right corner, and it redirects to the authentication page, for example, if the application is running the URL http://localhost:3000, then click in the link *Access*. The page should look similar to the Figure 7.

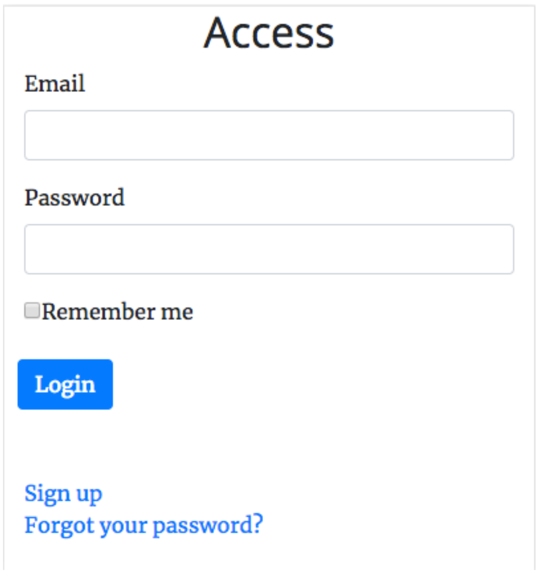
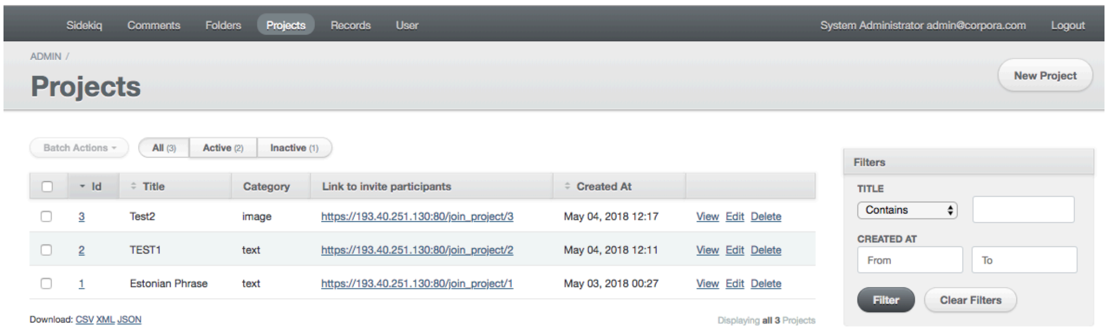


Figure 7 Participant Login UI

For both cases, after providing the email and password, the user needs to press login. So, it will authenticate the user and redirect to another page.

4.2.3 Projects

After the user passed the authentication, in the admin dashboard page the administrator needs to press the option “Projects” located in the menu at the top. After that, the user will get redirected to the project’s page where the administrator will see all existing projects according to Figure 8. It is possible to move between active and inactive projects, or to visualize all project by using the navigation tabs.



	Id	Title	Category	Link to invite participants	Created At	
☐	3	Test2	image	https://193.40.251.130:80/join_project/3	May 04, 2018 12:17	View Edit Delete
☐	2	TEST1	text	https://193.40.251.130:80/join_project/2	May 04, 2018 12:11	View Edit Delete
☐	1	Estonian Phrase	text	https://193.40.251.130:80/join_project/1	May 03, 2018 00:27	View Edit Delete

Download: CSV XML JSON

Displaying all 3 Projects

Figure 8 Project page

To add a new speech collection project, the administrator needs to press the button “New Project” located the top right. Then, it redirects to the project’s form, according to Figure 9 and Figure 10, and the user must fill out all required fields marked with “*” such as title, category, description and agreement message.

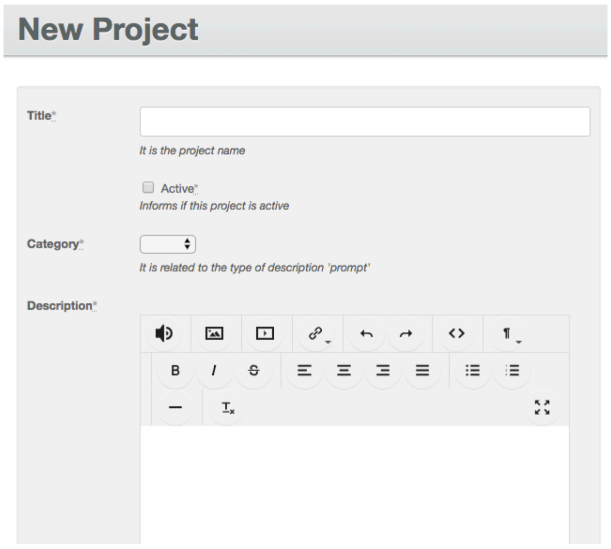

New Project
Title*
It is the project name
Active*
Informs if this project is active
Category*
It is related to the type of description 'prompt'
Description*

