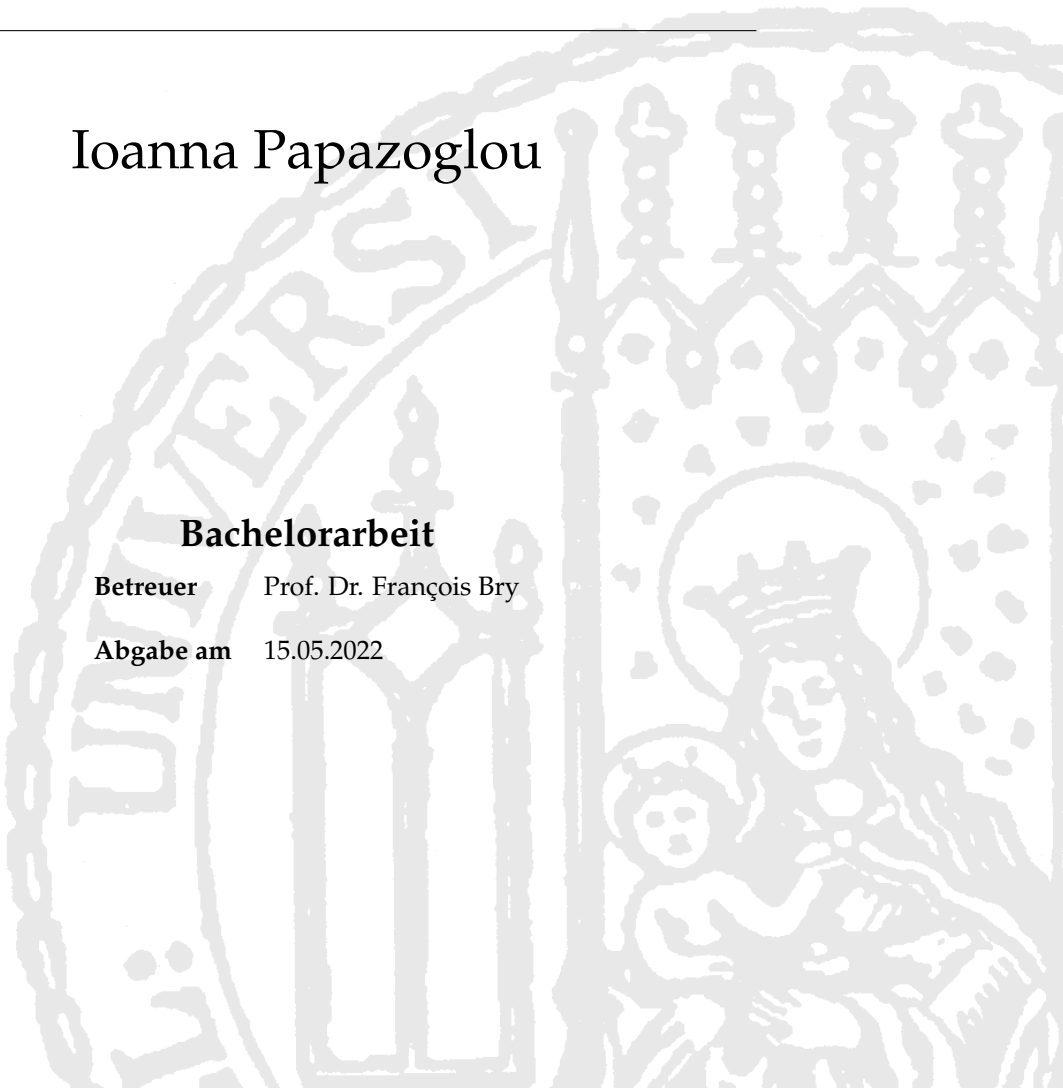
INSTITUT FÜR INFORMATIK
der Ludwig-Maximilians-Universität München

DIRI - A MOBILE SPEECH RECORDING GAME WITH A PURPOSE

Ioanna Papazoglou



Bachelorarbeit

Betreuer Prof. Dr. François Bry

Abgabe am 15.05.2022

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

München, den 15.05.2022 Ioanna Papazoglou, MatrNr:10975523

Abstract

Despite the growing importance of freely licensed pronunciation data for scientific and language training purposes, there is a lack of publicly available speech recordings.

This shortage is especially critical in projects that can legally only include creative commons licensed content, which leads to missing information in e.g. Wikipedia articles.

This work tries to show a possible solution by gamifying the recording and evaluation processes through the creation of a mobile app prototype.

During a short test phase a group of users generated and evaluated a number of usable quality audio recordings, which could be uploaded to open projects, like Wikimedia Commons.

Zusammenfassung

Trotz der wachsenden Bedeutung von frei lizenzierten Aussprachedaten für wissenschaftliche und sprachliche Trainingszwecke gibt es einen Mangel an öffentlich verfügbaren Sprachaufnahmen.

Dieses Defizit ist besonders kritisch bei Projekten, die rechtlich nur Creative-Commons-lizenzierte Inhalte einbeziehen können, was zu fehlenden Informationen in z.B. Wikipedia-Artikeln führt.

Diese Arbeit versucht eine mögliche Lösung aufzuzeigen, indem der Aufnahme- und Bewertungsprozess durch die Erstellung eines mobilen App-Prototyps gamifiziert wird.

Während einer kurzen Testphase generierte und bewertete eine Gruppe von Nutzern eine Reihe von Audioaufnahmen in brauchbarer Qualität, die in offene Projekte, wie Wikimedia Commons, hochgeladen werden konnten.

Acknowledgments

First and foremost, I would like to thank Prof. Dr. François Bry, who supervised and reviewed my bachelor thesis. I thank him for the wonderful conversations, his continued interest in the project, his helpful suggestions as well as his constructive advice during the creation process and writing of this thesis.

I also have to thank Dr. Christoph Draxler, for his support in all phonetics related questions.

A "Thank You" goes to all the testers, without whom this thesis would not have been possible and to the Wikimedia community for their amazing work in making free knowledge available to all.

Finally, I would like to thank my family, especially my partner, for their mental support and encouragement when I had doubts.

Contents

1 Introduction **1**
- 1.1 Pronunciation Databases – State of the Art . . . 1
- 1.2 Wikimedia Projects as a Hub for Free Knowledge and Data . . . 2
- 1.3 Games with a Purpose . . . 4

2 Related Projects **5**
- 2.1 Lingua Libre . . . 5
- 2.2 Common Voice . . . 7

3 Diri – Game Concept **9**
- 3.1 Similarities to Games and Language Learning Environments . . . 9
  - 3.1.1 Similarities to Language-Learning Applications . . . 9
  - 3.1.2 Similarities to Dictation Exercises . . . 10
  - 3.1.3 Similarity to Como . . . 10
- 3.2 Game Modes . . . 11
  - 3.2.1 Speak Mode . . . 11
  - 3.2.2 Listen and Write Mode . . . 11
- 3.3 Player Motivation . . . 13

4 Implementation **15**
- 4.1 Client . . . 15
  - 4.1.1 Application Software Structure . . . 15
  - 4.1.2 Audio Recording . . . 16
  - 4.1.3 Data Storage Mechanism . . . 16
  - 4.1.4 User Interface . . . 17
    - 4.1.4.1 Main Activity . . . 17
    - 4.1.4.2 Speak Mode Activity . . . 18
    - 4.1.4.3 Listen and Write Mode Activity . . . 18
    - 4.1.4.4 User Profile Activity . . . 18
    - 4.1.4.5 Info Page . . . 19
- 4.2 Server . . . 19
  - 4.2.1 Game Content Creation via Wikidata SPARQL . . . 19
  - 4.2.2 Hosting Platform Selection . . . 21
    - 4.2.2.1 Wikimedia Cloud Services . . . 23
    - 4.2.2.2 Toolforge Development Process . . . 23
  - 4.2.3 Database . . . 25
  - 4.2.4 Object Relational Mapping with Flask-SQLAlchemy . . . 26

4.2.5 Server – Client Communication via a HTTPS API 27

5 Evaluation **31**
5.1 Composition of the Test-group 31
5.2 Testing Phase 31
5.2.1 Speak Activity 32
5.2.2 Listen and Write Activity 33
5.3 User Survey Data 34
5.3.1 Results of the Survey 35

6 Conclusion and Perspectives **39**

7 Appendix **41**

Bibliography **45**

CHAPTER 1

Introduction

How is the name of US astrophysicist of Indian origin and Nobel laureate in physics, Subrahmanyan Chandrasekhar actually pronounced? An Indian speaker probably feels similar when stumbling across the Wikipedia article of the French artist of Dada, Marcel Duchamp. In the case of Chandrasekhar, one is already lucky enough to have an audio recording linked in the scientist's Wikipedia article. So listening to it can give for a better understanding regarding the pronunciation. But this is the exception so far. More often, there are representations in the phonetic transcription, which, however, are of little use for readers that have no experience with phonetic transcriptions without further assistance. Even more often, there are no hints at all on the pronunciation.

1.1 Pronunciation Databases – State of the Art

But there are actually pronunciation databases for the problem described above that could provide this information. A few databases are presented below and examined with regard to their licensing, although this list is of course not exhaustive [59].

The German public broadcaster ARD, for example, helps radio and television journalists finding correct pronunciations with the assistance of the ARD pronunciation database. As a joint institution under the leadership of Hessischer Rundfunk in Frankfurt it contains approximately 400,000 program-related entries (as of Aug. 18, 2021), including foreign-language proper names, designations and terms from fields such as geography, technology, politics, etc. [31]. Unfortunately it is only accessible for ARD employees.

Online audio dictionaries play a more and more important role for providing information about the right pronunciation of words, as there is no substitute for actually hearing how a word is pronounced. One of the largest online references for pronunciations is provided by Forvo [50]. The website allows access to, and playback of, pronunciation sound clips in many languages. More precisely, nearly 6 million pronounced words in over 430 languages are contained in its database, all created and maintained by native speakers [49]. For users who want to use the pronunciation data for their own projects, Forvo provides access to these datasets via an API. This usage comes with restrictions and additional costs. For example, the cheapest usage plan costs two dollars per month under the restriction that the data may not be used commercially and only 500 requests per day are possible [14].

Similar to Forvo, Howjsay is a website that aims to help users to learn English by providing over 180,000 British and American English entries in the dictionary [15]. It has been up and running since 2007. All the 180,000 entries are recorded by the owner of this website. Permission is granted to temporarily download one copy of the materials (information or software) for personal non-commercial transitory viewing only [18].

Tatoeba (Japanese phrase for *example*) is a large database of sentences translated into a significant amount of languages. It works like a multilingual translation dictionary, in which not the translation of a word can be found, but complete sentences in the language in which the searched word occurs. The translations often include a voice recording of the text read aloud and, rarely, phonetic transcriptions [20]. As of April 2022, almost 1,000,000 sentences in 38 languages had audio recordings, with the English language leading significantly [16]. Audio recordings of the sentences use the speaker's choice of license, such as Creative Commons Attribution-ShareAlike (CC BY-SA), Creative Commons Attribution (CC-BY), Creative Commons Attribution Non-Commercial (CC-BY-NC), or no public license at all [11]. Currently, (as of Mai 2022) the only way to download audio is by fetching each audio file one by one. There is no API or ZIP file data export that contains all audio, merely a ZIP file for English is available. Tatoeba is currently maintained collaboratively by a community of volunteers [13].

Unfortunately, all these databases do not offer free use of their datasets or do not make them available under a free license. Therefore, this pronunciation information cannot be integrated into Wikipedia or similar projects, which only accept CC-BY-SA or more permissively licensed content. Before discussing how audio recordings and thus pronunciation information can improve Wikipedia and its subsidiary projects, a brief introduction to the *Wikiverse* is given.

1.2 Wikimedia Projects as a Hub for Free Knowledge and Data

The Wikimedia Foundation (WMF) is a non-profit organization that promotes free knowledge and free content. It provides the essential infrastructure for contributing to all Wikimedia related projects [2].

The most well known project hosted by WMF is the free online encyclopedia Wikipedia, created, edited, and verified by volunteers. But there are plenty of other projects such as, Wikibooks - a project dedicated to build free learning materials, Wikisource - a library of freely-licensed source texts and historical documents, and the travel guide Wikivoyage, to name just a few [10]. Figure 1.1 shows the projects supported by the Wikimedia Foundation.

For the mapping and storage of linguistic content, including sound files, lexicographic data or descriptions of words of all languages, the Wiktionary, Wikimedia Commons and Wikidata projects are the most important and will therefore be discussed in more detail.

Wiktionary is a collaborative project to create an open-content, multilingual dictionary and a corresponding thesaurus in every language. The first Wiktionary was the English language Wiktionary, created by Brion Vibber on December 12, 2002. In 2004, other languages such as Polish and French and German followed. Since the creation of Wiktionary in 2002, the number of entries in Wiktionaries internationally has grown to over 35,5 million (as of April 29, 2022) [40]. All the content is dual-licensed under both the CC-BY-SA and the GNU Free Documentation License [68]. Beyond the Wikimedia community, Wiktionary is increasingly gaining influence in a wide variety of linguistic fields such as Natural Language Processing (NLP) and lexicography [69]. Sylviane Granger and Magali Paquot for example consider Wiktionary as "a competitor to dictionaries created by experts, opening up a wide range of possible applications" [48].

Figure 1.1: Wikimedia supported project logos the listed in order, starting from the top center clockwise: Wikimania, Wikibooks, Meta-Wiki, Wikiquote, Wikispecies, MediaWiki, Wikimedia Incubator, Wikidata, Wikivoyage, Wikiversity, Wiktionary, Wikinews, Wikisource, Wikimedia Commons and Wikipedia [55]

Wikimedia Commons is the largest freely licensed library of illustrations, photos, drawings, videos, music and sound files. It acts as a common repository for the various projects of the Wikimedia Foundation. Unlike media files uploaded to other projects, files uploaded to Wikimedia Commons can be embedded on pages of all Wikimedia projects without the need to separately upload them there. Launched on 7 September 2004, Wikimedia Commons currently contains 82,679,535 files and 80,424,197 media collections. Everyone is allowed to copy, use and modify any files freely as long as they follow the terms specified by the author; this often means crediting the source and author(s) appropriately [58].

The latest Wikimedia project, **Wikidata**, deployed in 2012, is a database of structured information, not only readable by humans but also by machines. This makes it easier to search, edit, curate and use data.

In the beginning the functionality of Wikidata was limited to linking between different language versions of Wikipedia pages on the same topic [38]. However, the project quickly evolved into a more flexible, general-purpose data modeling tool. After its initial launch, Wikidata was integrated into other Wikimedia Foundation projects serving as a central repository for the structured data of its Wikimedia sister projects such as Wikipedia, and Wiktionary [53].

But Wikidata has also gained relevance outside the Wikimedia community, becoming a topic of interest for researchers. As such, there is an ever-growing number of publications focusing on Wikidata, increasing every year since 2012 [44]. Due to the success of Wikidata, Google decided in 2014 to end its own structured knowledge project *Freebase* and offered its content to the Wikidata community, making efforts to migrate their data to Wikidata [42].

In an effort to expand its data model, lexicographical data structures were introduced as a new data type in 2018. Lexicographical data contains data similar to an entry in a traditional dictionary, like lexemes, forms, synonyms, etymology etc. [33]. All of Wikidata's Lexicographical data can be reused by multiple tools, queries and on other Wikimedia projects, because it is available under a CC0 (equivalent to Public Domain) license [54].

1.3 Games with a Purpose

Because of these license differences many of these crowd sourced projects rely on the help of volunteers and their contributions, as they cannot import content from sources with restrictive licenses. So it is all the more important to motivate these volunteers to help. Games are known to drive their users to many interactions. In the case of pronunciation recordings for Wikimedia related projects, the goal is to playfully encourage the volunteers to create voice recordings.

