



**MATEMATICKO-FYZIKÁLNÍ  
FAKULTA**  
Univerzita Karlova

**DIPLOMOVÁ PRÁCE**

Bc. Patrik Černý

**Kontrola výslovnosti v logopedické  
aplikaci**

Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: Mgr. Nino Peterek, Ph.D.

Studijní program: Informatika

Studijní obor: Softwarové systémy

Praha 2017

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne .....

Podpis autora

Děkuji Mgr. Ninu Peterkovi, Ph.D. za vedení diplomové práce. Velký dík také patří klinické logopedce Mgr. Kateřině Černé za její odbornou pomoc.

Název práce: Kontrola výslovnosti v logopedické aplikaci

Autor: Bc. Patrik Černý

Ústav: Ústav formální a aplikované lingvistiky

Vedoucí diplomové práce: Mgr. Nino Peterek, Ph.D., Ústav formální a aplikované lingvistiky

Abstrakt: Cílem této práce je navrhnout, realizovat a otestovat způsob validace výslovnosti na základě současných algoritmů pro rozpoznávání mluvené řeči. Výsledný software je logopedická aplikace umožňující trénování hlásek resp. slov se zpětnou vazbou, zdali bylo dané slovo vysloveno správně. Validace výslovnosti je postavena na nástrojích CMUSphinx a generování špatných výslovností (v podobě fonetických slovníků). Pro potřeby trénování a testování jsou nasbírány nahrávky správné a špatné výslovnosti. Ukázalo se že navržený přístup není vhodný. Výsledná aplikace se dá díky jejímu návrhu jednoduše rozšířit o techniky, které by mohly zvýšit efektivnost validace výslovnosti.

Klíčová slova: validace výslovnosti, rozpoznávání slov, dyslálie, logopedická aplikace

Title: Pronunciation Validation in Speech Therapy Application

Author: Bc. Patrik Černý

Institute: Institute of Formal and Applied Linguistics

Supervisor: Mgr. Nino Peterek, Ph.D., Institute of Formal and Applied Linguistics

Abstract: A goal of this thesis is to design, create and test speech validation method based on current speech recognition algorithms. Resulting software is a speech therapy application for sounds or words training with feedback about pronunciation accuracy. Speech validation is based on CMUSphinx tools and on inaccurate pronunciation generation (using phonetic dictionary). Records with accurate and inaccurate pronunciations has been collected for training and testing purposes. It has been shown, that this design is not appropriate. Thanks to the software design, application can be easily extended by techniques, that could improve validation efficiency.

Keywords: speech validation, word recognition, dyslalia, speech therapy application

# Obsah

<b>Úvod</b>	<b>3</b>
<b>1 Analýza</b>	<b>4</b>
1.1 Logopedie . . . . .	4
1.1.1 Posloupnost cvičného textu . . . . .	5
1.1.2 Věk pacienta . . . . .	5
1.2 Validace výslovnosti . . . . .	5
1.2.1 Klasifikátory řeči . . . . .	5
1.2.2 Vytvoření českého akustického modelu . . . . .	10
1.2.3 Generování špatné výslovnosti . . . . .	10
1.2.4 Pravděpodobnost špatné výslovnosti . . . . .	12
1.3 Existující řešení . . . . .	14
1.3.1 Logopedické aplikace . . . . .	14
1.3.2 Aplikace pro výuku cizích jazyků . . . . .	15
1.3.3 Ostatní aplikace zabývající se výslovností . . . . .	15
1.3.4 Knihovny pro validaci výslovnosti . . . . .	15
1.4 Zhodnocení a porovnání existujících řešení . . . . .	16
1.5 Specifika výsledné práce . . . . .	16
<b>2 Návrh řešení</b>	<b>19</b>
2.1 Architektura Systému . . . . .	19
2.1.1 Speech Validator Admin . . . . .	19
2.1.2 Speech Validator . . . . .	19
2.1.3 Speech Validator Server . . . . .	20
<b>3 Uživatelská a administrátorská dokumentace</b>	<b>35</b>
3.1 Speech Validator Admin . . . . .	35
3.1.1 Instalace aplikace . . . . .	35
3.1.2 Ovládání aplikace . . . . .	35
3.2 Speech Validator . . . . .	36
3.2.1 Instalace aplikace . . . . .	37
3.2.2 Ovládání aplikace . . . . .	37
3.3 Speech Validator Server . . . . .	37
3.3.1 Instalace aplikace . . . . .	38
<b>4 Programátorská dokumentace</b>	<b>44</b>
4.1 Speech Validator Admin a Speech Validator . . . . .	44
4.2 Speech Validator Server . . . . .	45
<b>5 Analýza použitelnosti Systému</b>	<b>47</b>
5.1 Testy pro výběr vah JSGF . . . . .	48
5.2 Testy pro výběr parametru LW . . . . .	52
5.3 Testy modelů . . . . .	52
5.4 Shrnutí výsledků . . . . .	54
5.5 Možnosti rozšíření . . . . .	54

Závěr	56
Seznam použité literatury	57
Seznam obrázků	60
Seznam tabulek	61
Přílohy	62

# Úvod

Ze statistických výkazů o činnosti pracovišť logopedie (a z jejich zpracování) uvedených na stránkách Ústavu zdravotnických informací a statistiky ČR (dále jen **ÚZIS ČR**) ÚZIS ČR (2010-2017) resp. v ÚZIS ČR (2014), ÚZIS ČR (2013), ÚZIS ČR (2012) je patrné, že v posledních letech je nejčastější poruchou komunikačních schopností u osob do 18 let dyslálie – patlavost, nepřesná výslovnost (procentuální zastoupení léčení dyslálie viz tab. A1).

K obdobnému závěru dochází i mnozí autoři zabývající se dyslálií – např. Anna Salomonová (Salomonová, 2003, kapitola 16.2) informuje o vysokém výskytu dyslálie během druhé poloviny dvacátého století.

Maria Gúthová a Daniela Šebianová ve své publikaci o terapii dyslálie (Gúthová a Šebianová, 2005, kapitola 5) zdůrazňují důležitost začít léčit dyslálii u dětí již v počátku, v předškolním věku. Své tvrzení doplňují negativními následky, které dyslálie má na výuku v základních školách, což by se dalo shrnout následovně. Děti s nedostatkem ve výslovnosti se častěji dopouštějí chyb při psaní a čtení.

Kompetentní osoby k léčení dyslálie jsou logopedi. Ti během logopedické intervence zadávají pacientům samostatná cvičení. Pacient během těchto cvičení často potřebuje zpětnou vazbu (tu v případě dítěte obvykle poskytuje jeho rodič), zdali cvičení provádí správně (např. hodnotí jeho správnou výslovnost hlásek). Logopedická intervence a samostatná cvičení jsou blíže popsány v kap. 1.1.

Všechna tato fakta a tvrzení poukazují na reálnou potřebu řešit dyslálii u osob do 18 let a to v co nejčasnějším stádiu a na potřebu zpětné vazby při samostatných cvičeních.

Tato práce má následující 3 cíle.

- Navrhnout způsob validace výslovnosti na základě současných algoritmů pro rozpoznávání mluvené řeči.
- Realizovat navržený způsob validace (dále jen **validátor výslovnosti**). Validátor výslovnosti bude součástí aplikace určené pro usnadnění zmíněné logopedické intervence (dále jen **System**). System je určen jako podpůrný prostředek pro předávání potřebných materiálů a zefektivnění samostatného cvičení pomocí zpětné vazby při trénování výslovnosti. Ze zpětné vazby by měl daný jedinec (dále jen **Uživatel**) poznat, zdali trénované slovo vyslovil správně.
- Vyhodnotit použitelnost validátoru výslovnosti.

První kapitola této práce je zaměřena na popis logopedické intervence a na možné přístupy k dosažení stanovených cílů. Jsou zde také popsána existující řešení a v závěru jsou specifikované požadavky na System včetně způsobu validace výslovnosti. Další část textu popisuje architekturu Systemu, podrobněji obzvláště validátor výslovnosti. Ve třetí a čtvrté kapitole se nachází uživatelská dokumentace a zajímavé části z implementace. Na závěr jsou uvedeny dosažené výsledky validátoru výslovnosti a návrhy možných rozšíření.

# 1. Analýza

Účelem této práce je poskytnout zpětnou vazbu pacientům s dyslálií během logopedické intervence resp. během jejich samostatných cvičení.

V rámci této kapitoly je nejdříve popsána logopedická intervence se závěrem, pro jaké její části a za jakých podmínek bude Systém sloužit jako vhodná pomůcka. Dále jsou rozebrány možnosti validace výslovnosti na základě současných algoritmů pro rozpoznávání mluvené řeči se závěrem, volby vhodného výběru.

## 1.1 Logopedie

Viktor Lechta ve svém díle o metodách logopedické intervence (Lechta, 2003, kapitola 3) popisuje 3 základní úrovně logopedické intervence:

- logopedická diagnostika,
- logopedická terapie,
- logopedická prevence.

Autor zmíněné publikace dále píše o uplatňování (v moderní logopedii) modelu symetrického poradenství – tj. jedinec s poruchou řeči (případně osoba, která mu pomáhá – v případě dětí se většinou jedná o rodiče) je partner logopeda se všemi právy i povinnostmi spoluterapeuta. Toto se často v praxi projevuje mj. tím, že se v terapii používají samostatná cvičení (tj. mezi jednotlivými logopedickými konzultacemi musí jedinec provádět cvičení).

Logopedická intervence by se tedy v praxi (podle výše popsaného) dala shrnout do 4 kroků.

1. Logoped provede vyšetření a stanoví diagnózu jedince.
2. Na základě stanovené diagnózy se určí plán terapie, který je pro každého jedince zcela individuální.
3. Logoped provede první kroky plánu a ukáže a předá kompetentní osobě (v případě dítěte se obvykle jedná o jeho rodiče) plán samostatného cvičení, terapie.
4. Jedinec (většinou pod dohledem osoby, která mu pomáhá při cvičeních a hodnotí jeho správnou výslovnost hlásek) provádí samostatné cvičení.

Jednotlivé kroky logopedické intervence se dle potřeby opakuji při každé opětovné konzultaci s logopedem.

Účel Systému je usnadnit zmíněnou logopedickou intervenci. Systém je určen jako podpůrný prostředek pro kroky 3. a 4. Neklade si za cíl tyto kroky nahradit, pouze je usnadnit, případně zefektivnit a to v rozsahu popsaném níže. Jeho hlavním úkolem v těchto bodech je zjednodušit předávání potřebných materiálů pro samostatná cvičení a jejich zefektivnění pomocí zpětné vazby, zdali trénované slovo vyslovil správně, a případné motivace.



### 1.1.1 Posloupnost cvičného textu

Samostatná cvičení mohou nabývat různých podob v závislosti na stanovené diagnóze a dalších aspektech jako je např. věk pacienta, intelekt nebo sociální zázemí.

Logopedická reedukace výslovnosti je sice každému pacientovi přizpůsobena individuálně, ale i přesto existují určitá pravidla, která by se měla dodržovat. Jedním z nich je posloupnost „kategorií“ vyvozovaných zvuků, slov případně i textů (dále jen **cvičný text**), o které se zmiňuje Anna Salomonová (Salomonová, 2003, kapitola 16.2). Nejprve je potřeba začínat od jednoduchého (ale přesného) vyvozování hlásky a postupně přidávat na složitosti až k dlouhým textům (toto pravidlo bude dále označováno jako **pravidlo posloupnosti cvičného textu**).

### 1.1.2 Věk pacienta

Forma cvičení je silně závislá na věku (resp. vyspělosti) pacienta. O tomto se zmiňuje i Anna Salomonová (Salomonová, 2003, kapitola 16.2). Rozděluje formy cvičení podle věku pacienta do 3 základních skupin:

- děti předškolního věku – forma hry (např. vhodná říkadla, memorování pohádek),
- děti školního věku – čtení, psaní,
- adolescenti a dospělí – je potřeba najít vlastní motivaci s čímž souvisí, že forma může být hodně individuální.

Jestliže má Systém zefektivnit samostatná cvičení, je důležité, aby bral v potaz věk Uživatele a to jednak při výběru cvičného textu, tak formy cvičení a zobrazení zpětné vazby.

## 1.2 Validace výslovnosti

Validátor výslovnosti bude postaven na základě současných algoritmů pro rozpoznávání mluvené řeči (dále jen **klasifikátory řeči**). V 1. části této kapitoly jsou popsány relevantní aspekty klasifikátorů řeči. Následně jsou rozebrány možnosti, jak využít klasifikátory řeči pro potřeby validace výslovnosti.

### 1.2.1 Klasifikátory řeči

V publikacích Komunikace s počítačem mluvenou řečí Psutka (1995) a Mluvíme s počítačem česky Psutka (2006) se jejich autoři zabývají problematikou klasifikátorů řeči. Zmiňují se o aspektech, které klasifikátory ovlivňují a rozebírají jednotlivá kritéria, podle nichž se mohou klasifikátory dělit do různých skupin. Relevantní z nich jsou popsány v následujících podkapitolách.

#### Složitost řešené úlohy

Klasifikátory řeči můžeme rozdělit podle složitosti řešené úlohy.

1. Rozpoznávání izolovaných slov
  - Jedná se o rozpoznávání vyslovených povelů (jedno slovo odpovídá jednomu povelu).
  - Většinou je dostatečný relativně malý **slovník** (tj. množina slov rozeznatelných klasifikátorem řeči).
2. Rozpoznávání mluveného diktátu
  - Slova jsou vyslovována izolovaně, s krátkou mezislovní pauzou.
  - Slovník může obsahovat i tisíce slov.
3. Rozpoznávání souvislé řeči
  - Slova jsou rozpoznávána při běžné promluvě (čtená řeč, přednes atd.).
  - Jedná se o rozsáhlé slovníky (desetitisíce slov).

