



**BRNO UNIVERSITY OF TECHNOLOGY**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF INTELLIGENT SYSTEMS**

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

**EDUCATIONAL APPLICATION FOR  
CZECH SIGN LANGUAGE**

VÝUKOVÁ APLIKACE PRO ČESKÝ ZNAKOVÝ JAZYK

**MASTER'S THESIS**

DIPLOMOVÁ PRÁCE

**AUTHOR**

AUTOR PRÁCE

**Bc. KRISTIÁN HEŘMAN**

**SUPERVISOR**

VEDOUCÍ PRÁCE

**Mgr. KAMIL MALINKA, Ph.D.**

**BRNO 2024**

# Master's Thesis Assignment



152044

Institut: Department of Intelligent Systems (DITS)  
Student: **Heřman Kristián, Bc.**  
Programme: Information Technology and Artificial Intelligence  
Specialization: Intelligent Devices  
Title: **Educational Application for Czech Sign Language**  
Category: Software Engineering  
Academic year: 2023/24

## Assignment:

1. Study the languages for the deaf in the Czech Republic.
2. Learn about existing projects and applications that facilitate communication between deaf and hearing people. Compare existing educational applications for sign language.
3. Propose an appropriate method of teaching sign language.
4. Design an educational application for teaching Czech sign language using the proposed method.  
Focus on practicing individual signs with checking the correctness of the demonstration (by comparing with reference videos).
5. Implement the application.
6. Test the application and evaluate the educational benefits.

## Literature:

- ISO 639-3 Czech Sign Language
- David, D., Alamoodi, A.H., Albahri, O.S. *et al.* Landscape of sign language research based on smartphone apps: coherent literature analysis, motivations, open challenges, recommendations and future directions for app assessment. *Univ Access Inf Soc* (2023).  
<https://doi.org/10.1007/s10209-022-00966-9>

Requirements for the semestral defence:  
Items 1 to 3.

Detailed formal requirements can be found at <https://www.fit.vut.cz/study/theses/>

Supervisor: **Malinka Kamil, Mgr., Ph.D.**  
Head of Department: Hanáček Petr, doc. Dr. Ing.  
Beginning of work: 1.11.2023  
Submission deadline: 17.5.2024  
Approval date: 6.11.2023

## Abstract

The aim of this thesis was to design and create an educational application for Czech sign language, which will be used as a application for learning and practicing vocabulary. The application uses the Flask web framework for the server side, Jinja2 for HTML templates and the Socket.IO library for real-time communication between client and server. The graphical user interface was designed using Bootstrap and JavaScript. The application was user tested.

## Abstrakt

Cílem této práce bylo navrhnout a vytvořit výukovou aplikaci pro český znakový jazyk, která bude sloužit jako nástroj pro učení a procvičování slovní zásoby. Aplikace využívá webový framework Flask pro serverovou část, Jinja2 pro HTML šablony a knihovnu Socket.IO pro komunikaci mezi klientem a serverem v reálném čase. Grafické uživatelské rozhraní bylo navrženo s využitím Bootstrap a JavaScriptu. Aplikace byla uživatelsky otestována.

## Keywords

Hearing impaired, Education system for the hearing impaired, Czech sign language, Web application, Educational application, Python, HTML, CSS, JavaScript

## Klíčová slova

Sluchově postižení, Vzdělávací systém sluchově postižených, Český znakový jazyk, Webová aplikace, Výuková aplikace, Python, HTML, CSS, JavaScript

## Reference

HEŘMAN, Kristián. *Educational Application for Czech Sign Language*. Brno, 2024. Master's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Mgr. Kamil Malinka, Ph.D.

## Rozšířený abstrakt

Cílem této práce je prozkoumat existující projekty a aplikace pro výuku českého znakového jazyka a následně vytvořit novou aplikaci, která bude implementovat předem navrženou vhodnou metodu výuky českého znakového jazyka. Tato metoda by se měla zaměřit na kontrolu předvedení správnosti daného znaku. Největší motivací pro tvorbu této aplikace je nedostatek či spíše úplná absence takovýchto aplikací v České republice, tedy pro český znakový jazyk. To ve velké míře souvisí s počtem projektů, které se českým znakovým jazykem zabývají v kombinaci s informačními technologiemi.

Pro vytvoření aplikace je potřeba nejdříve nastudovat problematiku komunikačních systémů neslyšících a především znakových jazyků včetně jejich výuky v České republice. Zmíněných komunikačních systémů pro neslyšící je na našem území je několik, z nichž tím základním systémem je právě český znakový jazyk, uzákoněný v české legislativě. Obecně se znakové jazyky liší od mluvených jazyků formou předání informace, ale také gramatikou či slovní zásobou. Zatímco mluvené jazyky považujeme za systémy audio-orální, znakové jazyky řadíme mezi systémy vizuálně-motorické. Vzhledem k tomu, že český znakový jazyk má gramatiku i slovní zásobu odlišnou od českého mluveného jazyka, musí být výuková aplikace znakového jazyka schopna se s tímto problémem vyrovnat. Pokud by totiž fungovala na principu překladače, potřebovali bychom paralelní korpusy obou jazyků, což je u českého znakového jazyka nereálný požadavek, jelikož zkoumání tohoto jazyka, jeho gramatiky a slovní zásoby, je stále trvající proces.

Otázka, na kterou se snaží práce odpovědět, je tak forma výuky a implementační přístup aplikace k možnosti kontroly a reflexe uživateli, jak správně daný znak českého znakového jazyka předvádí a jestli by mu takto rozuměli rodilí neslyšící mluvčí, komunikující tímto jazykem jako svým přirozeným.

Zvoleným přístupem pro kontrolu předvádění správnosti byl zvolen koncept strojového učení. Pomocí tohoto principu je třeba vytvořit modely, které se na referenčních videích, přesněji snímcích z těchto videí, naučí rozpoznat jednotlivé znaky českého znakového jazyka. Tomuto modelu tedy můžeme předložit snímky získané například z webové kamery znakového uživatele, přičemž model vyhodnotí, jestli je znakované gesto předvedeno s určitou mírou chyby správně. K tomuto účelu bylo využito nástroje ASL-Recognition, který byl součástí bakalářské práce Tomáše Děda, studenta fakulty jaderné a fyzikálně inženýrské Českého vysokého učení technického v Praze. Tento nástroj byl vyvinut jak pro tvorbu modelu neuronové sítě z vložených oštitkovaných dat, ale současně také pro klasifikaci tímto modelem nad proudem snímků z webové kamery v reálném čase. Ačkoli byl nástroj původně implementován pro práci se znaky ASL, tedy americký znakový jazyk, který je na světě považován za jeden z nejrozšířenějších znakových jazyků, bylo možné tento nástroj přizpůsobit pro podmínky českého znakového jazyka. ASL-Recognition byl napsán v jazyce Python, rozhraní pro práci s obrazem zajišťuje knihovna OpenCV.

Tyto poznatky byly proto vzaty do úvahy při definici architektury a technologií, které bude celá výuková aplikace využívat. Kromě toho je nutné před samotným návrhem aplikace stanovit profil typického uživatele aplikace, jeho požadavky a schopnosti. Jelikož se jedná o výukovou aplikaci českého znakového jazyka, je třeba za uživatele považovat libovolného člověka se zájmem o výuku tohoto jazyka. Jelikož je tato skupina velmi široká, lze u uživatelé aplikace uvažovat pouze základní schopnosti práce na počítači nabyté z jejich každodenního kontaktu – email, práce v informačním systému v zaměstnání a podobně. Zásadním požadavkem na aplikaci je tak jednoduchost – a to jak na dostupnost aplikace, tak na její užívání.

Z těchto důvodů byla vybrána forma webové aplikace. Narozdíl od desktopové aplikace odpadá z pohledu uživatele u webové aplikace nutnost cokoli instalovat, což s sebou může přinášet problémy, které potenciálního zájemce odradí. Technologie vybrané pro implementaci backendu je jazyk Python společně s knihovnou Flask pro realizaci webového serveru a knihovnou Socket.IO, která umožňuje oboustrannou komunikaci mezi serverem a klienty v reálném čase. Jazyk Python je multiplatformní, tudíž spuštění takovéto serverové webové aplikace nezávisí na operačním systému a lze ji tak spustit jak na OS Linux, tak Windows. Ve spojení s virtuálním prostředím pro Python jde o jednoduchou kombinaci, pomocí které lze software snadno distribuovat a v případě zájmu si tak každý může spustit svůj vlastní výukový server. K realizaci frontend části byly využity webové technologie HTML s šablonovacím systémem Jinja2, CSS framework Bootstrap a JavaScript pro komunikaci z klienta na server v reálném čase. Po výběru technologií tak bylo možné přistoupit k samotné implementaci navržené aplikace.

Po dokončení vývoje aplikace bylo přistoupeno k testování aplikace s uživateli. Z tohoto testování vyplynuly nové poznatky, z nichž některé byly zapracovány a ostatní by mohlo být vhodné realizovat v případě pokračujícího vývoje této aplikace v budoucnu. Aplikaci by mělo smysl dále vyvíjet a především připravit obrázková data znaků českého znakového jazyka do jednotlivých lekcí.

# Educational Application for Czech Sign Language

## Declaration

I hereby declare that this Master's thesis was prepared as an original work by the author under the supervision of Mr. Mgr. Kamil Malinka, Ph.D. I have listed all the literary sources, publications and other sources, which were used during the preparation of this thesis.

.....  
Kristián Heřman  
May 17, 2024

## Acknowledgements

I thank my thesis advisor for his valuable advice during the consultation. I also thank my family for their support throughout my studies. I would also like to thank Mr. Bc. Tomáš Děd from the Czech Technical University in Prague for his valuable advices.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Hearing impaired people, sign languages</b>	<b>4</b>
2.1	People with hearing impairment . . . . .	4
2.2	Sign languages . . . . .	5
2.3	Oral, total and bilingual methods . . . . .	5
2.4	The beginnings of Czech Sign Language . . . . .	6
2.5	Development in the 19th and 20th centuries . . . . .	6
2.6	Laws and standardization . . . . .	6
2.7	Communication systems of deaf and deaf-blind people . . . . .	7
<b>3</b>	<b>Education system for the hearing impaired the Czech Republic</b>	<b>9</b>
3.1	Teaching at primary school . . . . .	9
3.2	Education of special teachers . . . . .	11
<b>4</b>	<b>Existing solutions and projects</b>	<b>12</b>
4.1	Helpful applicatons for mutual communication . . . . .	12
4.2	Educational applications of sign language . . . . .	15
<b>5</b>	<b>Proposal of teaching method of Sign language</b>	<b>17</b>
<b>6</b>	<b>Requirements analysis and application design</b>	<b>18</b>
6.1	Application Design . . . . .	18
6.2	Application architecture . . . . .	20
6.3	Sequence diagram of communication . . . . .	20
6.4	ER diagram of application . . . . .	21
6.5	UI design of application . . . . .	22
6.6	SW Architecture and technology selection . . . . .	22
<b>7</b>	<b>Application Implementation</b>	<b>25</b>
7.1	Application ASL-recognition . . . . .	25
7.2	Web server . . . . .	28
7.3	Application Interface . . . . .	32
7.4	Web User Interface . . . . .	33
7.5	Usage of the application . . . . .	35
<b>8</b>	<b>Testing and evaluation</b>	<b>37</b>
8.1	User testing . . . . .	37

<b>9 Conclusion</b>	<b>39</b>
<b>Bibliography</b>	<b>40</b>



# Chapter 1

## Introduction

According to the World Health Organization, there are approximately 430 million people worldwide who require special care (hearing rehabilitation) due to hearing impairment. However, by 2050 this number will already exceed 700 million, which is one in ten people. [3] Hearing impairment then complicates a person's life in many other ways, such as education, employment, but also everyday life. This is due to the communication barrier between hearing impaired (deaf) and hearing people, as deaf people cannot use the speech, but use their own sign language to communicate. This can lead to loneliness as a result and social isolation. [41]

Any work that helps to break down communication barriers between deaf and hearing people, is indispensable precisely to avoid the social exclusion of these individuals. On the contrary, it is necessary to provide them with the support that will enable them to live a full life without barriers and without discrimination in the hearing environment. The development of information technology, together with the spread of smartphones then enabled the development of online sign language translation dictionaries language and other specific applications that facilitate communication between the two parties. Despite all this, there is still a shortage of such apps and aids, as hearing impaired are a minority among us and are therefore often neglected by mainstream society.

The author of this thesis decided to develop an educational application for Czech Sign Language, that will enable both special education students and the general public to learn in a simple way the signs of Czech Sign Language. The author's motivation was to help the community the hearing impaired community, to delve into the issue of communication with the hearing impaired and to use information technology to develop an application that can be used in everyday life (e.g., specifically in teaching teachers focused on hearing aid education hearing impaired). The aim is to develop an open system which, if desired can be developed further according to needs.

This thesis is divided into several chapters. The following chapter 2 discusses general view of the hearing impaired and their sign languages, including differences from spoken languages. Chapter 3 discusses both the teaching of hearing impaired children in primary schools and the training of special educators who teach in these schools. Chapter 4 summarises existing projects and available applications for teaching Czech sign language. Chapter 5 then proposes a suitable method of teaching Czech Sign Language that could be suitable for both the hearing impaired and the general public. In chapter 6, the requirements of the application are analysed based on the proposed method. Furthermore, application design is created, which is then implemented in chapter 7. In chapter 8, the results from user testing are presented and the benefits for educational purposes are evaluated.