An interesting concept in this context are games with a purpose, GWAPs for short. GWAPs are a kind of serious games, in which players generate useful data or solve problems, which are currently difficult to solve computationally, in a playful way [51]. The players' task may involve describing pictures, transcription of speech or recognizing hard-to-decipher writing. Two goals must be met in the design of a GWAP: Firstly, the game must be fun, to ensure sufficient player participation. Secondly, the game must ensure the collected data is correct [47].

Luis von Ahn and Laura Dabbish first introduced this concept of human computation in the ESP game, in which users label images on the web [52]. Von Ahn also was noted for his work on projects such as CAPTCHA and the very successful language learning platform Duolingo. In addition, there are also GWAPs in fields as diverse as computer vision, art, medicine and biology, Internet search, and linguistics (mainly NLP) [36]. GWAPs are also operated in the Wikimedia area on the web platform *The Distributed Game* [39], which allows less experienced users to edit and improve Wikidata entries playfully.

Among the areas where GWAPs exist, the absence of games dealing with naturally spoken language is striking. Mathieu Lafourcade, Alain Joubert and Nathalie Le Brun list examples of GWAPs from different categories in their book *Games with a purpose (Gwaps)*. GWAPs from the linguistic field are reduced to collecting text-based lexico-semantic information. Acoustic language data is not generated by these games [36].

In summary, there are many language databases with valuable information on pronunciation. Free knowledge projects and their users can profit from this data. However, this data is either not intended for private use, costly or difficult to export. Therefore, crowdsourcing projects like Wikipedia, Wikidata or Wiktionary rely on the contributions of volunteer editors. GWAPs seem to be successful in getting users to solve tasks and generate relevant data. Thus, it could be useful to simplify the creation of voice recordings through a GWAP and to encourage players to participate in open knowledge projects. Subsequently, the prototype Diri was developed and tested for a ten-day period to gain information about the user experience and interaction and more importantly the quantity and quality of generated speech recordings. With this in mind, this thesis first presents projects in Chapter 2 that share the same goal of generating speech recordings. Chapter 3 provides insight into the conceptualization of the prototype, followed by the implementation of the Server and Client in Chapter 4. Details on the analysis of the data collected during the testing phase are presented in Chapter 5. The thesis concludes with a final discussion of the project and possible future developments in Chapter 6.

CHAPTER 2

Related Projects

While there is a multitude of applications dealing with speech recording in some fashion, few of them actually reuse the recordings or make them accessible to the public. Companies like Google and Amazon for example collect millions of recordings and utilize them to improve their speech recognition software, however none of this training data is available publicly. Only a handful of projects collect speech recordings and publish their data under a free license. In this chapter, two examples of such projects are presented and, based on their characteristics, the need for Diri is further discussed.

2.1 Lingua Libre

Lingua Libre is an online collaborative project and web tool established by the Wikimedia France association. It was launched January 2015 with the goal of creating a multilingual and audiovisual corpus under free license, assembled by voluntary user contributions [29]. Figure 2.1 displays Lingua Libre's landing page.

Users can easily participate by recording words, phrases or sentences of any language with the help of the RecordWizard, a recording tool provided on the website of Lingua Libre. As seen in figure 2.2, words are presented to the speaker in the form of a list, created on the spot or in advance, or reusing an existing Wikimedia category.

A Wikimedia account is needed, if one wants to contribute. Also, a short speaker profile is created where speakers can give details about which languages they speak, their place of residence and under which license the recordings should be published. To create a recording the speaker simply reads the word displayed on the screen of the *RecordWizard*, and the software moves on to the next word when it detects silence for a defined interval. Before the recordings are uploaded from the web client to the Wikimedia Commons media library, speakers have the opportunity to listen to their audio recordings and, if necessary, replace bad recordings with improved ones. The recordings are mainly used on other Wikimedia projects, for example to illustrate entries on the different Wiktionaries or nouns in Wikipedia articles. But a great part of the recordings also are used for other projects like Common Voice.

The web application has overlapping goals with the Diri project, as it also tries to contribute to Wikimedia projects by providing the speech recordings under a free license. The main difference is in the creation of the voice recordings. In Lingua Libre a list of words can

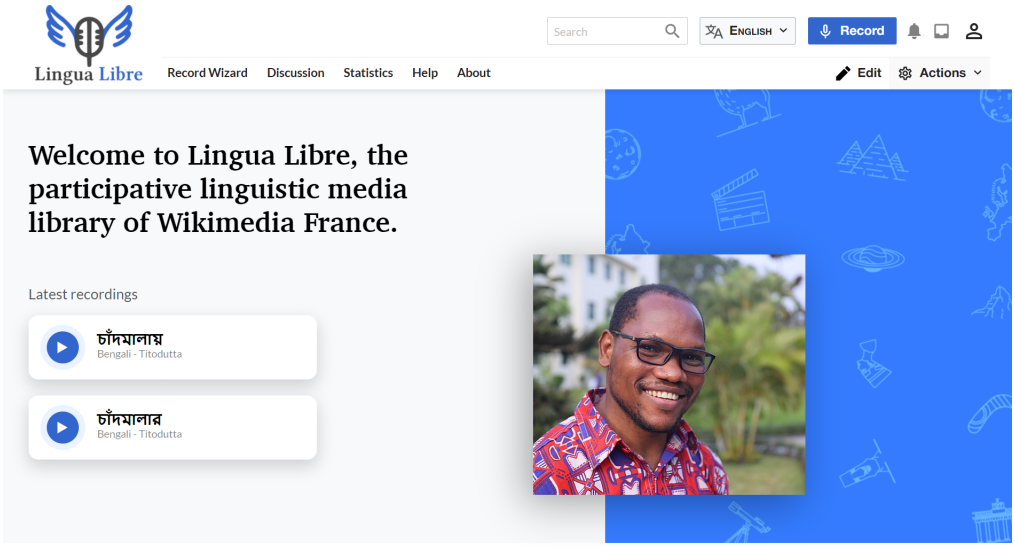


Figure 2.1: Lingua Libre homepage [57]

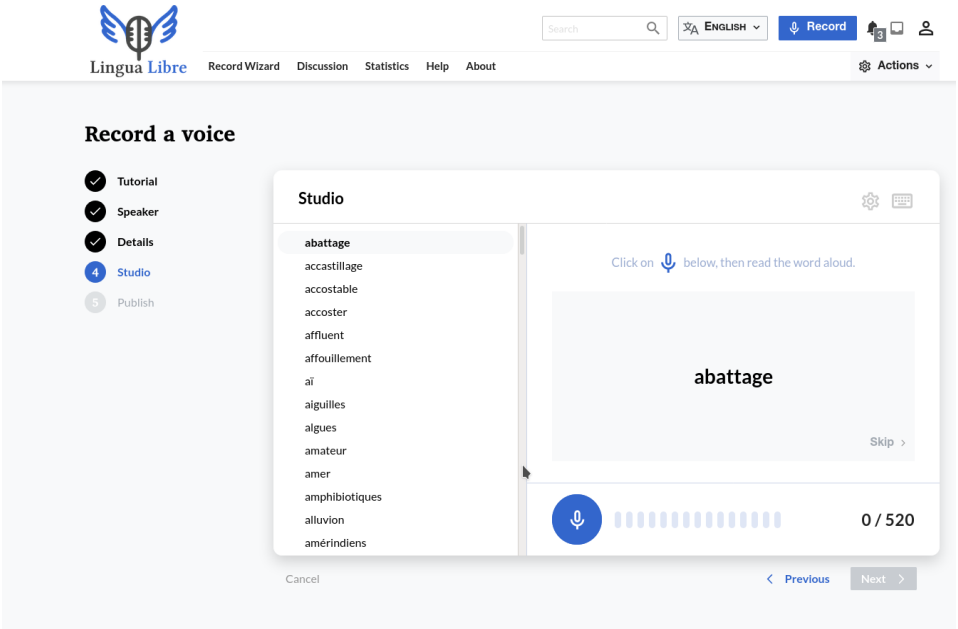


Figure 2.2: RecordWizard on Lingua Libre [56]

be chosen from to be read and recorded, whereas in Diri users are given lexemes in their phonetic representation, as the GWAP also aims to teach users about the International Phonetic Alphabet. This can be more time-consuming but also gives Diri more gamification aspects as reading words aloud is merely a simple and monotonous task.

2.2 Common Voice

Common Voice is a crowdsourcing project started by *Mozilla* which aims to create a free database of audio recordings for speech recognition software. As the currently available systems are expensive and proprietary the project is supported by voluntary speaker contributions [4]. The dataset currently consists of 11,192 validated hours of speech in 76 languages [5]. A strong focus lies on creating a database with recordings of all sexes, ages, languages and accents as awareness arises that speech recognition software can recognize male, accent free voices the best, due to stronger representation in datasets [70]. Anyone who wants to build speech recognition, speaker recognition, or any other type of application that requires voice data can use the Common Voice database, as the license of the recordings is CC0. To contribute to the Common Voice project a speaker can select between two kinds of tasks: recording sentences or listening to speech recording for validation (figure 2.3).

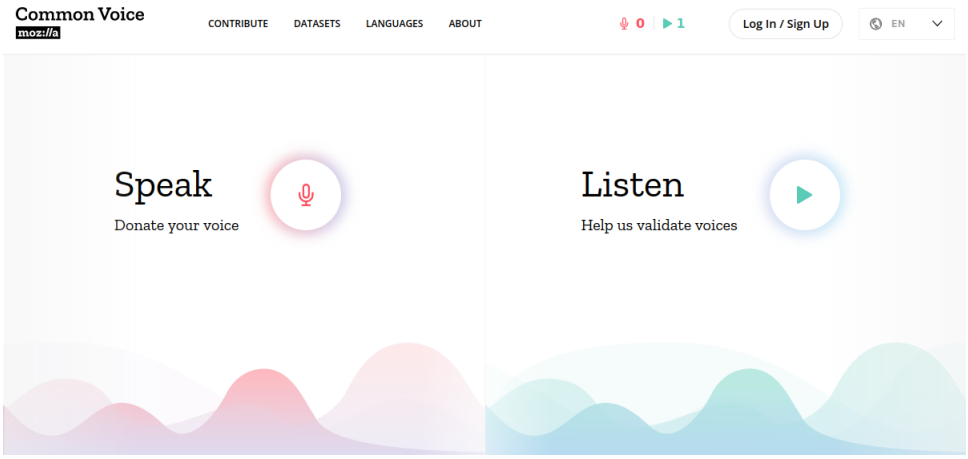


Figure 2.3: Main page of the Common Voice website

The text provided in the recording task is mostly generated from user submissions in Common Voice's Sentence Collector or scraped from Wikipedia articles with an extractor tool [22]. To make an audio recording users simply push the microphone (speak) button to access the recording page. There sentences are shown in a text view and speaker can start and end a recording after clicking on the record button (figure 2.4). Speakers can check their recordings before submitting them to the database. For the listening task, users listen to voice recordings and evaluate them for accuracy based on the displayed text (figure 2.5).

Users are offered a very reduced action set to complete the tasks of recording and listening, that can be executed quickly. There are no classic gamification attributes like rewards or points. Although there is a native Common Voice iOS application, Android users are excluded and are bound to their browsers.

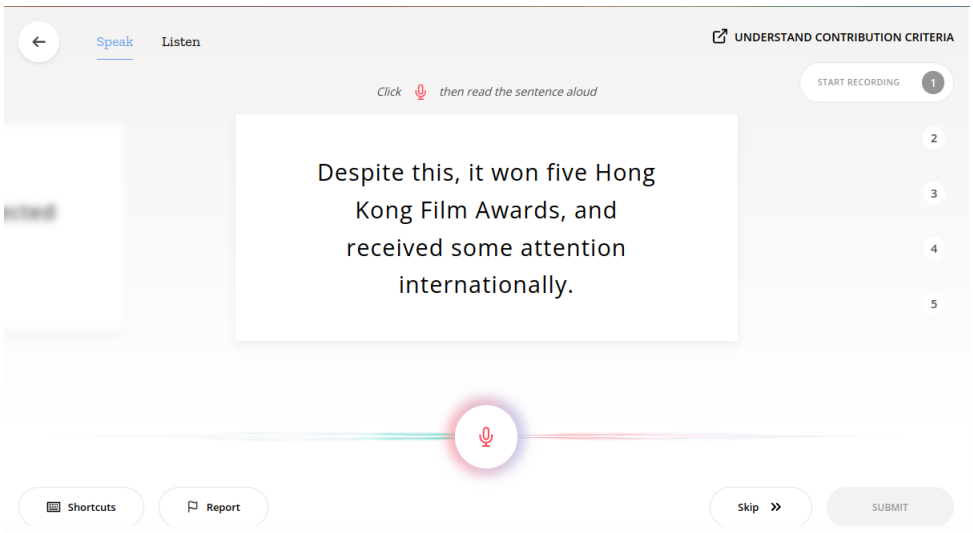


Figure 2.4: Common Voice's recording website, where speakers create voice recordings based on the displayed text

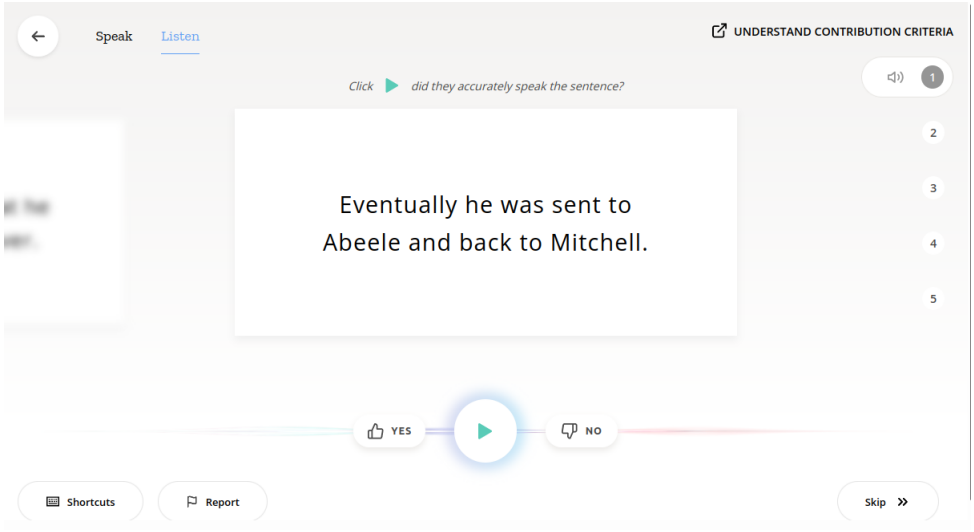


Figure 2.5: Main page of the Common Voice website

CHAPTER 3

Diri – Game Concept

Diri (the Esperanto word for *to speak*, *to tell*) is a language learning and training game for one player that focuses on pronunciation. It includes elements from the linguistic learning environment such as the use of language learning applications and dictation. More precisely, the game is based on the idea that the interaction with language learning elements and the accompanying practice of pronunciation provide the right environment to generate pronunciation audio files. At the same time, the app Como and its game modes are considered as a reference for the game structure. In the following, these elements are presented, and it is shown which aspects of them might apply to Diri. Afterwards, the game modes are explored and described.

3.1 Similarities to Games and Language Learning Environments

In contrast to the chapter Related works, where projects with similar goals were discussed, this section tries to emphasize the similarities in regard to game elements, data collection and use.