Samostatná cvičení se mohou skládat jednak z procvičování výslovnosti samostatných slov a následně i celých vět. Je tedy nutné, aby validátor výslovnosti uměl pracovat alespoň s izolovanými slovy.

### **Odlišnosti v hlase jednotlivých řečníků**

Hlas jedné osoby se liší od hlasu ostatních lidí. Způsobuje to více faktorů, mezi které patří např. parametry hlasového ústrojí jako tvar dutiny hrdelní, ústní, nosní, frekvence kmitání hlasivek aj. Klasifikátory řeči lze dělit dle rozdílnosti ve výslovnosti následovně.

1. Klasifikátory na řečníku závislé
  - Jsou závislé na hlase jednoho řečníka nebo menší skupiny řečníků.
  - Jsou potřeba před prvním použitím přizpůsobit, natrénovat, na hlas řečníka.
2. Klasifikátory na řečníku nezávislé
  - Jsou trénovány na hlasy mnoha řečníků (až tisíce).

Systém je určen pro osoby s dyslálií. Klasifikátor by tedy neměl být závislý na daném řečníku. V opačném případě by se klasifikátor mohl přizpůsobit špatné výslovnosti, kterou by následně nedokázal rozpoznat.

Pokud by tedy výsledný klasifikátor měl být na řečníku závislý, muselo by jeho přizpůsobení hlasu řečníka probíhat za přítomnosti logopeda, který by každou špatnou výslovnost ze systému odstranil.

### **Odlišnosti v hlase řečníka**

Hlas jedné osoby může být odlišný v různých situacích v závislosti na okolních vlivech, náladě řečníka, jeho zdravotním stavu atd. Málokdy se stane, aby jedno slovo vyslovené vícekrát bylo stejné. Hlavní příčina je v proměnlivosti časování a to jak časové délky celého slova, tak poměrné délky jeho jednotlivých částí.

Pro výsledný validátor řeči může být tento fakt kritický. Některé odlišnosti ve výslovnosti mohou znamenat ještě validní výslovnost a jiné již nevalidní.

## Akustické pozadí

**Akustickým pozadím** rozumíme všechny zvuky, v kterých by se klasifikátor neměl snažit rozeznávat slova ze slovníku. Jedná se např. o okolní šum nebo rušení přenosového kanálu. Pokud řečník bude ve větší vzdálenosti od mikrofonu, nebo bude mluvit potichu, pak může být vliv akustického pozadí o to větší. Pokud se bude řečník nacházet v oblasti, kde se značně mění akustické pozadí, pak bude těžší pro klasifikátor správně rozpoznávat mluvu řečníka. Obzvláště se bude hůře identifikovat začátek a konec slov a zpracovávat slabé frikativy.

Jedním ze způsobů, jak se s hlasitým akustickým pozadím vypořádat, je algoritmus echo cancellation. Tento algoritmus mají v sobě často zabudovány přímo mikrofony.

Vzhledem k povaze Systému je vhodné, aby samostatná cvičení probíhala v klidném prostředí. Není tedy potřeba předpokládat, že by akustické pozadí mělo být příliš rušivé.

## Metody rozpoznávání

Podle aplikovaných metod rozpoznávání se dají klasifikátory rozdělit do dvou základních skupin.

### 1. Klasifikátory, které porovnávají se vzory

Slovo se zpracovává jako celek.

Klasifikátor obsahuje funkci  $F$ , která určí vzdálenost mezi dvěma slovy. Slovo je klasifikováno jako ten vzor ze slovníku, ke kterému má (podle  $F$ ) nejmenší vzdálenost. Pro určení vzdálenosti se nejčastěji používá metody dynamického programování, která pracuje s nelineární časovou normalizací pomocí tzv. nelineární bortivé funkce **dynamic time warping** (dále jen **DTW**). Podle toho je také tento algoritmus označován jako DTW.

Využívá se pro rozpoznávání izolovaných slov a mluveného diktátu pro malé a středně velké slovníky. Dokáže rozpoznávat i souvislou řeč, ta ale musí být zanesena v přesné podobě do slovníku (tj. celé fáze, věty). V této podobě je tedy použitelný pouze omezeně.

Klasifikátor je obvykle na řečníku závislý.

Mezi jeho výhody patří jednoduché **trénování** klasifikátoru při malých slovnících, přičemž trénováním je chápán proces, při kterém se klasifikátor učí rozpoznávat jednotlivá slova.

Mezi jeho nevýhody patří, že nedokáže rozpoznávat souvislou řeč (resp. dokáže, ale pouze omezeně).

### 2. Klasifikátory, které využívají statistické metody

Slovo se zpracovává buď jako celek, nebo je rozděleno na menší subslovní jednotky (např. slabiky, fonémy).

Slova jsou modelována pomocí tzv. skrytých Markovových modelů a jsou klasifikována jako ten vzor ze slovníku, jehož model ho generuje s největší pravděpodobností.

Využívá se pro rozpoznávání izolovaných slov, mluveného diktátu i souvislé řeči s rozsáhlými slovníky.

Klasifikátor může být na řečníku závislý i nezávislý.

Mezi jeho výhody patří:

- svede rozpoznávat i souvislou řeč,
- je rozšířený, existuje několik jeho implementací, např. CMUSphinx (Shmyrev (a)) nebo HTK (Team).

Mezi jeho nevýhody patří, že je pro trénování i rozpoznávání řeči výpočetně náročnější než DTW. S výkonem dnešních počítačů už to není taková komplikace. Stále ale může být problém s výkonem v případě slabšího stroje (např. slabšího mobilního zařízení).

V dnešní době se nejčastěji využívají klasifikátory založené na statistických metodách a to především pro jejich schopnost rozpoznávat souvislou řeč.

DTW potřebuje mít pro svou funkčnost vzorová slova ve slovníku. Pokud bychom měli k dispozici dostatek správně i špatně vyslovených trénovaných slov, mohli bychom pro každé takové slovo vytvořit slovník vzorů všech jeho špatných i dobrých výslovností. Při validaci výslovnosti by se následně pomocí funkce  $F$  rozpoznal nejbližší vzor, na jehož základě by validátor rozhodl o správnosti dané výslovnosti.

Nevýhodou tohoto přístupu je, že je pro každé slovo potřeba provést jeho natrénování i na špatných výslovnostech. Pokud bychom tedy neměli k dispozici všechny nejčastější špatné výslovnosti daného slova, pak by je validátor nedokázal správně vyhodnotit. Špatné výslovnosti by také bylo potřeba mít od více řečníků, tak aby výsledný validátor nebyl na řečníku závislý.

Alternativou k tomuto přístupu, která by nepotřeboval pro každé nové trénované slovo nahrávky jeho špatné výslovnosti, je následující. Ve slovníku by se nacházela pouze správně vyslovená slova. V rámci iniciálního trénování by se pomocí funkce  $F$  spočetla vzdálenost správně a špatně vyslovených slov od jejich vzoru. Na základě těchto hodnot by se spočetl práh vzdálenosti, tj. hodnota vzdálenosti, která by od sebe oddělovala špatnou a správnou výslovnost.

U tohoto přístupu stále zůstává potřeba pro každé nové trénované slovo jeho zanesení do slovníku. K tomu je potřeba vícero nahrávek správné výslovnosti od více řečníků.

Klasifikátory využívající statistické metody mají zvlášť **akustický model**, tj. přiřazení akustických parametrů k jednotlivým subslovním jednotkám (např. fonémům, trifonémům), a zvlášť **fonetický slovník**, tj. přiřazení sekvence fonémů k jednotlivým rozpoznávaným slovům.

Pro správnou funkčnost je tedy potřeba nejdříve vytvořit správný akustický model. Následně je možné vytvořit fonetický slovník pro jakékoliv slovo skládající se z (v akustickém modelu) natrénovaných fonémů.

Výsledný validátor výslovnosti by takto mohl generovat špatnou výslovnost v podobě fonetických slovníků bez nutnosti nových nahrávek výslovnosti daných slov. Toto řešení má ale 3 následující předpoklady.

1. Akustický model obsahuje parametry pro všechny subslovní jednotky.

Tento předpoklad je splnitelný v případě rozpoznávání správné výslovnosti. V případě špatné výslovnosti však nelze zaručit ani vyvrátit, zdali řečník

nepoužije subslovní jednotku, která se vůbec v češtině (resp. v daném jazyce) nevyskytuje.

2. Umíme vygenerovat zápis špatné výslovnosti v podobě fonémů.

Špatnou výslovnost můžeme generovat hned několika způsoby a to nahrazením trénovaného fonému za jeho špatnou alternativu. U žádné varianty ale nelze dopředu určit, nakolik bude daný způsob vhodný pro výsledný validátor výslovnosti.

V kap. 1.1 jsou uvedeny navržené varianty pro generování špatných výslovností.

3. Klasifikátor řeči dokáže správně rozlišit mezi slovy, která se od sebe liší pouze v jednom fonému.

Klasifikátory řeči (resp. převážně jejich dostupné akustické modely) jsou připraveny tak, aby byli na řečníku nezávislé. To má za důsledek, že u jednotlivých slov mají větší toleranci ve způsobech jejich vyslovení. Jejich hlavním účelem totiž není rozpoznání pouze správných výslovností daného slova, ale rozpoznání většiny výslovností daného slova.

Klasifikátory řeči sice běžně obsahují slova lišící se pouze v jednom fonému, mají ale také možnost na základě omezení jazyka (pomocí tzv. jazykových modelů) omezit tato slova na pouze ta, u nichž se očekává jejich vyslovení. Nastavení této vlastnosti je pro přesnost klasifikátoru důležité (viz Shmyrev (e)).

Zůstává tedy otázkou, nakolik budou existující klasifikátory řeči schopny od sebe rozeznat navzájem „blízká“ slova lišící se pouze v jednom fonému. Toto nelze dopředu zaručit ani vyvrátit.

Alternativou k tomuto přístupu může být obdobné nastavení prahové hodnoty jako u DTW. Tj. klasifikátory, které využívají statistické metody, pro svůj výpočet přiřazují každému pravděpodobně vyslovenému slovu tzv. skóre. Toto skóre udává nakolik je pravděpodobné, že model slova generuje danou výslovnost. V rámci iniciálního trénování by se mohlo pomoci nahrávek správné a špatné výslovnosti stanovit práh skóre, který bude od sebe odlišovat správné a špatné výslovnosti.

Tato varianta je s předchozí podobná, tj. obě dvě využívají klasifikátorem spočtené skóre. V prvním případě získáme při validaci skóre pro každé slovo resp. pro správné i špatné výslovnosti a ty mezi sebou porovnáme. V druhém případě se při trénování stanoví práh skóre a při validaci získáme skóre pouze správné výslovnosti a to porovnáme s prahem. Hlavní rozdíl mezi nimi je:

- první varianta rozlišuje správnou a špatnou výslovnost na základě teorie o špatném vyvozování hlásek,
- druhá varianta rozlišuje správnou a špatnou výslovnost na základě testovacích dat.

Která z těchto variant je efektivnější, nelze dopředu jednoznačně určit.

## 1.2.2 Vytvoření českého akustického modelu

České akustické modely pro klasifikátory řeči využívající statistické metody nejsou příliš rozšířené. Např. CMUSphinx ani HTK na svých webových stránkách nemají žádné k dispozici.

Jejich vytvoření je možné 2 základními způsoby. Níže jsou uvedeny se specifiky pro CMUSphinx.

- Vytvoření nového modelu

Pro vytvoření nového modelu je potřeba velké množství nahrávek.

CMUSphinx na svých stránkách (Shmyrev (c)) uvádí, že pro potřeby rozpoznávání izolovaných slov a nezávislosti na řečníku je potřeba mít k dispozici 5 hodin nahrávek od alespoň 200 řečníků. Pokud by jich bylo méně, je velké riziko, že by klasifikátor byl nepřesný.

- Adaptace existujícího modelu

Je možné adaptovat již existující akustické modely, v našem případě pro jiný jazyk, např. angličtinu.

Tento postup předpokládá, že cílený český akustický model může být specifikován pomocí stejné fonémové sady jako anglický. Proces adaptace přizpůsobí parametry fonémů resp. **senonů**, což jsou nejmenší rozpoznávané jednotky řeči resp. jejich reprezentace, pro potřeby češtiny.

Přesné množství potřebných nahrávek není známé, resp. je závislé na rozdílnosti mezi původním anglickým a novým českým modelem. CMUSphinx na svých stránkách (Shmyrev (b)) uvádí, že může být dostatečné 5 minut nahrávek pro adaptaci na hlas řečníka.

## 1.2.3 Generování špatné výslovnosti

V následující kapitole jsou rozebrány možnosti generování špatné výslovnosti pro klasifikátory řeči využívající statistické metody. Tyto klasifikátory mají popsanou výslovnost pomocí fonetického slovníku (viz kap. 1.2.1).

Všechny navržené varianty generování jsou postaveny na nahrazení trénovaného fonému za jeho špatnou alternativu.

### 1. ALL

Trénovaný foném (resp. skupina fonémů zastupující trénovanou samohlásku - blíže viz kap. 2.1.3) se nahradí za libovolný jiný foném (resp. za skupinu fonémů zastupující některou samohlásku - blíže viz kap. 2.1.3), nebo se vynechá. Jedná se o největší z navržených modelů.

Níže je uveden příklad pro slovo „tři“ pomocí fonetické abecedy Arpabet (blíže o fonetických abecedách v kap. 2.1.3).

