BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

WEB SERVICE FOR PRONUNCIATION TRAINING AND FOREIGN LANGUAGE LEARNING
WEBOVÁ SLUŽBA PRO TRÉNOVÁNÍ A VÝUKU VÝSKLOVNOSTI CIZÍCH JAZYKŮ

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **AUTOR PRÁCE**	**JÁN STEHLÍK**
SUPERVISOR **VEDOUCÍ PRÁCE**	**Ing. IGOR SZŐKE, Ph.D.**

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačové grafiky a multimédií   Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Stehlík Ján**

Obor: Informační technologie

Téma: **Webová služba pro trénování a výuku výslovnosti cizích jazyků**
              Web Service for Pronunciation Training and Foreign Language Learning

Kategorie: Zpracování řeči a přirozeného jazyka

Pokyny:

1. Navrhněte jednoduchou webovou službu, která umožní uživatelům trénovat výslovnost opakováním krátkých videoukázek (např. z YouTube). Záznam jejich hlasu bude porovnán s referencí a bude vypočteno skóre podobnosti. Zabudujte též vhodné gamifikační techniky.
2. Seznamte se se základy porovnání vzorů dvou nahrávek (pomocí DTW). Nastudujte dodaný přehrávač videí se zabudouvanou funkcí pro porovnání dvou vzorů nahrávek.
3. Implementujte a zveřejněte webovou službu.
4. Sbírejte spětnou vazbu a chování uživatelů. Navrhněte úpravy, ty implementujte a vyhodnoťte.
5. Diskutujte dosažené cíle a navrhněte směry dalšího vývoje.
6. Vyrobte A2 plakátek a cca 30 vteřinové video prezentující výsledky vaší práce.

Literatura:
- Dle pokynů vedoucího

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese
http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Szőke Igor, Ing., Ph.D.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2016
Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Jan Černocký
vedoucí ústavu

Abstract

This article presents a design and an implementation of Web application for English pronunciation training. Nowadays, there are a lot of on-line applications that help to improve English level or skill level of any other foreign language. On the other hand, there are not so many applications that help to improve pronunciation of given language. This project was developed to fill the gap, and to enable users to practice pronunciation by repeating short video segments. Using their microphone device users will record their speech. This speech will be compared with the reference recording. Result of this comparison will be returned to the user as a similarity score. The similarity score is computed using Dynamic Time Warping algorithm, that is incorporated inside a video player used in this application.

Abstrakt

Táto práca prezentuje návrh a implementáciu webovej aplikácie, ktorá slúži na trénovanie výslovnosti Anglického jazyka. V súčastnosti existuje množstvo on-line služieb, ktorých cieľom je pomôcť užívateľom zlepšiť ich úroveň Anglického jazyka alebo akéhokoľvek iného cudzieho jazyka. Na druhej strane, aplikácií, ktoré by pomohli s výslovnosťou daného jazyka nieje mnoho. Tento projekt bol vyvinutý aby túto dieru zaplátal a aby umožnil užívateľom cvičiť výslovnosť opakovaním krátkych videoukážok. Reč užívateľa bude nahraná pomocou jeho mikrofónu a následne porovnaná s referenčnou nahrávkou. Výsledkom tohto porovnania bude skóre podobnosti, ktoré bude vrátene používateľovi. Skóre podobnosti bude vypočítané použitím algoritmu Dynamic Time Warping, ktorý je zabudovaný vo video prehrávači použitom v tejto aplikácii.

Keywords

Pronunciation training, user speech, English language, Web application, similarity score, video, video segment, DTW.

Klíčová slova

Trénovanie výslovnosti, reč užívateľa, Anglický jazyk, webová aplikácia, skóre podobnosti, video, video segment, DTW.

Reference

STEHLÍK, Ján. *Web service for pronunciation training and foreign language learning.* Brno, 2017. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Szőke Igor.

Web service for pronunciation training and foreign language learning

Declaration

I hereby declare that this project work is a record of an original work done by me under the guidance of Mr. Ing. Igor Szőke, Ph.D. I listed all bibliography sources and publications used in this project.

. . . .
Ján Stehlík
May 15, 2017

Acknowledgements

Contents

Chapter 1

Introduction

Topic of this bachelor's thesis is to develop Web service for pronunciation training and foreign language learning. Goal of the application is to enable users to train their pronunciation by repeating short video segments. Users will receive similarity score that will be calculated by comparing their speech with the original recording. Similarity comparison between reference recording and test recording is an embedded feature of glplayer[1], that is vital part of this application.

This project is focused on creating easy-to-use, interactive and multi-platform application, that will a)make it simple convenient for administrators to add new content and b)encourage and entertain users to use that content. Application is deployed on website *www.speakinglike.com*. Comparison between similar applications are described in chapter 2. That chapter will take closer look at three different web services and analyse their pros and cons. Design and decomposition is described in chapter 3. The implementation described in chapter 4 will delve into important implementation details. Chapter 5 is about testing of the application and evaluating feedback from the real users. Possible extensions of this project in chapter 6 suggest the possible development route this application might take.

1.1 Goals of the application

The final application has to fulfill number of important goals. The application has to consist of reliable and complex back-end for administrators and user-friendly and intuitive front-end for wide user base.

1.1.1 Key goals

1. Front-end

    (a) Authentic videos as means to train English pronunciation
    (b) Attract wide user base, language skill level not to be a barrier
    (c) Provide objective feedback to user while training
    (d) Implement search over subtitles to help users find the word/phrase they are looking for

---

[1]More about glplayer in chapter 4.2

2. Back-end

   (a) Simple and fast adding of new videos from YouTube, and setting up all necessary attributes

   (b) Perform all communication with SpokenData transcription service[2]

   (c) Simple and fast adding of attributes (difficulties, categories, tags)

   (d) Select segments of the video that will be highlighted and subtitled

   (e) Create entertaining exercises in a simple way

   (f) Authenticate to enter administration

---

[2]Communication between this application and www.spokendata.com is described in 4.3.1

Chapter 2

Characteristics of current state

English is the most widespread language in the world and is more widely spoken and written than any other language. Over 400 million people use the English vocabulary as a mother tongue. Over 700 million people, speak English, as a foreign language. The main language used throughout the world on the Internet is English. The media that make up the Internet are overwhelmingly American in origin, so it is no wonder that the mother tongue of the Web is English. All these facts indicate the necessity of relevant studying tools accessible all over the globe. There is plenty of applications available on the Internet which focus on improving user's grammar, vocabulary, writing and listening skills. Some even aim to enhance pronunciation using videos and subtitles to help users comprehend and imitate provided audio. We will take a closer look at three Web applications which use videos and subtitles as a primary learning tool.

2.1 FluentU

FluentU[1] is a subscription-based application that offers language immersion through big database of translated and subtitled YouTube videos in six languages (English, Chinese, Spanish, French, German, Japanese). The site provides intelligent learning system that allows user to learn the video content. Moreover, FluentU has a build-in flashcard centre, where all words and phrases user previously learnt are stored, available to be revised. Videos are sorted according to difficulty level (newbie, elementary, intermediate, upper-intermediate, advanced, native).

2.1.1 Video player

Features of video player:

1. Hover dictionary. The user can easily look up every word he does not understand by hovering over the word.

2. The possibility to listen to each individual word as many times as the user requires.

3. Ability to replay entire video or the video segment repeatedly.
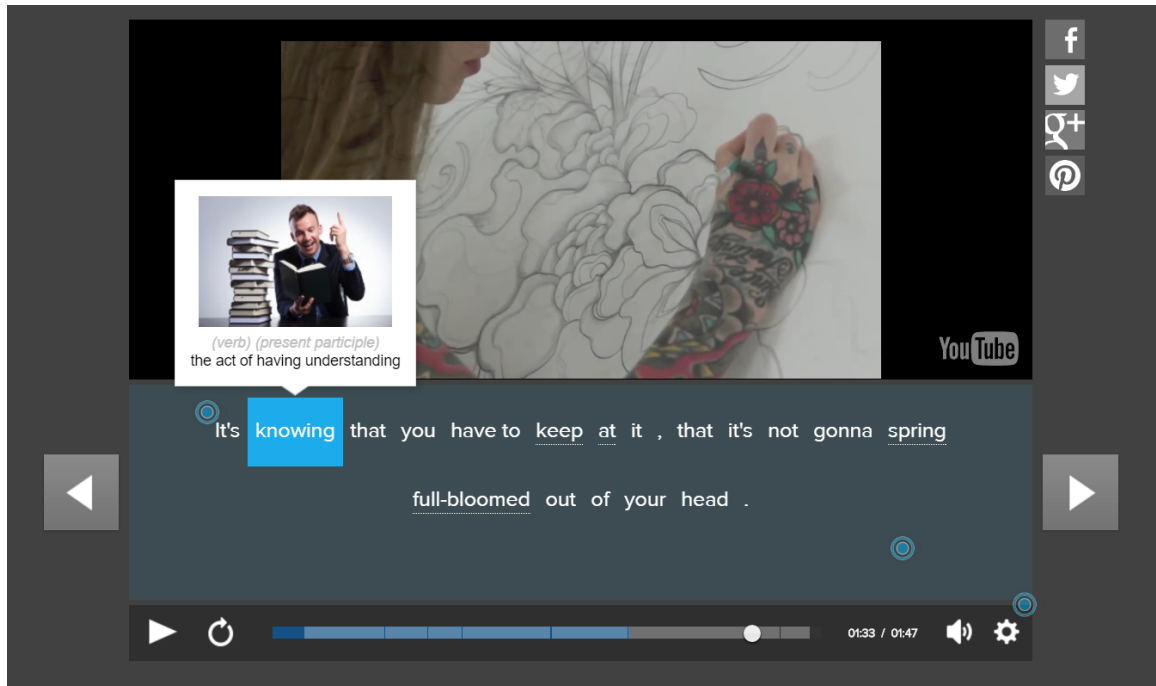
---

[1]http://www.fluentu.com/

Figure 2.1: Video player from FluentU. After hovering over a word a definition is displayed.