3.1.1 Similarities to Language-Learning Applications

In language learning platforms, users are asked to perform simple tasks related to language acquisition. Mostly the lesson system is build around specific topics (e.g. food, family, work, school, shopping) with each topic introducing new grammar and lessons, focussing mostly on building the learner's vocabulary. To practice the new vocabulary the exercises include translation, multiple-choice word recognition questions, and spelling [45]. Similar tasks are offered by language learning platforms such as Babbel, Busuu and Duolingo. Especially popular and most widespread is Duolingo due to the freemium business model [45]. In comparison, the platforms Babbel and Busuu are not available free of charge. Duolingo contains a variety of gamified features. According to Govender's and Arnedo-Morenoto's survey of gamification elements in mobile language-learning applications they identified 22 gamification elements in Duolingo's concept [30]. This includes "progress indicators (daily goal and experience points, unlocking levels), feedback

(correct/incorrect answer), fixed reward schedule (experience points), time-dependent rewards (streaks), customization (buying outfits for the owl mascot), challenges, knowledge sharing (forums), leaderboards, badges and achievements, and virtual economy ("lingots" and gems)"[45]. Although this amount of gaming features is supposed to make learning more motivating for its users, Duolingo's activities are considered to be lacking productive skills like writing and speaking. Or rather, these are only made available to the player in a later level [27]. To evaluate how close a learners' pronunciation is to the target the speech exercises use AI voice recognition. Similar AI voice recognition is provided at the Rosetta Stone learning platform, that focuses on the pronunciation of the language to be learned. The courses focus less on vocabulary and grammar learning and more on the spoken word and the contextualization of conversations. For example, the exercises involve the learner verbally describing an image - the microphone records what is said and the program's speech recognition gives feedback on pronunciation. In terms of Diri, the app could benefit from many of these language learning exercises. At least the environment and context of a learning app can set the right framework for Diri's tasks. As for now, the main focus should be on pronunciation recording rather than learning a new language. That's why the tasks are designed to give players the opportunity to work on their pronunciation. Unlike Duolingo and Rosetta Stone, however, these recordings will be evaluated by humans and not by speech recognition software.

3.1.2 Similarities to Dictation Exercises

A dictation is a transcription of a spoken text and is used as an exercise to improve spelling. Dictation requires at least two people: one to read the text aloud and one to write it down as it is spoken. There are two ways to evaluate the written text. The person reading the text checks at the text of the person writing it and looks for errors. Or the person who wrote the text can compare his or her solution with the text read aloud. This method can be helpful for verifying voice recordings with in the Diri app. It may prove to be a way to check the comprehensibility of the recordings by having the users listen to a word and write it down, as in dictation. Furthermore, the players can use such a method to check and improve their spelling.

3.1.3 Similarity to Como

Como is an Android app which tries to gamify the creation of lexeme senses, to attract new contributions. The application also checks the quality of the created senses by letting other players guess a lexeme from its sense. Como's focus is on the playful creation of descriptions of lexemes. There are two game modes for this. In the first game mode, players have to find the correct lexeme for a description four times. Only then are the players given the opportunity to create their own description for a lexeme. For the generation of game data, existing descriptions of lexemes from Wikidata are used, as well as lexemes that do not yet have a description. If a well-described lexeme is often recognized by players, this description will be uploaded to Wikidata [35]. Like Como, Diri uses data from Wikidata in an attempt to improve this project and other Wikimedia projects with the data generated from the game. Inspired by the game concept of Como, Diri also has two main game modes. A game mode where players generate data for Wikimedia and a game mode where this data is verified by players.

3.2 Game Modes

There are two major game modes in which players interact with pronunciation information in two different ways.

- **Speak mode**: In the *Speak mode* players are able to make voice recordings from phonetic transcriptions that are provided by the game. These voice recordings are then presented in other players *Listen and Write mode* later.
- **Listen and Write mode**: In the *Listen and Write mode* players are asked to listen to the previously made voice recordings and type the correct word. This is done to check the comprehensibility of the audio recordings.

3.2.1 Speak Mode

The main goal of the *Speak mode* is to allow players to create voice recordings by using a microphone button. For this purpose, the game suggests words in their phonetic representation by means of the transcription with the International Phonetic Alphabet (IPA) characters. This is intended to provide a more exciting game experience for the speakers, since simply reading words aloud is a monotonous task that players don't enjoy very much. The IPA is a notational standard for the phonetic representation of all languages, so that human speech sound can be visually represented. Its alphabet is based on the principles that there should be one symbol for each sound and that the same symbol should be used for one sound in every language. These symbols are summarized in the IPA chart as displayed in figure 3.1. It is mostly based on the Latin script, which has the advantage of being widely familiar, but also includes a variety of other letters and symbols from different sources (e.g. Greek)[23]. Important for Diri are the consonants and vowels. Some diacritical marks that are used for fine distinction sounds and to indicate nasalization of vowels, length, stress and tones are also used and displayed in the app but not considered for button generation.

By presenting words in their phonetic transcription players are given a better understanding of the exact pronunciation of the displayed word. Especially players who do not play the game in their native language could benefit from the detailed pronunciation information. Since not every player is familiar with the International Phonetic Alphabet, hint buttons are designed to help the player understand the phonetic characters. There are two types of hint buttons for this purpose. First, a keyboard of buttons is generated from the phonetic transcription, so that each sound can be listened individually. Additionally, another hint button shows the lexeme behind the transcription to the player, so that even if the phonetic representation cannot be understood, the player still has the possibility to create a voice recording. If a player has difficulties understanding the phonetic transcriptions, even with the help of the ´´hint" buttons it is possible to skip this task and continue the game with a new phonetic transcription suggested by the game. After players create a recording, they can listen to their audio recording and replace it with a new one if it is found to be incorrect. When the recording is confirmed by the speaker, the game uploads the audio file in the background and a new transcription is shown. Eeach confirmed recording is rewarded with a point.

3.2.2 Listen and Write Mode

In the *Listen and Write mode* players listen to voice recordings of other players made in the *Speak mode* and try to understand which word was spoken. To do so, there is a text field, where players type in the word they heard in the audio recording. This process serves three goals: On the one hand the *Listen and Write mode* checks if the audio recording matches the word and IPA transcription provided by the *Speak mode*. On the other hand it also checks

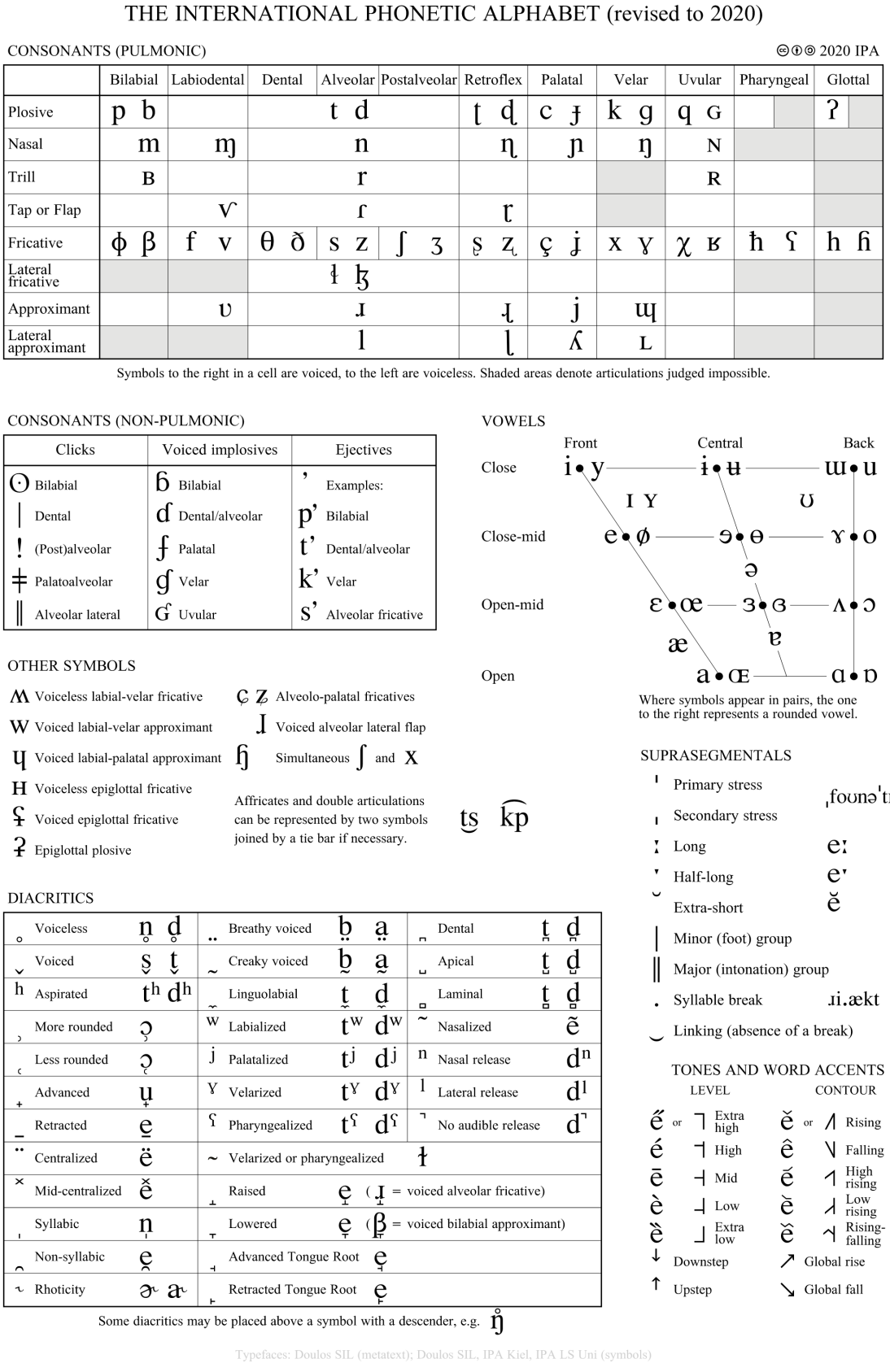


Figure 3.1: The IPA chart and its characters [32]

whether an audio recording is understandable. Additionally, by typing in the letters of the heard words, players can train their spelling skills. Once an audio recording has been assigned to the correct word, the game proceeds to the next audio recording and the player is rewarded with a point. Has a player entered the wrong word, they can try indefinitely to find the correct solution. If a voice recording is incomprehensible or a user wishes to stop guessing, the player can skip the task. User inputs and information about skipped tasks are sent to the server for evaluation of the player's performance.

Often played and correctly recognized voice recordings could then be uploaded to Wikimedia Commons and linked to Wikidata with the property (P443)[1].

3.3 Player Motivation

A game should primarily be an enjoyable pastime for its players. Therefore, a game with a purpose should a have sufficient gameplay component to attract and retain its players [36]. A key concept in gaming is motivation. In general, two categories of motivation can be distinguished: intrinsic and extrinsic motivation [34].

Intrinsic motivation is demonstrated when a person engages in activities purely for the joy of doing them. Completing a task is in itself rewarding for them, for example because it resonates with their personal beliefs or interests, or simply out of curiosity.

Extrinsically motivated people are encouraged by external rewards. This can be points, badges, good grades, monetary compensation, praise, a certificate, a prize or admiration. The nature of extrinsic motivation is temporary: once a goal is achieved, the person demands a new goal [34].

It is clear that motivation is highly subjective and most activities will have both intrinsic and extrinsic components [43]. Diri's concept of motivation was designed on the basis of this theory of motivation. The promotion of intrinsic motivation is of particular importance, as studies have shown the negative influence of strong extrinsic motivational elements on more intrinsically motivated individuals: When rewards become the goal, people are less likely to perform an activity without that reward, and their performance will generally be worse than in highly intrinsically motivated activities [21], [34]. As a result, Diri uses only some means of extrinsic motivation, mainly to encourage players, and otherwise relies on intrinsic motivation. In this way it could be possible to create a motivated player base that will play the game for an extended period of time and provide meaningful data.

Extrinsic motivation in Diri is based on a classical point system, where players receive points for correct inputs in the game. One point is awarded for each voice recording and for correctly typing the word in the *Listen and Write mode*. These points are an indirect measure of one's accomplishments and ability. In addition, a statistics page will list how many voice recordings have already been made. Since Diri is a Wikimedia-related project, it builds on the intrinsic motivation of the editors of these projects to contribute to open knowledge projects. Additionally, intrinsic motivation should arise by targeting people with an interest in languages who want to improve their pronunciation and practice their spelling skills. The learning experience and the feeling of contributing to a project are the main focus.

[1] https://www.wikidata.org/wiki/Property:P443

CHAPTER 4

Implementation

The software for the game is structured in a client-server model. The server side is a Python script hosted in the Wikimedia Cloud Services platform Toolforge. It connects to a MariaDB Database instance, which stores phonetic transcriptions, sound files and user data. The client side is implemented as an Android application written in Java, which connects via a standard HTTP API to the web-service. Starting with the client, the approach to the technical implementation of the prototype Diri is discussed is this chapter.

4.1 Client

Since a frequent use of a microphone is necessary to play the GWAP, an implementation of the client as a mobile application suits the purpose of the game the most. Mainly because integrated microphones in laptops, by comparison often pic up noise like fans or keyboard clicking. Also, the potential user-base is significantly larger than traditional desktop web browser users [28]. There are currently two major platforms for mobile application development, Android and iOS. For the purposes of Diri the Android platform was chosen because of lower overall deployment and development costs compared to iOS. The application's source code can be found online in Diri's Gitlab repository.

4.1.1 Application Software Structure

As writing all the application's code in one activity is a common mistake that causes many lifecycle-related problems it is beneficial to use an architectural pattern, that helps to separate the visible potions of the application from the back-end logic [3]. That helps to address numerous development issues and can make an application easier to test, maintain, and evolve [24]. For that reason Diri's software architecture follows the Model-View-ViewModel (MVVM) architectural pattern.

There are three core components in the MVVM pattern: the *Model*, the *View*, and the *viewModel*. Each component serves a distinct purpose.

The *View*, in Android called *activity*, represents the user interface of the application [24].

The *Model* holds the data of the application. Its logic is separated from the *View* with the *viewModel*.

The *viewModel* serves as a link between *View* and *Model*. It is responsible for transforming, converting and exposing all relevant data from the *Model* to the *View*. The *View* then only observes the *viewModel* [3], [7], [46], [24].

Figure 4.1 illustrates the structure of the MVVM pattern and the relationship between the components.

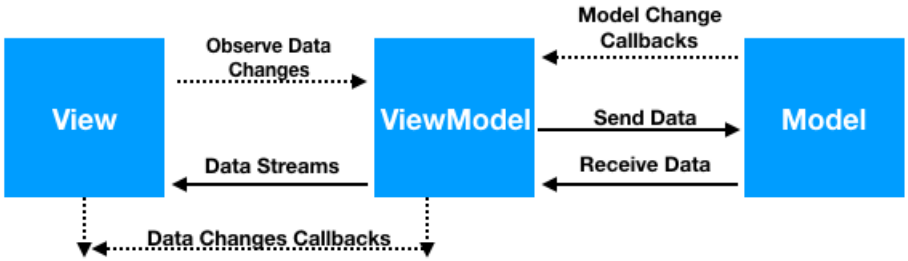


Figure 4.1: MVVM pattern overview [3]

4.1.2 Audio Recording

Since nearly every handheld Android device comes with a microphone, the Android SDK (Software Development Kit) includes support for audio recording out of the box. But the choice of the correct codec and sampling rate is still not an easy one, as support differs widely by Android version and manufacturer. Often there is also a difference between the encoding and decoding capabilities, which excludes some traditional lossless choices for speech analysis [19]. Additionally, some open source codecs and audio format are not available as encoders at all.