TRI T R SH IH
INACCURATE0 T AA IH
INACCURATE1 T AE IH
INACCURATE2 T AH IH

INACCURATE3	T	AO	IH
INACCURATE4	T	AW	IH
INACCURATE5	T	AY	IH
INACCURATE6	T	B	IH
INACCURATE7	T	CH	IH
INACCURATE8	T	D	IH
INACCURATE9	T	DH	IH
INACCURATE10	T	EH	IH
INACCURATE11	T	ER	IH
INACCURATE12	T	EY	IH
INACCURATE13	T	F	IH
INACCURATE14	T	G	IH
INACCURATE15	T	HH	IH
INACCURATE16	T	IH	IH
INACCURATE17	T	IY	IH
INACCURATE18	T	JH	IH
INACCURATE19	T	K~IH	
INACCURATE20	T	L	IH
INACCURATE21	T	M	IH
INACCURATE22	T	N	IH
INACCURATE23	T	NG	IH
INACCURATE24	T	OW	IH
INACCURATE25	T	OY	IH
INACCURATE26	T	P	IH
INACCURATE27	T	R	IH
INACCURATE28	T	S~IH	
INACCURATE29	T	SH	IH
INACCURATE30	T	T	IH
INACCURATE31	T	TH	IH
INACCURATE32	T	UH	IH
INACCURATE33	T	UW	IH
INACCURATE34	T	V~IH	
INACCURATE35	T	W	IH
INACCURATE36	T	Y	IH
INACCURATE37	T	Z~IH	
INACCURATE38	T	ZH	IH
INACCURATE39	T	AH AH	IH
INACCURATE40	T	T S~IH	
INACCURATE41	T	D Y	IH
INACCURATE42	T	EH EH	IH
INACCURATE43	T	N Y	IH
INACCURATE44	T	R ZH	IH
INACCURATE45	T	T Y	IH
INACCURATE46	T	IH	

## 2. CREATION

Samohlásky resp. fonémy lze rozdělit dle místa jejich tvorby a způsobem jejich tvoření. O tomto se pojednává např. v publikacích o Logopedie (Neubauer (2010)) a Klinické logopedie ((Salomonová, 2003, kapitola 16.2)).

V tab. 1.1 jsou po sloupcích rozděleny souhlásky do skupin podle místa

	retné		zubodásňové		tvrdopatrové		zadopatrové a hrtanové	
	N	Z	N	Z	N	Z	N	Z
<b>závěrové - ústní</b>	P	B	T	D	Ť	Ď	K	G
<b>závěrové - nosní</b>		M		N		Ň		NG
<b>polozávěrové</b>			C Ā					
<b>úžinové - středové</b>	F	V	S Š	Z Ž		J	CH	H
<b>úžinové - bokové</b>				L				
<b>úžinové - kmitavé</b>			Ř.	R Ř				

Tabulka 1.1: Rozdělení českých souhlásek

jejich tvoření a po řádcích pode způsobu jejich tvoření.

V tab. 1.1 jsou dohromady spojeny souhlásky:

- retoretné a retozubné jako retné,
- dásňové přední a dásňové zadní jako zubodásňové,
- měkopatrové a hrtanové jako „zadopatrové a hrtanové“.

Dále jsou zde přidány symboly pro:

- neznělé ř (v tabulce označeno jako „Ř.“),
- ng (např. ve slově mařinka).

Trénovaný foném (resp. skupina fonémů zastupující trénovanou samohlásku - blíže viz kap. 2.1.3) se nahradí za fonémy, které se tvoří na stejném místě (tj. podle sloupců tab. 1.1) nebo stejným způsobem (tj. podle řádků tab. 1.1), nebo se vynechá.

Níže je uveden příklad pro slovo „tři“ pomocí fonetické abecedy Arpabet (blíže o fonetických abecedách v kap. 2.1.3).

```

TRI T R SH IH
INACCURATEO T R IH
INACCURATE1 T R ZH IH
INACCURATE2 T T IH
INACCURATE3 T T S~IH
INACCURATE4 T CH IH
INACCURATE5 T S~IH
INACCURATE6 T SH IH
INACCURATE7 T IH

```

## 1.2.4 Pravděpodobnost špatné výslovnosti

Klasifikátory využívající statistické metody potřebují krom již zmíněného akustického modelu a fonetického slovníku i **jazykový model**. Ten udává jaká slova má klasifikátor v promluvě řečníka rozpoznávat.

Jazykový model může být ve většině existujících klasifikátorech řeči specifikován ve více formátech. Jedním z nich je pomocí gramatiky udávající v jakém pořadí, případně i s jakou pravděpodobností, mohou být jednotlivá slova rozpoznávána.



Pro validaci výslovnosti jednoho slova je možné vytvořit jednoduchý jazykový model, který udává, že je možné vyslovit trénované slovo buď správně, nebo špatně. Níže jsou uvedeny návrhy základních variant tohoto modelu.

### 1. NORM

Pravděpodobnost žádného slova resp. výslovnosti nebude specifikována. Tím klasifikátor předpokládá, že každá varianta výslovnosti má stejnou pravděpodobnost vyřčení.

Níže je uveden příklad pro variantu jazykového modelu CREATION pro slovo „tři“ zapsaný pomocí JSpeech Grammar Format (dále jen **JSGF**, blíže viz Hunt (2000)) zápisu.

```
#JSGF V1.0;
grammar speechvalidator;
public <speechvalidator> = TRI |
INACCURATE0 |
INACCURATE1 |
INACCURATE2 |
INACCURATE3 |
INACCURATE4 |
INACCURATE5 |
INACCURATE6 |
INACCURATE7;
```

### 2. 5050

Tato varianta předpokládá, že pravděpodobnost správného i špatného vyslovení trénovaného slova je stejná. Toho je docíleno přiřazením správné výslovnosti 50% pravděpodobnosti a rozdělením zbylých 50% procent všem špatným výslovnostem.

Níže je uveden příklad pro variantu jazykového modelu CREATION pro slovo „tři“ zapsaný pomocí JSGF zápisu.

```
#JSGF V1.0;
grammar speechvalidator;
public <speechvalidator> = /1000000000/ TRI |
/1250000000/ INACCURATE0 |
/1250000000/ INACCURATE1 |
/1250000000/ INACCURATE2 |
/1250000000/ INACCURATE3 |
/1250000000/ INACCURATE4 |
/1250000000/ INACCURATE5 |
/1250000000/ INACCURATE6 |
/1250000000/ INACCURATE7;
```

### 3. PREF

Tato varianta se snaží vyřešit možný problém klasifikátoru řeči při rozpoznávání „blízkých“ slov lišících se pouze v jednom fonému (viz kap. 1.2.1)

a to upřednostněním detekce správné výslovnosti. Toho je docíleno přiřazením značně vyšší pravděpodobnosti správné výslovnosti než špatným výslovnostem.

Níže je uveden příklad pro variantu jazykového modelu CREATION pro slovo „tři“ zapsaný pomocí JSGF zápisu.

```
#JSGF V1.0;  
grammar speechvalidator;  
public <speechvalidator> = /10000000000/ TRI |  
  /1.25e-11/ INACCURATE0 |  
  /1.25e-11/ INACCURATE1 |  
  /1.25e-11/ INACCURATE2 |  
  /1.25e-11/ INACCURATE3 |  
  /1.25e-11/ INACCURATE4 |  
  /1.25e-11/ INACCURATE5 |  
  /1.25e-11/ INACCURATE6 |  
  /1.25e-11/ INACCURATE7;
```

## 1.3 Existující řešení

V dnešní době existuje mnoho hotových řešení pro usnadnění logopedické intervence. Většina z nich ale neposkytuje uživatelům zpětnou vazbu určující, jestli dané cvičení bylo provedeno správně. Tím kladou nepsaný požadavek na uživatele, aby byl schopen vlastního sebehodnocení nebo aby prováděl cvičení v přítomnosti osoby, která je hodnocení schopná. Některá nejznámější řešení a další software zabývající se obdobnou tematikou jsou popsány v následujících podkapitolách.

### 1.3.1 Logopedické aplikace

Spolu s masivním rozšířením chytrých mobilních telefonů a tabletů se začalo vyvíjet velké množství aplikací určených pro tato zařízení. Pozadu nezůstávají ani logopedické aplikace. Těch je v dnešní době mnoho a často se zaměřují na cvičení formou hry. Jednou takovou je Logopedie [PMQ]. Na stránkách společnosti PMQ software (PMQ software (2013-2016)) popisují povahu této aplikace. Její hlavní účel je správné zafixování hlásek a jejich procvičení ve slovech. Cvičení je rozděleno do několika lekcí, které se snaží dítě pozitivně motivovat k vyslovení procvičovaných slov. Jedinou lekcí, v které uživatel dostane zpětnou vazbu o správnosti vyslovení slova, je “mluv”. Uživatel si nejdříve poslechne, jak zní správná výslovnost, pak slovo vysloví, přičemž aplikace nahrává zvuk. Ten je nakonec pro účely zpětné vazby přehrán.

Existují i další logopedické aplikace, které ale nijak nekontrolují správnou výslovnost. Jsou spíše určeny pro rozvoj slovní zásoby a dalších vlastností potřebných pro správnou komunikaci. Jednou z nich je např. Mluvídek, o kterém je možné se dočíst na jeho oficiálních stránkách TalkieBear s.r.o..

Na zahraničních trzích se objevují i další aplikace. Ty samozřejmě nejsou bez převodu do češtiny použitelné pro logopedii v České republice, nicméně mohou sloužit jako dobrá inspirace. Na stránkách společnosti Virtual Speech Center Inc.

(Virtual Speech Center, Inc. (2017)) je možné se dočíst o velkém množství logopedických aplikací z jejich tvorby. Na dalších stránkách zabývajících se aplikacemi pro korekci řeči [speech-languageapps.com](http://speech-languageapps.com) (2017) je seznam doporučených aplikací z dané oblasti, např. Articulation Station Pro, SmackTalk, VAST Autism 1 - Core.

Obdobných aplikací je možné najít mnoho. Většinou se soustředí na motivování dítěte vyslovit cvičný text. Občas nabízejí i možnost přehrát si správné znění, v některých případech i s video nahrávkou, na kterém jsou zachyceny artikulační orgány. Případně umožňují nahrát a přehrát svoji vlastní výslovnost (obdobně jako u Logopedie [PMQ] - viz výše). Žádná jiná zpětná vazba není poskytována.

### 1.3.2 Aplikace pro výuku cizích jazyků

Některé aplikace pro výuku cizích jazyků nabízejí cvičení pro zlepšení výslovnosti. Dosahují toho pomocí detekování špatné výslovnosti v promluvě. Uživatel se zobrazí text, který má vyslovit. Aplikace nahrává uživatelskou promluvu a následně v ní detekuje chyby.

U těchto aplikací se bez podrobnějších testů nedá zjistit, jestli jejich hodnocení je dostatečné i pro potřeby přesné výslovnosti, které se při logopedické intervenci snaží logoped docílit. Nicméně je dobré, alespoň pro inspiraci, vědět o existujících řešeních z této oblasti.

Existuje několik takovýchto komerčních aplikací – např. Pronunciation Power (Pronunciation Power Speech Test), NativeAccent English Speech Training (informace o nich jsou čerpány z internetových stránek zmíněných programů ECLI (2017), Carnegie Speech Company (2001-2017)). U těchto komerčních aplikací se nedá zjistit příliš mnoho podrobných informací o tom, jak je kontrola výslovnosti docílena, ani nakolik je efektivní.

Existuje i několik systémů zabývajících se správnou výslovností při výuce cizích jazyků, které jsou popsány v odborných článcích. Jeden z nich popisují autoři článku Automatic Pronunciation Evaluation And Mispronunciation Detection Using CMUSphinx Srikanth a kol. (2012).

Tyto aplikace (stejně jako jiné jim podobné) se při hodnocení výslovnosti snaží o toleranci akcentu mateřského jazyka. Pro uživatele totiž může být demotivující, pokud je jeho výslovnost neustále hodnocena jako nesprávná a to jen z důvodu jeho akcentu.

### 1.3.3 Ostatní aplikace zabývající se výslovností

Při některých činnostech je kladen velký důraz na dokonalou výslovnost (navíc splňující dodatečná pravidla). Jednou z nich je recitace Koránu. Autoři článku Computer Aided Pronunciation Learning System Using Speech Recognition Techniques Sherif (2006) popisují systém HAFSS, který se snaží detekovat chyby ve výslovnosti při recitaci Koránu v arabštině. Na tuto recitaci jsou navíc kladena jistá recitační pravidla, která se musí dodržovat.

HAFSS je dalším dílem, který sice neřeší stejnou oblast jako výsledný Systém. Nicméně svou snahou o co nejpresnější kontrolu výslovnosti se může kvalitou hodnocení blížit požadované úrovni Systému.

### 1.3.4 Knihovny pro validaci výslovnosti

Vhodná knihovna pro hodnocení výslovnosti by byla pro Systém velkým přínosem. V oblasti rozpoznávání řeči existuje několik známých knihoven, které jsou často i volně dostupné. Nicméně jejich hlavní zaměření je na rozpoznávání řeči, nikoliv na hodnocení výslovnosti. Z jejich řady vystoupila knihovna EduSpeak, která podle internetových stránek SRI International (SRI International (2016)) dokáže hodnotit výslovnost rodilých mluvčích i cizinců. EduSpeak v době započetí této práce nepodporoval český jazyk a nebyl volně dostupný.

## 1.4 Zhodnocení a porovnání existujících řešení

Žádný z výše popsaných softwarů nedokáže plně nahradit Systém. Mezi logopedickými aplikacemi (viz kap. 1.3.1) se sice objevuje snaha o zpětnou vazbu, ale žádná z nich nedokáže automaticky ohodnotit výslovnost. Zpětná vazba se zde maximálně vyskytuje v podobě přehrání vyslovené promluvy řečníka.

Ve všech uvedených aplikacích je zatelná snaha o motivaci řečníka k provedení cvičení, což poukazuje na potřebu přizpůsobovat formu cvičení věku (resp. vyspělosti) řečníka (toto je blíže popsáno v kap. 1.1.2).

Existují i další aplikace zabývající se hodnocením výslovnosti (viz kap. 1.3.2 a kap. 1.3.3). Je tedy zřejmé, že nějakého hodnocení se dá dosáhnout, není však jisté, jestli by toto hodnocení bylo dostatečné i pro potřeby logopedické intervence. U těchto aplikací také často bývá problém s dostupností bližších informací o jejich realizaci a přesnosti.