2.1.2 Examination

Every video has a programmed quiz which becomes available after watching video. Quiz focuses on few chosen words from the video, firstly showing the definition and usage in various sentences, secondly testing those words. Test consists of multiple choice questions and short written answer questions. Following the completion of the quiz, application displays final accuracy.

2.1.3 Overview

Overall FluentU has very interactive, easy to use environment with good vocabulary and flashback system. It is helpful to students with different English level, be it newbie or advanced user. However, FluentU does not provide any speaking examination, helping user only with passive vocabulary.

2.2 YouGlish

YouGlish[2] is a free of charge, YouTube-based pronunciation dictionary. It searches over YouTube videos that have transcripts. User enters the word he wants to practice and search algorithm of YouGlish will display big number of videos containing the searched word. The video will automatically start from the relevant sentence. Every video has embedded subtitles displayed as the video is playing. User can replay the sentence, or switch to another video with different example of usage.

---

[2]http://youglish.com/

2.2.1 Video player

YouGlish uses integrated YouTube video player with no further features.

Figure 2.2: Searching the word courage in YouGlish

2.2.2 Overview

Youglish service lack any kind of examination, so testing is entirely up to the user. However, YouGlish provide fast searching algorithm over subtitles and vast database of videos. Main focus of YouGlish site is to show the user correct pronunciation by native speakers. Big number of videos showcase many possible usages of English phrase or word. YouGlish is very easy to use, but lack of additional features makes it only pronunciation dictionary.

2.3 Yabla

Yabla[3], similar to FluentU, is an on-line subscription tool that offers database of subtitled videos in six languages (Chinese, Italian, Spanish, French, German, English). Videos are sorted according to three difficulty levels - beginner, intermediate, advanced. Additionally, videos are sorted according to many categories such as Action, Economy/Business, History,

---

[3]https://www.yabla.com/

Leisure, News, Travel etc. After choosing a specific video, user is asked to choose his native language. There are 30 different language to choose from including English.

2.3.1 Video player

Features of video player:

1. Integrated dictionary. Yabla application is looking for translation on various on-line dictionaries based on native language user chose (for Czech those are English-Czech dictionaries and Czech-English dictionaries). Translation is often incorrect or missing.

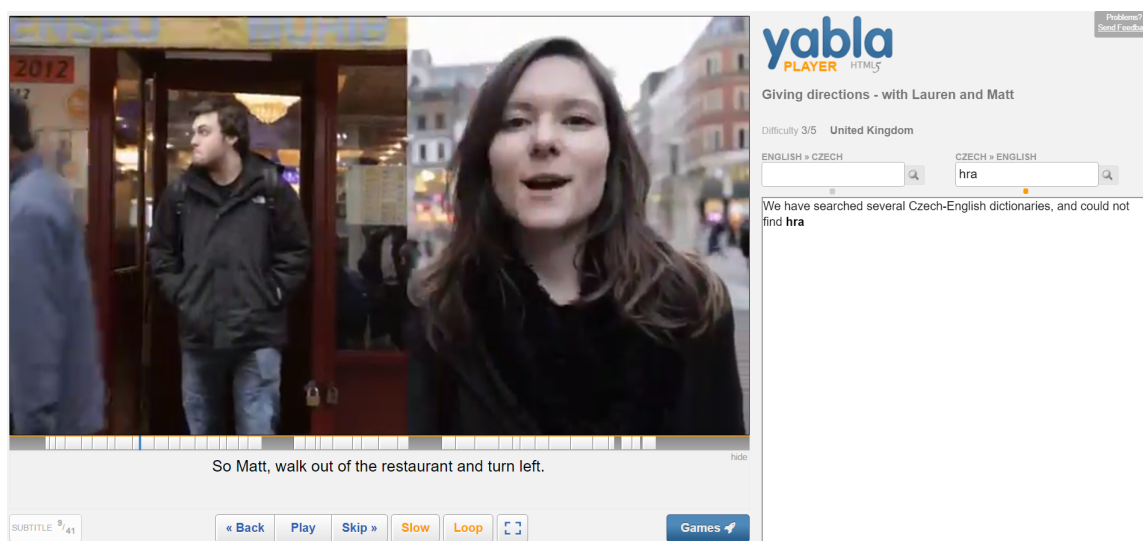2. Ability to replay entire video or video segment repeatedly.

Figure 2.3: Yabla video player with its unreliable integrated dictionary

2.3.2 Evaluation

Examination is done by playing games that are available for each video. There are two type of games to choose from - multiple choice or fill in the blank. Questions in these games are always sentences from subtitles missing one or two words, asking user to enter the correct answer. With each correct answer points are received. If the first attempt to answer the question was incorrect, user will get a second chance to correctly respond. The missing words are randomly generated. Each game consist of ten question. After all questions have been answered, user is evaluated with the final score.

2.3.3 Overview

Yablas primary focus is to improve listening skills by watching native speakers talk or sing. Yabla struggle to provide reliable translation tools, forcing user to use other party translation mechanisms. The missing words in games are any words including names of the places/people. This is due to random generating of words in games. Therefore, this evaluation process is far from optimal.

2.4 Summary

In general on-line learning foreign language applications primarily focus on grammar, passive vocabulary and listening skills. Many of them have difficulty to implement objective and reliable evaluation mechanisms that would provide a helpful feedback to users. This leaves a big room to potential Web services that would emphasize the active vocabulary by comparing users speech to expected output. Comparison between three examined services is shown in table 2.1.

	FluentU	YouGlish	Yabla
Video player	Interactive, a lot of features especially hover dictionary	Integrated YouTube video player	Interactive, dictionary feature that not always provide reliable results
Evaluation	Examining words of the videos, quizzes	None	Filling in missing words from the sentences in the videos
Gamification	Entertaining quizzes, score for completing tasks and quizzes	None	Score for completing build-in games

Table 2.1: Comparison between on-line services FluntU, YouGlish and Yabla

Chapter 3

Analysis and Decomposition

This chapter focuses on analysis of application requirements and application design. Entire problem is being decomposed into smaller parts, each part will be described in following sections: process of creating and managing videos and theirs interaction with user, graphic models, user behaviour, embedded gamification tools, and design of the database.

3.1 Video management and user interaction

There are two key elements of this application to enable user to train pronunciation. One is video, that is vital and central component, the other is video segment. Video segment is a short, logically connected part of the video. It is usually one sentence or two consecutive sentences. Every video contains number of segments depending on length of the video and occurrence of the human speech within the video. Every video segment has to be correctly subtitled and possible to played repeatedly if requested by the user. Video will behave as a parent of its segments, meaning segments will inherit most properties of the video, however, some of these properties may be set individually for each segment e.g. difficulty level of one segment might be different to the difficulty level of the video.
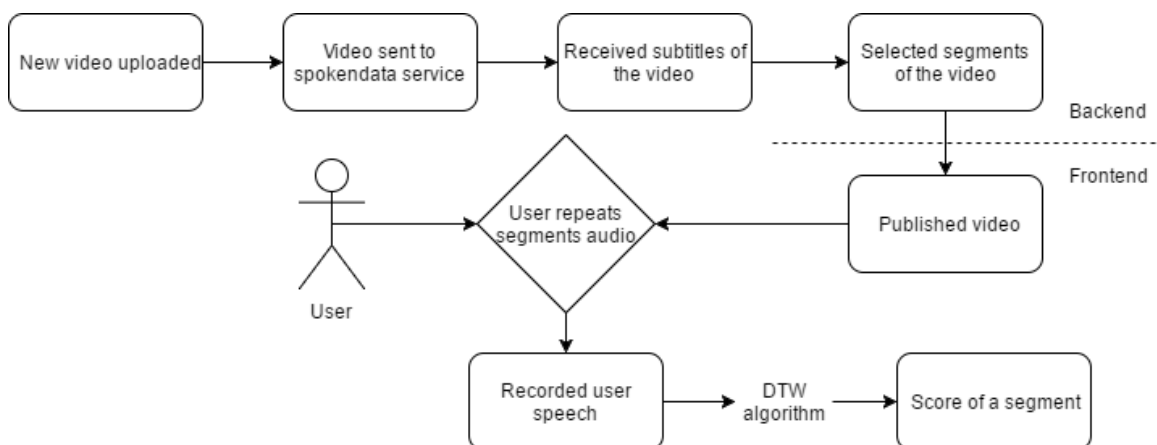

Figure 3.1: Process of video creation and score evaluation

Life-cycle of video is shown in the figure 3.1. Firstly, administrator uploads new video. Upon uploading the video it is send to spokendata service[1]. Spokendata will auto-generate

---

[1]Application for speech transcription https://www.spokendata.com/

subtitles for this video. Auto-generated subtitles need to be edited and segments of the video logically ordered. After these modifications the new edited subtitles will be downloaded. Administrator will then select segments he finds appropriate and change properties of each segment if necessary. Then video is published, thus getting visible to all users. Users play the video and after every segment they can repeat the audio of the segment using their microphone device. Recorded speech is compared to original recording via dynamic time warping algorithm. Similarity score, that is a result of this comparison will be returned back to the user.