Due to these limitations, the final encoding format AAC with an MPEG4 container was chosen. To allow for suitable quality audio recordings, a sampling rate of 44100 Hz was set, after a series of tests with lower rates proved unsatisfactory. To set the rate higher would increase the file size, without drastically improving the intelligibility of the recording. These settings should create audio files of comparable quality with samples from Wikimedia Commons, while of course still lagging behind professional studio recordings.

4.1.3 Data Storage Mechanism

Because Diri handles a great amount of structured data it benefits from managing that data locally. In Android there are two ways to access and modify data of the application's SQLite database: Using the SQLite API directly or through the *Room Persistence Library*. As the Android developers' documentation advises against the direct usage of SQLite database APIs, Diri's local database implementation is based on the *Room Persistence Library* approach [9]. Compared to using the SQLite API directly, Room offers tree major advantages:

- Compile-time verification of SQL queries
- Convenient annotations that minimize repetitive and error-prone boilerplate code
- Streamlined database migration paths

A short look at Room's architecture gives an overview of its functionality. There are three major components in Room as can be seen in figure 4.2:

The database class contains the database and serves as the main access point for the underlying connection to the application's persisted data.

Data entities represent tables in the app's database.

The data access objects (DAOs) provide methods that the app can use to query, update, insert, and delete data in the database [8].

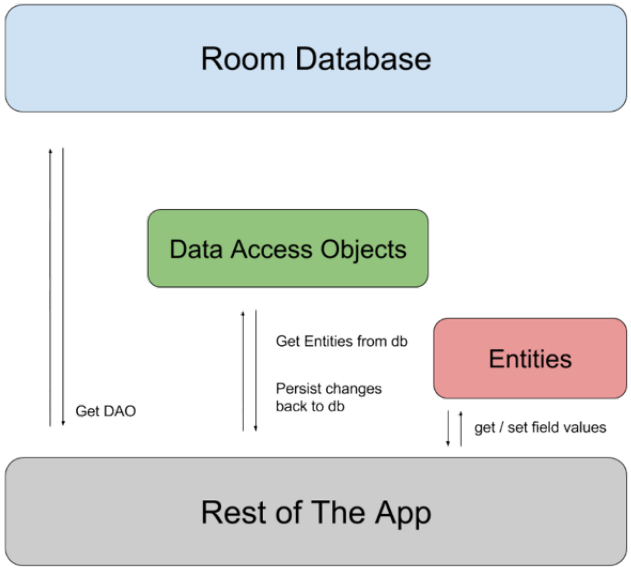


Figure 4.2: Room library architecture overview [8]

4.1.4 User Interface

Diri's user interface is kept simple and contains hardly any styled animations. The color scheme was generated with the help of a color pattern generator for harmonic color visualization at https://coolors.co/. It consists of eight colors as seen in figure 4.3. The whole application consists of four Android activities with which users can interact.

4.1.4.1 Main Activity

Diri's main activity, as shown in Figure 4.4, is the starting point for the user's interaction with the application. It shows at one glance all possible user activities. Without a tutorial, it tries to reduce the essence of the game to *Speaking* and *Listening and Writing* for the users. Those two tasks are the main activities of Diri, which is emphasized by the blue background, as these are the only buttons in those shades.

If players want to learn more about Diri, they are referred to the info page. Points and user statistics can be found on the profile page. All buttons are provided with text and additional images, so that users can immediately recognize what can be achieved in the respective activity, even without reading the text.

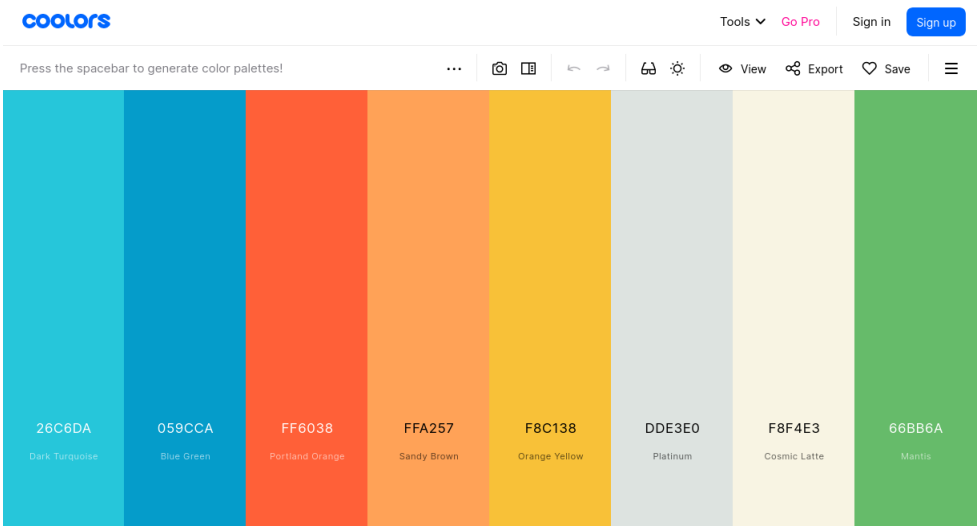


Figure 4.3: Generated color palette

4.1.4.2 Speak Mode Activity

The *Speak mode* is the most important activity of Diri, (see 4.5) where players have the task of creating a suitable speech recording for a word displayed in phonetic transcription. For this purpose, the interface presents a random word transcribed in IPA and prompts the user to create a voice recording. To accomplish this, the microphone button in the center of the interface should be pressed until the player finishes speaking the word. Players can listen to their voice recording by pressing the play button (marked as a triangle on the right side of the microphone). This button only appears after a recording was made. If necessary, a voice recording can be replaced by a new one by pressing the microphone button again. In order to be able to pronounce the word transcribed in IPA, there are two types of assistance. For each word, all the individual sounds are generated as buttons displayed under the word so that players can listen to them. With the hint button (marked with a question mark to the left of the microphone), players can have the word displayed in its written form. Furthermore, players can press the confirm and skip button. While the skip button can be activated throughout the game and brings up the next item, the confirm button only appears after a recording is made. By activating it the sound file is sent to the server.

4.1.4.3 Listen and Write Mode Activity

In the *Listen and Write mode* (4.6) players are tasked with writing the word they heard. For this reason, the interface presents the user with a text field in which the solution can be entered. The button with the speaker icon plays the voice recording of another player when pressed. Like in the *Speak mode*, the players can press the skip and check buttons (instead of confirm). With skip, the next possible sound file is loaded, while check writes the entered word to the database and synchronizes with the server.

4.1.4.4 User Profile Activity

Users can get an overview of their in-app activity and progress in the Profile activity 4.6. For this purpose, there is a scoreboard with the information how many voice recordings

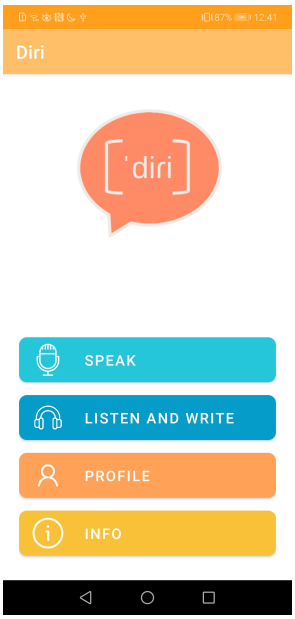


Figure 4.4: Diri's main activity

users have created and how many points they have achieved.

4.1.4.5 Info Page

The info page gives information about the app. As seen in figure 4.8 it describes on the one hand what the goal of Diri is and gives instructions on how the two main game modes are played. At the same time, the scoring system is explained. Finally, it mentions all the names that have contributed the phonemes for the speak activity, as they are retrieved from Wikimedia Commons.

4.2 Server

The server is a python script containing code to retrieve phonetic transcriptions from Wikidata as wells as the API for the client connection. The game data is created from these data sources and delivered to the client via the client interface (a JSON API). The following section takes a closer look at the generation of game data. Furthermore, special attention is paid to the technical implementation of the server providing information about the selection of the hosting platform and the used software. This is followed by an explanation of the database structure and its contents. Finally, the interfaces for server-client communication are discussed and explained in detail. The server source code can be found online in Diri's Gitlab repository.

4.2.1 Game Content Creation via Wikidata SPARQL

For game generation, information is downloaded from Wikidata using the SPARQLWrapper Python library [17].

This provides a simple abstraction for building queries and helps parse the returned information.

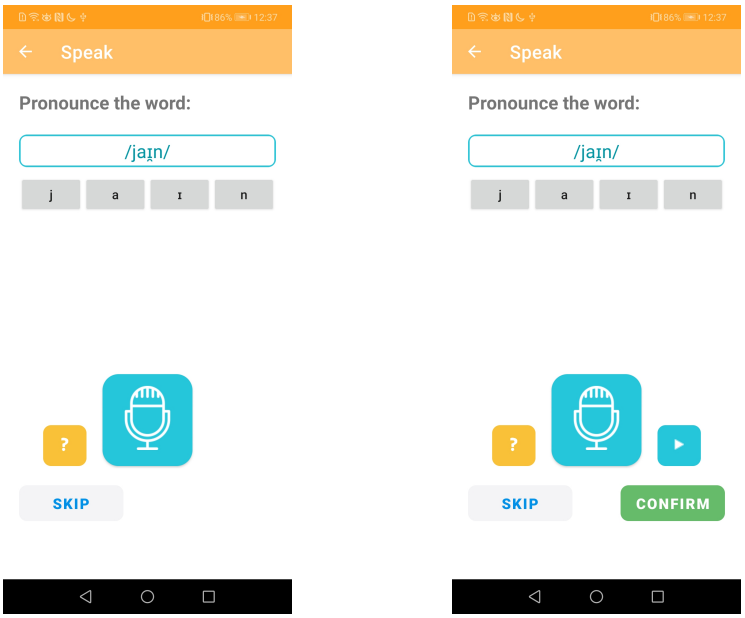


Figure 4.5: Diri's Speak Activity

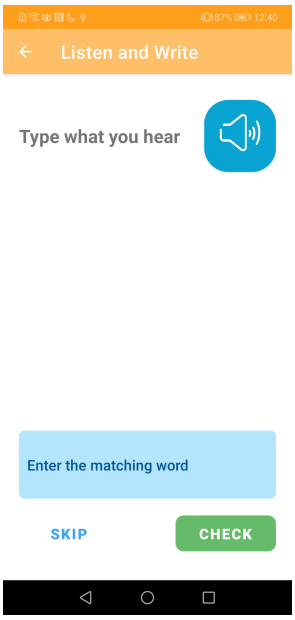


Figure 4.6: Diri's Listen and Write Activity

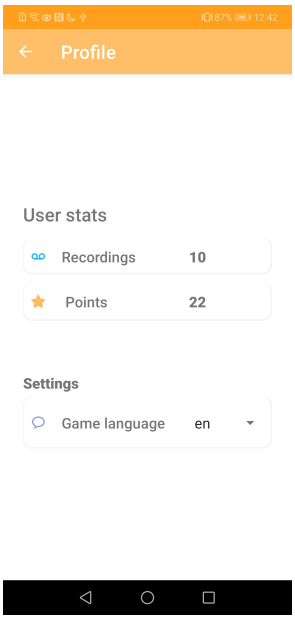


Figure 4.7: Diri user profile page

For easy ways to interact with and extract information from the Wikidata data pool, Wikidata provides a SPARQL endpoint at https://query.wikidata.org. Using this, the query depicted in figure 4.9 was created to generate the first dataset for the server, providing game data for the *Speak mode*. The query results in 698 items containing German lexemes, their IPA transcriptions, and partially linked speech recordings from Wikimedia Commons.

With these already existing voice recordings a comparison between the Diri and the existing Speak recordings in Wikimedia Commons could take place at a later time. Voice recordings made with IPA transcriptions are loaded into the pronunciation interaction table in the database and are used to generate game data for the *Listen and Write mode*.

4.2.2 Hosting Platform Selection

This part of the paper explains the criteria used to select the hosting platform for the project, followed by three of the most popular hosting products based on these criteria. Finally, the hosting platform Toolforge is examined in more detail. Since the project is implemented as a client-server model, the choice for the hosting platform is made under the following constraints:

- **Reliable and sufficient computational resources like CPU time and RAM**

  As the application needs to handle non-text data in a suitable manner, the limits should be set high enough to prevent lagging or crashes.

- **Availability of a static IPv4/IPv6 address**

  For the communication between the mobile application and the server a static IP address is advantageous. In addition, a static IP address ensures that the DNS record does not have to be constantly adjusted.

- **A domain name and SSL/TLS-Certificate**

  Domain names are used in various networking contexts and for application-specific addressing purposes. For Diri's purpose a domain name is needed, so the client can

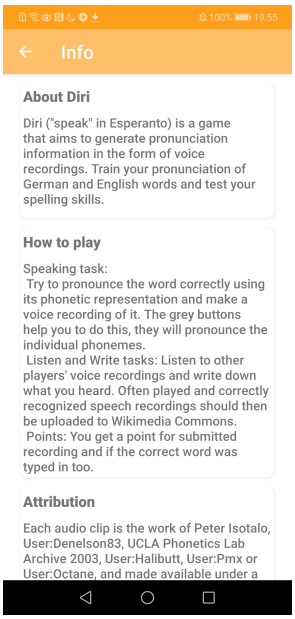


Figure 4.8: Diri info page

```
SELECT DISTINCT ?l ?lemma ?ipa ?audio
WHERE {?l dct:language wd:Q188;wikibase:lemma
      ?lemma; ontolex:lexicalForm ?form .
      ?form ontolex:representation ?lemma .
      ?form wdt:P898 ?ipa.
MINUS {?l wikibase:lexicalCategory wd:Q62155.}
OPTIONAL{ ?form wdt:P443 ?audio .}
   .FILTER regex(?lemma, "^[A-z].*")}
```

Figure 4.9: A SPARQL Query to download German lexemes, their IPA transcriptions and if available, links to Wikimedia Commons voice recordings.

connect with the server. Also, domain names are a requirement for easy TLS encryption.

- **Sufficient storage space and a backup solution**

  The server should provide a certain amount of storage space, as sound files are the main data, in addition to text data. The hosting platform should be able to handle this.

- **A safe automatic update process**

  To ease the burden of long-term maintenance, it should be possible to perform automatic updates.

- **Low or no cost**

  Since Diri is a small prototype based project, the costs for the server implementation should be kept as low as possible.

- **Python runtime**

As the server is implemented as a python script, a python runtime environment is necessary to run the script.

4.2.2.1 Wikimedia Cloud Services

When considering the requirements, there is a multitude of hosting strategies, e.g. self-hosting, web-hosting or cloud-hosting, to select from. Since the application should run for a long time, self-hosting is not eligible, since many of the requirements listed above are too complex to implement and maintain. Therefore, an online solution is chosen to run the python script on. Particularly popular are the web hosting tools from Amazon (AWS) [12] and Heroku for their free tokens. Another possible route is to use computing products offered by academia adjacent institutions like the Leibniz-Rechenzentrum (LRZ) [37]. As one of the few community driven projects of its size, Wikimedia also offers its own server environment to its more technically inclined contributors, called Wikimedia Cloud Services (WMCS) [60]. Its main purpose is to run bots, scripts and web applications related to the goals of the Wikimedia movement. Considering that Diri uses and contributes data from and to multiple Wikimedia projects (such as Wikimedia Commons and Wikidata) it is advantageous to implement the server in the Wikimedia ecosystem, as such using the Wikimedia Cloud Services.

The advantages offered by WMCS are a cost free use of the platform for Wikimedia related projects as well as a flexible computing ecosystem, that meets all requirements for hosting the server. WMCS also provides a wide range of tools, services and support for technical collaborators within the Wikimedia community [60]. An overview of WMCS' service concepts and products are depicted in figure 4.10. For requesting access to services like Toolforge or Cloud VPS a Wikitech account is needed, however their use is restricted per Wikitech Cloud Services Terms of Use. No proprietary software may be used, all project must be published under OSI approved licenses and the collected user information must be clearly communicated to the users [63].