V kapitole o knihovnách pro hodnocení výslovnosti (viz kap. 1.3.4) je zmíněna knihovna EduSpeak. Bohužel nepodporuje český jazyk a v době započetí této práce nebyla volně dostupná, což znemožňuje její použití pro potřeby Systému.

## 1.5 Specifika výsledné práce

V dnešní době nejsou validátory výslovnosti rozšířené a není známé, jak je navrhnout tak, aby dosahovaly přesných výsledků.

Existuje několik aplikací, případně článků zabývajících se validací výslovnosti, jejichž využití je cíleno pro výuku jazyků. Mají tedy větší toleranci k špatné výslovnosti. Není jisté, zdali by adaptace jejich řešení pro češtinu byla dostatečná pro potřeby logopedické intervence (viz kap. 1.4).

Z těchto důvodů je při návrhu a vývoji Systému kladen především důraz na korektnost výsledku (tento důvod bude dále označován jako **důvod korektnosti Systému**). Tj. bude upřednostňováno ověření korektnosti výsledku před komplexností celého Systému a jeho použitelností pro praxi. Např. motivace pro provedení samostatného cvičení bude brána v potaz pouze minimálně.

Dále bude kladen důraz na možnost jednoduchého rozšíření Systému tak, aby byl v případě dobrých výsledků rozšiřitelný do podoby atraktivní pro Uživatele (tento důvod bude dále označován jako **důvod atraktivního Systému**). Tj. tak aby ho bylo možné rozšířit pro potřeby praxe.

I přesto, že dyslálie je nejčastější poruchou komunikačních schopností u osob do 18 let (viz kap. 1), bude vzorovým Uživatelem osoba nespécifikovaného věku.

Bude se o ní předpokládat, že jí nejvíce vyhovuje efektivnost cvičení a korektnost jejich hodnocení.

Ze začátku této kapitoly (viz kap. 1) je popsáno, jak je důležité, aby byl Systém přizpůsoben věku Uživatele a to především při výběru cvičného textu, formy cvičení a zobrazení zpětné vazby. Při volbě Uživatele (viz výše) nebyl jeho věk specifikován. Nicméně vzhledem k jeho výše popsané povaze, budou na Systém kladeny následující požadavky.

- Při výběru cvičných textů nebude kladen důraz na jejich obsahovou stránku, ale na vhodnou kombinaci hlásek, z kterých se skládají.
- Forma cvičení bude vyřčení zobrazeného cvičného textu.
- Při zobrazení zpětné vazby bude kladen důraz na jednoznačnost výsledného hodnocení.
- Z důvodu atraktivního Systému bude možné jednoduše měnit cvičný text.
- Část Systému zodpovědná za hodnocení správné výslovnosti nebude závislá na zvolené formě cvičení a zobrazení zpětné vazby (tj. pokud by se změnila forma cvičení nebo zobrazení zpětné vazby, nebude nutné měnit zmíněnou část Systému).

Vzhledem k důvodu korektnosti Systému bude Systém rozpoznávat pouze izolovaná slova (viz kap. 1.2.1).

Dále bude snaha o vytvoření klasifikátoru na řečníku nezávislém, tak aby odpadla nutnost pracného přizpůsobení na hlas řečníka při dozoru logopeda (viz kap. 1.2.1). Pokud by se ukázal výsledek Systému jako nedostatečný, bylo by vhodným rozšířením této práce zapojení přizpůsobení na hlas řečníka.

Očekává se, že odlišnosti v hlase řečníka a akustické pozadí (viz kap. 1.2.1 a kap. 1.2.1) mohou zásadně ovlivnit výsledky. Obzvlášť bereme-li v potaz cílovou skupinu uživatelů, tj. osoby se špatnou výslovností s největší četností u dětí (resp. u věku do 18 let), můžeme očekávat, že někteří z nich budou mluvit tišeji. Pokud by se ukázal výsledek Systému jako nedostatečný, bylo by vhodným rozšířením této práce detekce tichého vyslovení, případně hlučného akustického pozadí, a následným upozorněním uživatele.

V kap. 1.2.1 jsou popsány 2 základní skupiny klasifikátorů podle metod rozpoznávání a návrh 4 možností jejich využití pro potřeby validace výslovnosti. Ve výsledném validátoru bude použit návrh pro klasifikátory využívající statistické metody založený na generování špatné výslovnosti a to z následujících důvodů.

- Tento typ klasifikátorů je více rozšířen.
- Rozšíření jeho využití i pro validaci celých vět je jednodušší než u DTW.
- Pro přidání nového cvičného slova není potřeba mít nahrávky se všemi (nebo alespoň nejčastějšími) jeho špatnými výslovnostmi.

- Vzhledem k tomu, že nelze dopředu jednoznačně určit, která z variant využívající statistické metody je efektivnější ve validaci výslovnosti, tak je prvně vybrána varianta založená na generování špatné výslovnosti. Pokud by se ukázala jako nedostatečná, bylo by vhodným rozšíření této práce vyzkoušet variantu druhou.

Konkrétně bude pro tyto potřeby použit software CMUSphinx.

Pro tento typ klasifikátoru řeči potřebujeme 3 modely: akustický, fonetický a jazykový (viz kap. 1.2.1).

Možnosti generování akustického modelu jsou popsány v kap. 1.2.2. Vzhledem k velkému množství potřebných nahrávek pro vytvoření modelu nového, bude ve výsledném validátoru použit model adaptovaný.

Fonetickým slovníkem resp. generováním špatné výslovnosti v podobě fonémového zápisu se zabývá kap. 1.2.3. Ve výsledném validátoru budou použity obě navržené varianty: ALL i CREATION. V závěru práce budou porovnány jejich výsledky.

Jazykovým modelem resp. pravděpodobností špatné výslovnosti se zabývá kap. 1.2.4. Ve výsledném validátoru budou použity všechny navržené varianty: NORM, 5050, PREF. V závěru práce budou porovnány jejich výsledky.

Systém tedy bude splňovat následující body.

1. Systém bude rozpoznávat pouze izolovaná slova.
2. Systém bude na řečníku nezávislý.
3. Systém bude využívat klasifikátor řeči využívající statistické metody a to CMUSphinx.
4. Systém bude pro CMUSphinx adaptovat nějaký již existující akustický model.
5. Systém bude pro CMUSphinx generovat fonetický slovník ve variantách:
  - ALL,
  - CREATION.
6. Systém bude pro CMUSphinx generovat jazykový model ve variantách:
  - NORM,
  - 5050,
  - PREF.
7. Systém bude obsahovat testy pro zjištění přesnosti validátoru výslovnosti a porovnání jeho jednotlivých variant (resp. variant jeho modelů).

## 2. Návrh řešení

Na konci minulé kapitoly (kap. 1.5) jsou shrnuty požadavky na Systém. V následujících podkapitolách se uvádí, jakým způsobem jsou jednotlivé požadavky naplněny a další podrobnosti týkající se návrhu Systému.

Výsledná aplikace se nazývá Speech Validator, pro českou lokaci překládáno jako Validátor Výslovnosti.

### 2.1 Architektura Systému

Celý Systém je rozdělen do 3 základních komponent.

- Speech Validator

Jedná se o hlavní frontend Systému, mobilní aplikaci určenou pro platformu Android.

- Speech Validator Server

Jedná se o backend Systému, webový server naprogramovaný převážně v Node.js a C++.

- Speech Validator Admin

Jedná se o mobilní aplikaci určenou pro platformu Android, jejíž hlavní účel je sběr nahrávek pro potřeby trénování validátoru a jeho následná testování.

Jednotlivé komponenty jsou blíže popsány v následujících podkapitolách.

#### 2.1.1 Speech Validator Admin

Speech Validator Admin je mobilní aplikace určená pro platformu Android.

Jedná se o samostatně funkční aplikaci určenou pro administrativní úkoly. Prozatím se administrace potřebná provádět na mobilním zařízení skládá pouze ze sběru nahrávek pro potřeby trénování a testování.

Architektura této aplikace je přizpůsobena možnému rozšíření (viz kap. 1.5), tj. možnosti rozšířit Systém pro rozpoznávání souvislejších promluv.

Toho je dosaženo pomocí Clean Architektury ve spojení s návrhovým vzorem MVC, o tomto se pojednává např. v Uncle Bob (2017). Díky zvolené architektuře je možné jednoduše přidat novou administrativní sekci pro případné rozšíření resp. pro sběr nahrávek souvislých promluv.

#### 2.1.2 Speech Validator

Speech Validator je mobilní aplikace určená pro platformu Android.

Jedná se o frontendovou aplikaci určenou pro samostatná cvičení resp. pro zaznamenání trénovaného slova, jeho odeslání na backend (Speech Validator Server) a následném zobrazení zpětné vazby udávající, zdali výslovnost byla správná.

Architektura této aplikace je stejně jako u Speech Validator Admin přizpůsobena možnému rozšíření (viz kap. 1.5), tj. možnosti rozšířit Systém:

- pro rozpoznávání souvislejších promluv,
- pro zavedení atraktivnější formy cvičení pro větší motivaci Uživatele provádět samostatná cvičení,
- pro přidání nových trénovaných slov.

Toho je dosaženo stejně jako u Speech Validator Admin a to pomocí Clean Architektury ve spojení s návrhovým vzorem MVC.

### 2.1.3 Speech Validator Server

Speech Validator Server je webový server naprogramovaný převážně v Node.js a C++.

Jedná se o backend Systému jehož hlavními úkoly jsou:

- specifikace podoby samostatných cvičení, tj. specifikace trénovaných hlásek resp. slov,
- validace výslovnosti,
- generování všech modelů potřebných pro CMUSphinx:
  - akustický model,
  - fonetický slovník,
  - jazykový model,
- testování validátoru výslovnosti.

Jednotlivé úkoly jsou rozepsány v následujících podkapitolách.

#### Trénované hlásky a slova

Pro potřeby specifikace trénovaných hlásek resp. slov vhodných pro trénování těchto hlásek byla navázána spolupráce s klinickou logopedkou Mgr. Kateřinou Černou. Ta navrhla seznam slov uvedený níže v tab. 2.1, který je v Systému použit. Celkem se jedná o 74 slov.

#### Sběr nahrávek

Sběr nahrávek správné a špatné výslovnosti všech výše zmíněných slov (viz kap. 2.1.3) probíhal ve spolupráci s klinickou logopedkou Mgr. Kateřinou Černou. Ta měla k dispozici aplikace Speech Validator Admin, pomocí které nasbírala pro každé slovo alespoň 10 správných výslovností a alespoň 10 špatných výslovností. Celkem se jedná o 1734 nahrávek. Přesný počet nahrávek pro jednotlivá slova je uveden v tab. 2.2.

#### Generování akustického modelu

V kap. 1.2.2 resp. v kap. 1.5 je popsáno, že výsledný akustický model bude vytvořen adaptací nějakého již existujícího.

Adaptovaným akustickým modelem je od CMUSphinx doporučený model pro angličtinu dostupný na webových stránkách Slashdot Media (2017). Tento akustický model je dostupný ve 4 základních variantách, které se dělí podle:



- frekvence nahrávek, pro které jsou přizpůsobeny:
  - 8 kHz,
  - 16 kHz,
- typu akustického modelu:
  - continues,  
jedná se o přesnější model, který je ale více výpočetně náročný (hodí se spíše pro backendové resp. serverové systémy),
  - semi-continues,  
jedná se o model, který je vhodný pro kompromis mezi výpočetní náročností a přesností (hodí se spíše pro frontendové systémy).

Pro Systém byl vybrán 16 kHz continues model a to z důvodu, že přesnost má vyšší prioritu než výpočetní náročnost resp. než doba odezvy Systému při validaci výslovnosti.

Adaptace modelu probíhá dle doporučení uvedeného v článku Shmyrev (d). Nejdříve je potřeba vytvořit **adaptační korpus**. Jedná se o:

- seznam vět, na kterých adaptace probíhá (v našem případě seznam slov navržených v kap. 2.1.3),
- fonetický slovník obsahující přepis všech použitých slov do sekvence fonémů,
- nahrávky vyslovení seznamu vět určených pro adaptaci.

Pro adaptaci je použita polovina nahrávek správně vyslovených slov nasbíraných pomocí Speech Validator Admin (viz kap. 2.1.3). Počet nahrávek použitých pro jednotlivá slova je uveden v tab. 2.3. Celkem je použito 471 nahrávek.

Seznam vět se skládá ze dvou souborů:

- speech-validator.transcription,
- speech-validator.fileids.

První z nich obsahuje na každém řádku přepis textu (v našem případě trénované slovo) obsaženého v nahrávce uzavřené do značek `<s>` a `</s>` následovaných odkazem na soubor s nahrávkou. Slova v přepsaném textu musí být psána velkými písmeny bez diakritiky. Stejný formát zápisu je použit napříč všemi modely, tj. ve fonetickém slovníku i jazykovém modelu.

Níže je uveden vzorový příklad souboru pro prvních 5 trénovaných slov, u každého jsou uvedeny 2 nahrávky.

```
<s> VANA </s> (record1)
<s> VANA </s> (record2)
<s> MAV A </s> (record10)
<s> MAV A </s> (record11)
<s> FOTI </s> (record19)
<s> FOTI </s> (record20)
<s> PANEV </s> (record25)
<s> PANEV </s> (record26)
<s> POFOUKA </s> (record31)
<s> POFOUKA </s> (record32)
```

Druhý soubor `speech-validator.fileids` obsahuje pouze seznam všech použitých nahrávek.

Níže je uveden příklad pro výše popsany vzorový případ.

```
record1
record2
record10
record11
record19
record20
record25
record26
record31
record32
```

Pro adaptační korpus nyní stačí vytvořit fonetický slovník `speech-validator.dic`. Bližší postup vytváření fonetického slovníku je popsán v kap. 2.1.3.