3.2   Graphic mock-up

There are two basic user interfaces: one for front-end, one for back-end. Both mock-ups evolved throughout the development of the application, improving little things and responding to follow-up needs. The general aim is to make it easy, intuitive, and enjoyable to work with the application.
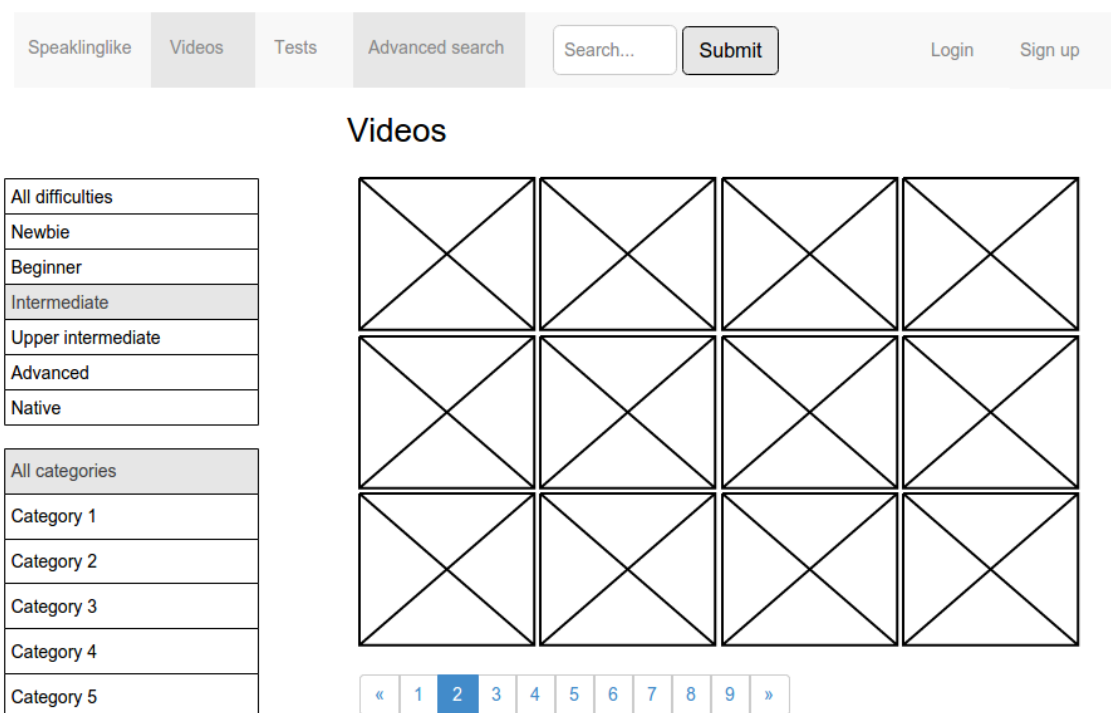
3.2.1   Front-End

Figure 3.2: User interface of front-end page 'speakinglike/videos'

Figure 3.2 shows graphic mock-up of the front-end page 'speakinglike/videos'. Main navigation bar is at the top of the page. This bar will be on all pages and subpages as a part of layout. On the left of the navigation bar, there are links to refer to all pages on the site. Links are followed by simple search form, that will help user to search among videos. On the right of the navigation bar, there are log in and sign up buttons, to get into the system or create user account. On the left of the page, there are two vertical menus, both

to help user choose videos for his skill level and point of interest. Thumbnails of the videos will fill in the central part.

3.2.2 Back-End

Figure 3.3: Graphic model of back-end page '/backend/videos'. This page is primary working page of an administrator.

Figure 3.3 displays design of the administration page 'backend/videos'. Again, top navigation bar will be primary navigation tool incorporated inside a back-end layout. It contains all important subpages as dropdown menus. Each dropdown menu will have options to create new item (video/tag...) and to view items (videos/tags/categories...). On the right of navigation bar, there is a dropdown menu with actions upon administrators account such as to change password, to log out, to view front-end page etc.

Videos will be displayed in the table. This table will provide all necessary information in a simple way and it will enable to perform various actions with ease. A filter above the table will help to extract only desired videos that administrator wants to work with. Mass action buttons will simplify operations on bigger number of videos.

3.3 User roles

There are two user roles in this application - administrator and common user. Figure 3.4 shows set of action for both these roles. Administrator and common user are not directly connected, neither one has a direct impact on the other role. However, they are indirectly connected, since administrator creates content of the application and user uses this content to train his pronunciation (videos and exercises).
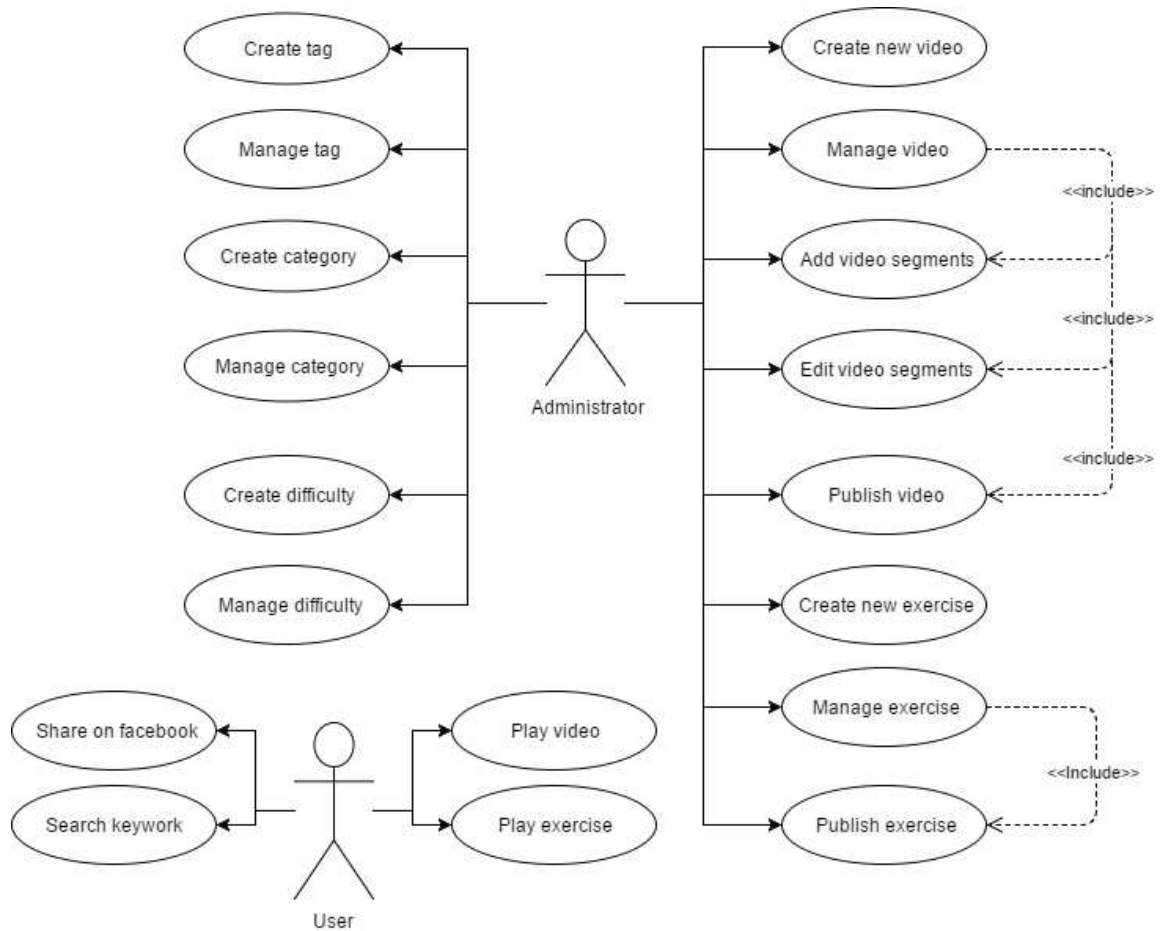
Figure 3.4: Use case diagram describes set of actions for administrator and user

3.4 Gamification tools

As mentioned in section 3.1 there are two primary entities in this application - video and video segments. Any gamification tool has to work with these elements and combine entertaining training experience with motivation and encouragement to play on. In spite of the fact that duration of videos vary, even the shortest videos will most likely be over minute in length to have any relevance. That is why I find more appropriate to make use of the video segments as the base of game called 'speak-like exercise'. This exercise will contain multiple video segments with similar properties. These segments will be played one after another, collecting part score during exercise. At the end of the exercise, the final score will be displayed with corresponding image. For example 'speak-like Donald Trump!' will be a collection of segments from different videos of Donald Trump. User will imitate every segment and at the end of the exercise user will be evaluated with total 'Donald Trump skill level'. If user reached very high similarity score he will be evaluated with very proud Trump picture, if user did mediocre, he will receive less happy Trump and so on. The user will be able to share the page on Facebook[2].