Figure 9 Project form part 1

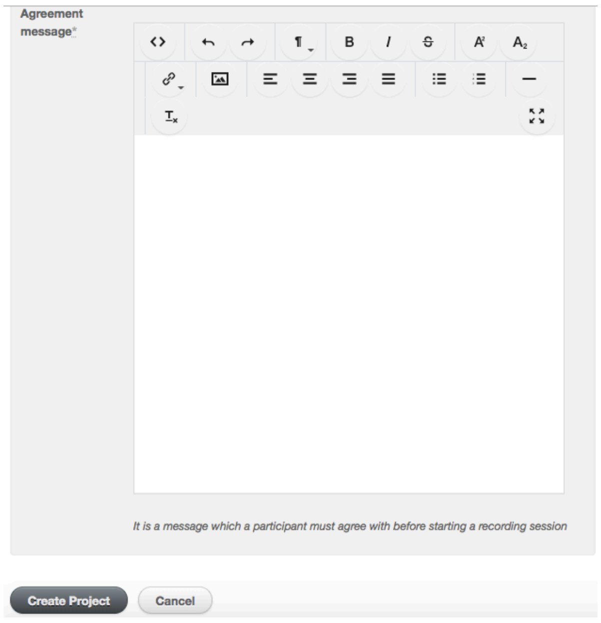

Agreement message
It is a message which a participant must agree with before starting a recording session
Create Project
Cancel

Figure 10 Project form part 2

When the required fields are filled out, the user must press the button “Create Project”. It saves the project into a database and redirects back to the projects page similar to Figure 8.

To edit a project the administrator needs to go to projects pages and clicks the link “edit” of a desired project which needs to be changed. The edit act occurs in the same project form used to insert a new project. The only modification is the down left side button which becomes “Update Project” instead of “Create Project” as seen the Figure 10.

A project only can be deleted if there is no record associated with it, otherwise it is not possible to delete it because a record must be associated with a project.

4.2.4 Add files to a folder

To add files to a folder it is mandatory to have an authenticated user and the authorized user must have an admin role. The interfaces s similar to Figure 11. The user needs to press the button “choose file”, select one file from the computer and press *Create folder* button. It saves a file to a folder.

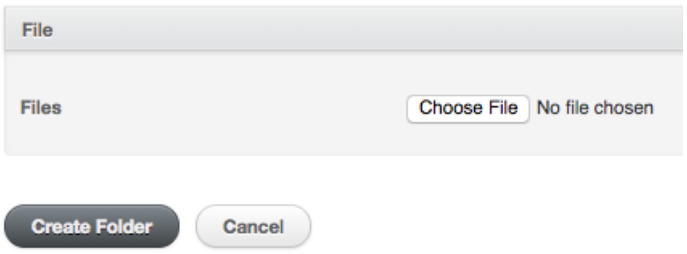
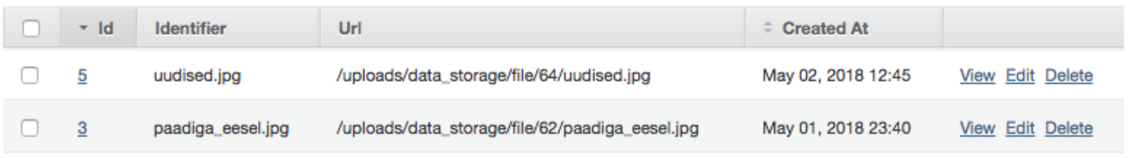