Developers can request access to those products depending on their needs, performance requirements and resources of their respective projects. As depicted in figure 4.11 these include Toolforge, Cloud VPS and Data Services. All three provide different technologies, resources and support level. In the case of Diri, Toolforge is the product that is used for hosting the server.

4.2.2.2 Toolforge Development Process

Toolforge is a shared hosting and platform as a service (PaaS) environment for volunteers who want to create Wikimedia related tools, administer bots, run web services and perform analytics [62]. Tools are software applications, web applications, gadgets, and bots that help people working on Wikimedia projects. A list of tools hosted in Toolforge can be found on on the project's website.

Toolforge, as a *Tool* hosting service offers an unusual development model, due to its multiuser and open nature. The workflow is closer to traditional web hosting environments than the more modern container-based development, but uses elements from both. Like in most shared web hosting environments users don't have access to superuser rights and are therefore limited to the preinstalled system packages. But because Toolforge tries to service a broader spectrum of web services it offers a lot of additional capabilities. These resources include web servers, databases and other data storage, and a distributed job processing system [64].

Additional features include:

- Shared management of tool accounts, where tools and bots are stored

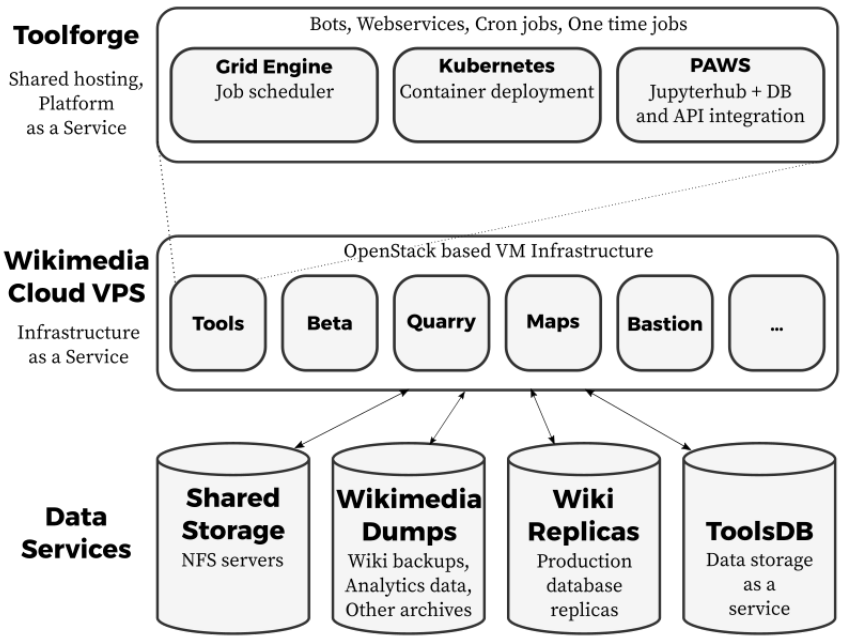


Figure 4.10: Poster describing Wikimedia Cloud Services components [26]

- A grid engine for dispatching jobs
- Support for SSH
- Version control via Gerrit and Git
- Domain names for user tools
- SSL/TLS Certificates
- User databases
- Proxy servers for handling all inbound requests [62]

Since the implementation of the server is based on Python, a Python runtime environment is required. A commonly chosen way is to run processes inside a Docker container, orchestrated by Kubernetes. To do so a Tool account must be created, which allows shared access by multiple maintainers to develop, deploy, and operate their tools [25]. In addition, a *Flask Web Server Gateway Interface* (WSGI) is created called by using the package uWSGI, which forwards requests to web applications or frameworks written in Python while preprocessing most HTTP details. For managing the application's external library dependencies a Python virtual environment is prepared. This allows the tool to install Python libraries locally without needing a Toolforge administrator's help [67]. At last Flask is added to the Python virtual environment. An overview of the Toolforge software Architecture is shown in figure 4.12.

To deploy a Python web service Toolforge uses a configuration design named *conventions rather than configuration*. *Conventions rather than configuration* is a software design paradigm that aims to reduce configuration complexity. As long as developers adhere to common conventions (e.g, identifiers of the same type, file structure) in all areas of software, there is no need to configure them, which simplifies configurations without limiting developers' options [41].

Activity / Needs	Quarry (DaaS)	PAWS (DaaS)	Toolforge (PaaS)	Cloud VPS (IaaS)
Browser based	✔	✔		
Terminal based			✔	✔
Write queries against replica databases	✔	✔	✔	
Run database dumps		✔	✔	
Write and run bots		✔	✔	
Run web services			✔	
Build tools to improve Wikimedia projects			✔	
Schedule or run continuous jobs			✔	
Administer your own virtual server				✔
Need your own subdomain			✔	✔
Write documentation and create tutorials		✔		
Work with co-maintainers and co-admins			✔	✔
User knowledge	curious —advanced	curious —advanced	intermediate —advanced	advanced
Service concept	Data as a service	Data as a service	Platform as a service	Infrastructure as a service

Figure 4.11: Table of WMCS products, their use and support level [65]

Since Toolforge expects these conventions to be followed the default web service configuration will automatically load libraries from `$HOME/www/python/venv` and the entry point of the uWSGI application must be located in `$HOME/www/python/src/app.py` in a variable named *app* [61]. The file structure shown in 4.13 must also be adhered to according to the specified convention design[67].

For version control and deploying new code a Git repository is established.

4.2.3 Database

All persistent information is stored in a MariaDB database instance on a shared server on `tools.db.svc.wikimedia.cloud` inside the Toolforge environment. Users have all privileges and have access to all grant options on their databases. The server gets access to the database through a user account with its own credentials.

The SQL database consists of four tables:

1. **User:** A simple table to keep track of users.
2. **Pronunciation:** Contains a list of German words and their IPA transcriptions without sound recordings.
3. **PronunciationInteraction:** Tracks which user made a sound recording to an IPA transcription of a word. Also tracks the sound file path.

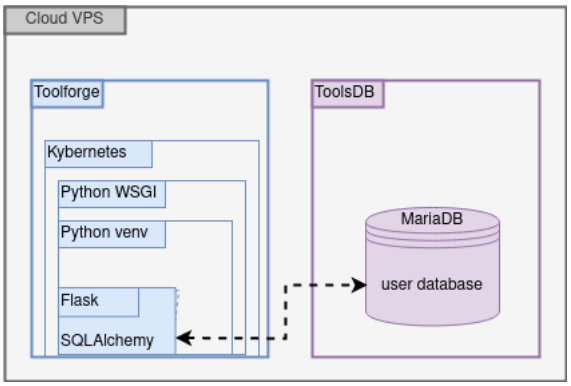


Figure 4.12: Simplified representation of the Toolforge deployment architecture

```
$HOME
├─ uwsgi.log
└─ www
   └─ python
      ├─ src
      │  └─ app.py
      └─ venv
```

Figure 4.13: Expected file layout [65]

4. **SpellingInteraction:** Tracks which user made which inputs for sound recordings.

As the game is producing and using audio files in addition to textual data, the storage requirements on the server side are not negligible. In the process of testing the app, the average file size of the audio recording averaged around 50 KB. On the Kubernetes cluster, all containers run with specific CPU and RAM limits, whereby the default values 0.5 CPU and 512 Mb RAM per container are used. The default storage size of a container is limited to 10 GB. For permanent storage, the `/data/project` directory, which is not inside the container, is used [66].

4.2.4 Object Relational Mapping with Flask-SQLAlchemy

For managing data in the database Flask-SQLAlchemy is used. SQLAlchemy is an open-source Python SQL toolkit and an Object Relational Mapper (ORM) that still gives application developers the full flexibility of SQL [1]. Flask-SQLAlchemy is an extension for Flask that adds support for SQLAlchemy to the application. It aims to simplify using SQLAlchemy with Flask by providing useful defaults and extra helpers that make it easier to accomplish common tasks [6]. The main reason for using the SQL-Alchemy ORM wrapper is that it enables to map the world of objects found in an object-oriented language to rows in tables found in relational databases. More precisely, it simplifies the process of storing, retrieving, updating and deleting from an object-oriented program in a relational database.

Figure 4.14 shows how data is organized in different environments.

Data is stored in two different ways. On the left the perspective of an object or class side is depicted, where a simple class diagram contains all properties. That means data is organized the way objects or classes in programming languages organize data. On the right

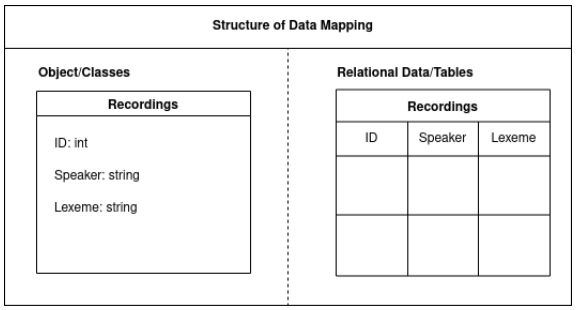


Figure 4.14: Structure of how data is organized in different environments.

side the relational perspective is shown, in which data is organized in tables, in particular in rows and columns. As shown, the structuring of data varies.

To avoid manual mapping, as this task is error-prone and time-consuming the ORM systems utilize a *data layer* to manage translation between the object-oriented program and the relational database. A schematic representation is shown in figure 4.15. The data layer translates the saving of objects as an *insert* operation and the retrieval or creation of new objects as a *select* operation. The *data layer* is typically a library written in the object-oriented language (here Python) that is part of, or works in conjunction with the web framework.

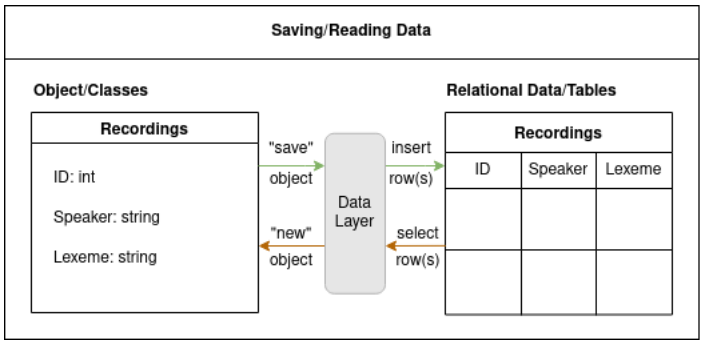


Figure 4.15: Schematic representation of the data layer managing the translation between the object-oriented program and relational database.

4.2.5 Server – Client Communication via a HTTPS API

The communication between the server and the client are established via an HTTP JSON API. As the API specification is quite simple it only handles two kinds of HTTP requests. Figure 4.16 illustrates the server-client communication via the HTTP JSON API.

The *GET /data/* request requires one parameter to send data to the client, the UUID. The UUID, "Universally Unique Identifier", is a 128-bit random string generated by the server to identify its user without a complex login process. This prevents sending duplicate game data to the user. The successful GET request returns a JSON object with two lists.

One list contains a set of words and their IPA transcriptions without audio files for the Speak Mode.

The other list contains user created sound files from the Speak Mode that are used in the Listen and Write mode.

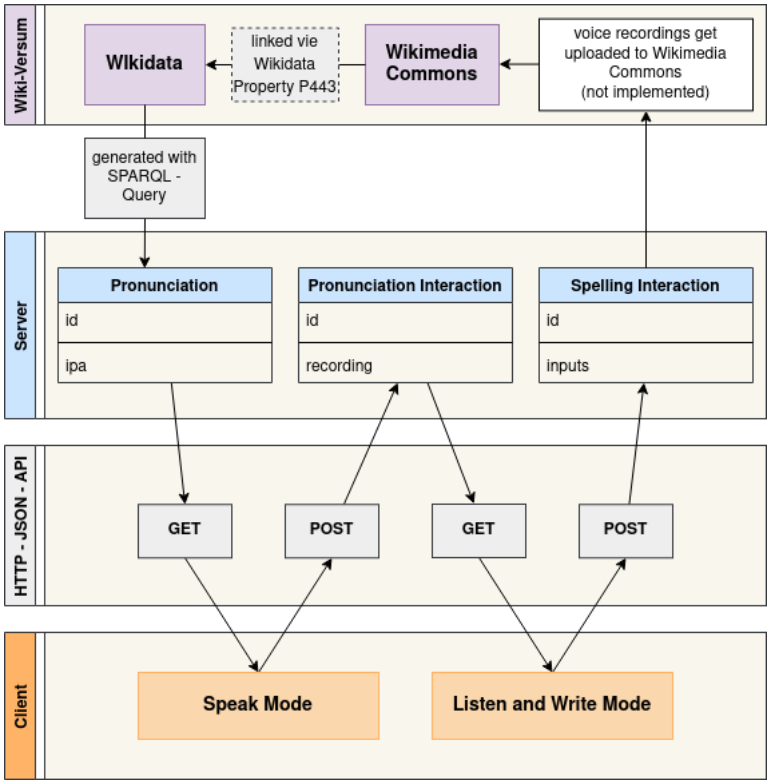


Figure 4.16: Schematic representation of server – client communication via HTTP API.

The *POST /data/* request sends user related data and interaction to the server. It also returns a JSON Object containing information about played games, including skipped games and user inputs from the *Listen and Write Mode*.

In the Speaker Mode a *Multipart POST* method is used for transmitting newly recorded audio files as binary data to the server. Multipart requests are a common way for file uploads and for transferring data of multiple types in a single request (for example, a file along with a JSON object). The additional data is returned in a JSON Object containing information about skipped games. If connection errors occur from the client to the server, client error messages are displayed in the respective game modes, prompting the player to re-establish the connection by pressing a button. This forces a *GET /data/* request and prevents app crashes.

CHAPTER 5

Evaluation

Following the implementation of the Diri prototype, a study was conducted to determine if users were motivated to create voice recordings and whether a GWAP is useful for evaluating these. Diri's testers could use the app for a ten-day period, between 04.05.2022 to 13.05.2022,to get an impression of the concept of the application, followed by an online survey. The testers were informed that Diri is a prototype and that further improvements would be necessary for a later productive operation.

In total 27 people downloaded the app from Google Play Store of which 14 actively tested the application and filled in the questionnaire. During the test phase all user interactions were recorded by the server to later be evaluated. However, no personally identifying information about the users was collected by the server, besides the voice recordings themselves. This was discussed and agreed to with the test group beforehand. The contributions can be measured in quantity and quality and will be evaluated in the following chapter. Some select data from the test period and the online survey can be found in Diri's Gitlab repository.

5.1 Composition of the Test-group

The composition of the test group cannot be determined precisely, as the app was freely available for download on the Google Play Store platform. Nevertheless, the data from the questionnaire provide an insight into the testers, as 14 out of 27 subjects participated in the survey.

50% of the participants were women and 50% men. The age range of the testers was between 17 and 66 years and most of them spoke German (some with a Greek or Russian background) while two testers have Greek as their mother language, as shown in figure 5.1. 50% of the respondents stated that they were familiar with the International Phonetic Alphabet, from which 4 have a background in linguistics (figure 5.2).