Níže je uveden příklad pro výše popsany vzorový případ.

```
VANA V~AH N AH
MAVA M AH AH V~AH AH
FOTI F AO T Y IY
PANEV P AH AH N EH F
POFOUKA P AO F OW K~AH AH
```

S takto připraveným adaptačním korpusem je možné započít proces adaptace. Prvně je potřeba nahrát do aktuální resp. pracovní složky:

- anglický akustický model,
- `speech-validator.transcriton`,
- `speech-validator.fileids`,
- `speech-validator.dic`,
- všechny nahrávky určené pro adaptaci.

```
cp -a /path/to/downloaded/pocketsphinx/model/en-us/en-us .
cp /path/to/generated/speech-validator.transcriton .
cp /path/to/generated/speech-validator.fileids .
cp /path/to/generated/speech-validator.dic .
cp -a /path/to/adaptation/records .
```

Nyní je potřeba pro každou nahrávku vytvořit soubor s akustickými příznaky. K tomu slouží nástroj `sphinx_fe`, který je součástí programu `SphinxBase`. Je třeba, aby všechny jeho parametry byly nastaveny správně.

- Parametr `argfile` musí obsahovat cestu k souboru `feat.params`, který se nachází uvnitř anglického akustického modelu. V tomto souboru se nachází parametry modelu potřebné pro extrakci akustických příznaků.

- Parametr `sample` musí být nastaven na frekvenci, s kterou byly nahrávky pořízeny, tj. 16 kHz.
- Parametr `c` ukazuje na vygenerovaný soubor `speech-validator.fileids`.
- Parametr `ei` udává koncovku souborů, v kterých jsou nahrávky uloženy.
- Parametr `raw` udává, že nahrávky byly pořízeny resp. uloženy v `raw` formátu.

```
sphinx_fe \
  -argfile en-us/feat.params \
  -sample 16000 \
  -c speech-validator.fileids \
  -di . \
  -do . \
  -ei raw \
  -eo mfc \
  -raw yes
```

Po provedení tohoto příkazu bychom měli mít v aktuálním adresáři vygenerované následující soubory.

```
record1.mfc
record2.mfc
record10.mfc
record11.mfc
record19.mfc
record20.mfc
record25.mfc
record26.mfc
record31.mfc
record32.mfc
```

Dalším krokem je vypočítat statistiky na základě vygenerovaných akustických příznaků. K tomu použijeme nástroj `bw` (tj. zkratka pro „Baum-Welch“), který je součástí programu `SphinxTrain`. Výsledné statistiky se uloží do souborů:

- `gauden_counts`,
- `mixw_counts`,
- `tmat_counts`.

Níže je uveden seznam zásadních parametrů `bw`.

- Parametr `ts2cbfn` musí být nastaven na `.cont.`, tj. používáme continues typ modelu.
- Parametr `dictfn` ukazuje na vygenerovaný fonetický slovník `speech-validator.dic`.

- Parametr `ctlfn` ukazuje na vygenerovaný soubor se seznamem nahrávek `speech-validator.fileids`.
- Parametr `lsnfn` ukazuje na vygenerovaný soubor s textem nahrávek `speech-validator.transcription`.

```
.bw \
-hmmdir en-us \
-moddefn en-us/mdef \
-ts2cbfn .cont. \
-feat ls_c_d_dd \
-cmn current \
-agc none \
-dictfn speech-validator.dic \
-ctlfn speech-validator.fileids \
-lsnfn speech-validator.transcription \
-accumdir . \
-lda en-us/feature_transform
```

Dále je potřeba připravit transformační matici pro adaptování skrytých Markovových modelů na základě získaných statistik. K tomu použijeme nástroj `mllr_solve` (tj. zkratka pro „Maximum Likelihood Linear Regression“), který je součástí programu `SphinxTrain`.

Níže je uveden seznam zásadních parametrů `mllr_solve`.

- Parametr `outmllrfn` udává, kam bude uložena transformační matice.
- Parametr `accumdir` ukazuje na adresář, v kterém jsou uloženy statistiky vypočítané pomocí `bw`.

```
./mllr_solve \
-meanfn en-us/means \
-varfn en-us/variances \
-outmllrfn mllr_matrix \
-accumdir .
```

Posledním krokem je pomocí transformační matice upravit skryté Markovovy modely. K tomuto účelu slouží nástroj `mllr_transform` (tj. zkratka pro „Maximum Likelihood Linear Regression“), který je součástí programu `SphinxTrain`.

Níže je uveden seznam zásadních parametrů `mllr_transform`.

- Parametr `mllrmat` ukazuje na soubor transformační matice.
- Parametr `inmeanfn` ukazuje na vstupní soubor.
- Parametr `outmeanfn` udává do jakého souboru bude uložen transformovaný výstup.

```
./mllr_transform \  
-inmeanfn work/en-us/means \  
-outmeanfn work/en-us/means \  
-mllrmat work/mllr_matrix
```

Nyní máme ve složce `./en-us` uložený adaptovaný akustický model. Tuto složku můžeme přejmenovat tak, aby se nepletla s původním názvem modelu na `en-us-cz-speech-validator`.

## Fonetická abeceda

V kap. 2.1.3 byl vybrán anglický akustický model, který se adaptoval na český. Tento model je přizpůsoben pro fonetickou abecedu Arpabet. Jedná se o abecedu používanou pro fonetickou transkripci, tj. přepis slov do jejich fonémové podoby. Nově adaptovaný model musí také používat Arpabet.

V Ústavu formální a aplikované lingvistiky byl vytvořen Alex Dialogue Systems Framework (dále jen **ALEX**) GitHub, Inc. (2017a). Součástí tohoto softwaru je i skript (GitHub, Inc. (2017b)) pro fonetickou transkripci, pro přepis českých slov do fonémů ČFA. Tento skript je v Systému použit pro generování fonetického slovníku. Nejdříve se s jeho pomocí přepíše trénovaná slova do ČFA a následně se tento zápis transformuje do Arpabetu.

V Arpabetu však není žádný vhodný symbol pro české „CH“. Z tohoto důvodu není možné pomocí adaptovaného modelu validovat správnou výslovnost hlásky „CH“. Odpadá tím možnost validace slov chata, duch a moucha z navržených v kap. 2.1.3.

V Arpabetu také nejsou vhodné fonémy pro některé další české hlásky, např. pro „Ť“. Toto se řeší přepisem do skupiny fonémů, např. foném v ČFA „tj“ je transformován na dvojici fonémů Arpabetu „T Y“.

Algoritmus pro transformaci ČFA do Arpabetu vychází ze dvou zdrojů PAVEL (2012) a O-W. V prvním z nich je popsáno mapování ČFA na Mezinárodní fonetickou abecedu (dále jen **IPA**). V druhé je popsáno mapování z IPA na Arpabet. Chybějící mapování byla navržena v rámci této práce. Výsledná transformace z ČFA na Arpabet je popsána v tab. 2.4.

## Generování fonetického slovníku

V Systému je pro každé trénované slovo vygenerovaný fonetický slovník zvlášť. Ten obsahuje jednak zápis správné výslovnosti, jednak všech špatných výslovností.

Nejdříve se vezme zápis daného slova a provede se fonetická transkripce popsaná v kap. 2.1.3. Tím se získá zápis pro správnou výslovnost.

Nyní se z tohoto správného zápisu vygeneruje špatná výslovnost. Bližší popis tohoto postupu je popsán v kap. 1.2.3.

Ve výsledku takto získáme 2 modely, jeden pro variantu ALL a druhý pro CREATION.

Ve fonetickém slovníku je možné jednomu slovu přiřadit více výslovností. Způsob tohoto zápisu je uveden v příkladu pro slovo „tři“, kde je uvedena správná výslovnost a 2 způsoby špatné výslovnosti.

```
TRI T R SH IH
INACCURATE T R IH
INACCURATE(2) T R ZH IH
```

Takto můžeme vygenerovat další 2 varianty fonetického slovníku, tj. jednu dle zápisu alternativní výslovnosti uvedené výše a druhou pomocí zápisu více slov, z čehož pouze první slovo představuje správný zápis a všechna další slova špatný.

```
TRI T R SH IH
INACCURATEO T R IH
INACCURATE1 T R ZH IH
```

Takto jsou v Systému vygenerovány 4 varianty fonetického slovníku:

- ALL2,
- ALLN,
- CREATION2,
- CREATIONN,

jejichž názvy udávají, zdali jsou v daném modelu pouze 2 slova (tj. 1 se správnou výslovností a 1 s vícero zápisů špatné výslovnosti), nebo zdali je zde  $N$  slov (tj. 1 se správnou výslovností a  $N - 1$  se špatnou výslovností).

Výsledky těchto modelů jsou v závěru práce porovnány (viz kap. 2.1.3 resp. kap. 5).

## Generování jazykového modelu

V Systému je pro každé trénované slovo vygenerovaný jazykový model zvlášť. Ten obsahuje možnost správně i špatně vysloveného slova.

V kap. 1.2.4 jsou popsány 3 základní varianty jazykového modelu: NORM, 5050, PREF.

Varianta NORM je jednoznačná, kdežto u zbylých dvou není zřejmé jak správně nastavit pravděpodobnosti. JSGF zápis umožňuje každému slovu přiřadit váhu. Pokud tuto váhu nastavíme 2 slovům stejně, mělo by to znamenat, že mají stejnou pravděpodobnost vyřčení. V CMUSphinx ale tato pravděpodobnost vstupuje do složitějšího výpočtu tzv. skóre a tím její nastavení ztrácí na intuitivnosti. CMUSphinx bohužel nikde oficiálně neuvádí doporučené hodnoty těchto vah. Z jednoduchých pokusů je zřejmé, že se tyto váhy musí nastavovat ve vysokých řádech, např.  $1e10$  nebo  $1e - 10$ .

Krom vah JSGF je možné CMUSphinx nastavit parametr Language Weight (dále jen **LW**). Skóre každého slova se skládá ze 2 složek, které se sčítají, a to z akustického skóre a jazykového skóre. LW vstupuje do výpočtu jazykového skóre jako činitel. Čím větší se tedy nastaví, tím větší dopad má jazykový model na výsledky klasifikátoru řeči. Pokud se ale nastaví příliš vysoké, pak akustické skóre bude mít na výsledek pouze minimální dopad, tj. klasifikátor řeči se bude rozhodovat převážně podle jazykového modelu.

V závěru této práce jsou popsány testy, pro různé hodnoty nastavení vah a LW (viz kap. 2.1.3 resp. kap. 5). Tyto testy jsou prováděny na menším množství slov za účelem vhodného nastavení vah pro celý Systém.

Jazykový model je vygenerován pro každou variantu fonetického slovníku zvlášť. Tím dostáváme 12 variant:

- ALL2-NORM,
- ALLN-NORM,
- CREATION2-NORM,
- CREATIONN-NORM,
- ALL2-5050,
- ALLN-5050,
- CREATION2-5050,
- CREATIONN-5050,
- ALL2-PREF,
- ALLN-PREF,
- CREATION2-PREF,
- CREATIONN-PREF.

Výsledky těchto modelů jsou v závěru práce porovnány (viz kap. 2.1.3 resp. kap. 5).

### Validace výslovnosti

Základní myšlenka validace výslovnosti pomocí CMUSPhinx je popsána v kap. 1.2.1. Vyslovené slovo se nechá rozpoznat klasifikátorem řeči CMUSphinx. Tomu se předloží připravený:

- adaptovaný akustický model,
- fonetický slovník, který obsahuje zápis správné výslovnosti a všech předpokládaných špatných výslovností,
- jazykový model, který udává, že je možné slovo vyslovit správně i špatně a dle zvolené varianty určí i jaké jsou pravděpodobnosti.

Jako výstup se do Systému vrátí seznam slov **nbest** obsahující  $N$  slov a u každého z nich i jeho skóre. Skóre udává nakolik je pravděpodobné, že dané slovo bylo vyřčeno. Tedy slovo s největším skóre má největší pravděpodobnost, že bylo vyřčeno.

Jedno slovo se v nbest může vyskytovat vícekrát. To je dáno způsobem výpočtu nbest klasifikátorem řeči. V tomto případě Systém skóre daných slov počítá. Toto má největší teoretický dopad na fonetické slovníky ALL2 a CREATION2, kde jsou špatné výslovnosti evidovány jako jedno slovo.

Pokud je slovo s největším skóre shodné s trénovaným slovem, Systém prohlásí, že dané slovo bylo vysloveno správně.

## Testování validátoru výslovnosti

Nedílným výsledkem této práce je i otestování výsledného návrhu validátoru výslovnosti. Systém obsahuje testy, kterými se dají výsledky porovnat.

Všechny testy se spouští nad nasbíranými nahrávkami aplikací Speech Validator Admin (viz kap. 2.1.3) krom těch, které byly použity pro adaptaci akustického modelu (viz kap. 2.1.3) a které slouží pro trénování hlásky „CH“ (viz kap. 2.1.3).

První testy jsou provedeny pro správné nastavení vah JSGF, které jsou popsány v kap. 2.1.3. Pro tyto testy je vybráno 6 testovacích slov:

- vana (ve výsledcích uváděno jako VANA),
- činka (ve výsledcích uváděno jako CINKA),
- liška (ve výsledcích uváděno jako LISKA),
- dříví (ve výsledcích uváděno jako DRIVI),
- voní (ve výsledcích uváděno jako VONI),
- magi (ve výsledcích uváděno jako MAGI)

a fonetický slovník ve variantě CREATIONN a jazykový model CREATIONN-PREF resp. CREATIONN-5050.

Slova byla vybrána tak, aby pokrývala všechny skupiny místa a způsobu tvorby hlásek pro variantu CREATION (viz kap. 1.2.3)..

Závěrem z testu jsou hodnoty vah JSGF:

- pro CREATIONN-PREF  $1e20$  a  $1e0$ ,
- pro CREATIONN-5050  $1e - 15$  a  $1e - 15$ ,

s kterými je dosaženo nejlepších výsledků. Tyto hodnoty jsou použity v modelech pro Systém.