Figure 11 Add files to folder

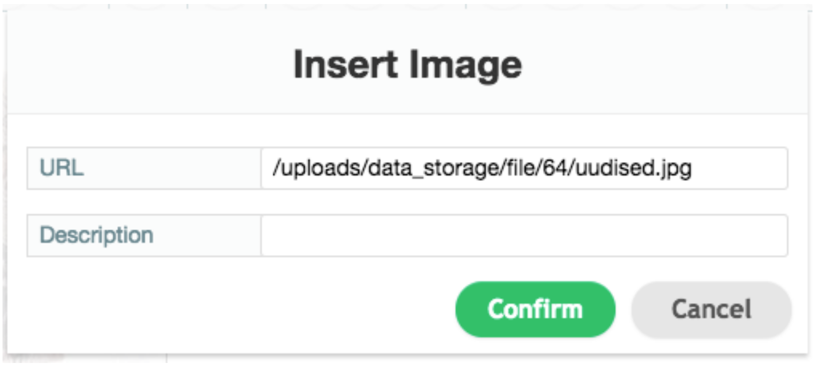
After adding a file, the user needs to copy the URL, which is the file's address in the application, as shown in the Figure 12.



	Id	Identifier	Url	Created At	
☐	5	uudised.jpg	/uploads/data_storage/file/64/uudised.jpg	May 02, 2018 12:45	View Edit Delete
☐	3	paadiga_eesel.jpg	/uploads/data_storage/file/62/paadiga_eesel.jpg	May 01, 2018 23:40	View Edit Delete

Figure 12 Files in the folder

And after copying the file's address paste it in the appropriate place. As an example, if the file is an image, it needs to be placed in the pop-up window to add an image as illustrated in Figure 13.



URL: /uploads/data_storage/file/64/uudised.jpg

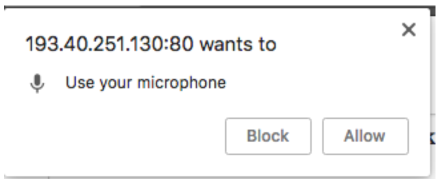
Figure 13 Pop-up window to insert an image file into a project description field

4.2.5 Collect audio

An audio collection session requires an existing and active project plus an authenticated user. Also, the user needs to receive the project URL that informs to which project she/he is invited to contribute. The audio collection action is illustrated in the Figure 16 and an explanation of each step is described below.

After the authentication, the user must agree with the agreement message before starting the recording session by pressing the button "Agree".

Next step, is to allow the microphone access from the web browser as shown in Figure 14 and adjust microphone's sensitivity.


193.40.251.130:80 wants to

Use your microphone

Figure 14 Example pop-up allow microphone access

To start a recording session the user needs to press the button located at bottom left corner "Start recording" see Figure 15. Then the user begins to read a transcript or explain the prompts provided, to complete the recording the user ends the recording session.

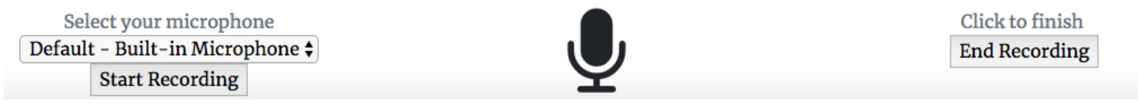


Figure 15 Start/End recording

When the user ends the recording session, the server starts to receive the audio file, saves it in a directory, and stores the audio file path into a database. After saving it, a process begins to align the transcript with the speech record using the AeneasWorker. This processing only happens when a project's category is text.