---

[2]https://www.facebook.com/

3.5 Database design

Implementation details to create ER diagram were obtained from scripts for course IDS[6]. ER diagram in figure 3.5 displays all tables of the database and relations between them. *Video* is the central entity of the database. This entity behaves as a parent to *Segment* table that inherits properties category, difficulty, and tags. However, properties difficulty and tags may be customized. There are three many-to-many relationships. The first is between entities *Video* and *Tag* where *VideoTagRelationship* behaves as an associative entity, the second is between entities *Segment* and *Tag* where *SegmentTagRelationship* is an associative entity, and the third is between entities *SpeaklikeExercise* and *Segment* where an associative entity is *SpeaklikeSegmentRelationship* table.
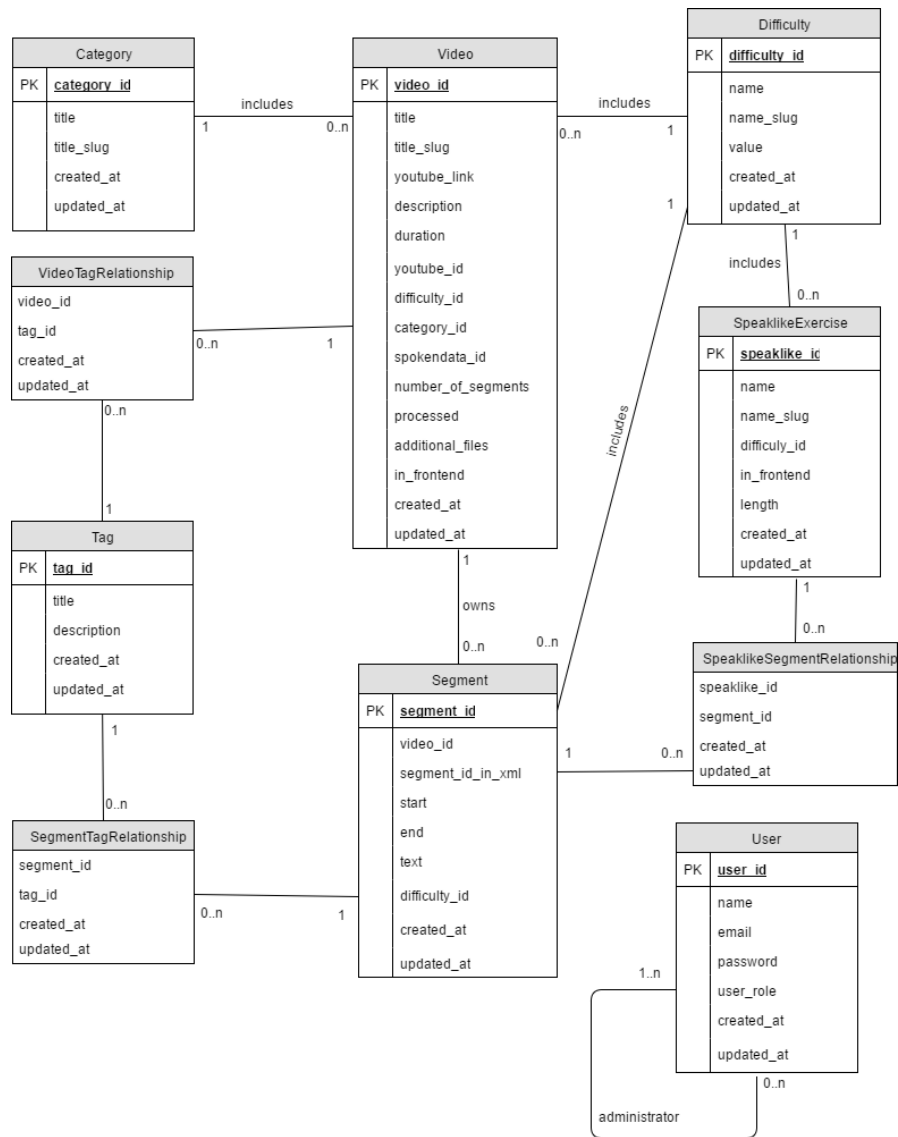
Figure 3.5: ER diagram of database

Chapter 4

Implementation

This chapter focuses on describing implementation details, moreover describing tools used in the application, namely dynamic time warping algorithm, third party services, supplied video player, developing environment, and other tools. Program has been developed over time which means that some changes/improvements have been incorporated after testing or natural change in requirements.

4.1 Brief introduction to Dynamic Time Warping (DTW) algorithm

Source materials for this chapter are ZRE (Processing of Speech Signals) course scripts [4], lecture about DTW algorithm of the same course [3], and publication by Institute Of Space Technology in Islamabad, Pakistan [1]. Isolated words in DTW are recognized by determining what reference word corresponds to the test word[1]. Reference words are stored in dictionary and every reference word is assigned parameter matrices:

$$\mathbf{R}_1...\mathbf{R}_N$$

Words are represented by sequence of vectors (more than one vector). Similarity between reference and test words is done by computing distance between reference sequence of the length R:

$$\mathbf{R} = [\mathbf{r}(1), ..., \mathbf{r}(R)]$$

and the test sequence of the length T:

$$\mathbf{O} = [\mathbf{o}(1), ..., \mathbf{o}(T)]$$

DTW aligns these two sequences to create path (figure 4.1). In this figure, the reference sequence (stored template in the dictionary) is on the y-axis, and the input sequence (to be compared) is on the x-axis. The input 'SsPEEhH' will be matched against the template 'SPEECH' in the system's dictionary. The distance between the sequences $\mathbf{O}$ and $\mathbf{R}$ is given as minimum distance over set of all possible paths (all possible lengths, all possible courses) *D(O, R)*. The global distance score for a path is a sum of local distances that make up the path. If *D(O, R)* is the global distance up to *(o,r)* and the local distance at *(o,r)* is given by *d(o,r)* then:

$$D(o, r) = min[D(o - 1, r - 1), D(o - 1, r), D(o, r - 1)] + d(o, r)$$

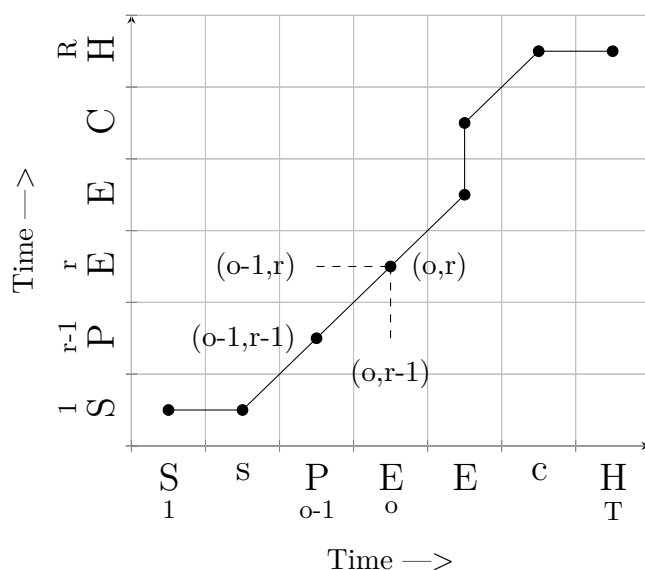[1]In this application test word is a word recorded by user using his microphone device

Figure 4.1: DTW alignment of two templates. The template on the x-axis is a test sequence of vectors, and the template on the y-axis is a reference sequence of vectors. Taken from [1].

D(1,1) = d(1,1) is the initial condition. The final global distance $D(O, R)$ gives us the overall matching score of the reference and the input. Feature extraction from audio file is out of the scope of this thesis.

4.2 Glplayer

Glplayer is video player programmed in JavaScript that is being developed by company ReplayWell[2]. Version used in this application is glplayer_0.8 that was the latest version at the time. It runs in two modes: *watch* and *speak*. Both modes have different features, functionality, and requirements. Both modes need source XML file to load source video, its subtitles, and other features. Source video may be local video or YouTube video. Glplayer has implemented algorithm to compare referential audio of video to recorded speech of user. Result of this comparison is returned as similarity score.

4.2.1 General functionality

Features   of glplayer shown in figure 4.2:

1. Play modes on top left corner. User can switch between these modes any time without pausing the video. These buttons also detect progress of video played in percentage. Default mode is explicitly configured in configuration file. Possible to enable only one mode if required (figure 4.2.1).

2. Play/Pause button. Clicking anywhere on the video screen has the same effect (figure 4.2.2).

---

[2]https://www.replaywell.com/

3. Repeat button to repeat video segment(figure 4.2.3).

4. Video speed customization button (figure 4.2.4).

5. Settings button where user can customize subtitles colour and size (figure 4.2.5).

6. Full-screen button. Or 'f'/'F'[3] keys (figure 4.2.6).

7. Segments section. Currently playing segment is marked yellow, already played segments are marked green, and segments not yet played are grey (figure 4.2.9).

Figure 4.2: Glplayer version 0.8. On the top left corner there are modes we can play the video in. 'Play/pause' button is on he bottom left corner. Next to play button is 'Play previous segment' button that is available only in *watch* mode. Next on the right is 'Repeat segment' button, followed by 'Play next segment' button that is available only in *watch* mode, then time played and total time of the video are displayed, then 'Speed customization button'. On the bottom right are two buttons: 'Change settings' button and 'Full-screen' button. Above this bar is a segments section, every segment is clickable. On the top of segments section are subtitles for currently played segment.

Source XML file is necessary to run video in glplayer. This file contains following informations: recording ID, recording title, video duration, video source (either local or YouTube link), details for all segments of the video, segments enabled in *watch* mode, segments enabled in *speak* mode, paths to three files necessary for *speak* mode - xml, wav, and bnfea files. Furthermore, detail of each segment consist of additional information. One such segment is shown below:

---

[3]ASCII characters 102 and 70

```
<segment>
        <id>29745_0000000audio_0000A_0006520_0006765</id>
        <start>65.2</start>
        <end>67.65</end>
        <text>A potential disaster for our country.</text>
</segment>
```

Segment ID is a predefined string with strict form:
{Recording ID}_0000000audio_{speaker ID}_{start time}_{end time}

1. {Recording ID} = Recording ID must correspond to ID of the same video in spokendata database.

2. {speaker ID} = Speaker ID that is always '0000A'.

3. {start time} = Starting time in seconds with fixed point to two decimals.

4. {end time} = Ending time in seconds with fixed point to two decimals.

It is necessary to have every segment ID in this form, since this ID will be referenced in *speak* mode by another XML file.

4.2.2   Watch mode

Apart from general features, *watch* mode provide following features:

1. Playing entire video without interruptions.

2. Previous segment button and next segment button (figure 4.2.7).

3. Subtitles displayed for all segments that are enabled in *watch* mode in source XML file (figure 4.2.8).

This mode is preparation for training process in speak mode.

4.2.3   Speak mode

Apart from general features, *speak* mode shown in figure 4.3 provide following functionality:

1. Subtitles displayed for all segments that are enabled in *speak* mode in source XML file (figure 4.3.1).

2. Speech recording. After a video segment is played, user is asked to repeat the audio of he segment. User need to enable microphone device to do so. After clicking the microphone timer appears, and glplayer start to record users speech. Recording stops either when user clicks the microphone again or timer runs out (figure 4.3.2).

3. Speech similarity evaluation. After it records users speech, glplayer evaluates similarity score between referential recording and users speech. After computing the score, glplayer returns the score to the user using DTW algorithm (figure 4.3.3).

4. Storing recorded user speech. User will be able to replay his latest speech after pressing 'My voice' button (figure 4.3.4).

5. Evaluation for every word of video segment. User will be able to see similarity score for every word of the segment (figure 4.3.5).