Druhé testy jsou provedeny pro správné nastavení parametru LW, který je popsán v kap. 2.1.3. Pro tyto testy jsou vybrána stejná testovací slova i modely jako pro předchozí testy.

Závěrem z testu jsou hodnota LW:

- pro CREATIONN-PREF 18,
- pro CREATIONN-5050 9,

s kterými je dosaženo nejlepších výsledků. Tyto hodnoty LW jsou použity v modelech pro Systém. Ve zbylých testech se porovnávají výsledky za použití různých variant fonetických slovníků resp. jazykových modelů:

- ALL2-NORM,
- ALLN-NORM,
- CREATION2-NORM,
- CREATIONN-NORM,



- ALL2-5050,
- ALLN-5050,
- CREATION2-5050,
- CREATIONN-5050,
- ALL2-PREF,
- ALLN-PREF,
- CREATION2-PREF,
- CREATIONN-PREF.

Tyto testy se spouští nad všemi nasbíranými nahrávkami aplikací Speech Validator Admin. Celkem se jedná o 1263 nahrávek, z toho 438 správných výslovností a 825 špatných. Přesný počet nahrávek pro jednotlivá slova je uveden v tab. 2.5.

Hláška	Slovo	Hláška	Slovo
C	cop	R	brambora
C	opice	R	dárky
C	pec	R	drak
C	pecka	R	metr
Č	činka	R	pere
Č	kočka	R	rohy
Č	míč	R	rýč
Č	míček	Ř	buřt
Č	oči	Ř	čtyři
CH	chata	Ř	dříví
CH	duch	Ř	řepa
CH	moucha	Ř	skřítek
D	dům	Ř	tři
D	jede	S	kost
Ď	dítě	S	miska
Ď	vidí	S	pes
F	fotí	S	pusa
F	pánev	S	sova
F	pofouká	Š	mašinka
G	gól	Š	myš
G	magi	Š	myška
H	duha	Š	šála
H	houpe	Š	tužka
K	kos	T	boty
K	luk	T	most
K	maska	T	sud
K	muška	T	táta
L	dělo	T	tudy
L	liška	Ť	kotě
L	pálka	Ť	sed
L	pil	Ť	tělo
N	brána	V	mává
N	buben	V	vana
N	nohy	Z	koza
Ň	nitě	Z	zuby
Ň	voní	Ž	ježek
		Ž	loupežník
		Ž	žába

Tabulka 2.1: Seznam slov

Slovo	# správně	# špatně	Slovo	# správně	# špatně
boty	15	10	miska	11	10
brambora	13	13	most	12	10
brána	10	10	moucha	15	10
buben	10	10	muška	10	11
buřt	10	17	myš	12	10
chata	16	10	myška	14	10
činka	14	10	nitě	10	10
cop	13	10	nohy	10	10
čtyři	10	20	oči	11	10
dárky	14	11	opice	12	10
dělo	13	10	pálka	11	10
dítě	14	10	pánev	11	10
drak	12	12	pec	12	10
dříví	11	19	pecka	11	11
duch	15	10	pere	14	11
duha	15	11	pes	11	10
dům	11	10	pil	10	12
fotí	12	10	pofouká	11	11
gól	10	10	pusa	12	10
houpe	19	10	řepa	10	20
jede	10	11	rohy	14	11
ježek	13	10	rýč	13	12
kočka	15	11	šála	12	10
kos	11	10	sedl	10	11
kost	11	10	skřítek	10	22
kotě	10	10	sova	10	11
koza	13	10	sud	12	10
liška	14	10	táta	14	11
loupežník	11	10	tělo	11	10
luk	10	11	tři	10	18
magi	13	10	tudy	15	10
mašinka	17	10	tužka	14	10
maska	10	10	vana	17	10
mává	17	10	vidí	14	10
metr	11	15	voní	11	10
míč	13	12	žába	12	10
míček	12	10	zuby	12	10

Tabulka 2.2: Počet nahrávek

Slovo	# nahrávek	Slovo	# nahrávek
boty	8	miska	6
brambora	7	most	6
brána	5	moucha	8
buben	5	muška	5
buřt	5	myš	6
chata	8	myška	7
činka	7	nitě	5
cop	7	nohy	5
čtyři	5	oči	6
dárky	7	opice	6
dělo	7	pálka	6
dítě	7	pánev	6
drak	6	pec	6
dříví	6	pecka	6
duch	8	pere	7
duha	8	pes	6
dům	6	pil	5
fotí	6	pofouká	6
gól	5	pusa	6
houpe	10	řepa	5
jede	5	rohy	7
ježek	7	rýč	7
kočka	8	šála	6
kos	6	sedl	5
kost	6	skřítek	5
kotě	5	sova	5
koza	7	sud	6
liška	7	táta	7
loupežník	6	tělo	6
luk	5	tři	5
magi	7	tudy	8
mašinka	9	tužka	7
maska	5	vana	9
mává	9	vidí	7
metr	6	voní	6
míč	7	žába	6
míček	6	zuby	6

Tabulka 2.3: Počet nahrávek pro adaptaci

<b>ČFA</b>	<b>Arpabet</b>
a	AH
aa	AH AH
aw	AW
b	B
c	T S
ch	CH
d	D
dj	D Y
e	EH
ee	EH EH
f	F
g	G
h	HH
i	IH
ii	IY
j	Y
k	K
l	L
m	M
n	N
ng	NG
nj	N Y
o	AO
ow	OW
p	P
r	R
rsh	R SH
rzh	R ZH
s	S
sh	SH
t	T
tj	T Y
u	UH
uu	UH
v	V
x	
z	Z
zh	ZH

Tabulka 2.4: Transformace ČFA na Arpabet

Slovo	# správně	# špatně	Slovo	# správně	# špatně
boty	7	10	miska	5	10
brambora	6	13	most	6	10
brána	5	10	moucha	7	10
buben	5	10	muška	5	11
buřt	5	17	myš	6	10
chata	8	10	myška	7	10
činka	7	10	nitě	5	10
cop	6	10	nohy	5	10
čtyři	5	20	oči	5	10
dárky	7	11	opice	6	10
dělo	6	10	pálka	5	10
dítě	7	10	pánev	5	10
drak	6	12	pec	6	10
dříví	5	19	pecka	5	11
duch	7	10	pere	7	11
duha	7	11	pes	5	10
dům	5	10	pil	5	12
fotí	6	10	pofouká	5	11
gól	5	10	pusa	6	10
houpe	9	10	řepa	5	20
jede	5	11	rohy	7	11
ježek	6	10	rýč	6	12
kočka	7	11	šála	6	10
kos	5	10	seď	5	11
kost	5	10	skřítek	5	22
kotě	5	10	sova	5	11
koza	6	10	sud	6	10
liška	7	10	táta	7	11
loupežník	5	10	tělo	5	10
luk	5	11	tři	5	18
magi	6	10	tudy	7	10
mašinka	8	10	tužka	7	10
maska	5	10	vana	8	10
mává	8	10	vidí	7	10
metr	5	15	voní	5	10
míč	6	12	žába	6	10
míček	6	10	zuby	6	10

Tabulka 2.5: Počet nahrávek pro testování

## 3. Uživatelská a administrátorská dokumentace

System se skládá ze 3 aplikací:

- Speech Validator Admin,
- Speech Validator,
- Speech Validator Server.

Každá z těchto aplikací je popsána samostatně v následujících podkapitolách.

### 3.1 Speech Validator Admin

Speech Validator Admin je mobilní aplikace určená pro platformu Android. Její účel je sběr nahrávek pro potřeby validátoru výslovnosti, který je součástí Speech Validator Server. Jedná se o zcela samostatnou aplikaci, tj. nijak nekomunikuje se Speech Validator Server.

Funguje offline, bez připojení k internetu, a obsahuje v sobě lokální kopii všech hlásek a slov, jejichž nahrávky je potřeba nasbírat.

Nahrávky se ukládají na lokální disk a to ve formátu vhodném pro Speech Validator Server:

- frekvence 16 kHz,
- kanál mono,
- kódování PCM 16 bit.

#### 3.1.1 Instalace aplikace

Instalační soubor aplikace `speech-validator-admin.apk` se nachází na CD ve složce `executable`.

Pro instalaci proveďte následující kroky.

1. Nahrajte instalační soubor do mobilního zařízení.
2. V mobilním zařízení spustě nahraný instalační soubor.
3. Pokud se instalace nepovedla je potřeba v nastavení mobilního zařízení povolit instalaci aplikací z neznámých zdrojů a zopakovat 2. krok.

#### 3.1.2 Ovládání aplikace

Po zapnutí aplikace se zobrazí uvítací obrazovka, viz obr. B1.

Po použití tlačítka Nahrát výslovnost je uživatel přesměrován na obrazovku pro výběr hlásky, viz obr. B2. Nachází se zde seznam všech hlásek pro nahrání. U každé z nich se v pravé části daného řádku zobrazují 2 čísla. Druhé z nich

udává kolik slov je potřeba k dané hlásce namluvit, kdežto první kolik jich již bylo namluveno.

Kliknutím na hlásku se zobrazí obrazovka pro výběr slova, viz obr. B3. U každého z nich se v pravé části daného řádku zobrazují 2 dvojice čísel. První dvojice, umístěná nahoře, udává kolikrát je dané slovo potřeba namluvit správně, resp. kolikrát již bylo správně namluveno. Druhá dvojice, umístěná dole, udává kolikrát je dané slovo potřeba namluvit špatně, resp. kolikrát již bylo špatně namluveno.

Kliknutím na slovo se zobrazí obrazovka se seznamem nahrávek, viz obr. B4. Nachází se zde všechny již provedené nahrávky daného slova. U každé z nich je uvedeno,

- zdali se jedná o správně vyslovené slovo (to je dáno typem smajlíku),
- datum a čas jejího pořízení.

Novou nahrávku je možné pořídít použitím tlačítka se symbolem plusu, které je umístěno v pravém dolním rohu. Tímto se zobrazí dialogové okno (viz obr. B5), v kterém je možné:

- nahrát vyslovení daného slova,
- nahrávku si přehrát,
- označit, zdali slovo bylo vysloveno správně
- a následně nahrávku uložit,
- nebo zrušit.

K nahrávce je možné se vrátit a to kliknutím na řádek této nahrávky. Tímto se zobrazí stejné dialogové okno jako pro přidání.

## 3.2 Speech Validator

Speech Validator je mobilní aplikace určená pro platformu Android.

Slouží k provádění samostatných cvičení, tj. trénování výslovnosti hlásek resp. slov. Aplikace zaznamená vyslovené slovo, odešlo ho do Speech Validator Server a následně zobrazení, zdali výslovnost byla správná.

Tato aplikace neumí fungovat bez internetu, resp. obsahuje buffer, díky kterému si dokáže po určitý čas zapamatovat načtená data. Nicméně vzhledem k tomu, že pro validaci výslovnosti je potřebné, aby byl Speech Validator Server dostupný, je tato funkcionality prozatím zbytečná. Její využití je připraveno pro potřeby případného rozšíření cvičení.

Nahrávky se pořizují ve formátu vhodném pro Speech Validator Server:

- frekvence 16 kHz,
- kanál mono,
- kódování PCM 16 bit.



### 3.2.1 Instalace aplikace

Instalační soubor aplikace `speech-validator.apk` se nachází na CD ve složce `executable`.

Pro instalaci proveďte následující kroky.

1. Nahrajte instalační soubor do mobilního zařízení.
2. V mobilním zařízení spustě nahraný instalační soubor.
3. Pokud se instalace nepovedla je potřeba v nastavení mobilního zařízení povolit instalaci aplikací z neznámých zdrojů a zopakovat 2. krok.

Instalační soubor předpokládá, že Speech Validator Server je dostupný na adrese `http://speechvalidator.localnet:3000/`. Ideální postup je připojit mobilní zařízení na lokální síť, v které je i Speech Validator Server, a nastavit DNS záznam pro `speechvalidator.localnet` na lokální IP adresu počítače, na kterém je Speech Validator Server provozován.

Pokud tento postup nebylo možné provést, pak je potřeba změnit konfigurační soubor pro Speech Validator a provést build nového instalačního programu. Blíže o tomto v kap. 4.

### 3.2.2 Ovládání aplikace

Po zapnutí aplikace se zobrazí uvítací obrazovka, viz obr. B6.

Po použití tlačítka Start je uživatel přesměrován na obrazovku pro výběr hlásky, viz obr. B7. Nachází se zde seznam všech hlásek pro nahrání.

Kliknutím na hlásku se zobrazí obrazovka pro výběr slova, viz obr. B8. U každého z nich se v levé části zobrazuje ikona aplikace. Jako vhodné rozšíření v rámci důvodu atraktivního Systému je zde vhodné umístit obrázek daného slova.

Kliknutím na slovo se zobrazí obrazovka pro samostatné cvičení, viz obr. B9.

Je zde možné nahrát vyslovení daného slova pomocí tlačítka Spustit a následně Zastavit. Tím se nahrávka automaticky odešle na Speech Validator Server a následně se ve spodní části obrazovky zobrazí, zdali se jedná o správně vyslovené slovo.

Nahrávku je možné zpětně přehrát.

## 3.3 Speech Validator Server

Speech Validator Server je webový server. Jedná se o backend Systému jehož úkoly jsou:

- specifikace podoby samostatných cvičení, tj. specifikace trénovaných hlásek a slov,
- validace výslovnosti,
- ukládání dat z validací výslovnosti,

- generování všech modelů potřebných pro CMUSphinx:
  - akustický model,
  - fonetický slovník,
  - jazykový model,
- testování validátoru výslovnosti.

### 3.3.1 Instalace aplikace

Speech Validator Server je Node.js server, ve většině případů pro spuštění a provoz stačí programy `node` a `npm`. Jejich instalaci je možné provést následujícími příkazy.

```
sudo apt-get update
sudo apt-get install nodejs
sudo apt-get install npm
```

Pro zprovoznění je potřeba celou složku s kódem zkopírovat do složky, odkud se bude server spouštět. Ta se nachází na CD ve složce `speech-validator`.