5.2 Testing Phase

During the test phase, subjects were asked to use the app for ten days. Due to timeout issues with the database connection, users were only able to test the app when the database

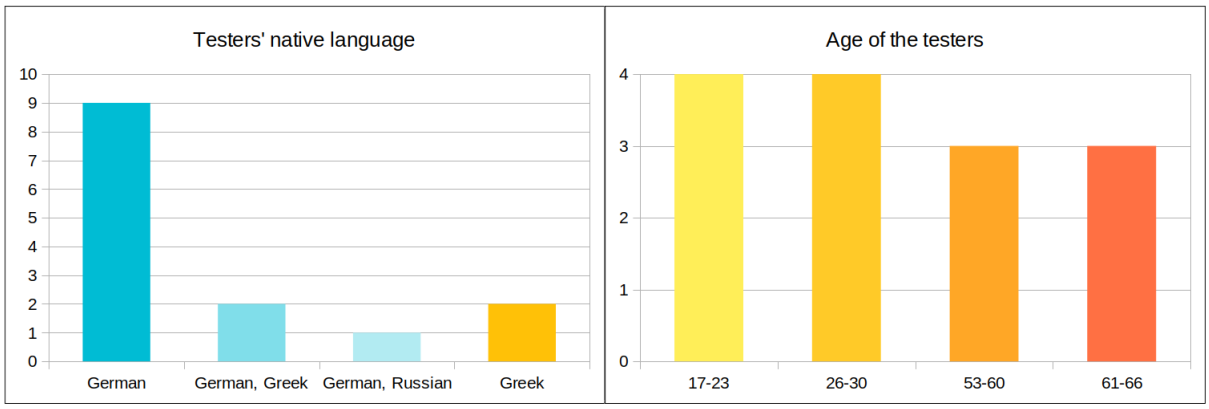


Figure 5.1: Age information of the testers and their spoken languages

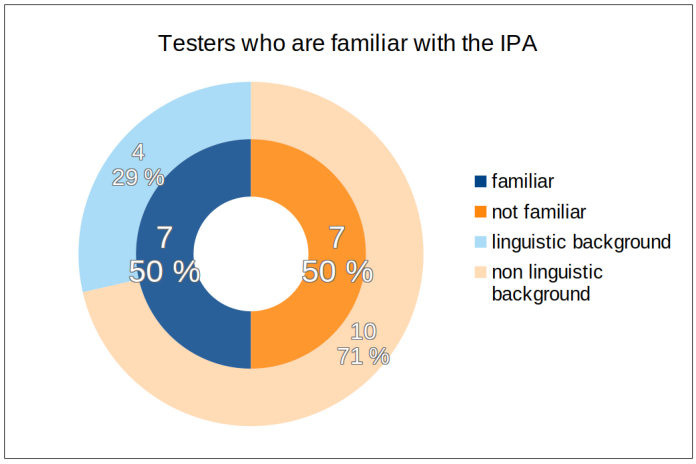


Figure 5.2: IPA transcription familiarity

was actively monitored for errors. Nevertheless, users used the app for extended periods of time and successfully made voice recordings.

The data demonstrated in the following sections was retrieved via a dump of the Toolforge database and a separate download process for the audio files.

5.2.1 Speak Activity

In *Speak mode*, players could record themselves speak reading IPA transcriptions. Of the 27 players, 16 made more than 0 recordings, which means that 11 users did not interact with this game mode and thus no voice recordings were made by these users. In the following only those 16 will be regarded as users. In total, 374 audio recordings were made in the Speak mode, an average of 23.4 recordings per user.

The five transcriptions with the best "skipped" to "recorded" ratio are shown in figure 5.3. In contrast to this, figure 5.4 shows the transcriptions with the worst ratio. Both tables were generated by using only results with at least eight interactions as a cutoff point (half the number of users). Since the number of users was far too small and their usage time with the app cannot be considered significant, it is impossible to draw hard conclusions from this short data excerpt. But there is a slight trend noticeable, as users preferred transcriptions,

that look lexicographically similar to the lexeme.

On the flip side, long transcription and unfamiliar non-alphabet characters seem to lead to a higher likelihood of rejection via skipping the task.

The data is also biased against some lexemes that ended up in the test set multiple times with similar transcriptions, as users didn't like interaction with the same tasks twice. This was remarked upon in the questionnaire and will be expanded upon further later. The lexeme "gehen" (German: to go) suffered explicitly from this as it was included three times with slightly different pronunciations.

Lexeme	IPA	Total interactions by users	# of times skipped	Wikidata-ID
Tafel	[ˈta:fl̩]	12	0	L226914
digital	digiˈta:l	12	1	L2213
Salz	[zalt͡s]	11	1	L35154
Buch	[bu:x]	10	0	L7895
Apfel	ˈapfl̩	10	1	L819

Figure 5.3: Top five transcriptions with the best "skipped" to "recorded" ratio

Lexeme	IPA	Total interactions by users	# of times skipped	Wikidata-ID
Kirchrode	kɪɐ̯çʁo:də	11	7	L407397
Quark	[kwɔ:ɐ̯k]	10	6	L478443
Grundstücksverkehrsgenehmigungszuständigkeitsübertragungsverordnung	ˈgʁʊntʃtʏksfɛɐ̯ˌke:ɐ̯sgəˌne:mɪgʊŋsˌtsu:ʃtɛndɪçkaɪ̯tsʔyːbɐˌtʁa:gʊŋsfɛɐ̯ˌʔɔʁdnʊŋ	9	5	L19614
gehen	ˈge:ən	11	6	L1026
mehrfach	[ˈme:ɐ̯fax]	10	5	L11535

Figure 5.4: Top five transcriptions with the worst "skipped" to "recorded" ratio

5.2.2 Listen and Write Activity

The *Listen and Write mode's* purpose is to check the intelligibility of the recordings made in the *Speak mode* and implicitly rate their quality. The review of the recordings is thus crowdsourced to the users with the help of this game mode.

As both activities could be played independently, the number of users differs. This mode was actively interacted with by 17 people during the evaluation period.

In total 1086 recordings where downloaded of which users actually played 537. The interaction with these recordings was either skipping or solving it by typing in the matching lexeme. The correct words were found for 429 audio recordings and thus solved. 108 recordings were skipped, of which users at least tried to input a guess 47 times.

This suggests the average user played around 31.6 rounds, and guessed correctly ca. 80% of the time.

Due to the nature of the *Speak mode*, which produces multiple recordings of the same transcription, the number of interactions per recording was drastically lower than in the first mode. Of the 374 produced recordings, each one was listened to on average only 1.4 times. Additionally, recordings that were made early in the test phase were listened to many more times than later ones, as is to be expected.

Regarding the qualitative evaluation of the recordings, the tables in figures 5.5 and 5.6 represent the top five best and worst examples from the data set. The cutoff point for the consideration in these tables was a minimum of 4 interactions.

Lexeme	Total interactions by users	# of times skipped	Filename
Abenteuer	7	0	254fde51-6f59-4c27-a2cb-92d8023e8b0d.mp4
Dienstag	6	0	afd514b6-76dd-4120-af14-1d39c834af83.mp4
Bierglas	5	0	a8f2cc01-0418-47df-9cb3-b1c186d9f1bf.mp4
Bindungsangst	5	0	3cde65b4-41a2-4dd5-bcfb-8e507ffb67d3.mp4
Mittwoch	5	0	ae9d8b78-9dab-40d6-8e09-a782d64c054c.mp4

Figure 5.5: Top five recordings with the best “skipped” to “solved” ratio. Link to audio files

Lexeme	Total interactions by users	# of times skipped	Filename
digital	5	4	9609d3cf-0197-41e4-99c9-4ebfece99307.mp4
Misburg-Nord	4	3	483e4959-f204-437d-a80c-933057df475d.mp4
Ricklingen	6	3	10e529e2-8868-4571-990c-c63ca157f130.mp4
Katze	4	2	ad5ea36e-2c68-4b52-8822-0c180bacf20e.mp4
Kirchrode	4	2	a31d8e9b-f7ee-42fc-9fc3-60af2feb18d3.mp4

Figure 5.6: Top five transcriptions with the worst “skipped” to “solved” ratio. Link to audio files

Most of the recordings with the best performance benefitted from the mentioned bias in favor of earlier recordings, as they were taken by the author. On the qualitative side of things they are all sufficiently clear in their audio quality, while not further interesting, as the test group and period was not long enough to create any stronger trends.

More valuable are the less well performing recordings, as they all show sings of technical and phonetic errors. In this small group nearly every file is cut of or too noisy to understand. This suggests that at least the negative implication of quality by only a few testers is enough to catch obviously inadequate audio files.

5.3 User Survey Data

Following the ten day testing period, users where asked to fill out a survey in order to further asses the concept of the game. The survey was conducted using the Framaforms.org

platform, which is operated by a French non-profit association, Framasoft. The questionnaire itself consisted of five parts. First, data about the test subjects was collected (results were already shown in the composition of the test group), followed by questions about the design. Third, questions about the game experience were asked. Next the learning experience was evaluated. The questionnaire ends with a general evaluation of the mobile application and feedback. Of the 27 users of the app, 14 completed the questionnaire.

5.3.1 Results of the Survey

In general, the users were satisfied with the simplicity of the app's design. In fact, the respondents gave only positive answers about the appearance and clarity. Additionally, respondents stated that they could navigate the app intuitively and that all elements were displayed correctly with a reasonable response time. An illustration of the respondents' answers can be seen in Figure 5.7.

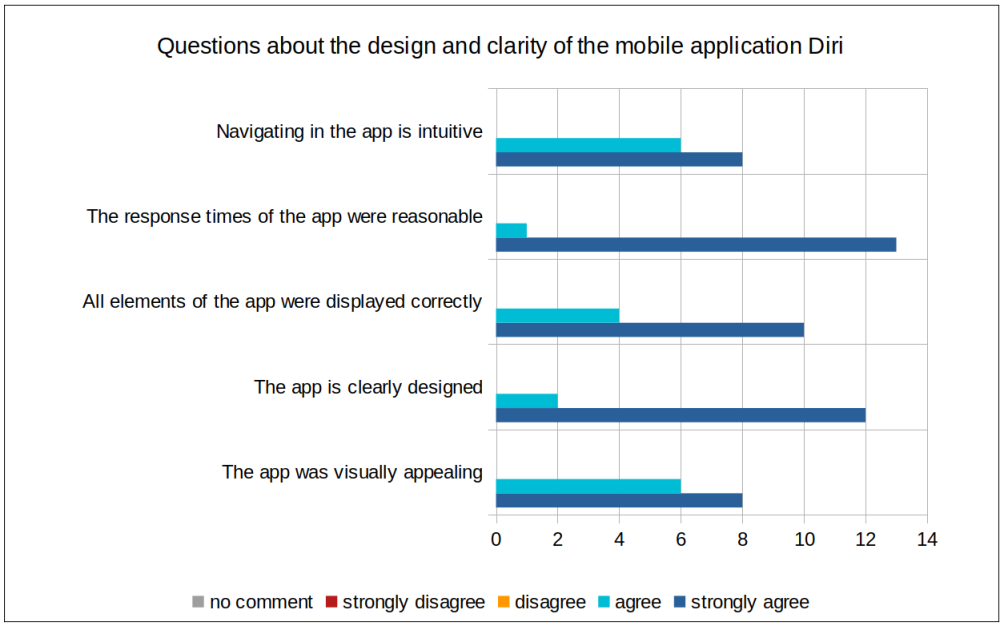


Figure 5.7: User responses about the design and clarity of the app

The next section of the questionnaire relates to usability in terms of Diri's gaming experience. Here, testers were able to answer questions in the area of the gaming modes' tasks, functionality of the app activities, sound quality and general enjoyment (see figure 5.8). Questions about the tasks were generally answered positively. 13 out of 14 testers stated that they understood the tasks proposed by the app. The testers also denied that they were faced with too difficult tasks (12 out of 14). This is also consistent, at least for the *Speak mode*, with the statement that nine of the respondents had no problems understanding the phonetic symbols. This statement is somewhat interesting, since only seven of the respondents stated that they were familiar with the International Phonetic Alphabet (see figure 5.2). In terms of functionality and interaction with the app, four users stated they were missing features that would have helped them use the app. If the respondents agreed with whether features were missing, they could also enter their suggestions in a text field. Six respondents used this option to make an entry, although only four respondents answered in the affirmative. The most desired features that could enhance the use of the app are listed in figure 5.9.

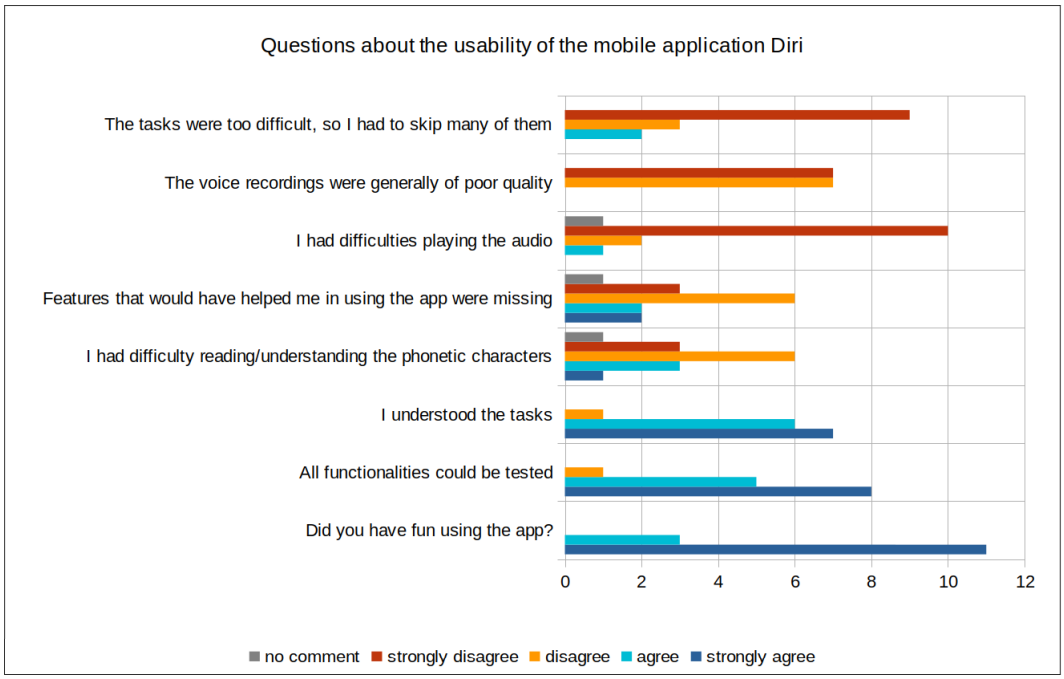


Figure 5.8: User responses about the usability of the app

These features were missing, which would have helped me in using the app	**# of users who made similar comments**
Short introduction/tutorial to the respective game modes	3
Microphone response: Visual sign as an indicator that the microphone is recording (e.g. change of background color).	3
Explain special characters/diacritics (characters on the phonetic letters)	1
Automatic playback of your own recording before confirming it	1

Figure 5.9: User feature requests

The testers noticed some minor problems during the test phase. For example, the phoneme /g/ could not be generated as a button, and the phoneme buttons generated from long words stretched across the entire screen. Two testers were unable to play sound for a short period of time. This error was resolved when the app was closed and reopened. All respondents found the sound quality of the voice recordings to be at least not bad. All testers stated that they had fun playing the game.