6. To replay entire segment or just one word. After hovering over any word in subtitles, user will be able to listen just to this particular word alone. After pressing 'Reference' button entire segment audio will be played (figure 4.3.6).
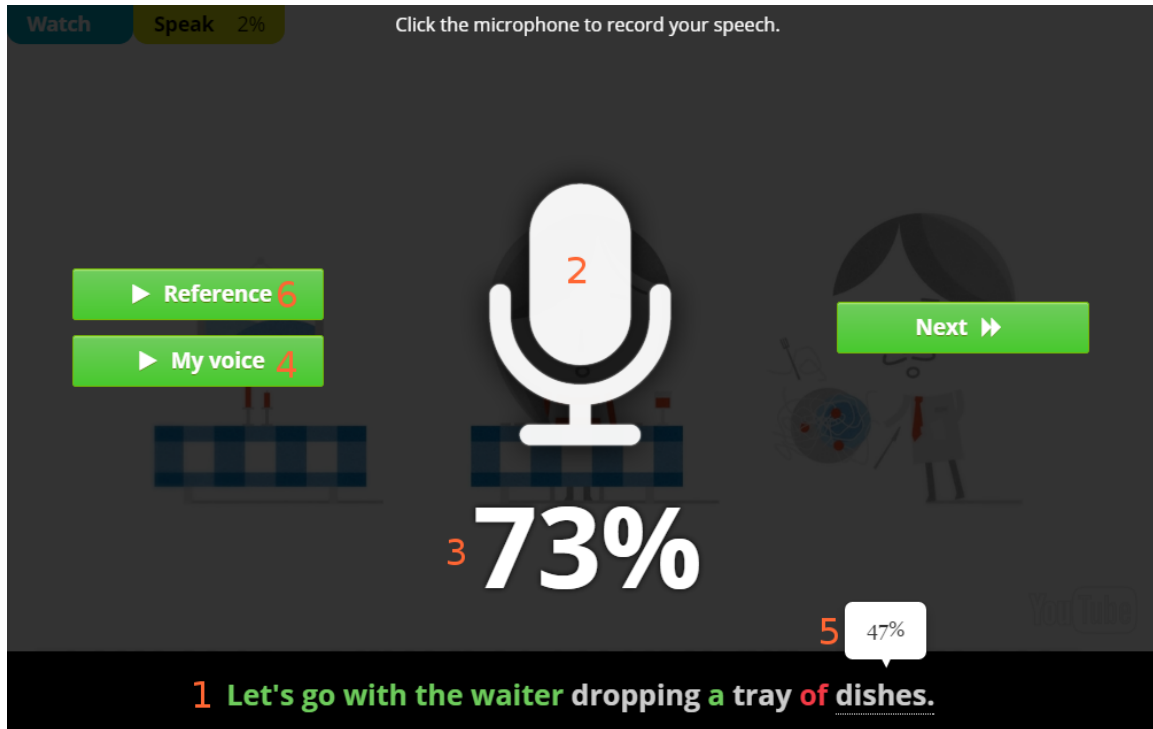

Figure 4.3: Speak mode. Microphone is used to record speech. Total similarity score is displayed above subtitles. Similarity score for particular word is shown after hovering over the word. User is able to replay reference recording and his recorded speech. Clicking 'Next »' button will cause video to continue.

Additional files required for this mode[4]:

- XML file containing phonemes of particular words and time alignment of these phonemes and words

- wav file containing audio of the video for word playback

- bnfea file containing pre-calculated features used in DTW

4.3 Video creation and video management

All videos page (figure 4.4) is the page that contains all created videos in the system. Since video is primary entity of this application, 'all videos' page is primary working page for administrator.

---

[4]These files are generated in spokendata service (https://www.spokendata.com/) with alignment process. Section 4.3.1 describes the process.
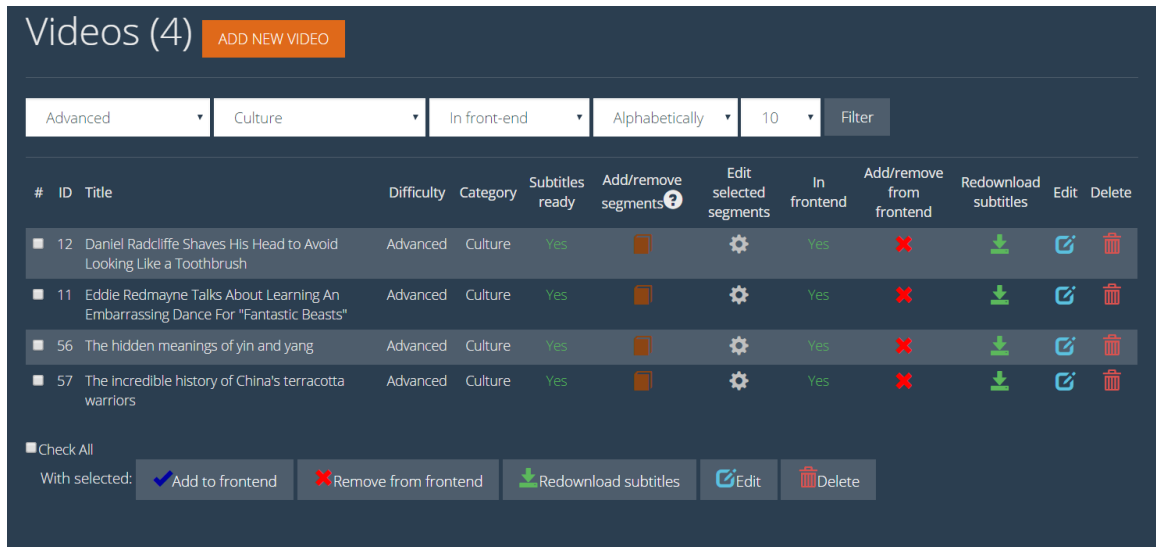
Figure 4.4: All videos page. Filter is located on the top to make it easier to work with desired videos. Table is the central part of the page. One row in the table represents one video. Some information about the video and all operation upon the video are available in every row. A brown book represents add/remove segments functionality, a grey cog represents edit segments operation, add/remove video from front-end is represented either by blue OK sign or by red remove sign depending on whether video is in front-end or it is not. The green download glyph is used to re-download subtitles for the video, light blue edit glyph is used to edit information about video and to view it. The last glyph on the right is a delete button. On the left of every video title there are check-boxes that enable to select videos we want to use for mass operations. Mass operations are positioned on the bottom of the page.

Operations upon created videos

Segment selection and attributes customization is an operation in which administrator selects what segments should be displayed for this video, furthermore, administrator customizes some of the attributes of selected segments (difficulty, tags). This operation is part of the process of publishing video described in chapter 4.3.2.

Add/remove video from front-end means to publish video and make it visible to users or remove already published video from front-end making it invisible to users. Only video that has subtitles ready can be published. All other operations (excluding segment selection) may be performed only on videos that are **not** in front-end.

Re-download subtitles is an operation that downloads the latest subtitles from the spokendata server. Application will recreate source XML file, and send request to spokendata server to start new alignment process to produce additional files (see 4.2.3). Old additional files will be deleted.

Video editing is a simple operation where administrator can update some information of video such as title, description, difficulty, category, and tags.

Video deleting is an operation where video is entirely removed from the system, including all its references and associative entities. Not every video, however, can be deleted. Requirements to safely delete video:

- Video must not be in front-end

- Video segments must not be part of any speak-like exercise.

If requirements are met following data are deleted:

- Tags of all video segments are deleted from '*segment_tag_relationship*' table.

- All local files are deleted (source XML, XML for speak mode, wav, bnfea).

- All video segments in are deleted from '*segments*' table.

- All video tags are deleted from '*video_tag_relationship*' table.

- Video itself is deleted from '*videos*' table.

Mass operations upon videos:

1. Add to front-end. Videos that are not in front-end will be published. Other videos without change.

2. Remove from front-end. Videos that are in front-end will be removed. Other videos without change.

3. Re-download subtitles. Download latest subtitles for all videos that are not in front-end. Other videos without change. Administrator will be notified how many unsuccessful re-downloads there were.

4. Edit videos. Only these attributes can be mass edited: category, difficulty, tags.

5. Delete videos. If video is in front-end or any video segment is part of one or more speak-like exercises, the video will not be deleted. All other videos will be deleted. Administrator will be notified with number of unsuccessful deletions.

Creating new video and publishing it

1. Filling in a form. Simple form with attributes: title, YouTube link[5], brief description, select difficulty, select category, and select tags for the video. Brief description and tags are not required, other attributes are compulsory.

2. Selecting segments and editing them (chapter 4.3.2).

3. Publishing video. After everything is done, video is ready to be published using 'Add to frontend' button.

---

[5]This application works exclusively with videos from YouTube (https://www.youtube.com/)