Dále je potřeba ve složce pro server nainstalovat závislosti.

```
cd server
npm install
```

Konfigurační soubor aplikace se nachází ve složce `server/config/config.example.js`. Tento soubor je potřeba zkopírovat a změnit mu jméno na `server/config/config.js`. Ve většině případů je dostatečné ho ponechat nezměněný. Je v něm možné změnit connection string do databáze, případně upravit logování nebo port, na kterém je server dostupný.

Nyní schází připravit databázi. Pro databázi je připraven Docker kontejner. Prvně je tedy potřeba nainstalovat program `docker-compose`. To je možné provést následovně.

```
curl -L https://github.com/docker/compose/releases/download/1.14.0/...
...docker-compose-'uname -s'-'uname -m' > /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

Připravený kontejner je pak možné spustit následovně.

```
cd ..
sudo docker-compose up
```

Tímto jsme spustili databázi. Její iniciální naplnění provedeme spuštěním migrací. Pro ty jsou připraveny skript `syncDb` a migrace.

```
cd server
node bin/syncDb
sequelize db:migrate
```

Tímto máme hlavní část severu připravenou pro spuštění.

Nyní je ještě potřeba připravit vše potřebné pro validátory. Na serveru jsou aktuálně připraveny 2 druhy validátorů.

- **test-validator**  
Jedná se o ukázkový příklad, určený jako vzor pro případné přidání nového typu validátoru. Neprovádí, žádnou validaci výslovnosti, pouze po každý vstup vrací konstantu udávající správnou výslovnost.
- **sphinx-word-validator**  
Jedná se o hlavní validátor výslovnosti postavený nad CMUSphinx. Tento validátor je použit pro všechny validace výslovnosti, tj. jak pro testy, tak pro mobilní aplikaci Speech Validator.

V následujících podkapitolách je popsána instalace jednotlivých validátorů.

### Instalace pro test-validator

Instalace validátoru `test-validator` je jednoduchá a slouží jako vzor pro instalaci `sphinx-word-validator`. Pro build C++ kódu do podoby použitelné pro Node.js je potřeba program `node-gyp`. Ten je vhodné mít nainstalovaný globálně.

```
sudo npm install node-gyp -g
```

Následně je potřeba nainstalovat zbylé závislosti.

```
cd validators/test-validator  
npm install
```

Nyní je vše připraveno pro build validátoru do podoby potřebné pro Node.js.

```
node-gyp configure  
node-gyp rebuild
```

Tím je celý build proces ukončen.

Validátor `test-validator` také obsahuje ukázkové testy. Pro jejich spuštění je potřeba programu `Mocha`. Ten je také vhodné mít také nainstalovaný globálně.

```
sudo npm install mocha -g
```

Nyní můžeme spustit test následujícím příkazem.

```
cd ../../server  
mocha ../validators/test-validator/test/ --timeout 720000
```

Je důležité, aby adresářem, v kterém se test spouští, byl adresář pro server (viz první příkaz `cd`). Také je vhodné nastavit dostatečně velkou časovou rezervu pomocí `timeout`. Pro tento konkrétní test to není nutné, ale jiné testy mohou zabrat hodně času a pokud trvají déle než nastavený interval, pak se nedokončí.

## Instalace nástrojů od CMUSphinx

Validátor sphinx-word-validator je závislý na nástrojích od CMUSphinx:

- Sphinxbase,
- Pocketsphinx,
- Sphinxtrain.

Všechny tyto nástroje je tedy potřeba lokálně nainstalovat. Jejich instalace má hodně závislostí, postup instalace je tedy hodně závislý na konkrétní distribuci operačního systému. Je vhodné postupovat dle oficiálního postupu Shmyrev (g) resp. GitHub, Inc. (2017d), GitHub, Inc. (2017c) a GitHub, Inc. (2017e).

Níže je uveden jednoduchý případ instalace balíčků:

- sphinxbase-5prealpha,
- pocketsphinx-5prealpha,
- sphinxtrain-5prealpha.

Nejdříve je potřeba mít nainstalovány následující závislosti gcc, automake, autoconf, libtool, bison, swig at least version 2.0, python development package, pulseaudio development package. Pro další postup budeme předpokládat, že všechny jsou již v operačním systému přítomny.

Ze začátku je potřeba z webových stránek Shmyrev (f) stáhnout všechny 3 zmíněné balíčky, rozbalit je a zkopírovat do pracovní složky. Následně stačí postupně ve všech složkách (ve stejném pořadí v jakém jsou výše uvedeny) spustit následující příkazy.

```
./autogen.sh
./configure
make
sudo make install
```

## Generování akustického modelu

Pro vygenerování akustického modelu je ve složce `validators/sphinx-word-validator/bin` připraven Node.js skript `amModelGenerator.js`.

Tento skript předpokládá že v daném operačním systému jsou nainstalovány nástroje od CMUSphinx:

- Sphinxbase,
- Sphinxtrain,

viz kap. 3.3.1. Pro správný průběh je potřeba mít v pracovní složce hlavní nástroje:

- `sphinx_fe` (ten je standardně dostupný z příkazového řádku, není ho tedy potřeba kopírovat do pracovní složky),

- `bw`,
- `mllr_solve`,
- `mllr_transform`

a další 2 pomocné:

- `map_adapt`,
- `mk_s2sendump`.

Tyto nástroje se nachází ve složce, kde je nainstalovaný Sphinxtrain, nejčastěji se jedná o složku `/usr/local/libexec/sphinxtrain`.

Při spuštění se tomuto skriptu zadává, jaký typ akustického modelu má vygenerovat. Lze mu zadat `ENUSCZSV`, což je adaptovaný akustický model.

```
cd validators/sphinx-word-validator/bin
node amModelGenerator.js --am ENUSCZSV
```

Výsledný adaptovaný akustický model se nachází ve složce `validators/sphinx-word-validator/bin/work/en-us-full`. Přenesením obsahu této složky do `validators/sphinx-word-validator/src/am/en-us-cz-speech-validator` je vše připraveno k tomu, aby mohl Speech Validator Server tento model začít používat.

## Generování fonetického slovníku a jazykového modelu

Pro vygenerování fonetického slovníku a jazykového modelu je ve složce `validators/sphinx-word-validator/bin` připraven Node.js skript `wordModelGenerator.js`.

Při spuštění je možné tomuto skriptu zadat následující parametry.

- `--dic`

Udává variantu fonetického slovníku.

Může nabývat hodnot:

- `ALL2`,
- `ALLN`,
- `CREATION2`,
- `CREATIONN`.

- `--lm`

Udává spolu s parametrem `--dic` variantu jazykového modelu.

Může nabývat hodnot:

- `PREF`,
- `5050`,
- `NORM`.

- `--accurate`

- `--inAccurate`

Tyto 2 parametry udávají jaké váhy mají být použity pro správnou a špatnou výslovnost.

Parametr má smysl zadávat pouze pro některé varianty `--lm` a to dle pravidel:

- PREF: hodnoty by měly být rozdílné a v intervalu od  $1e20$  do  $1e-20$ ,
- 5050: hodnoty by měly být stejné a v intervalu od  $1e20$  do  $1e-20$ .

Bez zadání těchto parametrů se vygenerují základní modely s váhami JSGF:

- $1e20$  a  $1e0$  pro PREF,
- $1e-15$  a  $1e-15$  pro 5050.

- `--allWeights`

Při nastavení na hodnotu `WEIGHTTEST` vygeneruje pro danou hodnotu `--lm` všechny potřebné varianty modelů pro otestování vah JSGF. Parametr má smysl zadávat pouze pro PREF a 5050.

Parametr by neměl být nastaven spolu s `--accurate` a `--inAccurate`.

Skript předpokládá že v je v pracovní složce dostupný skript `phonetic_transcription_cs.pl`, který je součástí frameworku ALEX. Tento skript je možné získat na webových stránkách GitHub, Inc. (2017b).

Níže je uveden příklad pro vygenerování fonetického slovníku CREATIONN a jazykového modelu CREATIONN-5050 s defaultními váhami JSGF  $1e-15$  a  $1e-15$ .

```
cd validators/sphinx-word-validator/bin
node wordModelGenerator.js --lm 5050 --dic CREATIONN
```

Výsledné modely se nachází ve složkách `validators/sphinx-word-validator/src/dic` a `validators/sphinx-word-validator/src/lm`, kde jsou rovnou připraveny pro použití v Speech Validator Server.

## Instalace pro sphinx-word-validator

Instalace validátoru `sphinx-word-validator` probíhá stejně jako u `test-validator`, viz kap. 3.3.1. Pouze je potřeba používat správnou složku, tj. `validators/test-validator` zaměnit za `validators/sphinx-word-validator`.

## Spuštění testů pro sphinx-word-validator

Validátor `sphinx-word-validator` obsahuje testy. Ty pokrývají:

- zjištění nejvhodnějších hodnot vah JSGF,
- zjištění nejvhodnějších hodnot parametru validátoru LW,
- otestování výsledků všech variant modelů validátoru:
  - ALL2-NORM,
  - ALLN-NORM,
  - CREATION2-NORM,

- CREATIONN-NORM,
- ALL2-5050,
- ALLN-5050,
- CREATION2-5050,
- CREATIONN-5050,
- ALL2-PREF,
- ALLN-PREF,
- CREATION2-PREF,
- CREATIONN-PREF.

Spuštění testů probíhá stejně jako u `test-validator`, viz kap. 3.3.1. Pouze je potřeba používat správnou složku, tj. `validators/test-validator` zaměnit za `validators/sphinx-word-validator`.

### **Spuštění aplikace**

Spuštění aplikace se provádí standardním způsobem pro Node.js.

```
cd server  
node bin/www
```

Případně je možné aplikaci spustit pomocí programu `gulp`.

## 4. Programátorská dokumentace

Systém se skládá ze 3 aplikací:

- Speech Validator Admin,
- Speech Validator,
- Speech Validator Server.

Speech Validator Admin a Speech Validator jsou Android aplikace. Některé části jejich kódu jsou společné. Resp. prvně vznikla aplikace Speech Validator Admin. Po její první verzi, která funguje zcela offline, bez potřeby internetu, se ze stejného kódu začala vyvíjet aplikace Speech Validator. Součástí kódu Speech Validator je tedy i předpřipravená verze pro Speech Validator Admin ve verzi 2.0, která by mohla být nad rámec této práce rozšířena i pro online chování. Tj. mohla by ze Speech Validator Server získávat informace o trénovaných hláskách a slovech a nasbírané nahrávky rovnou odesílat.

Zdrojový kód všech aplikací se nachází na CD v adresářích:

- `speech-validator-admin`,
- `speech-validator-m`,
- `speech-validator`.

Obě mobilní aplikace budou popsány ve společné podkapitole. Následně bude popsána i serverová aplikace Speech Validator Server.

### 4.1 Speech Validator Admin a Speech Validator

Speech Validator Admin a Speech Validator jsou Android aplikace naprogramované v Javě. Pro jejich build nejsou potřeba žádné nestandardní nástroje. Pro potřeby práce s kódem je možné pracovat s jakýmkoli nástrojem, který umí pracovat s build nástrojem Gradle.

Aplikace mají architekturu postavenou na Clean Architektuře a návrhovém vzoru MVC, o tomto se pojednává např. v Uncle Bob (2017). Struktura jejich kódu je rozdělena do následujících 3 základních modulů.

- `domain`

Tento modul obsahuje core business celé aplikace. Specifikuje hlavní datové struktury a operace s nimi (v terminologii kódu je označováno jako `UseCase`).

Kód není přímo závislý na žádné externí knihovně a je vytvořen v čisté Javě.

- `data`

Tento modul se stará o komunikaci s externími závislostmi. Provádí veškeré práce, které nejsou naprogramovány v čisté Javě. V tomto modulu se neřeší GUI.



Je zde řešena:

- případná komunikace se Speech Validator Server,
- případné ukládání a mazání záznamů z disku,
- nahrávání a přehrávání výslovnosti.

Pro komunikaci přes internet je zde použita knihovna Retrofit.

- presentation-collector resp. presentation

Tyto moduly obsahují veškerý potřebný kód pro GUI. Využívá se zde návrhového vzoru MVC.

Aplikace Speech Validator používá konfigurační soubor. Jeho příklad je uveden v souboru `gradle.properties.example`. Tento soubor je potřeba zkopírovat a přejmenovat na `gradle.example`. Je v něm možné nakonfigurovat následující parametry.

- `API_URL`

Tento parametr udává adresu, na které se snaží aplikace komunikovat se Speech Validator Server. Defaultně je aplikace nastavena pro komunikaci na adrese `http://speechvalidator.localnet:3000/`.

- `API_CACHE_SIZE`
- `API_CACHE_MAX_AGE`
- `API_CACHE_MAX_STALE`

Aplikace obsahuje offline cache, do které se načítají všechny odpovědi ze Speech Validator Server. Toho se dá prozatím využít pouze pro offline prohlížení načtených hlásek a jejich slov. Pro samotné provedení cvičení resp. odeslání zvukového záznamu do Speech Validator Server je internet potřeba vždy. Tato funkcionality je zde připravena pro možné rozšíření aplikace o uživatelsky atraktivnější typy samostatného cvičení.

Těmito parametry se nastavuje zmíněná cache.

- `API_TIMEOUT`
- `API_OFFLINE_TIMEOUT`

Těmito parametry se nastavuje doba timeoutu při komunikaci se Speech Validator Server.

## 4.2 Speech Validator Server

Speech Validator Server je backendová, serverová aplikace naprogramovaná v Node.js a C++. Její build, build validátorů výslovnosti, zprovoznění skriptů pro generování modelů a spouštění testů je popsáno v kap. 3.3.

V rámci této kapitoly jsou popsány některé implementační detaily.

Kód je rozdělen do 2 základních složek: `server` a `validators`. Ve složce `server` se nachází veškerý kód REST serveru. Ve složce `validators` se nachází veškerý kód validátorů výslovnosti.

Aplikace využívá databáze PostgreSQL, která běží v Docker kontejneru. V databázi jsou uloženy následující data.

- Trénované hlásky, tabulka `speechSound`  
U každé hlásky je evidován její název a fonémy, které se procvičují.
- Trénovaná slova v tabulce `speechWord`  
U každého slova je evidován pouze jeho název.
- Záznamy o nahrávkách určených pro trénování a testování, tabulka `speechWordRecord`  
U každého záznamu o nahrávce se eviduje, zdali se jedná o správnou výslovnost a zdali byla tato nahrávka použita pro potřeby trénování.
- Záznamy o validacích, tabulka `speechWordValidation`  
Všechny validace a nahrávky se ukládají pro případné potřeby dalšího trénování a testování validátorů výslovnosti. U každého záznamu o validaci se eviduje, zdali daná nahrávka byla detekována jako správná výslovnost.
- nahrávky, tabulka `record`  
Zde jsou fyzicky uloženy všechny nahrávky. V ostatních tabulkách je pouze odkaz na tuto.

Pro každou tabulku existuje její stejnojmenný model. Ten se nachází ve složce `server/models`. Tyto modely se používají pro abstrakci databáze na úrovni entit. V zbytku kódu se díky tomu pracuje s těmito entitami a nemusí se vytvářet SQL dotazy.

Ve složce `server/routes` se nachází všechny routy REST API. Jedná se o 2 routy, první pro operace s hlásky `speechSound.js`, druhá pro slova `speechWord.js`. Na routě pro slova se provádí i validace výslovnosti.

Kód výsledného validátoru výslovnosti se nachází ve složce `validators/sphinx-word-validator`. Rozhraní validátoru je naprogramováno v Node.js. v souboru `validator.js`. Ten volá C++ kód v souboru `validator.cpp`. Díky předpřipravenému společnému C++ kódu je možné validátor programovat jako klasický C++ kód. Kód validátoru se nachází v souboru `SphinxWordSpeechValidator.cpp` pod složkou `src`. Validátor musí dědit od abstraktního předka `SpeechValidator`, který udává rozhraní pro následující metodu.