To get an idea of whether Diri can also give testers an understanding of the phonetic alphabet and thus an overview of pronunciation, the user study included questions about the learning experience. A look at figure 5.10 shows that there is a general positive perception towards the learning experience with the app. For example, all 14 participants said

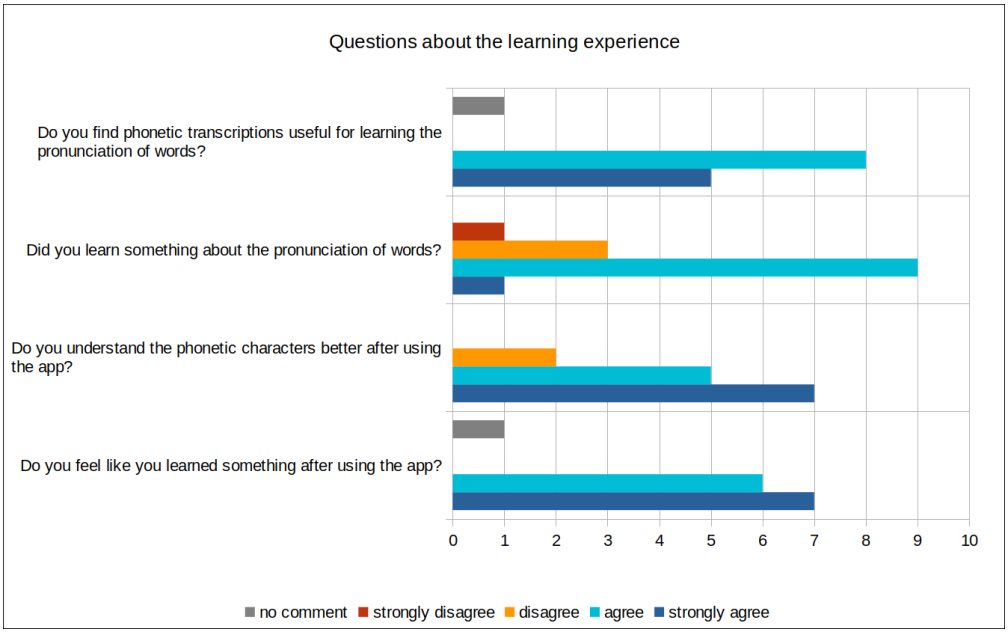


Figure 5.10: Testers responses to questions about their learning experience

they had learned something with the app, 12 of whom reported that they understood phonetic symbols better after using the app, and 10 participants agreed that they had learned something about the pronunciation of words. A selection of the testers' feedback on their learning experience can be seen below in figure 5.11.

The app motivates users to create voice recordings. It's especially fun to engage with the phonetic transcriptions instead of just reading off words.
This app stimulates interest to continue playing. The transcription of phonetic speech into spoken language is particularly exciting. Practicing this activity is certainly useful in foreign language learning.
An entertaining game for in between for people who are interested in language.
Very good for practicing spelling

Figure 5.11: Feedback on the learning experience gained by the users

In general, the concept of Diri was rated as positive. Not only were all users able to test basic functionality, but all respondents found the interaction with the app fun. Thus, 46% of the users rated the app with 5/5 stars and 46% with 4/5. Furthermore, almost all respondents were positive about further use of a finished version of the app (46% agree and 46% strongly agree). Also, 50% of the testers agreed to recommend the app and 29% agreed strongly.

CHAPTER 6

Conclusion and Perspectives

Games with a purpose are research tools that allow to use the mass of Internet users to collect data and solve crowdsourcing problems in many fields, playfully.

In this thesis, Diri, a game with the purpose to create voice recordings and letting them be evaluated by other users, was conceptualized, implemented and tested.

In Diri, players can create speech recordings based on phonetic transcriptions. These are then, as the game progresses, evaluated for intelligibility by other players. This serves several purposes: On the one hand, players can engage in new learning experiences in the area of word pronunciation and spelling, in which they have to interact with phonetic transcriptions, while their spelling skills are put to the test. On the other hand, Diri generates pronunciation audio via a mobile platform and crowdsources the evaluation process, so that these recordings then could be reused in different applications.

By conducting a study, it was possible to determine that users are entertained by Diri's concept and that the app is capable of generating voice recordings from a range of different speakers. Especially the evaluation process in *Listen and Write mode* was the most played by the users and shows that Diri has the potential to effectively evaluate speech recordings in a playful way.

The number of testers and the duration of the testing period do not provide sufficient data to draw definitive or even strong conclusions. Further research is needed to confirm the validity of the concept to engage users long term. Therefore, future studies should both recruit more participants and test the app over a longer period of time. For example, a survey could qualitatively compare the speech recordings from Diri with recordings from other projects or professional speech corpora to see if similar recordings can be produced with a mobile phone.

Since Diri is still in the prototype phase, certain issues should be addressed in order to make the app usable for the public and thus start the next testing phases.

One of the main issues are the unpredictable connection timeouts due to the MariaDB database on Toolforge. This problem should be further investigated to ensure proper operation of the app.

Since voice data is sometimes considered sensitive data, the concept of Diri should be changed so that only with the users' consent in the app their voice recordings can be uploaded to Wikimedia Commons. Also, the audio file format should be changed to comply with Wikimedia Commons rules, as only formats like Ogg, WAV, Mp3, FLAC are supported. In comparison, Diri records in MPEG4 format.

Furthermore, duplicate words should be removed from the game generation of Wikidata, as there is a risk that they will be skipped due to frequent occurrence. The duplications are caused by lexemes with various IPA transcriptions in the Wikidata database.

In addition, it should be examined if the skip option in the *Listen and Write mode* is a good indicator for whether a word was spoken understandably. Also, the text input that players make is not always useful for drawing conclusions about intelligibility. The *Listen and Write mode* could be extended so that more buttons are available encourage more detailed statements about the recording. Also, different game modes for the evaluation of the voice recordings could be considered, in which different criteria are to be paid attention to, e.g. whether other noises can be heard in the background or whether the recording is cut off.

This work only generally presented a prototype implementation and not a fully functional app with differentiated game elements. Suggestions of the users during the study should therefore be implemented in the future development of the game.

Numerous suggestions were made that are related to the scoring system. For example, players wanted a better scoring mechanism so that different points would be awarded depending on the difficulty of the transcribed word. Also, it was suggested to introduce a level system according to certain scores, so that players have the incentive to reach a higher one. Moreover, the hint button, which reveals the word behind the phonetic transcription, was criticized, and a suggestion was made that this button could be used after spending a certain amount of points. Other user requests were related to the feedback behavior of the app. The suggestions to implement a leaderboard to see how many recordings other users have created could encourage users do the same.

Due to time constraints, it was not possible to adapt Diri to the additional collection of other language learning features. For instance, the game could be extended to include multiple languages, so that more diversified voice recordings could be generated. It would be particularly interesting to see differences in the pronunciation of people who record outside their native language. By implementing an additional correction mode, the pronunciations of players recording in non-native languages could be evaluated by other players with some kind of feedback. This advanced implementation could enable social interactivity aspects in Diri. To extend this idea, users could have other users re-record one of their voice recordings and then rate it or provide feedback later. This would allow more voice recordings to be created and players to get feedback from other players about their recordings or pronunciations. Users who are interested in learning other languages might particularly enjoy such game modes.

To summarize, this work has investigated and explored the potential of using a GWAP to create voice recordings along with their evaluation by users. The next step is to improve Diri's gameplay, solve technical problems, and conduct studies on a larger scale to further evaluate the qualities of the Diri application.

CHAPTER 7

Appendix

Figure 7.1: User survey: personal questions

Fragen zur App: Design und Übersichtlichkeit : 2 / 6

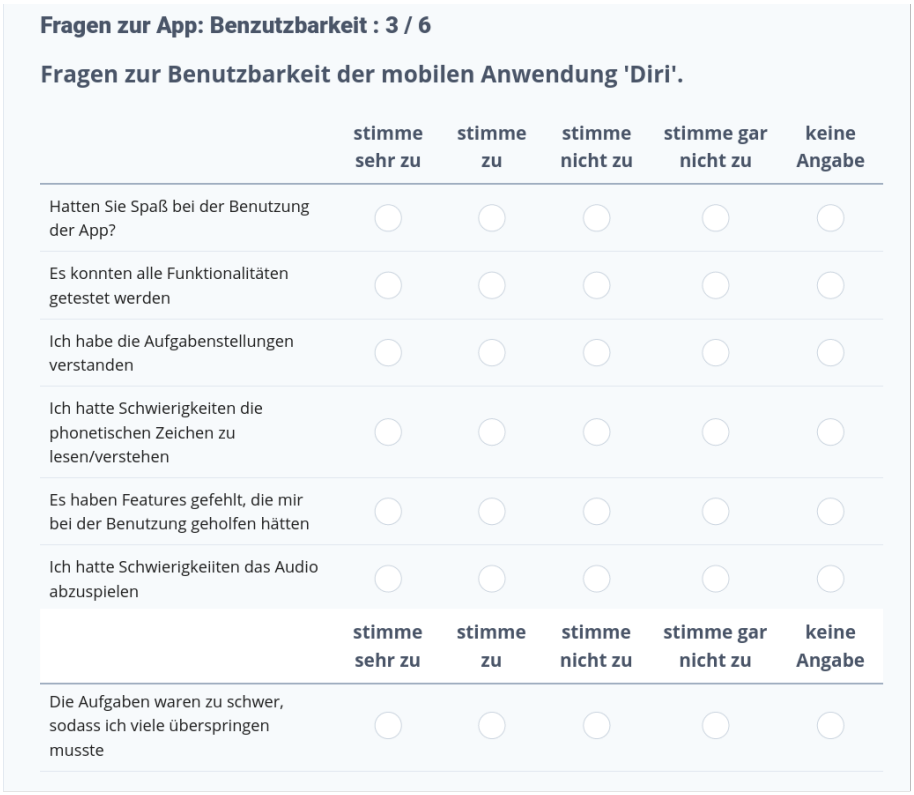
Fragen zum Design und Übersichtlichkeit der mobilen Anwendung 'Diri'.

	stimme sehr zu	stimme zu	stimme nicht zu	stimme gar nicht zu	keine Angabe
Die App war optisch ansprechend gestaltet	○	○	○	○	○

	stimme sehr zu	stimme zu	stimme nicht zu	stimme gar nicht zu	keine Angabe

Alle Elemente der App wurden korrekt angezeigt	○	○	○	○	○
Die Antwortzeiten der App waren angemessen	○	○	○	○	○
Das navigieren in der App ist intuitiv	○	○	○	○	○

Figure 7.2: User survey: questions about design and clarity



Fragen zur App: Benzutzbarkeit : 3 / 6

Fragen zur Benutzbarkeit der mobilen Anwendung 'Diri'.

	stimme sehr zu	stimme zu	stimme nicht zu	stimme gar nicht zu	keine Angabe
Hatten Sie Spaß bei der Benutzung der App?	○	○	○	○	○
Es konnten alle Funktionalitäten getestet werden	○	○	○	○	○
Ich habe die Aufgabenstellungen verstanden	○	○	○	○	○
Ich hatte Schwierigkeiten die phonetischen Zeichen zu lesen/verstehen	○	○	○	○	○
Es haben Features gefehlt, die mir bei der Benutzung geholfen hätten	○	○	○	○	○
Ich hatte Schwierigkeiiten das Audio abzuspielen	○	○	○	○	○

	stimme sehr zu	stimme zu	stimme nicht zu	stimme gar nicht zu	keine Angabe

Die Aufgaben waren zu schwer, sodass ich viele überspringen musste	○	○	○	○	○

Figure 7.3: User survey: questions about the usability of the app

Figure 7.4: User survey: questions about the usability of the app

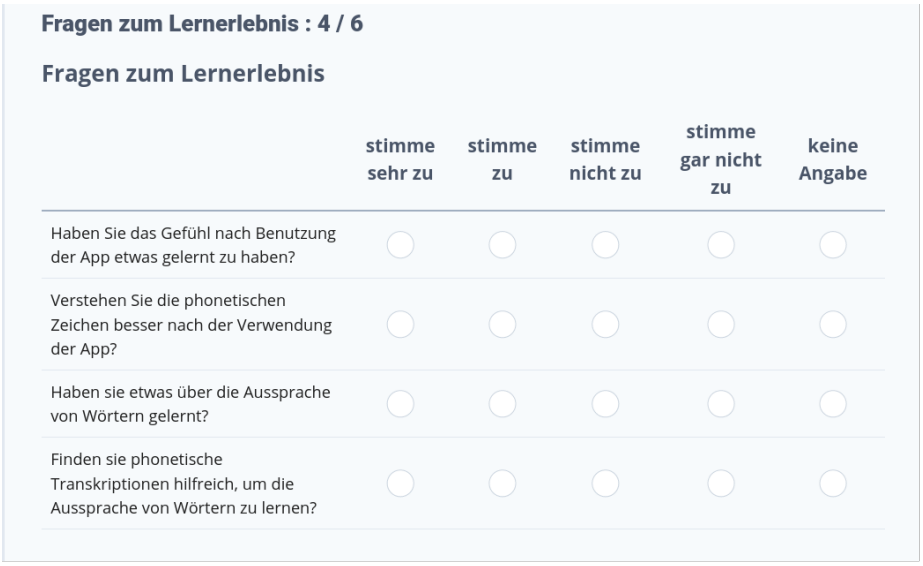


Figure 7.5: User survey: questions regarding the learning experience

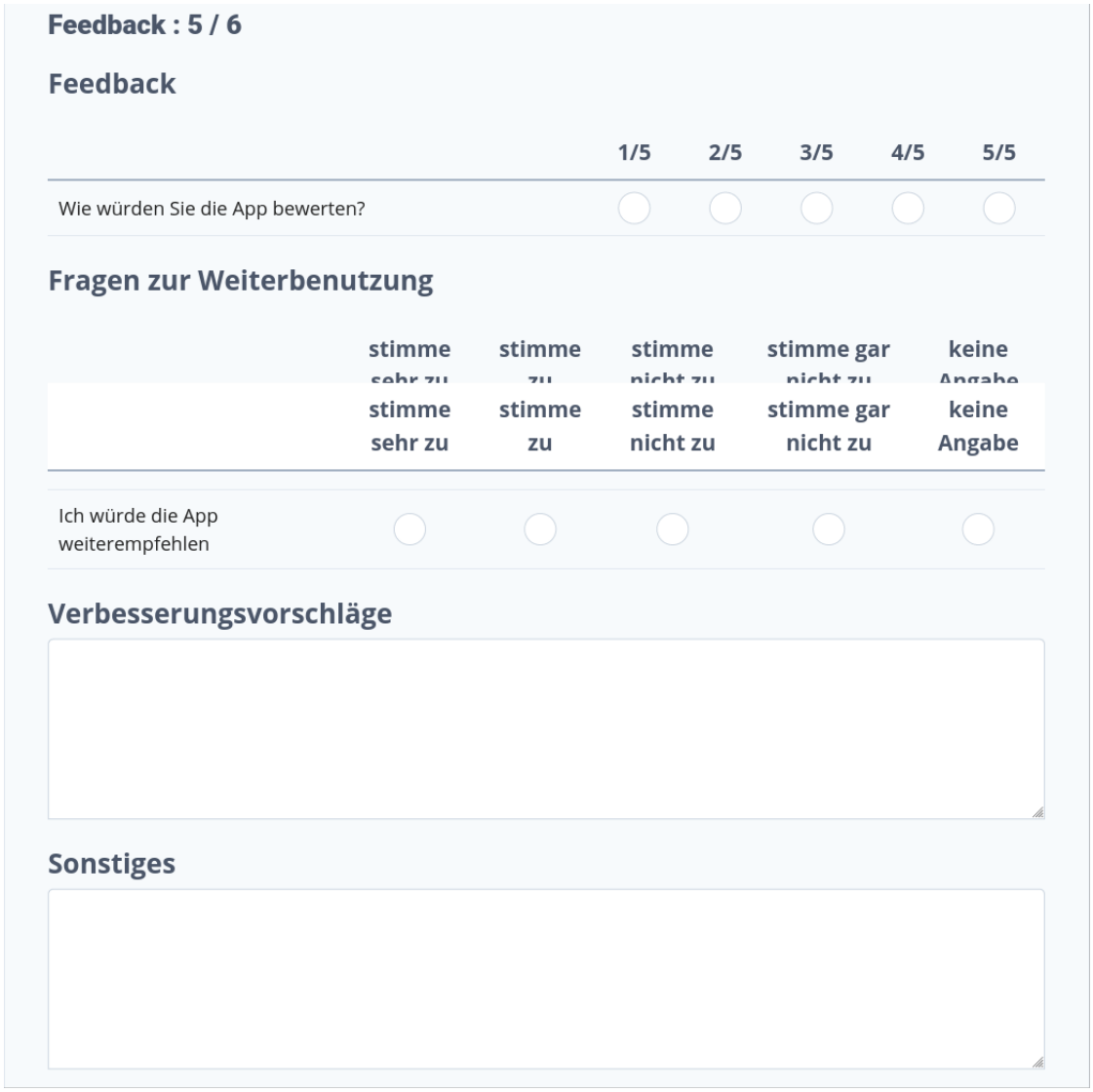


Figure 7.6: User survey: User feedback

Bibliography

[1] *SQLAlchemy - The Database Toolkit for Python*, https://www.sqlalchemy.org, [Online; accessed 12. Apr. 2022].