## Chapter 2

# Hearing impaired people, sign languages

### 2.1 People with hearing impairment

The hearing impaired can be divided into several groups according to the severity of their disability. Michaela Novotná in her thesis divides people with hearing impairment into four categories according to the following criteria [23]:

**Hard-of-hearing persons** Persons who are able to perceive spoken speech with the help of compensatory aids (hearing aids), from these reasons, most of them are seniors.

**Deaf persons** Persons who are *pre-lingually deaf* and *post-lingually deaf* depending on the age at which they lost their hearing – prelingual hearing loss may be congenital defect or deafness by the age of 5 to 6 years.

**Post-lingually deaf persons** Persons who are so-called *postlingually deaf*, i.e. partially or totally lost their hearing after speech development was complete.

**Persons with cochlear implants** Persons whose speech defect is so severe that it cannot be compensated by use of hearing aids. But if the auditory nerve has not been impaired, there could be implanted a hearing prosthesis, called in another words a *cochlear implant*.

In the following, we will focus on deaf people, especially prelingually deaf people who – unlike other groups of hearing impaired people – did not come through sufficient development of spoken language, and are therefore dependent on their own language – the sign language. At this point, however, it is necessary to mention the term **Deaf**, written with a capital initial letter. Josefína Kalousová defines it as follows:

This is a group of people who feel they are members of a linguistic and cultural minority. This is the reason why the term is capitalized. They are a linguistic minority because they use Czech sign language – a language that is different from Czech – in their everyday communication with each other. They are a cultural minority because they have a culture that is different from the Czech one. They

have different rules of behaviour, different customs, different values, their own specific art, their own history, different humour and much more. Belonging to the Deaf community does not depend on the severity of the hearing loss, it can include both deaf and hard-of-hearing people, but especially it is important to communicate in Czech sign language, to know the Deaf culture and to feel like a member of this community. [13]

## 2.2 Sign languages

Unlike hearing people who communicate in spoken languages (also called *audio-oral languages*), deaf people communicate in sign language, which is a visual-motor language produced in the three-dimensional space around the speaker. Other specific features of sign languages are: simultaneity and the absence of written form [24].

Although it might seem that there is one sign language for the deaf all over the world, this is not the case. As Czech sign language interpreter Daniela Vacková says, sign language has the same attributes as spoken languages – it has its own grammar, regional differences, and so on. For example, Czech sign language in Bohemia and Moravia has (just minimally) different vocabulary. However, there is a certain kind of similarity between sign languages. So there is a good chance that deaf people will be able to communicate with each other internationally, at least to some degree.

At international events such as EU meetings, United Nations (UN) meetings or the Deaflympics, the *International Sign* (IS) is sometimes used [39]. It puts together the signs of several sign languages together with other sufficiently clear signs or pantomimic movements.

However, this thesis focuses directly on Czech Sign Language as the natural language of the Deaf people coming from the Czech Republic. (see chapter 2.2) [24].

## 2.3 Oral, total and bilingual methods

At this point we should also mention the oral method, total communication and bilingual education, which are used for teaching the deaf.

**Oral method** In this method of teaching, communication is used mainly in spoken form. If we speak of the purely oral method, it refuses any manual aid systems in teaching speech. [35] Nevertheless, this is too demanding if specifically deaf children should be taught using this method. Thus, this method is likely to be used in teaching hard-of-hearing children.

**Total communication** Total communication can be defined as a philosophy of approach to people with hearing impairment rather than a simple teaching method [22]. The principle of this method is to use all of the available means and combine them. Among others, speech, sign language, natural signs and facial expressions, body language, finger alphabet, reading or writing are used. **Sign Languages** and **Signed Languages** are the most used languages in Czech educational system. Both concepts will be explained in detail in the chapters 2.7.2 and 2.7.3.

**Bilingual education** It is based on the use of sign and spoken language, but sign language is considered the primary means of communication [29] There is no simultaneous use of both languages and the mother tongue methods of language teaching are followed [22].

## 2.4 The beginnings of Czech Sign Language

According to Lenka Vokrouhliková, the origins of Czech Sign Language date back to the end of the 18th century, more precisely to 1786, when the *Institute for the deaf-mute* was founded in Prague. [25] Furthermore, Vokrouhlikova mentions Johan Mücke, who published the first dictionary of the Czech Sign Language in 1834. Due to the situation in the Czech country, it was written in German, but the signs described were part of the Czech Sign Language.

Even before the establishment of this institute, the deaf were creating their own signs to facilitate their communication with each other. Signs were shared between pupils and pupils, as well as between pupils and their teachers. [38]. In this situation, teachers were intermediaries who acquired a vocabulary of signs from many deaf pupils over time. These signs were then shown on to new deaf children. Mázerová writes in her book *History of the Deaf Community to 1938* that the teachers themselves created new signs and also gained more signs from their study stays – in the Czech Republic and also in abroad. [19]

## 2.5 Development in the 19th and 20th centuries

In the two following centuries, i.e. in the 19th and 20th centuries, Czech Sign Language was developing and its role in the education of the hearing impaired people began to be important. Despite this, we should mention the period of communist regime in Czechoslovakia, during which all development of languages for the deaf almost stopped. The regime of the time prioritized the oral method for all severities of hearing impairment (why this is inappropriate approach were discussed in the chapter 2.3). Moreover, there was an attempt to isolate the hearing impaired completely from mainstream society in the same time. [38]

## 2.6 Laws and standardization

After the establishment of the independent state of Czech Republic, the first draft of law was made in 1995. This law was approved in 1998 as Act 155/1998 (*Sign Speech Act*)<sup>1</sup>. In 2008, Act 384/2008 on *Communication Systems for Deaf and Deaf-Blind Persons*<sup>2</sup> was approved, modifying some parts of the original version. First of all, the terminology has been changed – the term sign speech (in Czech “znaková řeč”) has been replaced by the new term sign language (in Czech “znakový jazyk”). Vybíhalová Žlábková in her thesis describes the difference as follows: “*speech* is an instrument of communication, while *lanaguage* is term of its systemic means.” [38] We should add that Czech deaf people use several types of Czech sign languages – all of them are defined in the law and each will be described in the following chapters.

As far as the standardization of Czech Sign Language is concerned, it is included in the list of language codes of the international standard ISO 639, more precisely the ISO 639-3 version, which collects language codes of all natural languages – whether they are living, extinct, written or unwritten. [40] For example, the Czech language has the abbreviation *ces* in ISO 639-3; the Czech sign language is listed here under the abbreviation *cse*<sup>3</sup>.

<sup>1</sup><https://www.zakonyprolidi.cz/cs/1998-155>

<sup>2</sup><https://www.zakonyprolidi.cz/cs/2008-384>

<sup>3</sup>[https://iso639-3.sil.org/code\\_tables/639/data](https://iso639-3.sil.org/code_tables/639/data)

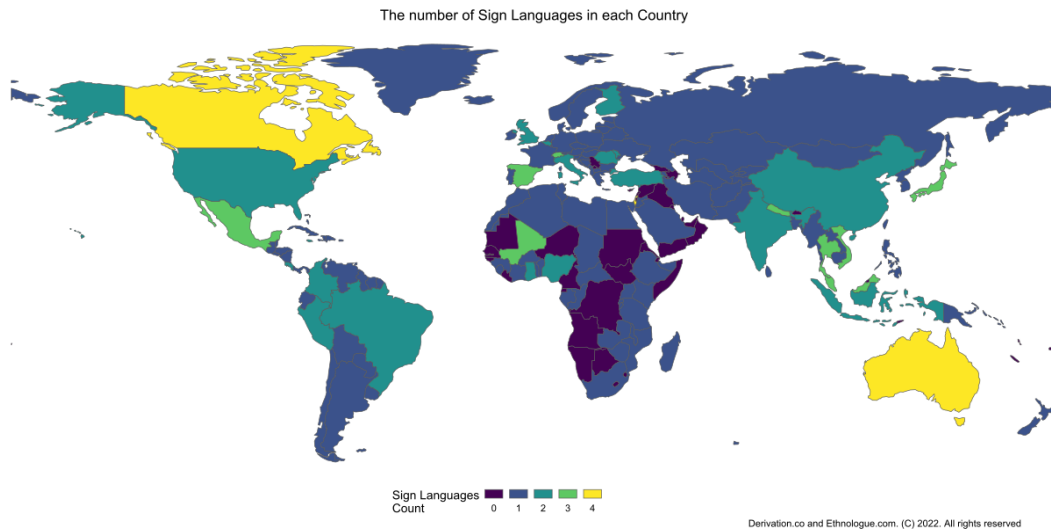


Figure 2.1: The number of living sign languages in the world. Source: [5]

## 2.7 Communication systems of deaf and deaf-blind people

Act 155/1998, and its amendment 384/2008, divides communication systems for deaf and deaf-blind persons into *Czech sign language* and *communication systems based on the Czech language*. We will mention here the best known and most used systems based on the Czech language, then we will focus on the Czech sign language, since it is the important theoretical part of this thesis.

### 2.7.1 Finger alphabet

The finger alphabet uses the formalized and established positions of the fingers and palm of one hand or the fingers and palms of both hands to represent the individual letters of the Czech alphabet. The finger alphabet is mainly used to spell out foreign words, technical terms, or other terms. [37]

Existují dvě základní varianty prstové abecedy. Podle využití jedné či obou rukou tedy tzv. jednoruční abeceda a obouruční abeceda. V České republice se v prostředí sluchově postižených používá převážně obouruční prstová abeceda. Jednoruční prstová abeceda se používá na východní Moravě a na Slovensku. [8]

There are two basic variants of the finger alphabet. Depending on the use of one or both hands, there is the *one-handed alphabet* and the *two-handed alphabet*. In the Czech Republic, the two-handed finger alphabet is largely used in the hearing impaired environment. The one-handed finger alphabet is used in regions of East Moravia and Slovakia. [8]

### Dvouruční prstová abeceda

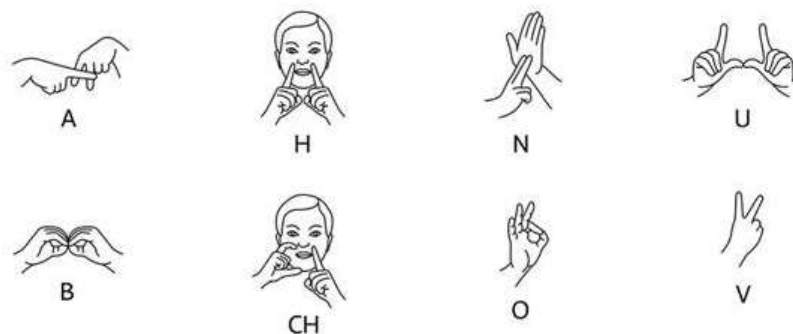


Figure 2.2: A sample of the signs of the finger alphabet. Source: [17]

### 2.7.2 Signed Czech

Signed Czech uses the grammatical sources of Czech. The sentence is simultaneously with the signing articulated loudly or voicelessly. Together with articulating Czech words, individual signs taken from vocabulary of Czech sign language are shown by movement and hand positions. [37]

The fact that signed Czech uses the means of regular spoken Czech makes it suitable for teaching and communicating with the post-lingually deaf or with hard-of-hearing people. On the contrary, it is not suitable for communication with the pre-lingually deaf, because the grammar of spoken Czech and Czech sign language is not interrelated and does not build on each other. The only justifiable exception for the use of Signed Czech in teaching the pre-lingually deaf is the teaching of Czech language. Deaf people are also taught regular Czech language and for reasons as such explanation of grammatical details and so on, the use of Signed Czech is needed.

In addition to these two systems based on Czech sign language, the law also mentions others – we could mention at least the *visualization of spoken Czech* (articulation when pronouncing words for the possibility of lip-reading), *Lorm's alphabet* (palm-touch alphabet) or the *Tadoma vibration method* (based on the perception of speech by feeling and touching the vocal cords, jaws, lips, etc.).

### 2.7.3 Czech Sign language

Just as the Czech language can be classified as a Slavic language, Czech Sign language belongs to the family of languages of the French sign language [6]. Therefore, we can speak of the similarity of Czech sign language directly with the French equivalent. Compared to that, despite the similarity of Czech and Slovak sign languages, they do not have much in common. From this reason, we can observe that although the grammars of the two languages are very similar and native speakers of spoken languages (e.g. Czech and Slovak) can understand each other without much difficulty, the grammars of the sign languages are strictly different from the grammars of the common (spoken) languages.

## Chapter 3

# Education system for the hearing impaired the Czech Republic

The educational system in the Czech Republic has been developing since the first institute in 1786, see the previous chapter. In the beginning, schools were divided according to the degree of hearing impairment into schools for the hard-of-hearing, for pupils with residual hearing and for the deaf. According to these criteria, individual pupils were then assigned to these schools [16].