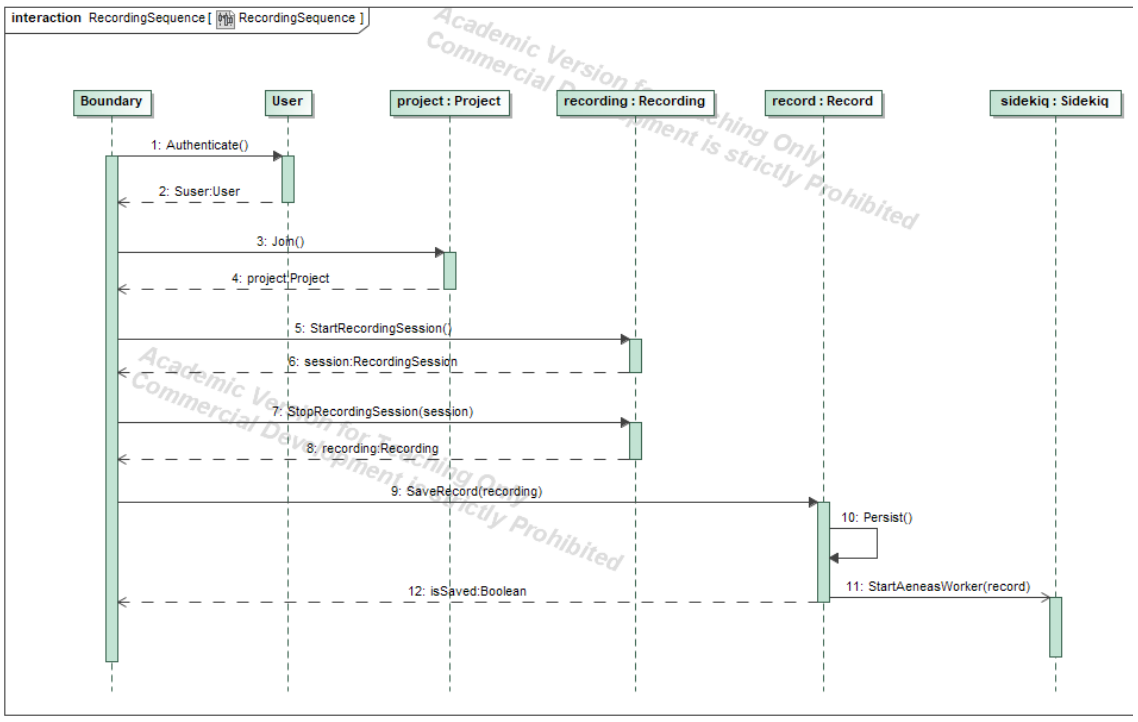


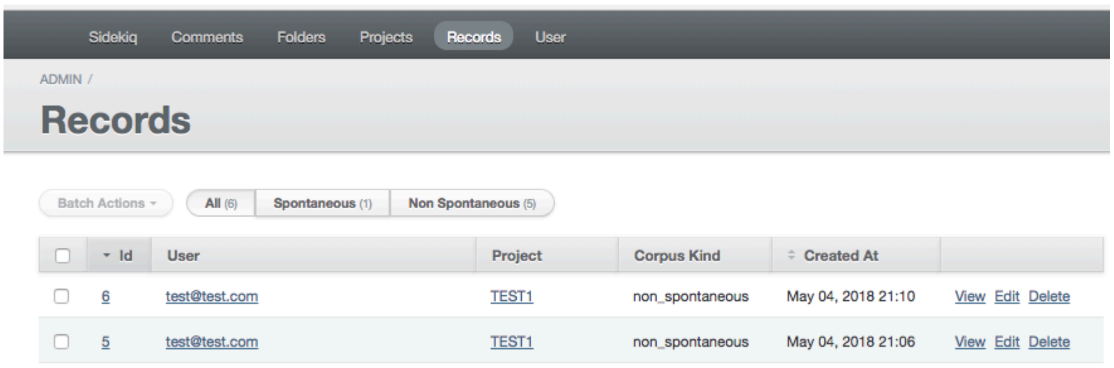
Figure 16 Audio collection sequence diagram

4.2.6 Records

Records are created during the audio collection phase by a subject participating in speech corpus collection participant. To access the records, it is required to have an authenticated admin and from the administrator press the option “Records” located on the menu on top.

After pressing “Records”, it redirects to the records’ page as shown in Figure 17. This is the page where the admin user can see all existing records.

It is possible to filter the records according to the nature of the speech corpus that can be either spontaneous or non-spontaneous.



Records

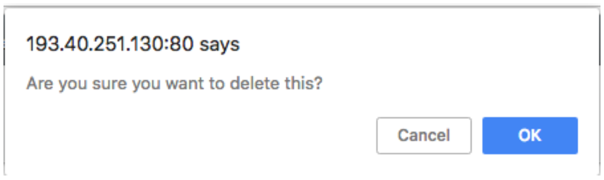
Id	User	Project	Corpus Kind	Created At	
6	test@test.com	TEST1	non_spontaneous	May 04, 2018 21:10	View Edit Delete
5	test@test.com	TEST1	non_spontaneous	May 04, 2018 21:06	View Edit Delete

Figure 17 Records page

From the records page, the administrator may want to listen to speech files, to see their automatically generated transcripts in Praat TextGrid format or both. The transcript is only created when the project type is set as a text.