4.3.1 Communication with spokendata

All HTTP communication in this application is done via Guzzle 6 PHP HTTP client[6]. Documentation of guzzle [2] provided all implementation details. Figure 4.5 displays communication between the speakinglike (SL next) server and spokendata (SD next) server via spokendata API[7].
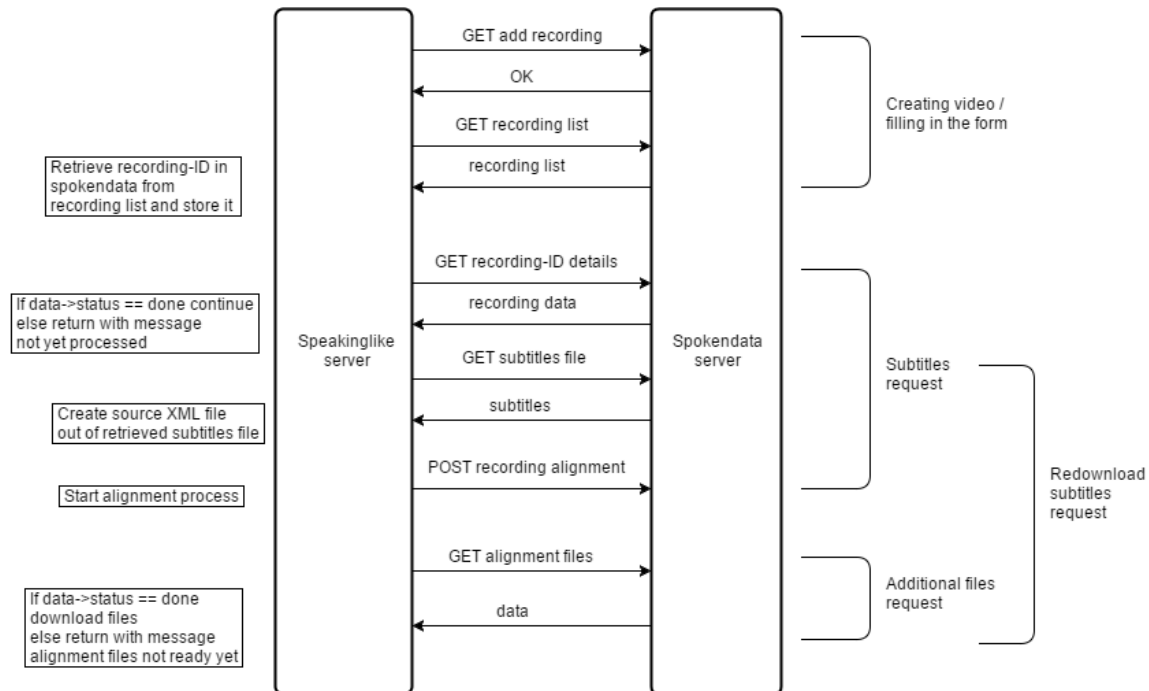

Figure 4.5: HTTP communication between speakinglike server and spokendata server. Arrows from speaklike server to spokendata server represent GET or POST requests, and arrows from spokendata server to speaklike server represent expected responses (code 200).

Speaklike server communicates with spokendata server on four occasions:

1. Upon video creation.

2. Download subtitles of newly added video.

3. Download additional (alignment) files download of newly added video.

4. Re-download subtitles of old video.

Video creation

When a new video is added, SL server sends GET request to add recording to SD database. SD will assign ID to this recording. SL then sends another GET request to get list of all recordings. SD server will send in response message with body containing list of all recordings as an XML file. In this recording list SL will look for just added recording and get its SD ID. If everything went well a new video record is added to *videos* table with attribute spokendata_id corresponding to SD ID obtained from recording list.

---

[6]https://github.com/guzzle/guzzle
[7]https://www.spokendata.com/api-for-developers

Download subtitles - new video

After a new video is added, it is not immediately ready to be worked with. Subtitles need to be downloaded to create source XML file. It takes some time for SD to process video and generate subtitles, that is why subtitles are not ready at first. After administrator checks for subtitles availability[8], SL server will send GET request to acquire recording details. Response body from SD server will contain data file that will include status of this recording. In state of processing a video, this status attribute will have value 'processing' and administrator will be informed that video is not yet processed. If state attribute has value 'done', video is already processed and SL server will send another GET request to obtain subtitles file. Response body from SD server will contain subtitles file. This subtitles file will be used to create source XML file. SL server will then send POST request to start alignment process.

Download additional (alignment) files - new video

Additional or alignment files are files that are necessary for correct functionality of *speak* mode (see 4.2.3). Administrator is allowed to add video to front-end only after these files are downloaded. Alignment process is already started and it takes some time (about 3 minutes). When administrator checks for additional files availability[9], SL server will send GET request to get these files. Body of the response from the SD server will contain data XML file with attribute status. If value of this attribute is 'processing', files are not yet generated and administrator is informed with such message. If the value is 'done', data XML file will also contain links to additional files. Subsequently, these files (xml, wav, bnfea) will be downloaded using curl library[10] and stored on SL server.

Re-download subtitles - old video

In case of inaccuracies in subtitles or illogical segmentation of subtitles, administrator can edit subtitles and align segments on SD website as shown in figure 4.6. To update outdated subtitles on SL server, re-download functionality is implemented. Constraints to re-download subtitles:

- Video must not be in front-end

- Old segments of the video must not be part of any speak-like exercises.

Re-download functionality if constraints are overcome: SL server will send GET request to obtain new subtitles file. Response body of the SD server will contain latest subtitles file[11]. SL server will receive new subtitles, SL will update source XML file for this video and send POST request to start alignment process for new subtitles. When administrator checks for availability of additional files, SL server will send GET request to obtain additional files. Response body from SD server will contain status value. If this value is 'done' new

---

[8]Administrator can do so by clicking specific button that appears when subtitles are not yet downloaded.

[9]Administrator can check for availability of additional files either 1)by clicking specific button that appears when source XML file is created but additional are not yet acquired or 2)by attempting to add video to front-end. Second approach will do: check for additional files, and if available, download them, update source XML and add video to front-end.

[10]https://curl.haxx.se/

[11]Note that there is no need to check for value of status. It is always 'done' because transcription process had been done previously - it is performed only when new videos are added
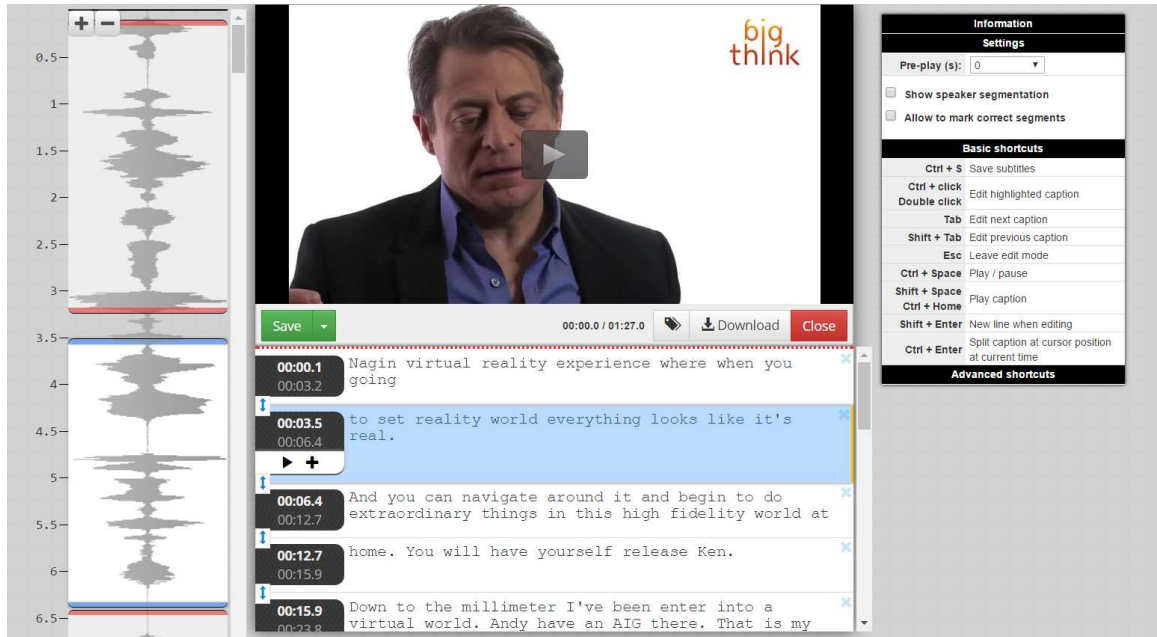
Figure 4.6: Editing of subtitles in the SpokeData application. On the left we can align segments so they contain logical sequence, on the bottom we can edit subtitles themselves or add another segment with '+' button. Some useful shortcuts are displayed on the right. We can play video with video player in the center.

additional files will be downloaded using curl library and stored on SL server. Old additional files will be deleted and paths to additional files in source XML file will be updated.

4.3.2 Segment selection and editing

Segment selection is a process when administrator chooses what segments will be displayed in glplayer. Selected segments will be stored in *segments* table. Such segments then have following functionality:

- Displayed in glplayer. Subtitles of segment shown while playing the segment.

- Fully clickable and referenceable.

- Independent attributes: difficulty and tags.

- Ready to be used for speak-like exercise.

Process of selecting segments is very easy and quick. Requirements for this operation is having source XML file[12] already created for the video.

Editing of segments is follow-up operation. Two attributes can be customized for every segment: difficulty and tags. Implicit values of these attributes are inherited from the video to which segment belongs. Every difficulty have different value (e.g. beginner=2, intermediate=3 etc.) and when difficulty of particular segments are edited according to their

---

[12]Requirements to create source XML file is to retrieve subtitles file from spokendata server. Note that alignment files are not necessary for this operation.

real difficulty, the calculation of overall difficulty for group of segments is more accurate. Unique tags for group of segments will make it a lot easier to filter out these segments and use them accordingly.