As already mentioned, during communist regime in the Czech Republic, pupils with disabilities were suppressed and hidden. Hudáková adds that before 1991, all children with hearing impairment were educated monolingually and monoculturally. In other words, their mother tongue should be strictly spoken Czech and they had to identify with the values of hearing Czechs. Here we are getting into conflict with the previously discussed rights of the deaf to their own values, their own culture, etc., as it was said in the description of community of the Deaf with a capital letter. [11]

In 2005, the common school curriculums for the hearing impaired were canceled. Since this year, the school curriculum (in Czech “ŠVP – školní vzdělávací plán”), based on the Framework Education Plan (in Czech “RVP – rámcový vzdělávací plán”), became to be in responsibility of the headmaster of the school. [15]

Next, we will focus on the education of pupils in primary school. We do so primarily because elementary school is part of compulsory schooling that applies to all children in the Czech Republic. Let us add here the information that until 1948, according to Andrea Hudáková, there were many hearing-impaired children in our country who did not attend school at all. The reason was following: compulsory schooling was applied to all children – but except those who “could not be educated because of severe illness or physical defect”. [12]

### 3.1 Teaching at primary school

Šedivá writes, that the main goal in educating students with hearing impairments is to help them to achieve an education that matches their intellectual and motivational abilities. [28] The state should provide them sufficient opportunities to do so. Common education in the Czech Republic is described in School Act, i.e. Act 561/2004, or its amendment 178/2016. This also sets out the conditions for the education of pupils with special educational needs. Among other impairments, this amendment includes pupils with hearing impairment. In



addition to this law, there is also Decree 416/2017 on the education of pupils with special educational needs and talented pupils, which includes other regulations. This decree contains between others the years of compulsory schooling or the recommended number of pupils per class. [27]

According to Bartoňová and Vítková [1], it is possible to educate pupils in schools for pupils with hearing impairments as well as in mainstream schools, but only if the conditions and aids available to these pupils are appropriate to the degree and type of the individual's disability. One of the support measures may be for example the provision of a teaching assistant for the pupil himself.

At the beginning of this chapter we mentioned that the form of education at each school for the hearing impaired is individual, according to the school curriculum and ideas of each institution. These decisions include, for example, the method of teaching (oral method, total or bilingual), the length of compulsory schooling (which can be extended up to 10 years) or the use of various assistive and compensatory aids. [22] According to Pospíšilová [27], majority of primary schools in the Czech Republic use the possibility of extending the educational process to 10 years.

Let us focus briefly on a comparison of the number of pupils enrolled in programmes for pupils without special needs (i.e. students in mainstream schools) with the number of pupils enrolled in programmes for pupils with special educational needs (i.e. pupils in special schools).

Looking at the graph of the number of children in primary schools (see figure 3.2) of both types, we can see a change in the dominant type of school between years 2010 and 2013. From 2013, we can see that the majority of pupils attend schools for pupils without special educational needs – i.e. pupils integrated into mainstream education because of the school inclusion.

In contrast, we can look at the chart 3.1 of the same comparison, but now for kindergarten pupils. Here, although the trend (integration of pupils into mainstream school groups) is also visible, the latest data from academic year 2022/2023 shows that the majority of hearing impaired pupils are still educated in schools for children with special educational needs. [12]

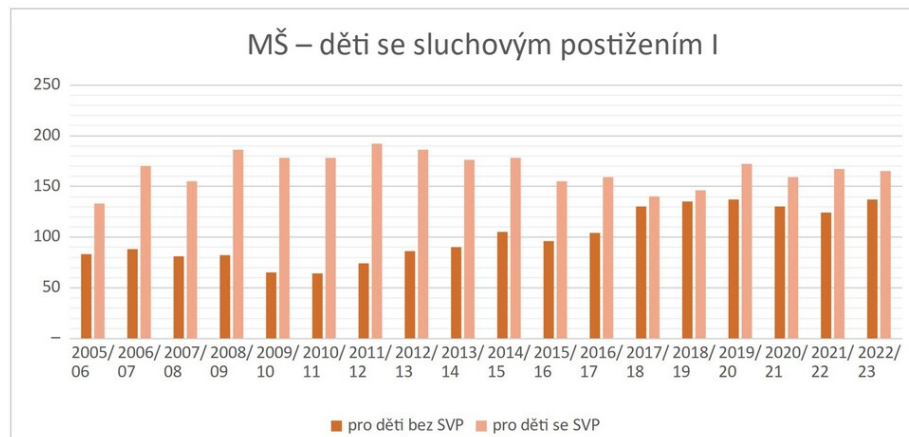


Figure 3.1: Number of children with hearing impairment in kindergarten with and without special educational needs. Source: [12]



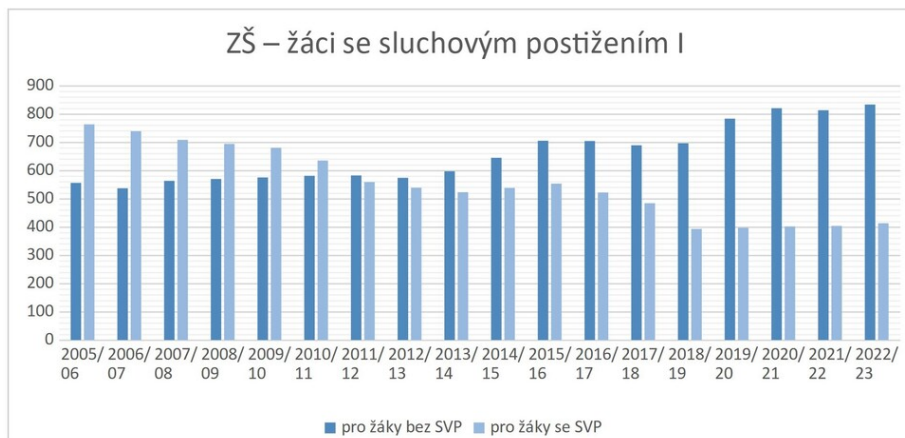


Figure 3.2: Number of children with hearing impairment in primary school with and without special educational needs. Source: [12]

It should be added that the integration of pupils with special educational needs into mainstream schools is big process in recent years because of the trend of school inclusion. It is for these reasons that the aim of this thesis is to design an application that will help people without any previous experience an effective way to overcome the communication barrier between hearing and hearing impaired. That is undoubtedly the problem of common language.

### 3.2 Education of special teachers

An equally important part of the education system is the education of teachers of the hearing impaired. This can be studied in the Czech Republic at faculties of education, in the field of special education, which except the hearing therapy (surdopedy) includes other disciplines for teaching pupils with special educational needs, i.e. speech therapy (logopedics), psychopediatry and others. During the high school education, these students take courses focusing on sign language. The problem is still the constant lack of support materials and aids.

## Chapter 4

# Existing solutions and projects

Because of the limited count of aids for the hearing impaired, the author of the thesis visited an associate professor at the Institute of Special Education Studies at Palacký University in Olomouc, doc. Jiří Langer, PhD., an expert in surdopedy, inclusive education and also the deputy director of the Support Centre of Students with Special Needs at the university. The discussion was held on the topic of viable computer aids that could help not only the hearing impaired persons in teaching and in everyday life, but possibly also students of special education or the broad society.

### 4.1 Helpful applicatons for mutual communication

#### 4.1.1 Sign language translators

At this point it is worth mentioning that the original idea of this master thesis was to create a translator from Czech language into Czech sign language. According to Associate Professor Langer, this is much more demanding and the results of the work are not guaranteed. From his words, researchers from the Department of Cybernetics at the University of West Bohemia in Pilsen started working on such a translator a few years ago. Since the best variant of the translator seems to be *statistical automatic translation*, it is necessary to build parallel corpuses for this method first. These are large datasets of phrases in both languages – i.e., both in Czech and Czech sign language.

In the 2013 study *Assistive Technology* by authors Lhotská, Štěpánková, Novák and Kužílek, researchers from the University of West Bohemia comment on the sign language translator as follows:

Translating Czech into Czech sign language is even more difficult than translating Czech into another, foreign spoken language. This is because in the case of translating Czech into another spoken language, it is a translation between languages that express semantic relations and grammatical organization in the same way in a sentence – i.e. by using words – and not by using visual-motor systems in the space around the speaker, as it is in case sign languages. [18]

#### 4.1.2 Czech Sign Language Dictionaries

Dictionaries of Czech sign language signs can also be included into the basic aids between the hearing and the hearing impaired people. Although this is not an aid that can be used by the nonspecialist public, for people with basic knowledge of interpreting into Czech

Sign Language, these dictionaries can greatly assist in explaining a message to a hearing impaired person correctly.

Among many Czech sign language dictionaries we can mention at least the multilingual dictionary **SpreadTheSign**<sup>1</sup>, the dictionary of the Teiresias Centre for Assistance to Students with Specific Needs at Masaryk University in Brno **Dictio.info**<sup>2</sup> or **Dictionary of Selected Technical Terms**<sup>3</sup>, provided by the Alfons Counselling Centre at Brno University of Technology. [36]

### 4.1.3 Tiché zprávy

*Tiché zprávy* (in English „Silent News“) is a website which brings news to the hearing impaired in their native language – Czech Sign Language. It works like a ordinary news website, but with small difference, that every article is also accompanied with a recorded video with translation into Czech sign language (signed by an interpreter).

The news are divided into categories – in addition to sections *Home*, *World*, *Sport*, *Interesting News*, *People*, there is also a *Sign guidepost* (in Czech “Znakovník”) section. There are translations of more complex Czech language terms that the hearing impaired might have trouble understanding. [33]

### 4.1.4 Záchranka application

To tell the truth, Záchranka app does not only target hearing-impaired people. Apart from many other apps that are unusable for these people, developers of this application thought about them and implemented support for the deaf in 2020. Users just have to set *the deaf checkbox* in settings. When the critical situation occurs, they will get user interface with table of icons from which they can select a specific injury instead of speaking with operator. [42] An additional option in this application is also a video call with the emergency operator. With help of this video, the deaf person can use signs to describe the realtime situation.

---

<sup>1</sup><https://www.spreadthesign.com/cs.cz/>

<sup>2</sup><https://www.dictio.info/>

<sup>3</sup><https://slovníkczj.vutbr.cz/>

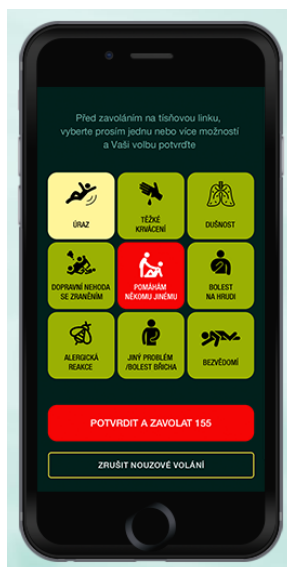


Figure 4.1: User interface for hearing-impaired people in Záchranka app. Source: [43]

#### 4.1.5 Online translator – Tichá linka

In addition to translators, which exist only with great limitations, we can also mention online translators from *Tichá linka* (in English „Silent Line“). With today’s online video calling, it is much easier to use an interpreter without having to be physically present with the client. This can be used e.g. at the doctor’s office or for personal purposes. For emergency calls, a 24-hour line is available. *Tichá linka* offers 4 services for hearing impaired:

- phone call
- personal call
- editing of text
- translation of text

Let us mention the procedures of at least the first two, which are the most important for the hearing impaired.

**Phone call** In this service, the hearing impaired person is in contact with the interpreter via video call, at the same time the interpreter calls the required telephone number and interprets the called person into Czech sign language on the video call to the client and interprets their answers back to the called person in Czech language.

**Personal call** In this case, the hearing impaired is in contact with the interpreter again via video call. In this case the hearing person is next to the hearing impaired person. The interpreter interprets the client from Czech sign language into Czech and vice versa.

These descriptions were adapted from official website of project *Tichá linka*<sup>4</sup>.

<sup>4</sup><https://www.tichalinka.cz/druhy-sluzeb-olt>

## 4.2 Educational applications of sign language

### 4.2.1 Znakujte s Tamtamem („Sign with Tamtam“)

Sign with Tamtam is an edition of mobile applications developed by the Czech Centre for Children’s Hearing Tamtam<sup>5</sup>. It is divided into three individual applications, each targeting a different age audience and each teaching sign language in its own specific way.

**Medvěd („Bear – Sign with Tamtam“)** This variant is suitable for smaller children. Not only that students will learn concepts and signs from basic areas of life while using the app. Children have to take care of a bear character, so in this way, it can keep their attention for a longer period of time.



Figure 4.2: User interface of application *Bear – Sign with Tamtam*. Source: [2]

**Sign with Tamtam** According to the description, this version of the mobile learning application is designed primarily for hearing parents and their hearing impaired children or other people interested in learning Czech sign language. [2] The app contains large vocabulary with corresponding sign videos, and. Moreover, it offers complete sentences with corresponding translations (videos). These words and sentences are divided into different levels of difficulty in the app.

**Sign with Tamtam – pairs** The app tries to teach children the basic signs in a playful way, using a children’s game called *pairs*(in Czech “pexeso”). The signs are always connected with a specific picture on the card. The game can be played by 1 or 2 players.