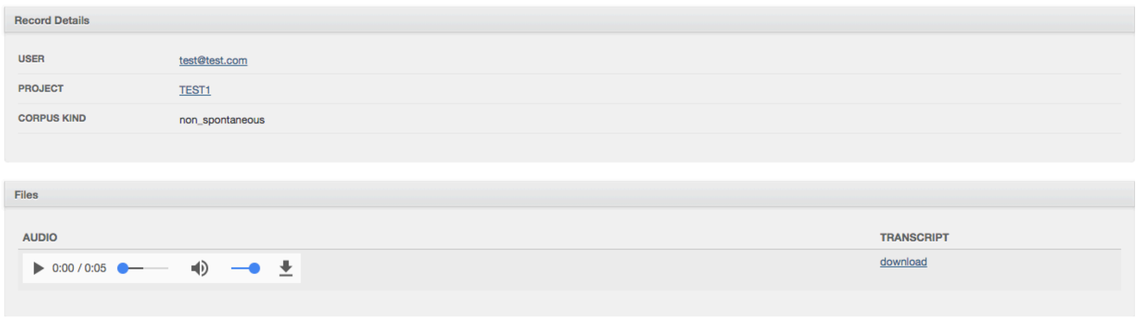
In the record details' page, similar to Figure 19, it is possible to download the available files and see some other information related to the record such as the participant and the type of the speech corpus.

To delete a record, the administrator needs to go to the records page (Figure 17) and clicks the link "Delete" in the row of the table row of the record supposed to be removed. A confirmation pop-up window similar to Figure 18 will appear and the admin user has to confirm the action. This confirmation is necessary because deleting a record accidentally are not welcome. After deleting a record, it is impossible to recover it.


193.40.251.130:80 says

Are you sure you want to delete this?

Figure 18 Record deletion confirmation


Record Details

USER	test@test.com
PROJECT	TEST1
CORPUS KIND	non_spontaneous

Files

AUDIO	TRANSCRIPT
0:00 / 0:05	download

Figure 19 Record details

4.3 Installation

This section describes the requirements and procedures to install and execute the application.

4.3.1 Requirements

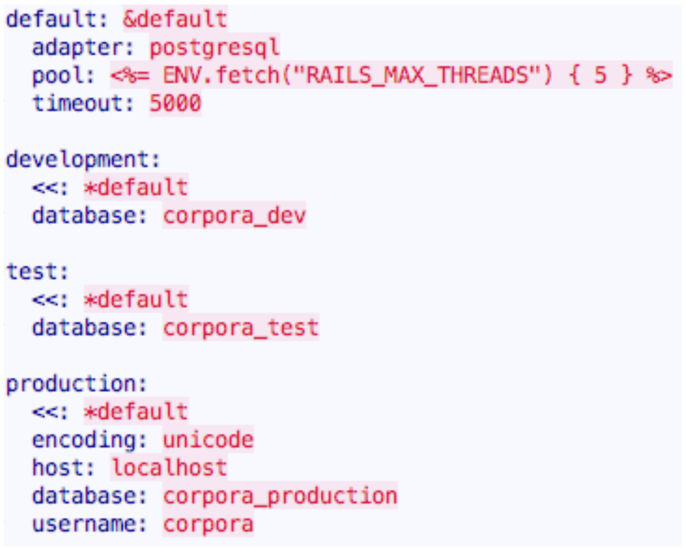
The application requires the pre-requisites listed in Table 8 plus always keep your browser updated[1]. Install a Ruby version manager (RVM) which makes easier to have a different Ruby version on your machine and to upgrade the releases as well. Also, obtain the project source-code mentioned in section 4.4.

Table 8 Hardware Requirements

Minimum	**Recommended**
1 GB memory RAM	2 GB memory RAM
1 core processor	2 core processor
200 MB disk space	600 MB disk space

The application requires a relational data management system (RDMS) Postgres version 9.4 or higher. To configure the application to use a database, it is required to modify the database.yml file located in the config directory, the database.yml is similar to the Figure 20.