[2] *Wikimedia Foundation Mission*, https://wikimediafoundation.org/about/mission, Sep 2018, [Online; accessed 14. Mar. 2022].

[3] *Android MVVM Design Pattern - JournalDev*, https://www.journaldev.com/20292/android-mvvm-design-pattern, Mar 2019, [Online; accessed 17. Nov. 2021].

[4] *Common Voice by Mozilla*, https://commonvoice.mozilla.org/en, Nov 2021, [Online; accessed 29. Nov. 2021].

[5] *Datasets — Common Voice by Mozilla*, https://commonvoice.mozilla.org/en/datasets, Jul 2021, [Online; accessed 29. Nov. 2021].

[6] *Flask-SQLAlchemy — Flask-SQLAlchemy Documentation (2.x)*, https://flask-sqlalchemy.palletsprojects.com/en/2.x, Dec 2021, [Online; accessed 12. Apr. 2022].

[7] *Guide to app architecture | Android Developers*, https://developer.android.com/jetpack/guide#mobile-app-ux, Aug 2021, [Online; accessed 18. Nov. 2021].

[8] *Save data in a local database using Room | Android Developers*, https://developer.android.com/training/data-storage/room, Oct 2021, [Online; accessed 19. Nov. 2021].

[9] *Save data using SQLite | Android Developers*, https://developer.android.com/training/data-storage/sqlite, Oct 2021, [Online; accessed 19. Nov. 2021].

[10] *Wikimedia Projects*, https://wikimediafoundation.org/our-work/wikimedia-projects/#a1-reference, Jun 2021, [Online; accessed 14. Mar. 2022].

[11] *All public lists containing "audio" (156) - Tatoeba*, https://tatoeba.org/en/sentences_lists/index/audio, 2022, [Online; accessed 10. May 2022].

[12] *Amazon Web Services AWS – Server Hosting & Cloud Services*, https://aws.amazon.com/de, Feb 2022, [Online; accessed 10. Feb. 2022].

[13] *FAQ — Tatoeba*, https://en.wiki.tatoeba.org/articles/show/faq#, 2022, [Online; accessed 10. May 2022].

[14] *Forvo Pronunciation API, plans and princing*, https://api.forvo.com/plans-and-pricing, Feb 2022, [Online; accessed 16. Feb. 2022].

[15] *Notes on how to use this free online pronunciation dictionary | howjsay*, https://howjsay.com/donations, Feb 2022, [Online; accessed 16. Feb. 2022].

[16] *Number of sentences per language — Tatoeba*, https://tatoeba.org/en/stats/sentences_by_language, 2022, [Online; accessed 10. May 2022].

[17] *PostgreSQL driver for Python — Psycopg*, https://www.psycopg.org, May 2022, [Online; accessed 15. May 2022].

[18] *Privacy Policy - Howjsay*, https://howjsay.com/terms-conditions, Feb 2022, [Online; accessed 16. Feb. 2022].

[19] *Supported media formats Android Developers*, https://developer.android.com/guide/topics/media/media-formats, May 2022, [Online; accessed 2. May 2022].

[20] *What is Tatoeba? — Tatoeba*, https://tatoeba.org/en/about, 2022, [Online; accessed 10. May 2022].

[21] Kohn Alfie, *Punished by Rewards: The Trouble with Gold Stars, Incentive Plans, A's, Praise, and Other Bribes*, Houghton Mifflin Company, 1993.

[22] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, *Common Voice: A Massively-Multilingual Speech Corpus*, CoRR **abs/1912.06670** (2020), https://doi.org/10.48550/arxiv.1912.06670.

[23] International Phonetic Association, *Handbook of the international phonetic association : A guide to the use of the international phonetic alphabet*, Cambridge University Press, 1999.

[24] David Britch, *Preface to Enterprise App Development - Xamarin*, https://docs.microsoft.com/en-us/xamarin/xamarin-forms/enterprise-application-patterns/preface, Nov 2021, [Online; accessed 18. Nov. 2021].

[25] Bryan Davis, *Introducing the Cloud Services Team: What we do, and how we can help you*, https://diff.wikimedia.org/2017/09/11/introducing-wikimedia-cloud-services, Sep 2017, [Online; accessed 11. Apr. 2022].

[26] Wikimedia Commons, *File:WMCS components.svg - Wikimedia Commons*, https://commons.wikimedia.org/w/index.php?curid=62184119, Mar 2022, [Online; accessed 11. Apr. 2022 – Licence Bryan Davis - Own work, based on diagram initially created by Madhumitha Viswanathan, CC BY-SA 4.0].

[27] Emily Moline, *Covering all the bases: Duolingo's approach to speaking skills*, https://blog.duolingo.com/covering-all-the-bases-duolingos-approach-to-speaking-skills, Oct 2020, [Online; accessed 7. Mar. 2022].

[28] Eric Enge, *Mobile vs. Desktop Usage in 2020*, https://www.perficient.com/insights/research-hub/mobile-vs-desktop-usage, 2022, [Online; accessed 11. May 2022].

[29] Wikimedia France, *LinguaLibre:About - Lingua Libre*, https://lingualibre.org/wiki/LinguaLibre:About, May 2021, [Online; accessed 20. May 2021].

[30] Terence Govender and Joan Arnedo-Moreno, *A Survey on Gamification Elements in Mobile Language-Learning Applications*, Eighth International Conference on Technological Ecosystems for Enhancing Multiculturality (New York, NY, USA), TEEM'20, Association for Computing Machinery, 2020, https://doi.org/10.1145/3434780.3436597, ISBN – 9781450388504, p. 669–676.

[31] Hessischer Rundfunk, *ARD-Aussprachedatenbank*, HR (2017), https://www.hr.de/unternehmen/rolle-innerhalb-der-ard/ard-einrichtungen-im-hr/ard-aussprachedatenbank,ard-einrichtungen-aussprachedatenbank-100.html, [Online; accessed 14. Feb. 2022].

[32] International Phonetic Association, *IPA Chart*, https://www.internationalphoneticassociation.org/content/ipa-chart, July 2020, [Online; accessed 19. Nov. 2021 – License International Phonetic Association CC BY-SA 3.0 Unported].

[33] Jens Ohlig, *Lexicographical data on Wikidata: Words, words, words – Wikimedia Deutschland Blog*, https://blog.wikimedia.de/2019/03/25/lexicographical-data-on-wikidata-words-words-words, Mar 2019, [Online; accessed 14. Mar. 2022].

[34] Karl M. Kapp, *The gamification of learning and instruction: Game-based methods and strategies for training and education*, Pfeiffer essential resources for training and HR professionals, Wiley, 2014.

[35] Maximilian Kristen, *Como: A Lexicographical Data Structuring Game With A Purpose – Institute of Computer Science – LMU, Munich*, http://www.pms.ifi.lmu.de/publikationen/#PA_Maximilian.Kristen, 2021, [Online; accessed 7. Mai. 2022].

[36] Mathieu Lafourcade, Alain Joubert, and Nathalie Le Brun, *Games with a Purpose (GWAPs)*, Wiley Online Library, Jul 2015, https://doi.org/10.1002/9781119136309, ISBN – 978-1-84821803-1.

[37] Leibniz-Rechenzentrum, *Webhosting-Angebot*, https://doku.lrz.de/display/PUBLIC/Webhosting-Angebot, 2022, [Online; accessed 10. Feb. 2022].

[38] Lydia Pintscher, *First steps of Wikidata in the Hungarian Wikipedia*, https://blog.wikimedia.de/2013/01/14/first-steps-of-wikidata-in-the-hungarian-wikipedia, Jan 2013, [Online; accessed 14. Mar. 2022].

[39] Markus Manske, *Wikidata – The Distributed Game*, https://wikidata-game.toolforge.org/distributed/#, June 2020, [Online; accessed 2. May 2022].

[40] Meta, *Wiktionary — Meta, discussion about Wikimedia projects*, https://meta.wikimedia.org/w/index.php?title=Wiktionary&oldid=23129642, 2022.

[41] Jeremy Miller, *Patterns in Practice - Convention Over Configuration*, https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-in-practice-convention-over-configuration, Feb 2009, [Online; accessed 10. Apr. 2022].

[42] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher, *From Freebase to Wikidata: The Great Migration*, Proceedings of the 25th International Conference on World Wide Web, WWW '16, International World Wide Web Conferences Steering Committee, 2016, p. 1419–1428.

[43] R.M. Ryan and E.L. Deci, *Self-determination theory: Basic psychological needs in motivation, development, and wellness*, Guilford Publications, 2017.

[44] Scholia, *Scholia information on Wikidata*, https://scholia.toolforge.org/topic/Q2013, Mar 2022, [Online; accessed 7. Mai. 2022].

[45] Mitchell Shortt, Shantanu Tilak, Irina Kuznetcova, Bethany Martens, and Babatunde Akinkuolie, *Gamification in mobile-assisted language learning: A systematic review of Duolingo literature from public release of 2012 to early 2020*, Computer Assisted Language Learning (2021), https://doi.org/10.1080/09588221.2021.1933540.

[46] Josh Smith, *Patterns - WPF Apps With The Model-View-ViewModel Design Pattern*, https://docs.microsoft.com/en-us/archive/msdn-magazine/2009/february/patterns-wpf-apps-with-the-model-view-viewmodel-design-pattern, Feb 2009, [Online; accessed 17. Nov. 2021].

[47] Bartholomäus Steinmayr, Christoph Wieser, Fabian Kneißl, and Fraçois Bry, *Karido: A gwap for telling artworks apart*, 2011 16th International Conference on Computer Games (CGAMES), 2011, 10.1109/CGAMES.2011.6000338, pp. 193–200.

[48] Sylviane Granger and Magali Paquot (eds), *Electronic Lexicography.*, OUP Oxford, 2012.

[49] Team, *About Forvo, the audio pronunciation guide*, https://forvo.com/about, Feb 2022, [Online; accessed 15. Feb. 2022].

[50] _____, *Forvo: the pronunciation dictionary. All the words in the world pronounced by native speakers*, https://forvo.com, Feb 2022, [Online; accessed 15. Feb. 2022].

[51] Luis von Ahn, *Games with a purpose*, IEEE Computer Magazine **39** (2006), 96–98, https://www.cs.cmu.edu/~biglou/ieee-gwap.pdf.

[52] Luis von Ahn and Laura Dabbish, *Labeling images with a computer game*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (New York, NY, USA), Association for Computing Machinery, 2004, https://doi.org/10.1145/985692.985733, p. 319–326.

[53] Wikidata Contributors, *Wikidata — Main Page*, https://www.wikidata.org/w/index.php?title=Wikidata:Main_Page&oldid=1086709037, Dec 2019, [Online; accessed 14. Mar. 2022].

[54] _____, *Wikidata:Lexicographical data - Wikidata*, https://www.wikidata.org/wiki/Wikidata:Lexicographical_data, Apr 2022, [Online; accessed 14. Mar. 2022].

[55] Wikimedia Commons, *File:Wikimedia logo family complete-2022.svg — Wikimedia Commons, the free media repository*, https://commons.wikimedia.org/w/index.php?title=File:Wikimedia_logo_family_complete-2022.svg&oldid=647124640, [Online; accessed 12. Mar. 2022 – License CC BY-SA 3.0].

[56] _____, *File:Lingua Libre - Record Wizard - Studio.png — Wikimedia Commons, the free media repository*, https://commons.wikimedia.org/w/index.php?title=File:Lingua_Libre_-_Record_Wizard_-_Studio.png&oldid=508762774, 2020, [Online; accessed 19. Nov. 2021 – Licence Pamputt, CC BY-SA 4.0].

[57] _____, *File:Lingua Libre home page 2020-12.png — Wikimedia Commons, the free media repository*, https://commons.wikimedia.org/w/index.php?title=File:Lingua_Libre_home_page_2020-12.png&oldid=575848707, 2021, [Online; accessed 19. Nov. 2021 – License Wikimedia France, CC BY-SA 4.0].

[58] _____, *Commons:Welcome — Wikimedia Commons*, https://commons.wikimedia.org/w/index.php?title=Commons:Welcome&oldid=524107325, Mar 2022, [Online; accessed 13. Mai. 2022].

[59] _____, *List of Wiki audio projects*, https://commons.wikimedia.org/w/index.php?title=List_of_Wiki_audio_projects&oldid=632474901, 2022, [Online; accessed 2-May-2022].

[60] Wikipedia, *Wikipedia:Wikimedia Cloud Services — Wikipedia*, https://en.wikipedia.org/w/index.php?title=Wikipedia:Wikimedia_Cloud_Services&oldid=1014141994, 2021, [Online; accessed 21. May 2021].

[61] Wikitech, *Help:Toolforge/Web/Python — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Help:Toolforge/Web/Python&oldid=1935041, 2021, [Online; accessed 10. Apr. 2022].

[62] _____, *Portal:Toolforge/About Toolforge — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Portal:Toolforge/About_Toolforge&oldid=1939404, 2021, [Online; accessed 19. May 2021].

[63] _____, *Wikitech:Cloud Services Terms of use — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Wikitech:Cloud_Services_Terms_of_use&oldid=1880338, May 2021, [Online; accessed 19. May 2021].

[64] Wikitech, *Help:At a glance: Cloud VPS and Toolforge — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Help:Cloud_Services_introduction&oldid=1962985, May 2022, [Online; accessed 8-May-2022].

[65] Wikitech, *Help:Cloud Services introduction — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Help:Cloud_Services_introduction&oldid=1962985, 2022, [Online; accessed 6. Apr. 2022].

[66] _____, *Help:Toolforge/Kubernetes — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Help:Toolforge/Kubernetes&oldid=1949058, 2022, [Online; accessed 10. Apr. 2022].

[67] _____, *Help:Toolforge/My first Flask OAuth tool — Wikitech*, https://wikitech.wikimedia.org/w/index.php?title=Help:Toolforge/My_first_Flask_OAuth_tool&oldid=1949040, 2022, [Online; accessed 10. Apr. 2022].

[68] Wiktionary, *Wiktionary, the free dictionary*, https://en.wiktionary.org/w/index.php?title=Wiktionary:Main_Page&oldid=66535345, 04 2022, [Online; accessed 30. Apr. 2022].

[69] Sascha Wolfer and Carolin Müller-Spitzer, *How Many People Constitute a Crowd and What Do They Do? Quantitative Analyses of Revisions in the English and German Wiktionary Editions*, Lexikos **26** (2016), 347–371.

[70] Yates, Kieran, *Why do we gender AI? Voice tech firms move to be more inclusive*, the Guardian (2020), https://www.theguardian.com/technology/2020/jan/11/why-do-we-gender-ai-voice-tech-firms-move-to-be-more-inclusive [Online; accessed 29. Nov. 2021].