### 4.2.2 Tichý jazyk

*Tichá linka* and *Tiché zprávy* projects mentioned earlier in this paper also include the *Tichý jazyk* (in English „Tichý jazyk“) project. It offers Czech sign language courses to the general public. It also provides a mobile application of the same name, *Tichý jazyk*, as a support tool. This works as e-learning for practising signs and sentences in Czech sign language, when not in classes. The disadvantage of the app is that the user must be a participant in *Tichý jazyk* course to log in. [34]

---

<sup>5</sup><https://www.tamtam.cz/>

### 4.2.3 *Tableto* or *LMS Eliška*

During the research work on this chapter, the *Tableto* project and the *Eliška* E-learning system were also discovered.

The *Tableto* application was an interactive book for tablet installation in which children went through a story accompanied with pictures, videos in sign language and other animations. [14] The application worked until 2019.

The *Eliška* e-learning portal was part of a larger project funded by the European Social Fund between 2005 and 2007. It contained many interactive lessons, including lessons on Czech sign language also with dictionaries of signs in form: word—video sign. The *Eliška* portal was in operation for a short time after the project ended, but these days, it is no longer working.

In conclusion, according to this research, there is a real shortage of educational applications for Czech sign language and any other application that will add value over the basic Czech sign language dictionary will certainly be appreciated. Especially, if it would be used in school or so.

## Chapter 5

# Proposal of teaching method of Sign language

This chapter aims to suggest an appropriate method for teaching sign language, or rather for teaching the hearing impaired in general. Since the author of this thesis does not feel knowledgeable to suggest the proper teaching of sign language from a didactic point of view, he discussed this question with a teacher of a primary school for the hearing impaired in Olomouc. Based on this discussion, the author finds the following facts.

The fundamental finding is the need for individuality. Hearing impairments have different levels and at each level pupils need to be taught sign language in a different style. While sign language is a minority form of communication for post-lingually students or hard of hearing (as they are mostly able to communicate orally), for children who are deaf, sign language is the essence of their communication.

For deaf learners, sign language is also much more important with its grammar, as it is often the only component of their communication. They cannot use the approach of the post-lingually deaf and hard of hearing – i.e., that they express grammar verbally using Czech language and Czech sign language is used only to express the vocabulary.

It makes perfect sense to focus on vocabulary related to the topic of the particular lesson and the particular subject being taught. That means, for example, in a primary school lesson, we can first focus on the vocabulary (in sign language) of animals and vehicles that will be used in the rest of the class. In this case, the children will learn the vocabulary early. In the the rest of class, they will associate use of the particular sign with context, in which the word/sign was used.

Another part of an appropriate method should be interconnection of the signs with visual perception. This means that the interpretation of the signs should be accompanied by pictures. In this way, pupils can better associate the abstract form of the sign with the object of real world.

Let us mention one last component. This is the role of the native speaker in teaching sign language. However, only a native speaker can teach the grammar and vocabulary in a completely natural way, because sign language is their mother tongue. It is also probable that the children will be more psychologically prepared to this way of teaching. To tell the truth, this fact can be found in the learning of any foreign language – working with a native speaker can often be more effective than teaching with a regular teacher.

## Chapter 6

# Requirements analysis and application design

When analysing the requirements for the education application, it is necessary to start from the previous chapter 5, which discusses a suitable method for teaching sign language.

If we focus on the usable components of the method, it will mainly be about expanding vocabulary in a given theme by connecting a sign with a picture or word. The application should focus on practising and knowing these signs. Compared to the existing applications (see chapter 4), the benefit of the newly proposed application should be a programmatic check of the correctness of the sign presentation.

## 6.1 Application Design

### 6.1.1 Functional requirements

The application will provide the user dashboard with an overview of lessons (modules), organized in topics – e.g. animals, means of transport, food, zoo, etc. If the user starts a particular module, he will be given a word (alternatively a picture) to sign in Czech sign language. The application will then use the webcam to track the signer motions in real time. If the signer correctly interprets the sign and the application recognizes it, it will move on to the next word from the module. The user will be able to view a help in some form of a short recorded video of the tested sign to recall the correct sign movement.

A major limiting requirement is that, due to the expected lack of annotated data – i.e. videos of signs – the application will be limited to a certain number of topics that it will be able to offer for practice. Similarly, the number of signs in each lesson (or topic) will be limited to e.g. tens of signs. However, if there will be an interest in expanding the application in the future, it should be ready for this and it will depend only on the quantity and quality of the annotated data.



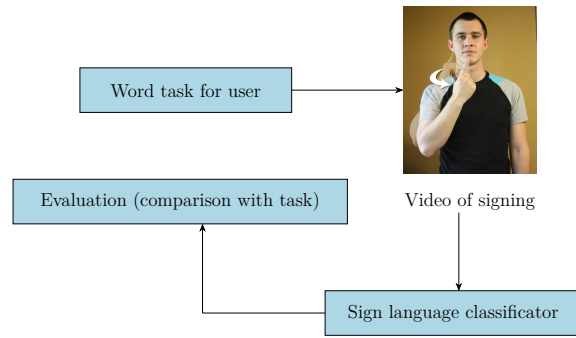


Figure 6.1: Basic principle of application functionality

### 6.1.2 Non-functional requirements

The system will be implemented as a web application usable on PC and mobile devices (that means it will be mobile responsive). There will be no need to download any application to the user’s device. Due to its architecture, the scalability of the application will depend only on the server solution used. In the case of low performance (e.g. large number of users in the same time), this can be solved by increasing the hardware components of the server easily.

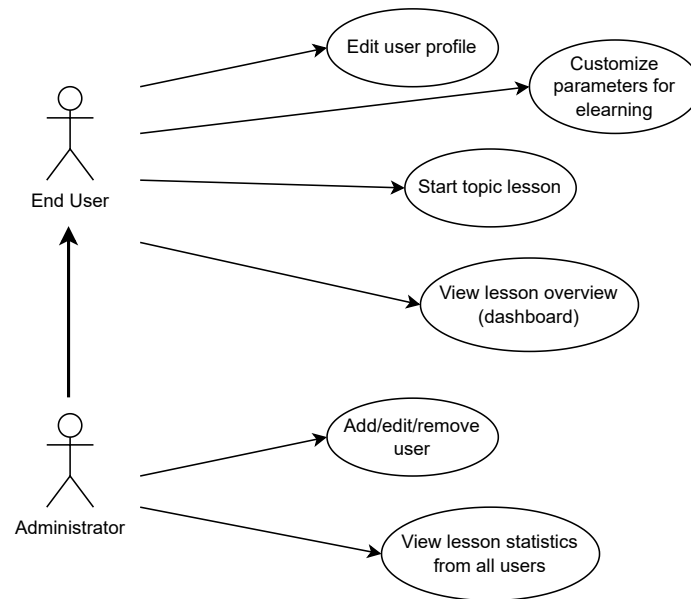


Figure 6.2: Use-case diagram of the application

For the purposes of the thesis, the server and client can be implemented on the same device – it will be a local web server. However, a virtual private server (VPS) with a public IP can also be used to run the application.

The application will include an authentication interface through which the user logs into their user account. Inside, they will see their progress in each of module.

## 6.2 Application architecture

At the highest level, the architecture can be called a classic server-client form. The clients here will be the individual users who run the web application on the server. The server will then run the main program, which will take care of displaying the web pages and working with the video camera footage. In real time, these images will be sent to the Czech Sign Language sign classifier, which will be implemented by a pre-trained neural network. This classifier will return the classification result back to the main program. In addition to this main task, an authentication service should also be provided on the server, which will prompt the user to log in to his account whenever he accesses the web application.

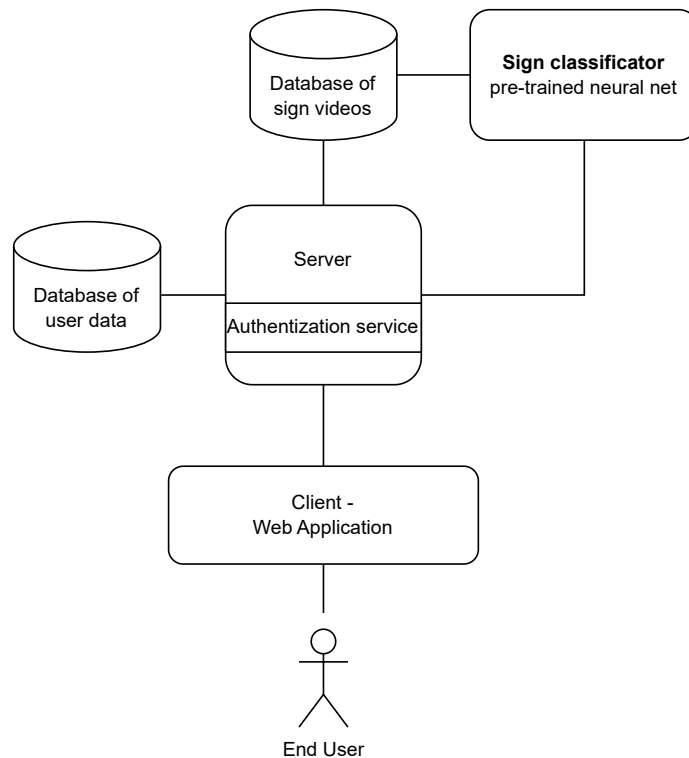


Figure 6.3: Application architecture design – high level of abstraction

## 6.3 Sequence diagram of communication

Based on the functional, non-functional requirements and high-level architecture design, a sequence diagram was constructed. It shows the most important processes that will have to be executed when running the core function – **online lesson with evaluation of correct signing**.

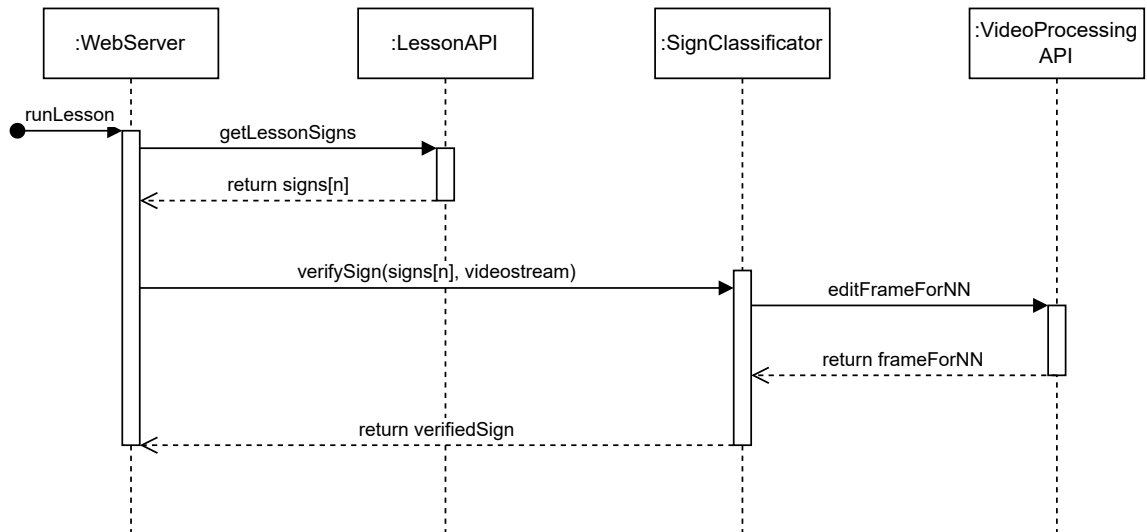


Figure 6.4: Communication Sequence Diagram – running signing lesson

## 6.4 ER diagram of application

Based on the application data storage requirements, such as storing users or storing the success rate of each trial in the test, an ER diagram was created for faster orientation and easy implementation in the next stage.