1 https://whatbrowser.org/



```
default: &default
  adapter: postgresql
  pool: <%= ENV.fetch("RAILS_MAX_THREADS") { 5 } %>
  timeout: 5000

development:
  <<: *default
  database: corpora_dev

test:
  <<: *default
  database: corpora_test

production:
  <<: *default
  encoding: unicode
  host: localhost
  database: corpora_production
  username: corpora
```

Figure 20 Database configuration file

4.3.2 Linux

The installation of RVM in most of Linux distributions can be done by following the steps

- gpg --keyserver hkp://keys.gnupg.net --recv-keys 409B6B1796C275462A1703113804BB82D39DC0E37D2BAF1CF37B13E2069D6956105BD0E739499BDB
- curl -sSL https://get.rvm.io | bash -s stable
- Run rvm install 2.5
- Run gem install bundler
- In the application’s directory run the command bundle install
- Execute bundle exec rake db:create to create a database instance
- Execute bundle exec rake db:migrate to create the database tables
- Execute bundle exec rake db:seed to add the default user

4.3.3 MacOS

It is recommended to have installed the package manager for macOS homebrew[1]. The steps to install the application are:

- Install gpg by running brew install gpg

1 https://brew.sh

- curl -L https://get.rvm.io | bash -s stable
- rvm install 2.5
- Run gem install bundler
- In the application's directory run the command bundle install
- Execute bundle exec rake db:create to create a database instance
- Execute bundle exec rake db:migrate to create the database tables
- Execute bundle exec rake db:seed to add the default user

4.3.4 Windows

For Windows platform, it is recommended to use a different ruby version's manager instead of RMV it is suggested to use Uru.

- Install uru by downloading the executable at https://bitbuket.org/joinforums/uru/wiki/Downloads
- Run uru 250 which install ruby 2.5
- Run gem install bundler
- In the application's directory run the command bundle install
- Execute bundle exec rake db:create to create a database instance
- Execute bundle exec rake db:migrate to create the database tables
- Execute bundle exec rake db:seed to add the default user

4.3.5 Starting the application

To start the application in your machine, go to the application directory and run the command bundle exec Rails server, and the application can be accessed at the address http://localhost:3000

When starting the application on another computer than your machine it requires an SSL certification because the Web Audio API is only allowed to run in a secure domain. Besides that, there is no other restriction.

To start the application in a server it needs to provide an SSL certification and it be achieved by executing the command, in the terminal and application's directory, bundle exec passenger start --ssl --ssl-certificate /your/path/cert.pem --ssl-certificate-key /your/path/key.pem

To start the Sidekiq, see section 3.3, requires the key-value database Redis[1] and to execute the following command bundle exec sidekiq -C config/sidekiq.yml.

4.3.6 Aeneas

Aeneas, see section 3.1, is a tool to synchronize a text prompt with an audio, and generate a Praat compatible file. Its requirements are listed below:

- python3
- pip3
- tgt
- numpy
- ffmpeg
- espeak

After installing the requirements, install Aeneas tool via pip3 executing the following command in a terminal *pip3 install aeneas*.

4.4 Source-code

Application's source-code are licensed under MIT[2] open-source license and it is available at https://github.com/thalessr/corpora

4.5 Default user

The default user has admin role and use it, after the installation process, to add a new administrator. To access the application with the default user, the credential is the following:

Email: admin@corpora.com

Password: password

1 https://redis.io

2 https://opensource.org/licenses/MIT

5 Summary

This work describes the application for collection of speech corpus without requiring any extra software installation from the client side. So, the person who interacts with the application does not need to install any Java nor Flash script to collect audio. Currently, it is possible to setup a speech corpus collection project, invite volunteers to contribute to it, download the record files for further phonetics analysis.

5.1 Future Work

Nowadays the use of mobile phones to access internet reaches 51.2%, and the use of desktop is 48.7% [17]. As the use of smartphones surpasses the use of a personal computer in that way, it would be interesting to use mobile phones for collecting data and sending them through the application's API.

Currently, during the audio collection all prompts are displayed at the same time and speech is store in single audio file containing all utterances. It would be more practical to display one prompt at a time and record speech related to each prompt into separate files. In this way, it will facilitate further processing of signals and transcripts utterance by utterance.

The current authorization is simple and broad. So, it may not authorize the access to services and resources as needed. For solving this issue, there exists a Rails authorization framework named Pundit[1] which does authorization by resource.

1 https://github.com/varvet/pundit

References

[1] "Speech and Voice Recognition Market by Technology (Speech Recognition, Voice Recognition), Vertical (Automotive, Consumer, Banking, Financial Services and Insurance (BFSI), Retail, Education, Healthcare & Government) and Geography - Global Forecast to 2023 : ReportsnReports." [Online]. Available: http://www.reportsnreports.com/reports/590849-speech-voice-recognition-market-by-technology-speech-recognition-voice-recognition-application-ai-based-non-ai-based-vertical-automotive-consumer-finance-retail-military-healthcare-government-and-geography-global-forecast-to-2022.html. [Accessed: 24-Jan-2018].