4.4 Creating speak-like exercise

Speak-like exercise (figure 4.7) is a simple game that contain several segments from different videos with similar properties. These segments are played in defined order. For each segment user is assigned score. This score will contribute to total score that is calculated with formula:

$$total\ score = \frac{\sum_{n=1}^{n=number\ of\ segments} score\ for\ segment\ n}{number\ of\ segments}$$

Formula is based on equal weight of all segments. Total score range is 0-100. After completing the exercise total score for the exercise will be displayed with corresponding picture. There are four different pictures for four different results: score above 80, score 60-80, score 40-60, and score below 40. User will be able to try again this exercise or share it on facebook. Every part of this exercise is played in *speak* mode. Although, user can play one part many times, score will be calculated only from the first try. Creating of speak-like exercise consist of three steps:

1. Adding segments

2. Sorting segments

3. Setting final properties
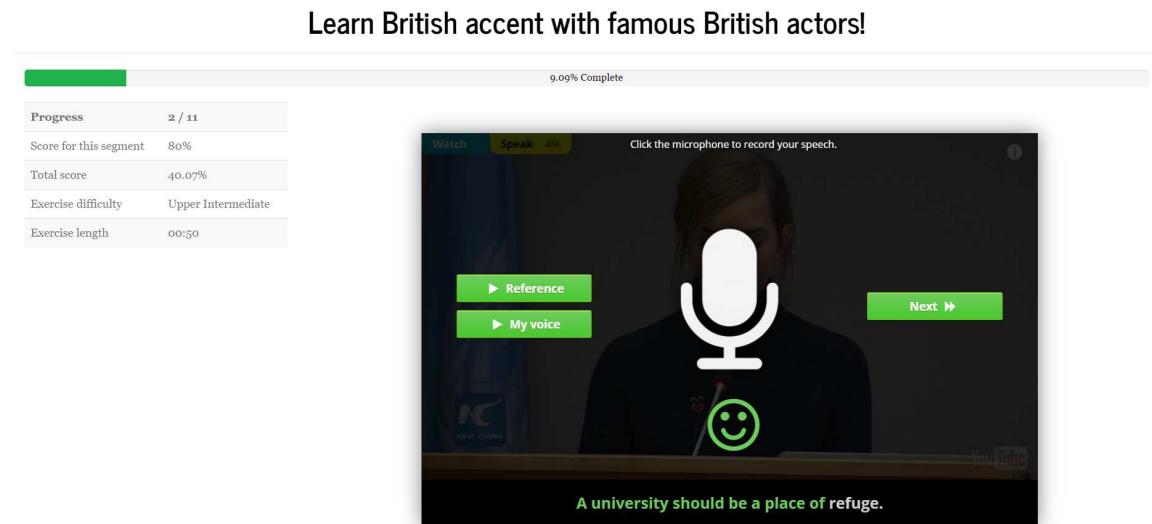
Figure 4.7: Preview of speaklike exercise. On the top is a progress bar with current progress, on the left is the table that provide general information about exercise and information about score for current part and total score that is updated every part. After playing the current part click 'Next »' to progress in exercise.

4.4.1 Adding segments

First step to create speak-like exercise consists of choosing appropriate segments (figure 4.8). As database contains huge amount of segments there is a build-in filter to help with finding the right ones. To add segments only check checkboxes corresponding to desired segments and click 'Add segments' button to stay on the page and add some more or 'Continue' to proceed to next step. Added segments will be displayed on the top of the page.

Figure 4.8: Step 1 - adding segments. Filter (difficulty='Beginner', category='Politics and Society') applied to segments producing 3 results. On the top are IDs of the segments that have already been added (456,459).

4.4.2 Sorting segments

Second step is to order segments. Position of segments can be changed by arrows. Unwanted segments are easily removed by bin button. To get back to step one and add additional segments press 'Return' button. Press 'Continue' to proceed to third step.

Figure 4.9: Step 2 - sorting segments. Segments are ordered using arrows. Unwanted segment can be easily removed. If administrator wants to add another segments, he can do so using 'Return' button. Segments previously added will be remembered within session.

4.4.3 Finishing

Final step consists of filling in the form. Average difficulty value is pre-calculated mean value of all added segments. Difficulty can be edited. Four uploaded picture will be stored locally and one of them will be displayed at the end of the exercise. If users total score will exceed 80 the first picture will be displayed, second picture is for score 60-80, third for score 40-60, and fourth for 0-40. If no pictures are uploaded, no picture will be displayed after finishing the exercise.

Figure 4.10: Step 3 - finalizing. Final properties to be set and pictures uploaded. Pictures should reflect users result. For instance for exercise 'Speak like true narrator', picture for score above 80 should represent very happy and proud narrator, for score 60-80 less happy narrator and so on.

4.5   Authentication

Authentication used in this application is build-in Laravel authentication[13] (see section 4.7). This authentication provide all necessary tools: registration, login, remember token. There are two roles:

- Administrator - role 1

- Common user - role 2

---

[13]https://laravel.com/docs/5.3/authentication

Role middleware[14] was created to assign permission for these roles. After logging in, administrator is redirected to administrator page, and common user is redirected to home page. No account can intervene in any other account, be it administrator account or user account. Registration will always result in creating role 2 - common user account. Naturally, non-administrator account is not allowed to enter administration.

4.6 Search

To search for particular word or phrase, there is implemented search functionality. There are two kinds of search:

- Quick search

- Advanced search

Both kinds of search look for keyword(s) among selected segments of all published videos. The difference is that in advanced search user can restrict difficulty and category of desired output. Quick search is implemented with Laravel command (see section 4.7) that sends select statement to MySQL:

$$select \; [all \; parameters \; necessary] \; from \; segments \; where \; text \; like \; '\%query\%'$$

Advanced search is very similar but extended select statement, that performs join on *videos* table first, because *segments* table does not contain category information. Then a function is implemented such that if difficulty and category were entered, it selects only corresponding results. Result of the search shown in figure 4.11 will produce videos where the searched keyword is present. Sentence with the keyword is displayed under the video. Video will start at the relevant sentence.
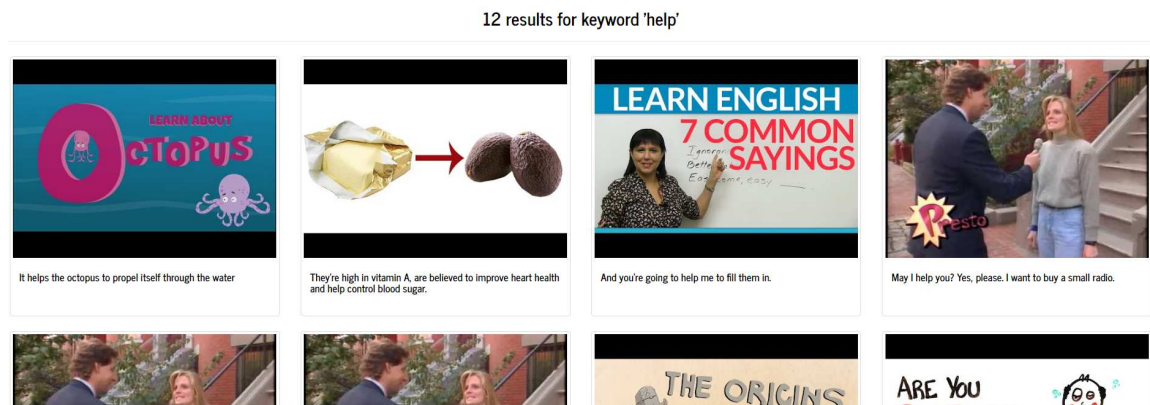

Figure 4.11: Output of search. After opening any link, video will start at sentence containing keyword.

4.7 Tools used

Following tools have been used for implementation.

Laravel 5.3 [15] Laravel is one the most popular PHP frameworks. It has excellent documentation [5] that was primary source of implementation details. Some features:

- Very fast blade template engine

- Fast and simple ORM (Object-relational mapping) providing very convenient way to work with database

- Modularity across application is build with composer packages. Any developer can contribute his own package.

- Directory structure configurable.

- Reverse routing.

- A lot of build in functionality using artisan[16] (authentication, migration...)

I have decided to use Laravel since I had previous experience with it and it is easy to use and learn. It has excellent reviews too. Additional packages used: laravel debugbar[17], intervention image[18], guzzle 6[19], html extension[20].

Bootstrap [21] Both front-end and back-end use bootstrap. It has a lot of pre-coded features that are easy to use and customize and it guarantees responsiveness. For back-end a superhero[22] theme was used and for front-end journal theme[23]. Both themes were created by *Thomas Park* under MIT licence[24] and are based on bootstrap.

PHP [25] PHP is a server scripting language that is used in laravel framework. Concrete release used in this application is 7.0.6.

JavaScript [26] JavaScript is a client programming language. Glplayer is programmed in JavaScript and other small features were implemented mainly using jQuery[27] library.

MySQL [28] MySQL is open source SQL database and is developed by Oracle Corporation[29]. Laravel mediates all database communication. MySQL is used to store all tables and adminer[30] is used to manage database.

---

[15] https://laravel.com/
[16] https://laravel.com/docs/5.3/artisan
[17] https://github.com/barryvdh/laravel-debugbar
[18] https://github.com/Intervention/image
[19] https://github.com/guzzle/guzzle
[20] https://github.com/LaravelCollective/html
[21] http://getbootstrap.com/
[22] https://bootswatch.com/superhero/
[23] https://bootswatch.com/journal/
[24] https://github.com/thomaspark/bootswatch/blob/gh-pages/LICENSE
[25] https://secure.php.net/
[26] https://www.javascript.com/
[27] https://jquery.com/
[28] https://www.mysql.com/
[29] https://www.oracle.com/
[30] https://www.adminer.org/

4.8   Final look

Final outlook of the main front-end part is shown in figure 4.12. Main navigation bar is at the top of the page and is part of layout. Part of the navigation bar is quick search form that enables to search over subtitles of videos. Another navigation is on the left. This navigation help users to find videos of their interests and for their English skill level. Central part contains thumbnails of videos sorted according to their popularity. Every video thumbnail contains video title and difficulty.
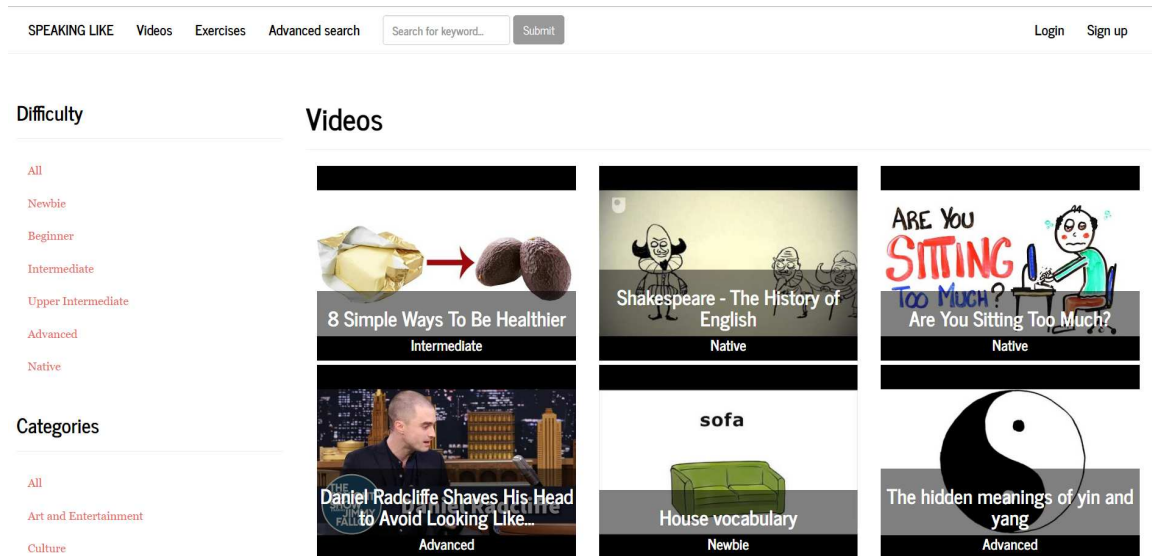

Figure 4.12: Home page.