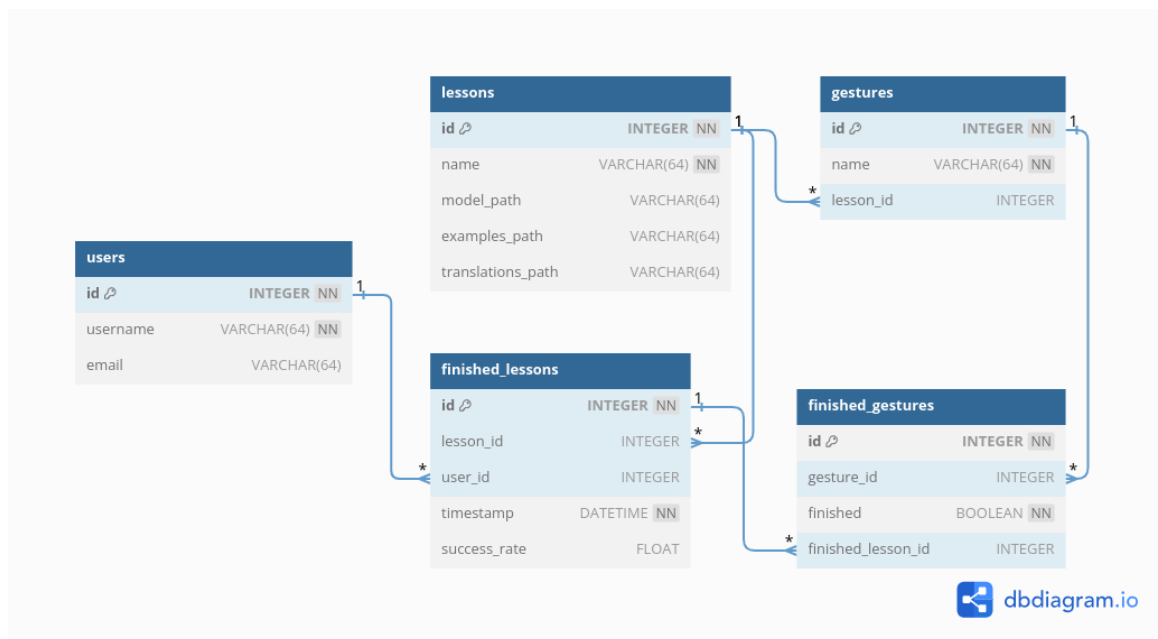


Figure 6.5: Entity relationship diagram. Diagram was made in DBdiagram.io [10]

## 6.5 UI design of application

### 6.5.1 Wireframes

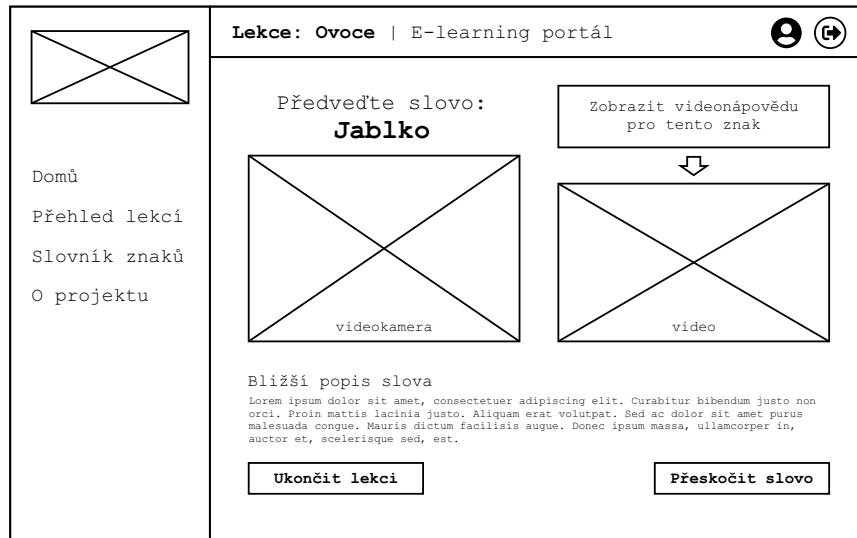
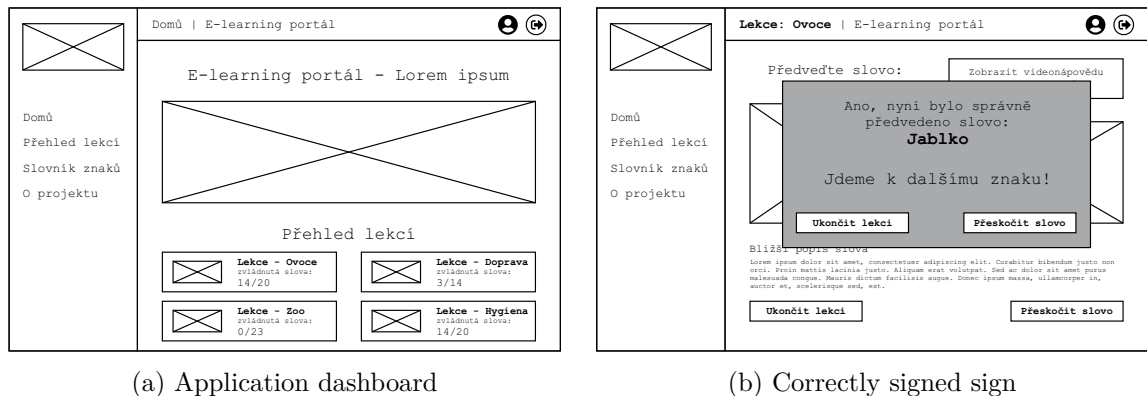


Figure 6.6: Wireframe – running signing lesson



(a) Application dashboard

(b) Correctly signed sign

Figure 6.7: Wireframe – dashboard and modal window when the sign is correctly signed

## 6.6 SW Architecture and technology selection

The application architecture can be divided into backend and frontend parts.

### 6.6.1 Backend architecture

To be able to run on a server, we need library support from the programming language to create a web server. In addition to the web server, we also need a library for real-time processing of video frames due to the requirements. Furthermore, we need an interface for calling the classifier of each sign of the sign language. In this application that will be a pre-trained neural network model on enough data.

**Web server framework** For a web server in Python it can be used one of the libraries like Flask or Django. Since the author has more experience from previous projects with the Flask library, he chose this framework for this application.

**Video Processing** The OpenCV<sup>1</sup> library will be used to manipulate the video. which is written in C++. The library can be used in C++ language, moreover it offers versions for Python, Java or Matlab/OCTAVE. Since the application will be implemented in Python, we can use the `opencv-python` library.

**Neural network – sign classifier** The classifier, that will decides if the presented sign in the video matches the given word, will be a pre-trained neural network model, and this model will most likely be built and trained with the available artificial intelligence and machine learning libraries Pytorch or Tensorflow. Because the application ASL-recognition<sup>2</sup> was found during the research of existing application, the author of this thesis contacted the author of ASL-Recognition to see if it was possible to continue working with the application. The author agreed without any problem, just to leave the code open-source. The work was implemented as a bachelor thesis at the CTU in Prague, moreover the source code of this application was open and publicly available from the GitHub repository.

**Application ASL-Recognition** This is an application for real-time recognition of ASL, American Sign Language. The recognition is based on pre-trained model on input data of sign language images. The tools used for this are mainly TensorFlow, Keras, OpenCV and Python. In addition to real-time sign recognition, the application also provides an interface to collect images of individual signs or possibility of training a model on this data [4]. Because of these facts, it is possible to use this application, upload example data and images of Czech sign language signs, train a new model and use the existing interface for real-time webcam prediction in our application.

As it was already mentioned, the Python programming language will be chosen for the backend of this application, because all of the important libraries, that are needed for the system functionality are available in Python. The application has to provide bidirectional communication and data transfer from and back to the web server, also real-time video processing and it will have to communicate with a separate module for the sign classifier. A separate module for authentication will also have to be prepared. Or this can be handled by some external service and the application would only work with some type of login token. There can be added another modules according to the needs of implemented application.

## 6.6.2 Frontend architecture

To implement the frontend, Flask provides us the Jinja2 templating system for HTML language. We will use this system to create all designs of end-points, or let us say individual web pages. To HTML markup language we will also add CSS framework Bootstrap, which will help us with a simple implementation of the graphical user interface. For realtime communication between the client and the server we will use JavaScript, more precisely the Socket.IO library, which will be run both on the client-side and on the server.

---

<sup>1</sup><https://opencv.org/>

<sup>2</sup><https://github.com/t-ded/ASL-Recognition>

**Jinja2** This templating system is used for its ability to create dynamic websites. Big advantage is that Jinja2 is a built-in templating system for Flask. From this reason it allows really good interface between these two systems. As an example we can mention passing variables from Python to templates, which is solved easily. Jinja2 makes it easy to create HTML templates that can use advantage of inserting variables, using control constructs like `if` or `for` to dynamically create pages, and so on. The system allows us to separate the logic from the appearance of the application, which significantly simplifies maintenance.

**CSS framework Bootstrap** We will use the CSS framework Bootstrap for styling and responsive design of web pages. Bootstrap provides many predefined styles – or rather, CSS classes – that make it quick and easy to create modern and user-friendly GUI. With Bootstrap and its principle of 12-columns webpage layout, we can easily control the appearance on different sized devices – from small mobile phones to 4K monitors. All of these without having to write a large number of CSS media queries, but also with a few predefined classes.

**Socket.IO** Socket.IO library will be used to make real-time communication between the client and the server. This library allows real-time two-way communication using interface of WebSockets [31]. The principle of this library is sending events [30] that are captured by the other side. This can be used both from client to server or in the other way direction. In our case, Socket.IO will be deployed both on the client side (using JavaScript), but also on the server side. Here it will be integrated with Flask. This is another advantage of the Flask framework, that it has an integration for SocketIO – `flask-socketio` Python library. With this library, we easily process images from a webcam stream or we can evaluate the correctness of the presented sign.

## Chapter 7

# Application Implementation

This chapter is devoted to the implementation of the whole web application according to the design described in chapter 6. As it was already mentioned, it was necessary to solve the configuration of the web server, its individual end-points (web pages), the sign classifier and the associated application interface for passing frames from the client and sending the evaluation back from the server to the client. In addition, a login module had to be implemented to ensure that the application would be able to store data on individual lesson attempts. Furthermore, it was necessary to solve and implement the appearance of the user interface based on wireframes prepared in chapter 6.5.1.

The first stage of the implementation was to explore and understand the application of **ASL-recognition**, discussed in the 6.6.1 chapter. The result of this part should be a guide for preparing the data for the subsequent training of the model. In addition, the trained model from this data should be ready to be used in our application in real-time prediction. There should be used interface of **ASL-recognition** connected to the newly implemented app.

### 7.1 Application ASL-recognition

The **ASL-recognition** application was used for data collection and training. It is publicly available from the repository at GitHub.com<sup>1</sup>, where the source code of the application can be downloaded. A more detailed description of all the components of the original application is given directly in the project README. Therefore, only the basic skeleton of the application which needed to build the model into our application will be described here.

For our need to train the model, we first need to define the Czech sign language signs in one lesson for which we will train our model. Since this is mainly demoversion of application, one lesson will contain units of signs. Each lesson will have its own pre-trained model for recognizing signs from that lesson only, minimizing overlap in the form of similarities in some of the sign language signs. The sign definitions are saved in the `gestures.txt` file. For the purposes of original **ASL-recognition** application, we also need to fill in the English translations of the signs in the file `translations.txt`, although in our case we will not use these translations. However, the used application checks the contents of these two files for consistency and quits the script if an error occurs.

---

<sup>1</sup><https://github.com/t-ded/ASL-Recognition>

1

```
Motorka, Auto, Jízdní kolo, Vlák, N-A, Autobus
```

Source code 1: Example content of file `gestures.txt`.

1  
2  
3  
4  
5  
6

```
Autobus      Bus
Jízdní kolo  Bicycle
Auto         Car
Motorka      Motorcycle
Vlák         Train
N-A         N-A
```

Source code 2: Example content of file `translations.txt`.

Note that the N-A sign occurs in both of mentioned text files. This is a routine taken from the original author of the application. This sign means a so-called empty sign (N-A – not available), i.e. a case where the signer does not show any sign or is not in the screen. Once we have the signs we want to have in our lesson – in other words the signs to train – we can move on.

Now it is necessary to prepare a virtual Python environment for running the application, which is ensuring compatibility of the versions of all libraries. We could create a new virtual environment in command line using command:

```
python -m venv /path/to/new/virtual/environment [32].
```

After that we activate it using command `. venv/bin/activate`. In the running virtual environment, we can use the pip packaging system for Python to install all the necessary libraries to run the application using `pip install -r requirements.txt`.

Important functions of the application can be accessed using the Python script `run.py`, which is located in the root directory of the project. The script provides five different run options. From these, we will use approx. the following three in our work:

1. **Image Collection** run by `python run.py -col` or `python run.py -collect` starts the image collection process to train the model.
2. **Environment showcase** run by `python run.py -show` or `python run.py -showcase` starts the environment familiarization mode.
3. **Training** can be started with `python run.py -tr` or `python run.py -train`, of course many other parameters can be set for training, by default the `Data` folder is used with prepared snapshots from the webcam.
4. **Prediction** run by `python run.py -pred` or `python run.py -predict` will use the pre-trained model for real-time prediction.

### 7.1.1 Image Collection

The script starts a window with the camera view and a hint window of the sign to be shown. These help images must be saved in advance in the `Examples` folder. The images are saved in the `Data` folder. In the `config.json` file, another important parameters, e.g. paths to



the `translations.txt` and `gestures.txt` files, the resolution of the saved images (default is 196px), and more can be specified. There is a rectangle in the camera view that indicates which part of the image is being saved, and shows the name of the currently collected sign. When a new sign is collected, the user is shown a larger sample of the sign, followed by a short warm-up period during which images are not saved. Once the rectangle and text colors change to green, the images begin to save.

At this point, the environment demonstration should be mentioned. As in the image collection case, the script launches the webcam preview window and the sign image help. Compared to the previous case, it does not save any images. This mode is the default when running the script without parameters so that the user cannot overwrite yet existing data.