[2] D. O'Shaughnessy, "Introduction," in *Speech communications: human and machine; 2nd ed.*, New York, 2000, pp. 1–8.

[3] "What is Speech?" [Online]. Available: http://www.speechlanguage-resources.com/what-is-speech.html. [Accessed: 01-Mar-2018].

[4] F. Schiel and D. Christoph, *Production and validation of speech corpora*. 2003.

[5] E. Meister, *Promoting Estonian speech technology: from resources to prototypes.*, vol. 4. Tartu University Press, 2003.

[6] "Web Audio API." [Online]. Available: https://webaudio.github.io/web-audio-api/. [Accessed: 01-Feb-2018].

[7] G. Anthes, "HTML5 Leads a Web Revolution," *Commun. ACM*, vol. 55, no. 7, pp. 16–17, Jul. 2012.

[8] C. Draxler, J. Harrington, and F. Schiel, "Towards the next generation of speech tools and corpora.," *Computer Speech & Language*, vol. 46, pp. 175–178, 2017.

[9] J. Guðnason and M. Pétursson, "Building ASR corpora using Eyra," presented at the Interspeech 2017, 2017, pp. 2173–2177.

[10] I. Lane, M. Eck, K. Rottmann, and A. Waibel, "Tools for Collecting Speech Corpora via Mechanical-Turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Los Angeles, 2010, pp. 184–187.

[11] W. Y. Wang, D. Bohus, E. Kamar, and E. Horvitz, "Crowdsourcing the acquisition of natural language corpora: Methods and observations," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 73–78.

[12] R. Winkelmann, "Managing Speech Databases with emuR and the EMU-webApp," presented at the Sixteenth Annual Conference of the International Speech Communication Association, 2015.

[13] X. W. JunLan Feng and D. LiMin, "Data Collection and Processing in a Chinese Spontaneous Speech Corpus IIS_CSS," presented at the Sixth International Conference on Spoken Language Processing, 2000.

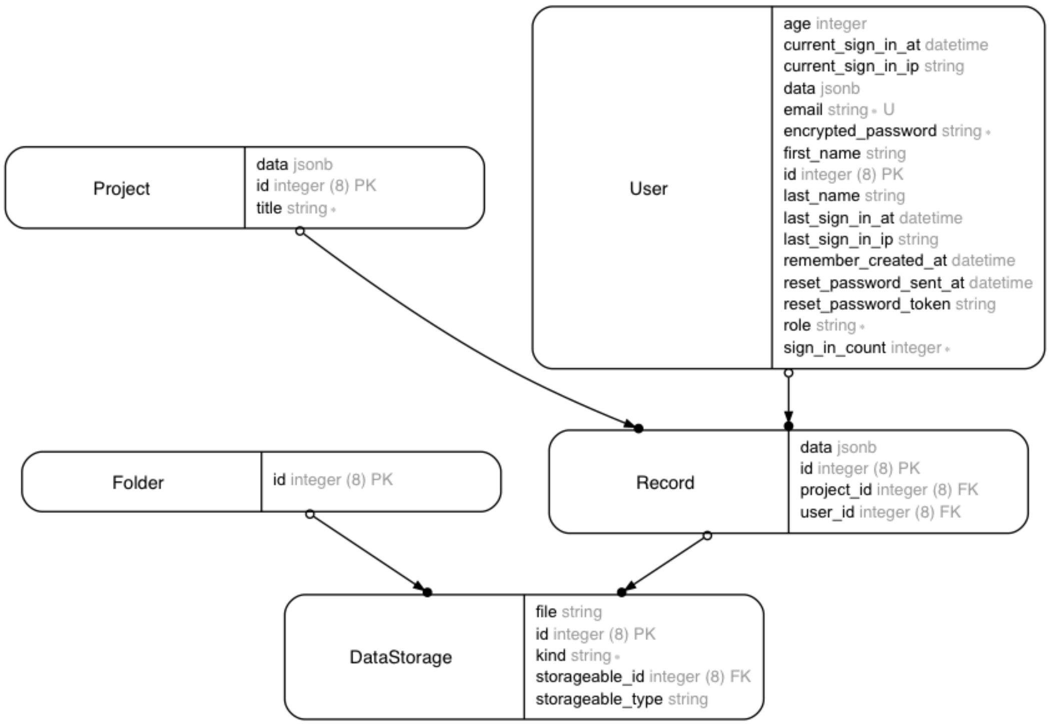
[14] C. Draxler and K. Jansch, "Draxler, Christoph, and Klaus Jänsch. WikiSpeech-A Content Management System for Speech Databases.," presented at the Ninth Annual Conference of the International Speech Communication Association., 2008.

