

“ALEXANDRU IOAN CUZA” UNIVERSITY OF IAȘI

MASTER OF COMPUTATIONAL OPTIMIZATION

FACULTY OF COMPUTER SCIENCE

ADVANCED SOFTWARE ENGINEERING TECHNIQUES 2019 PROJECT

- REQUIREMENTS ANALYSIS -

Audio Tagging

**Automatically recognize sounds and apply tags of varying
natures**

PROPOSED BY: COJOCARU GABRIEL-CODRIN

DINU SERGIU ANDREI

LUNCAȘU BOGDAN CRISTIAN

RACOVITĂ MĂDĂLINA-ALINA

VÎNTUR CRISTIAN

SCIENTIFIC COORDINATORS: PHD ASSOCIATE PROFESSOR ADRIAN IFTENE

PHD ASSOCIATE PROFESSOR MIHAELA ELENA BREABAN

Contents

Contents	2
I. Introduction	3
I.1. Application's overview	3
I.2. Constraints	3
II. Minimum viable product	3
II.1. Functionalities	4
II.2. Use cases	4
II.2.1 Get an audio file tagged	4
II.2.2 Search for an audio file by tags	4
II.2.3 Listen to an audio file on the platform	5
II.2.4 Download an audio file from the platform	5
III. Additional features	5
III.1. Functionalities	6
III.2. Use cases	6
III.2.1 Get a video file tagged	6
III.2.2 Get a video from youtube tagged	6
III.2.3 Add or flag tags when the tagging algorithm fails	6

I. Introduction

I.1. Application's overview

Clink is a web application that specializes in uploading and retrieving audio files. On upload, a number of tags are generated automatically and associated with the given audio file which can then be used for queries.

I.2. Constraints

Any further described functionality will be guaranteed to work on Google Chrome 77.0.3865 (or newer) running on machines using as operating system Windows 10 build 18362 (or newer) or Ubuntu 18.04.3 (or newer). At the same time the width of the screen used while browsing the application must be at least 1200 pixels. For any other configurations the application is not guaranteed to work as described further.

II. Minimum viable product

II.1. Functionalities

Functionalities present in the minimum viable product (MVP) include being able to upload, listen or download audio files. On upload, a number of tags will be automatically generated and associated with the given file. A user can also query the existing base of audio files with filters using the tags. An example of a query is 'give me all audio files on the platform that contain bark and applause'. The available tags an user will be able to choose from are: Accelerating and revving and vroom, Accordion, Acoustic guitar, Applause, Bark, Bass drum, Bass guitar, Bathtub (filling or washing), Bicycle bell, Burping and eructation, Bus, Buzz, Car passing by, Cheering, Chewing and mastication, Child speech and kid speaking, Chink and clink, Chirp and tweet, Church bell, Clapping, Computer keyboard, Crackle, Cricket, Crowd, Cupboard open or close, Cutlery and silverware, Dishes and pots and pans, Drawer open or close, Drip, Electric guitar, Fart, Female singing, Female speech and woman speaking, Fill (with liquid), Finger snapping, Frying (food), Gasp, Glockenspiel, Gong, Gurgling, Harmonica, Hi-hat, Hiss, Keys jangling, Knock, Male singing, Male speech and man speaking, Marimba and xylophone, Mechanical fan, Meow, Microwave oven, Motorcycle, Printer, Purr, Race car and auto racing, Raindrop, Run, Scissors, Screaming, Shatter, Sigh, Sink (filling or washing), Skateboard, Slam, Sneeze, Squeak, Stream, Strum, Tap, Tick-tock, Toilet flush, Traffic noise and roadway noise, Trickle and dribble, Walk and footsteps, Water tap and faucet, Waves and surf, Whispering, Writing, Yell, Zipper (clothing).

II.2. Use cases

II.2.1 Get an audio file tagged

1. On opening the application, an user lands on the home page where he can find an upload button
2. By clicking the upload button he is prompted to select an audio file from his device
3. After selecting a valid audio file he will see a loader with the progress made on uploading the file
4. Upon finishing the upload the user will be able to see on the screen the tags associated with his file

II.2.2 Search for an audio file by tags

1. On opening the application, an user lands on the home page where he can find a tags input
2. The user can start writing the name of a tag into the input and by pressing enter, the tag will be added to the tags filter list
3. After the tags filter list is complete, the user can click the search button
4. Upon finishing retrieving the results, the user will be able to see audio files matching the specified filter

II.2.3 Listen to an audio file on the platform

1. By following II.2.2 the user can get a list of audio files matching a set of filters
2. Each audio file retrieved will have a play button associated on the page
3. By pressing the play button, the user can listen to the audio file
4. The user can also press the pause button while an audio file is playing to stop it

II.2.4 Download an audio file from the platform

1. By following II.2.2 the user can get a list of audio files matching a set of filters
2. Each audio file retrieved will have a download button associated on the page
3. By pressing the download button the user can download the audio file on his device

III. Additional features

III.1. Functionalities

A set of additional features can be added to the MVP based on the time remaining after finishing it. These features include uploading videos, tagging videos from youtube by providing only an url to the video, users being able to manually specify when the tagging algorithm is not working properly by adding or flagging tags, an authentication system or generating subtitles for a video based on the generated tags.

III.2. Use cases

III.2.1 Get a video file tagged

1. An user will be able to follow steps described in II.2.1 but additionally he will be able to select video files

III.2.2 Get a video from youtube tagged

1. On opening the application, an user lands on the home page where he can find an input for a youtube url
2. After writing a valid url to the input specified above, the user will be able to click a tag youtube video button
3. After clicking the button specified above the user will see a loader with the progress made on tagging the specified video
4. Upon finishing the user will be able to see the the generated tags on screen

III.2.3 Add or flag tags when the tagging algorithm fails

1. After following the steps described above to tag a multimedia file, the user will be able to flag tags outputed by the tagging algorithm as wrong or add new tags the algorithm didn't catch
2. While browsing other uploaded multimedia files the user can flag tags outputed by the tagging algorithm as wrong or add new tags the algorithm didn't catch