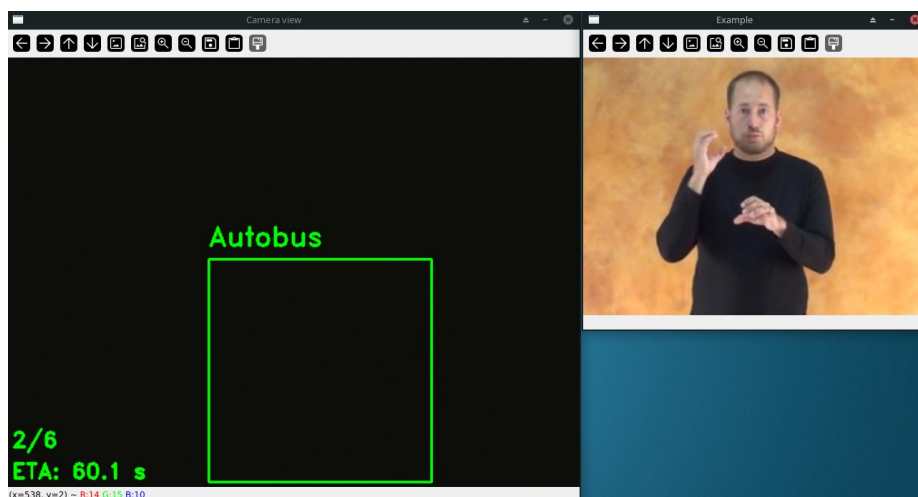


Figure 7.1: User interface of ASL-recognition app collecting data.

### 7.1.2 Model Training

The script starts training on the data folder. Many parameters are available in the documentation (README) of the project [4]. In our project, however, we run it with the parameters suggested in the documentation, and the training ran everytime for 50 epochs.

```

1 Model training:
2 Epoch 1/50
3 7/7 [=====] - 2s 203ms/stepposs: 1.4501 - accuracy: 0.4330 -
  ↳ recall: 0.1780 - precision: 0.7941 - f1_score: 0.3437 - auc: 0.7382
4 15/15 [=====] - 18s 1s/step - loss: 1.4501 - accuracy: 0.4330 -
  ↳ recall: 0.1780 - precision: 0.7941 - f1_score: 0.3437 - auc: 0.7382 - val_loss: 1.8611
  ↳ - val_accuracy: 0.2732 - val_recall: 0.1186 - val_precision: 0.2072 - val_f1_score:
  ↳ 0.1603 - val_auc: 0.7903 - lr: 0.0100
5 Epoch 2/50
6 9/15 [=====>.....] - ETA: 5s - loss: 1.2078 - accuracy: 0.5437 -
  ↳ recall: 0.3346 - precision: 0.7586 - f1_score: 0.4005 - auc: 0.8553

```

Source code 3: Example output during training procedure in command line.

### 7.1.3 Prediction

The most important part of the application is the real-time prediction script module. The original application launches a webcam window and processes the images in the green square, classifying them with a pre-trained model and returning the three most similar signs. It is already worth mentioning here that we had to redesign this module in some ways. Large changes were made to the application interface, because we had to adapt original module to our application needs.

Individual procedures and their individual settings can be specified using additional parameters, which are listed in the documentation, but will not be mentioned in this thesis.

## 7.2 Web server

As mentioned in the previous chapter on application design (6), the web server was implemented using the Flask library for Python. This section will describe how to configure routes and work with session variables.

### 7.2.1 Routes configuration

The configuration of routes in Flask is very clear and quite simple. The individual routes define which functions are executed when any URL is accessed. The following example source code shows the route configuration, which was created to handle requests to the URL `/lesson/<lesson_id>`. Note that the parameter after the slash – `lesson_id` – is parameter sent into the function in form of variable. Although there is communication with the database in the following code, this will be described more in detail in the following sections.

```
1 @app.route('/lesson/<lesson_id>')
2 def lesson(lesson_id):
3     if current_user.is_anonymous:
4         return redirect(url_for('index'))
5     ...
6     gestures = Gesture.query.filter_by(lesson_id=lesson_id).all()
7     session['sequence_list'] = np.random.permutation([gesture.name for gesture in
8     ↪ gestures]).tolist()
9     ...
10    session.modified = True
11    return render_template('lesson.html')
```

Source code 4: Example configuration of route in Flask web framework.

### 7.2.2 Session variables

The `session` variable in Flask allows you to store data between requests from the user. Compared to global variables, which are shared between all users, session variable stores data specifically for an individual user. This ensures that each user has its own context, so

the data will not interact with each other and the server can easily serve multiple clients at one time.

That is the reason, why the `session` variable was used to store information for each user. Here, in the example, it is `sequence_list`. That is a variable for a list of signs to practice, generated by random permutation of signs in the lesson. This variable is newly generated each time the lesson is run.

The use of session is most proper because the information is specific to the individual user at the time, and some of it also needs to be kept between requests.

The following section will discuss how to work with a database to store and manage the data needed in the application.

### 7.2.3 Database

This section will describe how the SQLite database was used. The communication between the application and the database was made using the SQLAlchemy library in Flask. SQLAlchemy<sup>2</sup> is a library that is ease to use to work with databases directly in Python code without having to write SQL queries. That is because of the object-related approach, which the library is using. SQLite was used because of its low-maintenance technology and also because of its suitability for smaller projects or for development purposes, which the application is.

Database models in Flask were defined as Python classes that inherited from class `db.Model`. Each class represented a single table in the database. The file with database scheme definition and with the data is stored in the `instance/db.sqlite` file. In the following example, there are defined some of the tables used in the application. The

```
1 db = SQLAlchemy(app)
2
3 class Lesson(db.Model):
4     __tablename__ = 'lessons'
5     id = db.Column(db.Integer, primary_key=True)
6     name = db.Column(db.String(64), nullable=False)
7     model_path = db.Column(db.String(64), nullable=True)
8     examples_path = db.Column(db.String(64), nullable=True)
9     translations_path = db.Column(db.String(64), nullable=True)
10
11 class Gesture(db.Model):
12     __tablename__ = 'gestures'
13     id = db.Column(db.Integer, primary_key=True)
14     name = db.Column(db.String(64), nullable=False)
15     lesson_id = db.Column(db.Integer, db.ForeignKey('lessons.id'))
16
```

Source code 5: Example of database table definition in SQLAlchemy library.

When initializing the database it was necessary to create all tables according to the example above, at the same time it was necessary to insert basic data.

---

<sup>2</sup><https://www.sqlalchemy.org/>

```

1 with app.app_context():
2     db.drop_all()
3     db.create_all()
4
5     db.session.add(Lesson(name="Veřejná doprava", model_path="model/verejna_doprava",
6     ↪ examples_path="Examples_verejna_doprava",
7     ↪ translations_path="translations_verejna_doprava.txt"))
8     db.session.commit()
9
10    lesson = Lesson.query.order_by(Lesson.id.desc()).first()
11    db.session.add(Gesture(name="Autobus", lesson_id=lesson.id))
12    db.session.add(Gesture(name="Jízdní kolo", lesson_id=lesson.id))
13    db.session.add(Gesture(name="Auto", lesson_id=lesson.id))
14    db.session.add(Gesture(name="Motorka", lesson_id=lesson.id))
15    db.session.add(Gesture(name="Vlak", lesson_id=lesson.id))
16    db.session.commit()

```

Source code 6: Initialization of database with basic data of lecture and its signs.

Two different approaches were used to modify and display the current data in the SQLite database. The first possibility was using `sqlite3` tool. The tool allows executing SQL queries directly from the terminal, for example entering the command in the root folder of the app: `sqlite3 instance/db.sqlite`.

The second option for easier database management was to use the SQLite Browser graphical user interface. This tool is an opensource application that allows user to visually create, design and edit SQLite compatible database files [26].

#### 7.2.4 Realtime server-client communication

To implement realtime communication in our application we used the Socket.IO library. For interactive applications that require immediate response, this part is an important element of the overall communication between client and server. During the implementation, we quickly stopped using only static end-points from the Flask library, since we needed to make the communication more dynamic. Also we needed to exchange information between client and server on a single web page, without having to refresh the page or send data from for example HTML `<form>` elements.

Socket.IO allowed us to set up events senders and handlers on both client and server side. Below you can see how we implemented a server-side function to skip a sign if the user does not know or if the user fails to show it on the webcam in a way that our pre-trained model would recognize it. This function is activated when the button is clicked from graphical user interface. Functions made with the Socket.IO library can be identified by the `@socketio` decorator.

In this function it was necessary to first update the structure with the solutions `gesture_solution_dict`. The value `False` indicates that the user skipped this sign. If it is the case that the current sign is the last one in the lesson, a signal is sent to redirect the user to the evaluation page. In other case, the variable `current_gesture` is updated to

```

1 @socketio.on('skip_gesture')
2 def skip_gesture():
3     session['gesture_solution_dict'][session['current_gesture']] = False
4
5     if session['current_gesture'] == session['sequence_list'][-1]:
6         emit('redirect', '/evaluation')
7         return
8     else:
9         session['current_gesture'] = session['sequence_list']
10        [session['sequence_list'].index(session['current_gesture']) + 1]
11        print("next gesture: ", session['current_gesture'])
12
13    emit('skip', session['current_gesture'])
14

```

Source code 7: Socket.IO endpoint which provides skipping of sign during lesson.

the next sign in the sequence and then the signal `skip` is sent back from the server to the client to continue in the lesson.

On the client side it was necessary to set up the event handler to catch the user action and send the signals to the server. The code below handles the click on the skip button and sends the signal `skip_gesture` to the server.

```

1 document.getElementById('skipGesture').addEventListener('click', function() {
2     socket.emit('skip_gesture');
3 });
4

```

Source code 8: Emitting signal from client application with SocketIO method.

With a very simple interface, we are able to handle many events that could occur on the client side in the case of running lesson, whichs need to be handled immediately on the server side with the result sent back. For example, premature ending of a lesson or skipping a gesture. Most of all the library simplified the exchange of images from the webcam on the client side with the server. And finally, it equally helped with sending the processed frame back from the server to the web page.

### 7.2.5 Authentication Module

Logging into user accounts was another important part of our sign language learning application. The main reason for this functionality was to enable storing of individual user progress in all lessons.

The user accounts were also stored in the SQLite database, which was discussed earlier in the section 7.2.3. The `users` database table was designed to contain only one piece of information, i.e. their email.

Login module was implemented using OAuth. In our application, the source code of Miguel Grinberg’s `flask-oauth-example`<sup>3</sup> was used [9]. OAuth allows users to log in using their existing accounts on known platforms such as Google, GitHub and so on, which really simplifies the login process from the user’s perspective. Moreover, it is eliminating the security risks that can be present if the login is implemented ourselves.

Unfortunately, the OAuth implementation had one major disadvantage: the functionality was limited to a application running at the URL `localhost`. API keys for OAuth authentication from providers like Google or GitHub must be linked to a specific URL, which in our case was only `localhost`. This is sufficient when running a server on our own device and running the app from a browser on the same device. Also, it is no problem to add a domain name to the API keys if the app would be deployed live and the authentication would work after that. This has been tested by running the app via the `ngrok` tool.

### Tunnelling into the public network

We tried to make the application available in the public Internet using `ngrok`<sup>4</sup> tool. This programme is used to tunnel a local server through a secure URL in the Internet. This method had the potential to solve the problem of testing OAuth authentication outside of the local environment. However, `ngrok` provides URLs that change after every 2 hours, which means that the URL would need to be periodically updated in the Google/Github OAuth API key settings. Due to this fact, this solution was abandoned.

However, the login problem limited the ability to test the application in real-world conditions outside of the development environment – for example, from a mobile phone connected to a web server running on another computer in local network.

### Alternative login for testing

In order to test the application easily, the option to log in as a test user was created. This login method does not use OAuth, but a direct login, which allowed to bypass the limitations of OAuth authentication. Implementing this feature ensured that it was possible to simulate real-world use of the application, while providing a way to verify that the user progress tracking was working correctly.

## 7.3 Application Interface

The interface between the web client, the server and the sign classification module was designed so that the functionality of module from `ASL-recognition` could be easily integrated into our educational application. The classification module was originally implemented using the `showcase_model()` method in a separate file. The function contained a `while` loop that periodically retrieved images from the webcam at the beginning of each loop using the `OpenCV` library and performed classification over them. For a server application, however, this approach would mean that the frames would be read from the webcam on the server, which was not proper solution.

For the purposes of the application, the images needed to be retrieved from the client webcam and then sent to the server for classification. From that reason, the `while` loop was removed, and the `showcase_model()` function was modified to receive the webcam image

---

<sup>3</sup><https://github.com/miguelgrinberg/flask-oauth-example>

<sup>4</sup><https://ngrok.com/>

directly on input, perform classification over this using a pre-trained neural network model, and return the result frame with result of prediction.

Getting image from the webcam was done in a web browser. Periodically every 100 milliseconds, a signal was emitted on the client using the `Socket.IO` library, after which the application encoded a webcam image and sent it as a signal parameter to the server for further processing by our application.

On the server side, a function was implemented that receives this frame, decodes it from the URL data into the `OpenCV` library format and then calls the mentioned `showcase_model()` function. As output of this function, the server expects the processed image with a decision if the sign was signed correctly. This was also added to the original code of `ASL-recognition`. The application originally just listed the three most likely signs it detected in the image. For our application, we added a condition that returns `True` if the probability of the correct sign is above 85 %. If so, the application allowed the user to continue to the next sign in the lesson.

## 7.4 Web User Interface

The graphical user interface of the application was created with an accent on simplicity, efficiency and user-friendliness. We used the `jinja2` templating system for HTML to implement the frontend. This combination allows easy manipulation of Flask variables and session variables directly in HTML templates. The individual template files for `jinja2` in HTML format are stored in the `templates` folder in the application root.