[15] H. Rawlinson, S. Nevo, and J. Fiala, "Meyda: an audio feature extraction library for the web audio api.," presented at the The 1st Web Audio Conference (WAC), Paris, Fr, 2015.

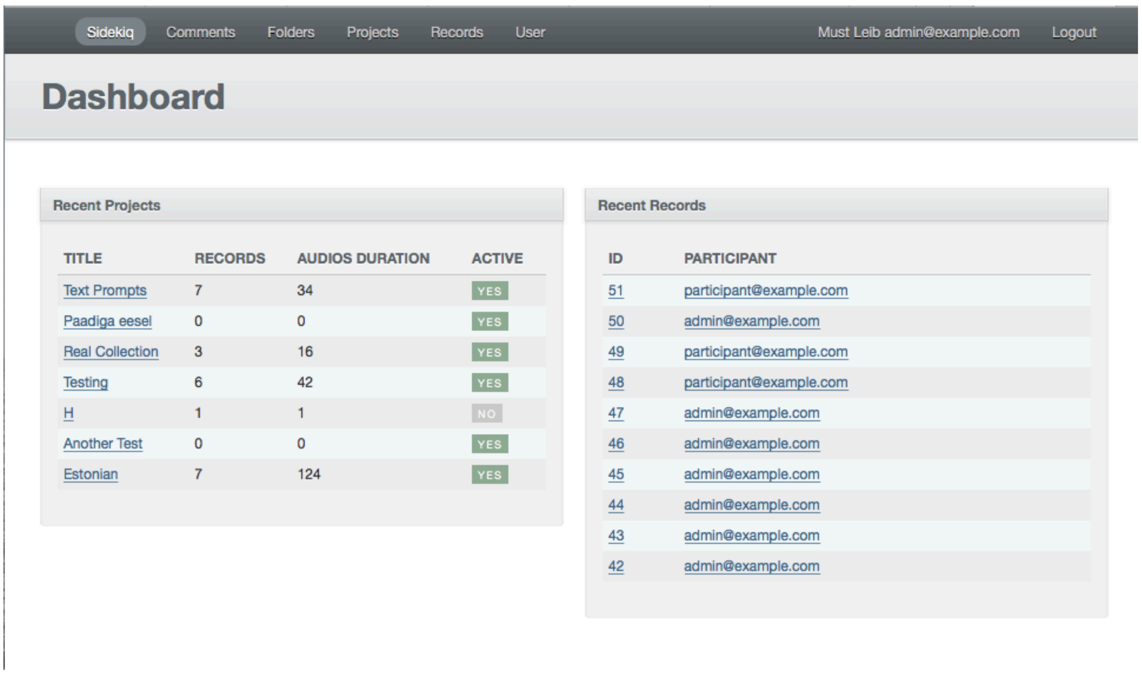
[16] M. Bächle and P. Kirchberg, "Ruby on Rails," *IEEE Software*, vol. 24, no. 6, pp. 105–108, Nov. 2007.

[17] "Mobile internet use passes desktop for the first time, study finds," *TechCrunch*, 01-Nov-2016. .

Appendix 1 – Class Diagram



Appendix 2 – Admin application's dashboard



Dashboard

Recent Projects

TITLE	RECORDS	AUDIOS DURATION	ACTIVE
Text Prompts	7	34	YES
Paadiga eesel	0	0	YES
Real Collection	3	16	YES
Testing	6	42	YES
H	1	1	NO
Another Test	0	0	YES
Estonian	7	124	YES

Recent Records

ID	PARTICIPANT
51	participant@example.com
50	admin@example.com
49	participant@example.com
48	participant@example.com
47	admin@example.com
46	admin@example.com
45	admin@example.com
44	admin@example.com
43	admin@example.com
42	admin@example.com