Result of speak-like exercise is shown in figure 4.13. User will be awarded with points and picture. What picture will be displayed depends on user's final score. The user can try again the exercise or share the page on Facebook.

Figure 4.13: Page that is displayed after the user finishes speak-like exercise.

Chapter 5

Deployment of application and testing

The application was deployed on 12th of April on the address *https://www.speakinglike.com/*. However, it took some time to create enough video content, so application was not published at once. On 19th of April with over 40 videos added to front-end, application was published. Usability testing for this application consisted of:

- remote usability testing using on-line analytics tools

- usability testing using on-line questionnaire

5.1 Smartlook

First on-line tool used to observe user behaviour was Smartlook[1]. Smartlook captures what users exactly do on website. This tool does not gather data, so I decided to watch some recordings to find out whether users are behaving in way I expect. This revealed that users generally found website easy to navigate and use.

Smartlook also introduced heatmaps functionality. This is still in beta so few heatmaps did not work. Those that did work revealed that primary focus of users were first few videos. Users very rarely scrolled down to see more videos, they usually clicked on the first ones. Also, users were a lot more interested to filter videos according to their difficulty on contrary to filter according to their category. Furthermore, videos that caught the most attention were the ones that had some information value. For instance '8 simple ways to be healthier' was the most popular at the time. Because of this I decided to add attribute *number of views* to every video. Videos will be sorted according to number of views. That will result in having the most popular videos at top positions. I explicitly incremented value *number of view* for videos that have been popular before and for videos that had potential to be appealing (they possess additional informational value).

5.2 Google Analytics

Google Analytics[2] was used as a main tool to gather data about user behaviour at the website. Google Analytics was deployed immediately after publishing the application. Figure

---

[1]https://www.smartlook.com/
[2]https://analytics.google.com/

shows data about users from 19th of April to 13th of May. A lot of new applications struggle in bounce rate department, which means that a lot of visitors tend to leave website immediately after visiting it. However, bounce rate of this application was 34.47% that is very encouraging number. Of course, taking into account very brief period that website was available, it might be highly misleading. Average 5.74 pages per session is another hint that users found the website worth investigating.
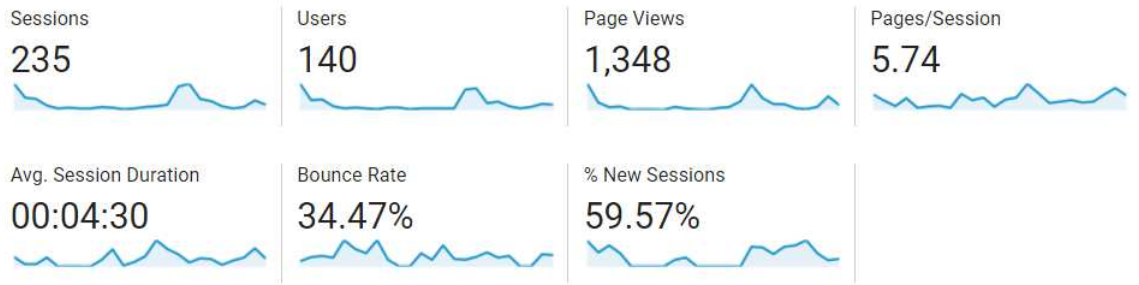

Figure 5.1: Data for web application *speakinglike.com* 25 days after deploying.

5.3 Questionnaire

The last part of testing included simple questionnaire to get feedback from the users. Twenty-two respondents answered following ten questions:

Question 1   IS ENGLISH YOUR MOTHER TONGUE?
Yes = `9.1%`
No = `90.9%`

Question 2   HOW OFTEN DO YOU USE ENGLISH?
Every day - in work/school & at home = `45.5%`
Every day - in work/school = `31.8%`
Every week = `22.7%`
Every month = `0%`
Less than once a month = `0%`

Question 3   HOW OFTEN DO YOU TALK IN ENGLISH?
Every day = `40.9%`
Every week = `22.7%`
Every month = `9.1%`
Less than once a month = `27.3%`

Question 4   I FIND SHORT VIDEOS A GOOD WAY TO PRACTICE ENGLISH (scale 1-5, 1-absolutely agree, 5-absolutely disagree)
1 = `59.1%`
2 = `31.8%`

3 = 9.1%
4 = 0%
5 = 0%

Question 5 I find content of this application helpful to practice English pronunciation (scale 1-5, 1-absolutely agree, 5-absolutely disagree)
1 = 45.5%
2 = 31.8%
3 = 13.6%
4 = 4.5%
5 = 4.5%

Question 6 I find this app intuitive (scale 1-5, 1-absolutely agree, 5-absolutely disagree)
1 = 40.9%
2 = 31.8%
3 = 18.2%
4 = 9.1%
5 = 0%

Question 7 I find feedback from the video player precise (scale 1-5, 1-absolutely agree, 5-absolutely disagree)
1 = 0%
2 = 36.4%
3 = 54.5%
4 = 9.1%
5 = 0%

Question 8 Have you come across similar application focused on pronunciation training?
Yes = 22.7%
No = 77.3%

Question 9 What do you like about the application?
Three most common answers:

1. idea / it's different

2. simplicity, various difficulties and categories / interactive content

3. easy to start and fun / informative videos

Question 10 WHAT YOU DON'T LIKE ABOUT THE APP / WOULD LIKE TO IMPROVE (AND HOW)?

Three most common answers:

1. lack of content, better design

2. similarity score too generous / easy to achieve 100%

3. lack introduction page / some kind of tutorial

Feedback from the questionnaire was mostly positive. Some users found score evaluation 'too generous'. They think it is very easy to reach 100% score. Glplayer have implemented five score calibrations. The one set in the application is medium, so as part of the further development, it is possible to customize calibration value according to demands of the user.

Chapter 6

Further development

This project have potential to grow a lot bigger than it is. New functionality might aim to make the application educative as well as entertaining.

Development of user account

Although basic authentication is implemented, registered users do not benefit from creating new account. This might be the course of further development: to help users measure their progress, remember their history and suggest new videos based on acquired data, to track achievements etc.

Development of glplayer

Another features of glplayer might include:

- Add volume button - it was requested by users

- Implement hover dictionary - currently hovering over the word in *speak* mode leads to playing the word. Additionally it might be good idea to implement window that will display the definition of the word. This will make it a lot easier for user to understand unknown words.

Add new languages apart from English

Current application works for English language only. Speakinglike application is closely connected to spokendata service, and spokendata currently support transcription for following languages (excluding English): Russian, Chinese - mandarin, Spanish, Czech, Slovak. These languages could be easily added to speakinglike application as well. Only simple changes would be necessary.

Chapter 7

Conclusion

Web service for pronunciation training and foreign language learning is deployed on the address *www.speakinglike.com.* The application offer database with over 40 videos that are fully subtitled and are available for user in front-end. Another more than 40 are created, however, not yet in front-end, because subtitles of these videos were not yet corrected. Additional 2 speak-like exercises were created and published. The application provides easy to use interface for administrator to add videos/exercises and manage them. A lot of bigger and smaller bugs were fixed throughout development. The application was published only very recently and it is not yet indexed by Google, that means it cannot be found by Google search engine. Nonetheless, basic sitemap[1] was created and submitted to Google on 4th of May. It takes some time[2] for Google to assign index, thus website *www.speakinglike.com* is not yet foundable on Google.

Feedback from users who visited this application was in majority very positive and most of them expressed that they have not seen similar application and found the very idea very good and interesting. This leaves a lot potential to keep building on and design a perspective, educative, entertaining, and a lot bigger multilingual and multiplatform application in future.

---

[1]Sitemap is a way to tell Google about a new website that is not yet referenced across Internet https://support.google.com/webmasters/answer/156184?hl=en

[2]Time to assign index vary a lot, no official guarantees are given by Google. Unofficial experience ranges from 4 days to 1 month.

Bibliography

[1] Amin, T. B.; Mahmood, I.: *Speech Recognition using Dynamic Time Warping*. IEEE. 2008. ISBN 978-1-4244-3299-8.

[2] Dowling., M.: *Guzzle Documentation*.
Retrieved from: http://docs.guzzlephp.org

[3] Černocký, J.: *Speech recognition - Intro and DTW*.
Retrieved from:
http://www.fit.vutbr.cz/~grezl/ZRE/lectures/08_reco_dtw_en.pdf

[4] Černocký, J.: *Zpracování řečových signálů — studijní opora*.
Retrieved from:
http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf

[5] Otwell, T.: *Laravel 5.3 Documentation*.
Retrieved from: https://laravel.com/docs/5.3/

[6] Zendulka, J.; Rudolfová, I.: *Databázové systémy - studijní opora*.
Retrieved from: https://wis.fit.vutbr.cz/FIT/st/course-files-st.php.cs?file=
%2Fcourse%2FIDS-IT%2Ftexts%2FIDS_predn.pdf&cid=9969

Appendices

Appendix A

Poster

Figure A.1: Poster promoting application speakinglike.

Appendix B

Video

Short video to introduce this Web application is located on the address https://www.youtube.com/watch?v=VcLKqXb9fsA.