The application contains several templates, each of which is linked to a specific URL. Some templates are simple and use only basic `for` loops and conditions to list content from a table in the database – for example summary of completed lessons.

However, `lesson.html` is the most important of these templates as it implements all the logic of the currently running lesson on the client side. It includes a graphical user interface for the webcam on the left side of the screen and any video help on the right side. There are also user elements for ending/skipping the current sign. The template also contains code to get video from the browser webcam using the HTML5 API and JavaScript method `getUserMedia()`.

### 7.4.1 Camera Stream

The HTML5 API method `getUserMedia()` was used to get the video from the browser webcam. This method allows access to multimedia devices, such as a webcam, and provides a stream of data in video format. Using JavaScript, we called this method directly in the client browser. When the user starts the lesson, the browser displays a dialog box in which the user could confirm access to the webcam. Once the stream from the webcam is available, it needs to be periodically sent to the server in single frame format. To do this, the HTML element `<canvas>` was used, into which the current frame from the video was periodically copied. After that, the canvas with the image could be encoded into the data URL format, which is used to embed small files directly in the code [20]. In this application, it allows us the image to be simply transferred via `WebSocket` to the server in the form of data URL string.

However, we have hidden the `<video>` element, which is used to get the video stream from the webcam, and replaced it with the `<img>` element with the `src` attribute. We want to show the images that are sent back from the server with the detected signs, not just the



raw webcam stream. The images are also sent as URL data, so this variable value could be directly set to the value of the `src` attribute. Due to the periodic detection every 100 milliseconds, the image elements seems to be as smooth as the original video.

**SSL server security** Although SSL security is not part of the Web User Interface section, there was a problem when using the `getUserMedia()` method. The browser blocked access to the webcam when accessing the page via the unsecured HTTP protocol. A warning about this problem was subsequently found in the documentation of this method [21].

For security reasons, the `getUserMedia()` method is only available on pages accessed through the HTTPS protocol, which means that the application must be running on a secure server to use this method. At the start of developing the application, the Flask server was running on HTTP port no. 5000. So a solution had to be found that would allow the application to run on a secure protocol.

One tested approach was to use ad-hoc SSL security when running the Flask server using command `app.run(app, debug=True, host='0.0.0.0', ssl_context='adhoc')`. Unfortunately, this solution was unreliable and often caused a timeout when connecting to the server. Finally, solution was to first generate SSL certificates using utility `OpenSSL`<sup>5</sup> and then run the Flask server on port 443, which is used for HTTPS traffic.

The certificate and key are stored in the root folder of the application. The paths to them are then inserted in the code when the web server is started using `app.run(ssl_context=('cert.pem', 'key.pem'), port=443, host='0.0.0.0', debug=True)` Note that in this case the application must be run with superuser (`sudo`) privileges, because ports up to 1024 are privileged. The `host='0.0.0.0'` parameter means that the server will be visible in the whole local network and will be accessible from other devices via the local IP address of the device.

## 7.4.2 Screens of the app

This section will show the user interfaces of prepared web pages of the application. In addition to the appearance of the home page (7.2), which provides an overview of the available lessons to complete, there is also a page containing all the help videos from all the lessons (7.3). Users can watch these videos and learn the individual sign language signs if they want by watching the videos from the signers.

At this point, it is important to mention that the signing videos are obtained from the SpreadTheSign<sup>6</sup> web dictionary, which has already collected more than 400,000 sign language signs around the world, divided into many categories [7].

Another important screen is the overview of completed attempts within all lessons of the logged in user (7.4). Here users can track their progress for each lesson.

The last screen is the user interface of the current lesson (7.5). Here the user is given a complete overview of the current lesson status, including the option to view a hint, which can be used if the user does not remember the exact signing of the sign.

---

<sup>5</sup><https://www.openssl.org/>

<sup>6</sup><https://www.spreadthesign.com/>



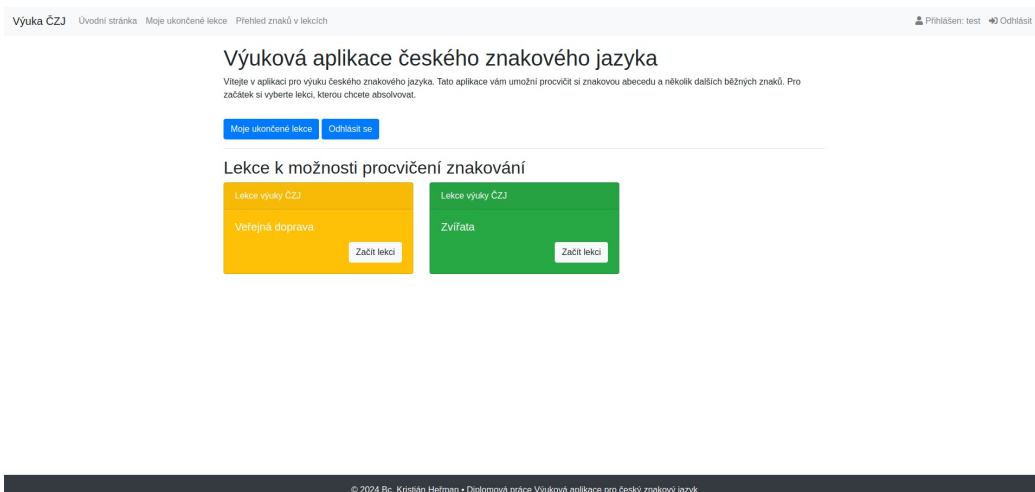


Figure 7.2: Screenshot of homepage of the app.

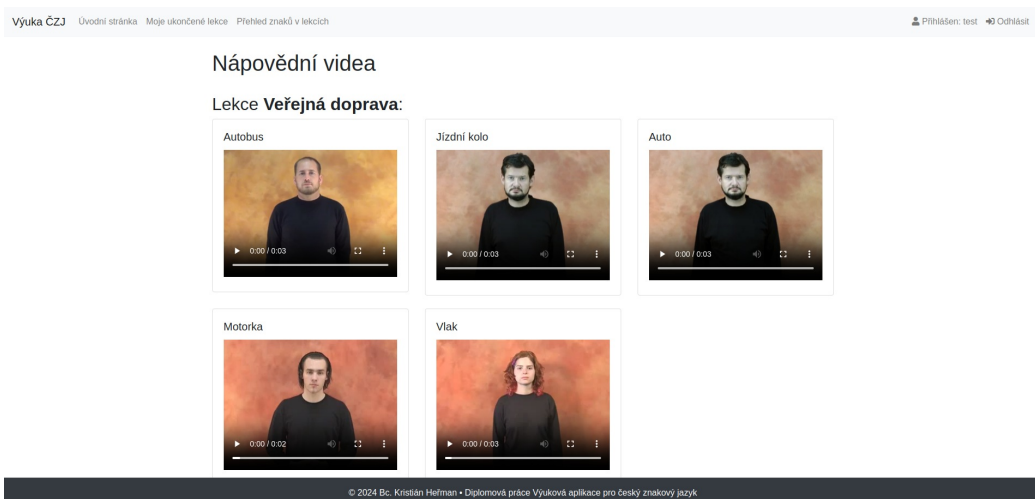


Figure 7.3: Screenshot of page with all help videos of signs.

## 7.5 Usage of the application

### 7.5.1 For developers

To run the server with the educational application, there is need to have a VPS (Virtual Private Server) or a computer with a virtual environment containing all the necessary Python libraries listed in the `requirements.txt` file. If such an environment is available, you can run the script in sudo mode. The `sudo python run.py -pred -guided` command can be used to start the web server with initialization needed. The server then starts listening on port 443. After the startup, the application can be opened at the URL that will be provided when the server starts – typically `localhost:443` when running on local machine.

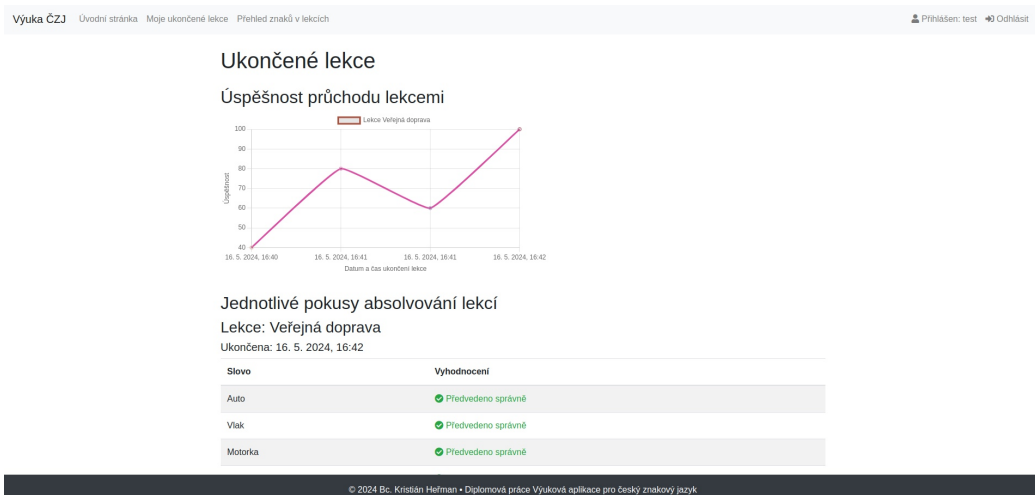


Figure 7.4: Screenshot of page with finished lessons overview.

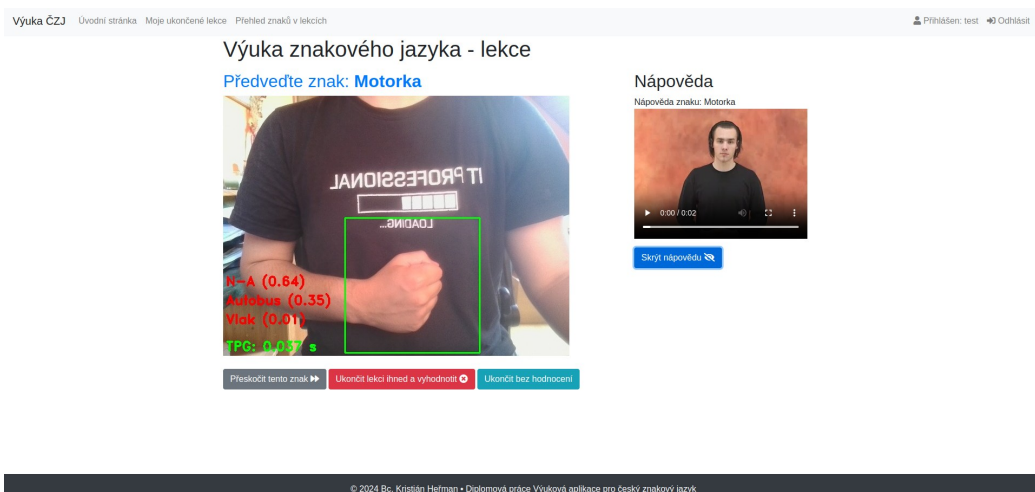


Figure 7.5: Screenshot of currently running lessons.

### 7.5.2 For users

If a web server with a learning application is already running, the user can simply open the web page with the application in the browser (for example, in the case of a local run, this would be `https://localhost/`). Here he has the option to register or log in (including a test login option). Then the user can start browsing the video database and learning the demonstration of each sign or he can start the lesson directly from the main page. The user is able to run the lesson repeatedly, and is not limited in number of attempts. Each attempt is saved in the database with a percentage success rate, unless the user quits the lesson earlier before the end without evaluation.

## Chapter 8

# Testing and evaluation

Part of the development of any application is its testing, which is important to ensure the quality and functionality of the application. Testing can reveal problems and faults that the developer may not encounter during the work, but which can be a major barrier for the users. Although the Czech sign language educational app is smaller application, the testing is important.

Even though we have limited the scope of the application to two lessons and five signs in each lesson, and the pre-trained models may not always classify the presented signs well due to the lack of data, the user-friendliness of the application is important for potential users. We need to make user testing to get feedback from actual users so that we can use this knowledge to fix specific problems and set further goals for the development of the application.

### 8.1 User testing

In the user testing, 5 people aged 28–65 years with normal technical abilities were approached, none of them had previous experience with Czech sign language. Before the test, the author of thesis described them the purpose of the application and the principle of its functionality. Subsequently, these users were given several tasks to attempt in the web application running on a local device. For each person, the author observed their actions without interfering with the users' process.

#### Overview of tasks in user testing

- Visit the website and log in to the application for the test user.
- Watch videos of the signs in the videos database and try to learn a few of them.
- Start the lesson of their choice and try to complete it.
- Take the same lesson a second time, but this time end it before the end.

All test participants completed their assigned tasks. None of the participants had problems navigating the application, perhaps because of its compactness and focus on one main function. The graphical elements were understood well and the user controls seemed intuitive to the users. Among the younger participants, it was mentioned that the app reminded them of the Duolingo app, but with a focus on sign demonstration instead of speaking into the application or entering the input from keyboard.

One bigger problem was found during testing. Some users had difficulty understanding the green and red colors of the detected signs in the webcam image. Also they have problems understanding, that the sign has to be presented into the dedicated green square in the image. This problem was solved by a simple tutorial that was then created and inserted as an additional web page in the application menu.

However, users said that the idea of an educational app for Czech sign language was very innovative and expressed excitement about the possibility that this app could offer a sufficient number of lessons and data to make it usable by anyone. For example, one of the users could imagine trying out Czech sign language lessons, just as she would like to try out, for example, the Spanish lessons in the Duolingo application, even though she has no major relationship to both languages. All in all, this application would certainly find its place among the public. When the author asked if they could imagine using such an application in education, especially in special schools, all participants answered positive.

So if we should evaluate the benefits of the application for educational purposes, the implemented application is a suitable example project that could be used both by the general public and for students at universities, who study special education. Part of their subjects are focused on Czech sign language, in which they could need to extend their vocabulary. According to one user, the advantages of the application are both the checking of the correct signing, which is lacking in self study, and also the app feature which shows probability of the correctness of the given sign. This feature gives the feeling that the user is closer to the success when signing the sign.

Overall, the app is still only for demonstration purposes. To achieve the overall goal of the application, it would need to be filled up with classification models for individual lessons. This means making a huge amount of labeled data that would have to be supplied to the application. The lack of such data is same around the entire world of artificial learning. However, the lack of data for Czech sign language signs is even more noticeable. The datasets basically do not exist, so everything has to be created from scratch.

## Chapter 9

# Conclusion

The aim of the thesis was to get oriented of the problems of teaching the hearing impaired. It was also necessary to learn about their educational system and the means of communication that hearing impaired people use to communicate with other hearing impaired people as well as with other hearing people.

After researching existing solutions that help or helped in communication and teaching, a suitable method was proposed to learn Czech sign language – the mother tongue of the deaf. Teaching grammar is very difficult and the best option is to learn directly with a native speaker. As far as vocabulary is concerned, it can be expanded by simpler methods, and for these reasons, an application for learning and practicing Czech Sign Language vocabulary was designed.

Then the application was implemented according to the prepared design, using the `ASL-recognition` project, which was the output of a bachelor thesis on a similar topic from the last year. This provided the a framework for sign language recognition that was adapted to the specific needs of the app. The application implements an educational system with the possibility to pass individual lessons and also it stores data from these attempts in order to monitor the user progress. The application was then tested on users, which confirmed its sufficient functionality and user-friendliness.

Such an application is unique and innovative in the Czech Republic, and although it works only on a demonstration sample of data, it would be advisable to continue development and fill the application with enough data to make it work as it was intended. The aim of the thesis was not to develop the whole application with all the needed data, but to prepare the skeleton that should be further developed.

The potential of this application is considerable, so it could become a valuable tool for teaching and practicing Czech sign language, not only in schools for the hearing impaired, but also among students of special education, or among the general public interested in learning the language.

# Bibliography

- [1] BARTOŇOVÁ, M. and VÍTKOVÁ, M. *Vzdělávání se zaměřením na inkluzivní didaktiku a vyučování žáků se speciálními vzdělávacími potřebami ve škole hlavního vzdělávacího proudu*. Brno: Masarykova Univerzita, 2013. ISBN 978-80-210-6678-6.
- [2] CENTRUM PRO DĚTSKÝ SLUCH TAMTAM. *Mobilní aplikace*. 2024. Available at: <https://www.tamtam.cz/co-delame/mobilni-aplikace/>.
- [3] DAVID, D., ALAMOUDI, A. H., ALBAHRI, O. S., ZAIDAN, B. B., ZAIDAN, A. A. et al. Landscape of sign language research based on smartphone apps: coherent literature analysis, motivations, open challenges, recommendations and future directions for app assessment. *Universal Access in the Information Society*. January 2023. DOI: 10.1007/s10209-022-00966-9. ISSN 1615-5289.
- [4] DĚD, T. Recognition of the American Sign Language in real time. 2023. Available at: <https://github.com/t-ded/ASL-Recognition>.
- [5] DERIVATION LLC. *International Day of Sign Languages (2022)*. 2022. Available at: <https://derivation.co/international-day-of-sign-languages-2022/>.
- [6] EBERHARD, D. M., SIMONS, G. F. and FENNIG, C. D. *Czech Sign Language*. 2023. Available at: <https://www.ethnologue.com/language/cse/>.
- [7] EUROPEAN SIGN LANGUAGE CENTER and LYDELL, T. *Slovník znakového jazyka / SpreadTheSign*. 2018. Available at: <https://www.spreadthesign.com/cs.cz/search/>.
- [8] FIKEJS, J. *Prstová abeceda*. 2005. Available at: <http://ruce.cz/clanky/3-prstova-abeceda>.
- [9] GRINBERG, M. *OAuth Authentication with Flask in 2023*. 2023. Available at: <https://blog.miguelgrinberg.com/post/oauth-authentication-with-flask-in-2023>.
- [10] HOLISTICS SOFTWARE. *dbdiagram.io - Database Relationship Diagrams Design Tool*. 2024. Available at: <https://dbdiagram.io/home>.
- [11] HUDÁKOVÁ, A. Žáci s vadami sluchu u státních maturit? *Speciální pedagogika*. 2004, vol. 14, no. 3, p. 177–198. ISSN 1211-2720.
- [12] HUDÁKOVÁ, A. *Kde se vzdělávají děti se sluchovým postižením?* September 2023. Available at: <https://www.idetskysluch.cz/integrace/vzdelavani/kde-se-vzdelavaji-deti-se-sluchovym-postizenim/>.
- [13] KALOUSOVÁ, J. *Sluchové postižení*. 2011. Available at: <https://spv.skauting.cz/jak-na-to/sluchove-postizeni/>.

- [14] KOUTSKÁ, M. *TABLETO, interaktivní kniha pro neslyšící děti*. April 2014. Available at: <https://www.helpnet.cz/aktualne/mariana-koutska-masarykova-univerzita-tableto-interaktivni-kniha-pro-neslysic-deti>.
- [15] LANGER, J. *Základy surdopedie*. Olomouc: Univerzita Palackého v Olomouci, 2013. ISBN 978-80-244-3702-6.
- [16] LANGER, J. and SOURALOVÁ, E. *Výchova a vzdělávání osob se sluchovým postižením*. Olomouc: Univerzita Palackého v Olomouci, 2013. ISBN 978-80-244-3701-9.
- [17] LENCOVÁ, R. *Putujeme za písmem*. Comenia Scriptth ed. Praha: Doležal Stanislav - Nakladatelství Svět, 2015. ISBN 9788087201077.
- [18] LHOTSKÁ, L., ŠTĚPÁNKOVÁ, O., NOVÁK, D. and KUŽÍLEK, J. *Asistivní technologie*. Praha: České vysoké učení technické v Praze, 2013. Available at: <https://docplayer.cz/7959394-Asistivni-technologie.html>.
- [19] MÁZEROVÁ, R. *Nástin historie komunity neslyšících do roku 1938*. Budapest: Open Society Institute, march 1999. Available at: [https://is.muni.cz/el/ped/podzim2017/SP2BK\\_SP2c/um/historie\\_neslysicich\\_Mazerova.pdf](https://is.muni.cz/el/ped/podzim2017/SP2BK_SP2c/um/historie_neslysicich_Mazerova.pdf).
- [20] MOZILLA FOUNDATION. *Data URLs - HTTP | MDN*. 2023. Available at: [https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics\\_of\\_HTTP/Data\\_URLs](https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Data_URLs).
- [21] MOZILLA FOUNDATION. *MediaDevices: getUserMedia() method - WebAPIs | MDN*. 2023. Available at: <https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia>.
- [22] NEŠPOROVÁ, T. *Změny v základním vzdělávání žáků se sluchovým postižením v kontextu novely školského zákona*. Hradec Králové, 2018. Dissertation. Univerzita Hradec Králové.
- [23] NOVOTNÁ, M. *Problematika profesního uplatnění osob se sluchovým postižením*. Olomouc, 2014. Dissertation. Univerzita Palackého. Available at: [https://theses.cz/id/84qqmh/Novotn\\_Problematika\\_profesneho\\_uplatnn\\_osob\\_se\\_sluchovm\\_po.pdf](https://theses.cz/id/84qqmh/Novotn_Problematika_profesneho_uplatnn_osob_se_sluchovm_po.pdf).
- [24] OKROUHLÍKOVÁ, L. *Notace znakových jazyků*. Praha: Karolinum, 2016. ISBN 978-80-246-2958-2.
- [25] OKROUHLÍKOVÁ, L. Historical Roots of Czech Sign Language the first half of the 19th century. *E-Pedagogium*. august 2021, vol. 21, no. 2, p. 46–65. DOI: 10.5507/epd.2021.018. ISSN 12137758.
- [26] PIACENTINI, M. *sqlitebrowser/sqlitebrowser: Official home of the DB Browser for SQLite*. 2014. Available at: [sqlitebrowser/sqlitebrowser](https://sqlitebrowser.org/).
- [27] POSPÍŠILOVÁ, I. *Vzdělávání dětí se sluchovým postižením*. 2023. Available at: <https://sancedetem.cz/vzdelavani-deti-se-sluchovym-postizenim>.
- [28] ŠEDIVÁ, Z. *Psychologie sluchově postižených ve školní praxi*. Praha: Septima, 2006. ISBN 80-7216-232-2.

- [29] SKÁKALOVÁ, T. *Dítě se sluchovým postižením*. 1st ed. Hradec Králové: Gaudeamus, 2014. ISBN 978-80-7435-502-8.
- [30] SOCKET.IO. *Emitting events | Socket.IO*. 2024. Available at: <https://socket.io/docs/v4/tutorial/step-4>.
- [31] SOCKET.IO. *Introduction | Socket.IO*. 2024. Available at: <https://socket.io/docs/v4/>.
- [32] THE PYTHON SOFTWARE FOUNDATION. *venv — Creation of virtual environments*. 2024. Available at: <https://docs.python.org/3/library/venv.html>.
- [33] TICHÝ SVĚT - CHRÁNĚNÁ PRACOVNÍŠTĚ. *O tichých zprávách*. 2021. Available at: <https://www.tichezpravy.cz/o-nas/>.
- [34] TICHÝ SVĚT - CHRÁNĚNÁ PRACOVNÍŠTĚ. *Přehledná aplikace pro učení českého znakového jazyka*. January 2022. Available at: <https://www.tichezpravy.cz/prehledna-aplikace-pro-uceni-ceskeho-znakoveho-jazyka/>.
- [35] TRMALOVÁ, H. *Komunikace sluchově postižených v rámci vzdělávacího procesu*. České Budějovice, 2009. Dissertation. Jihočeská univerzita v Českých Budějovicích. Available at: [https://theses.cz/id/zhn1z/downloadPraceContent\\_adipIdno\\_14536](https://theses.cz/id/zhn1z/downloadPraceContent_adipIdno_14536).
- [36] ČESKÁ KOMORA TLUMOČNÍKŮ ZNAKOVÉHO JAZYKA. *Slovníky*. 2024. Available at: <https://www.cktzj.com/co-vas-zajima/slovniky/>.
- [37] ČESKO. *Zákon č. 155/1998, o komunikačních systémech neslyšících a hluchoslepých osob*. 1998. Available at: <https://www.zakonyprolidi.cz/cs/1998-155>.
- [38] VYBÍHALOVÁ ŽLÉBKOVÁ, R. *Vývoj a šíření českého znakového jazyka*. Olomouc, 2012. Dissertation. Univerzita Palackého.
- [39] WIKIPEDIA.ORG. *International Sign*. 2023. Available at: [https://en.wikipedia.org/wiki/International\\_Sign](https://en.wikipedia.org/wiki/International_Sign).
- [40] WIKIPEDIA.ORG. *ISO 639-3*. 2024. Available at: [https://en.wikipedia.org/wiki/ISO\\_639-3](https://en.wikipedia.org/wiki/ISO_639-3).
- [41] WORLD HEALTH ORGANIZATION. *Deafness and hearing loss*. February 2023. Available at: <https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss>.
- [42] ZÁCHRANKA. *Neslyšící a aplikace záchranka*. September 2020. Available at: <https://www.zachrankaapp.cz/cs/neslysici-a-aplikace-zachranka>.
- [43] ZÁCHRANKA. *Novinky v aplikaci: podpora pro neslyšící*. 2020. Available at: <https://www.zachrankaapp.cz/cs/novinky-v-aplikaci-podpora-pro-neslysici>.



# Contents of the attached media

List of files and folder at the top level on the included SD card:

- `README.txt` – description of application installation and usage
- `text` folder – text of the thesis in *pdf* and its source files in *tex* format,
- `src` folder – application source code and scripts,

Detailed directory tree of the included SD card:

```
├── README
├── text
├── src
│   ├── Data
│   ├── Data_verejna_doprava
│   ├── Data_zvirata
│   ├── Examples
│   ├── Examples_verejna_doprava
│   ├── Examples_zvirata
│   ├── flask_session
│   ├── instance
│   │   └── db.sqlite
│   ├── model
│   │   ├── verejna_doprava
│   │   └── zvirata
│   ├── templates
│   ├── venv
│   └── Videos
```