```
ValidationResult validate(ValidationInput* input)
```

Společný kód pro všechny validátory výslovnosti se nachází ve složce `validators/src`. Jsou zde nadefinovány vstupní a výstupní parametry pro validaci, základní třída `AddonValidator` pro propojení JavaScriptu s C++ a již dříve zmíněná třída `SpeechValidator`.

## 5. Analýza použitelnosti Systému

Z důvodu korektnosti Systému jsou hlavní požadavky na Systém kladeny na korektnost validátoru výslovnosti, resp. na jeho správné otestování.

Hlavní požadavky na zbylé části Systému (tj. mimo validátor výslovnosti) jsou následující.

- Při výběru cvičných textů nebude kladen důraz na jejich obsahovou stránku, ale na vhodnou kombinaci hlásek, z kterých se skládají.

Toto je splněno výběrem vhodných hlásek a slov, viz kap. 2.1.3.

- Forma cvičení bude vyřčení zobrazeného cvičného slova.

Toto je splněno ve výsledné aplikaci Speech Validator (resp. i v odpovídající části Speech Validator Server), viz kap. 3.2.

- Při zobrazení zpětné vazby bude kladen důraz na jednoznačnost výsledného hodnocení.

Toto je splněno ve výsledné aplikaci Speech Validator, viz kap. 3.2.

- Z důvodu atraktivního Systému bude možné jednoduše měnit cvičný text.

Toto je splněno v Speech Validator Server. Ten má všechna data specifikována v databázi. Změnou dat se změní i seznam hlásek a slov uvedený v Speech Validator.

- Část Systému zodpovědná za hodnocení správné výslovnosti nebude závislá na zvolené formě cvičení a zobrazení zpětné vazby.

Toto je splněno jednak zvolenou architekturou celého Systému (tj. rozdělení na frontend Speech Validator, který udává formu cvičení, a backend Speech Validator Server, který provádí validaci výslovnosti), jednak architekturou Speech Validator, díky které je možné jednoduše provádět změny ve cvičení, aniž by se tím ovlivnila komunikace (resp. validace) se Speech Validator Server viz kap. 3.2.

Tyto požadavky jsou považovány za splněné.

Požadavky kladené na validátor výslovnosti jsou následující.

1. Systém bude rozpoznávat pouze izolovaná slova.

Toto je splněno výběrem slov pro samostatná cvičení a způsobem vytváření jazykového modelu, který vždy obsahuje pouze jedno slovo s jeho správnou a špatnými výslovnostmi, viz kap. 1.2.4, kap. 2.1.3.

2. Systém bude na řečníku nezávislý.

Pro validátor výslovnosti je vytvořen adaptovaný akustický model z anglického kap. 2.1.3. Původní anglický slovník je na řečníku nezávislý a adaptace probíhala na nahrávkách od více řečníků nasbíraných aplikací Speech Validator Admin. Takto vytvořený slovník by měl být na řečníku nezávislý.

3. Systém bude využívat klasifikátor řeči využívající statistické metody a to CMUSphinx.

Toto je jednoduše splněno použitím CMUSphinx.

4. Systém bude pro CMUSphinx adaptovat nějaký již existující akustický model.

Toto je splněno adaptací anglického akustického modelu, viz kap. 2.1.3.

5. Systém bude pro CMUSphinx generovat fonetický slovník ve variantách:  
- ALL,  
- CREATION.

Systém generuje 4 typy fonetických slovníků a to ALL2, ALLN, CREATION2 a CREATIONN, viz kap. 2.1.3.

6. Systém bude pro CMUSphinx generovat jazykový model ve variantách:  
- NORM,  
- 5050,  
- PREF.

Systém generuje 12 typů jazykových modelů a to ALL2-NORM, ALLN-NORM, CREATION2-NORM, CREATIONN-NORM, ALL2-5050, ALLN-5050, CREATION2-5050, CREATIONN-5050, ALL2-PREF, ALLN-PREF, CREATION2-PREF, CREATIONN-PREF, viz kap. 2.1.3.

7. Systém bude obsahovat testy pro zjištění přesnosti validátoru výslovnosti a porovnání jeho jednotlivých variant (resp. variant jeho modelů).

Tento bod je splněn v následující podkapitolách.

V kap. 2.1.3 byl navíc přidán jeden požadavek na výběr vhodných vah JSGF a parametru LW. Odpovídající testy pro tento výběr i testy výsledného validátoru výslovnosti se nacházejí v následujících podkapitolách.

## 5.1 Testy pro výběr vah JSGF

V kap. 2.1.3 je popsána povaha vah JSGF a byl navržen test pro vybrání nejvhodnějších vah pro jazykové modely. V rámci tohoto bylo testováno 6 slov:

- vana (ve výsledcích uváděno jako VANA),
- činka (ve výsledcích uváděno jako CINKA),
- liška (ve výsledcích uváděno jako LISKA),
- dříví (ve výsledcích uváděno jako DRIVI),
- voní (ve výsledcích uváděno jako VONI),
- magi (ve výsledcích uváděno jako MAGI).

Pro tato slova byly provedeny testy s použitím fonetického slovníku CREATIONN a 2 jazykových modelů CREATIONN-PREF a CREATIONN-5050.

V kap. 2.1.3 je uvedena charakteristika vah JSGF s tím, že by se pro ně měly používat hodnoty vyšších řádů. Testy jsou prováděny v intervalech hodnot od  $1e20$  do  $1e-20$ . Rozdíly mezi správnou výslovností a špatnou jsou po skocích o 5 řádů. Navíc jsou pro zajímavé intervaly dodatečně provedeny testy s rozdíly po skocích o 1 řád.

Pro CREATIONN-PREF byly testovány dvojice vah JSGF (první je vždy uvedena hodnota pro správnou výslovnost a druhá pro špatnou):

- $1e20, 1e15,$
- $1e20, 1e10,$
- $1e20, 1e5,$
- $1e20, 1e4,$
- $1e20, 1e3,$
- $1e20, 1e2,$
- $1e20, 1e1,$
- $1e20, 1e0,$
- $1e20, 1e-10,$
- $1e20, 1e-15,$
- $1e20, 1e-20,$
- $1e10, 1e5,$
- $1e10, 1e3,$
- $1e10, 1e2,$
- $1e10, 1e0,$
- $1e10, 1e-5,$
- $1e10, 1e-10,$
- $1e10, 1e-15,$
- $1e10, 1e-20,$
- $1e5, 1e0,$
- $1e5, 1e-10,$
- $1e5, 1e-15,$
- $1e5, 1e-20,$

- $1e0, 1e - 5,$
- $1e0, 1e - 10,$
- $1e0, 1e - 15,$
- $1e0, 1e - 20,$
- $1e - 5, 1e - 10,$
- $1e - 5, 1e - 15,$
- $1e - 5, 1e - 20,$
- $1e - 10, 1e - 15,$
- $1e - 10, 1e - 20,$
- $1e - 15, 1e - 20,$
- $1e1, 1e - 1,$
- $1e2, 1e - 2,$
- $1e3, 1e - 3,$
- $1e4, 1e - 4,$
- $1e5, 1e - 5,$
- $1e6, 1e - 6,$
- $1e7, 1e - 7.$

Pro CREATIONN-5050 byly testovány váhy JS GF:

- $1e20, 1e20,$
- $1e15, 1e15,$
- $1e10, 1e10,$
- $1e5, 1e5,$
- $1e - 5, 1e - 5,$
- $1e - 10, 1e - 10,$
- $1e - 15, 1e - 15,$
- $1e - 20, 1e - 20,$
- $1e4, 1e4,$
- $1e3, 1e3,$
- $1e2, 1e2,$

- $1e1, 1e1,$
- $1e0, 1e0,$
- $1e - 1, 1e - 1,$
- $1e - 2, 1e - 2,$
- $1e - 3, 1e - 3,$
- $1e - 4, 1e - 4.$

Výsledky testů pro variantu CREATIONN-5050 jsou uvedeny v tab. C1. V tabulce jsou pro každé slovo uvedeny 3 sloupce. První udává u kolika procent nahrávek správné výslovnosti byla validátorem řeči detekována správná výslovnost. Druhý obdobně udává u kolika procent nahrávek špatné výslovnosti byla validátorem řeči detekována špatná výslovnost. Poslední je spočítaný průměr z dvou předchozích.

Z výsledků je patrné, že tato varianta nedává pro zvolená slova správné výsledky. U všech slov krom vana vyšla 100% úspěšnost při validaci špatné výslovnosti a 0% úspěšnost při validaci správné výslovnosti. Jinými slovy, výsledky z validace byly vždy, že se jedná o špatnou výslovnost.

Jediné slovo, podle kterého můžeme zvolit váhu JSGF je vana. U něj bylo nejlepších průměrných výsledků dosaženo při hodnotě  $1e - 20$ , přičemž k detekci správné výslovnosti došlo pouze v 25%. K nejlepší detekci správné výslovnosti došlo při hodnotě  $1e - 15$  a to v 37,5%. Průměrný výsledek pro tuto hodnotu je 49%, což je druhý nejlepší.

Výsledky testů pro variantu CREATIONN-PREF jsou uvedeny v tab. C2. V tabulce jsou pro každé slovo uvedeny 3 sloupce. Významy jednotlivých sloupců jsou totožné jako u předchozího testu.

Na konci tabulky je uvedeno shrnutí napříč všemi slovy. Shrnutí opět obsahuje 3 sloupce. V sloupcích pro shrnutí je červeně zvýrazněna každá hodnota menší než 50%.

Nejlepších průměrných výsledků 55.17% je dosaženo při dvojici hodnot  $1e20$  a  $1e - 20$ , přičemž k detekci špatné výslovnosti došlo pouze v 12.72%. Tato varianta jednoznačně až příliš upřednostňuje správnou výslovnost. K obdobně vysoké průměrné přesnosti (tj. 51.88%) došlo u hodnot  $1e20$  a  $1e0$ . Jedná se o nejvyšší průměrnou hodnotu, u které je detekce správné výslovnosti nad 50% (tj. 70.6%) a současně došlo k nárůstu špatné detekce na 33,16%. Obdobný charakter s horším průměrem je dosaženo u hodnot  $1e20$  a  $1e1$ . Ve zbylých hodnotách vah JSGF dochází k příliš velkému upřednostňování špatné nebo správné výslovnosti.

Jako výsledek z těchto testů byla pro:

- CREATIONN-5050 zvolena hodnota vah JSGF  $1e - 15$  a to pro nejvyšší detekci správné výslovnosti,
- CREATIONN-PREF zvolena dvojice hodnot vah JSGF  $1e20$  a  $1e0$  a to pro rozumný kompromis mezi detekci správné a špatné výslovnosti se zachováním jednoho z nejvyšších průměrů.

## 5.2 Testy pro výběr parametru LW

V kap. 2.1.3 je popsán význam parametru LW a byl zde navržen test pro vybrání nejvhodnější hodnoty. V rámci tohoto bylo testováno 6 slov. Jedná se o stejná slova jako u předchozího testu, viz kap. 5.1.

Pro tato slova byly provedeny testy s použitím fonetického slovníku CREATIONN a 2 jazykových modelů CREATIONN-PREF a CREATIONN-5050. Tyto jazykové modely měly nastaveny váhy JSGF dle navržených na základě předchozího testu, viz kap. 5.1.

Pro parametr LW je doporučeno nejčastěji nastavovat hodnoty v rozmezí od 6 do 20, přičemž základní přednastavená hodnota je 6.5. Testy byly provedeny pro hodnoty od 6 do 20 po skocích o velikosti 1. Navíc je proveden test pro přednastavenou hodnotu 6.5.

Výsledky testů pro obě varianty CREATIONN-PREF i CREATIONN-5050 jsou uvedeny v tab. C3. V tabulce jsou pro každé slovo uvedeny 3 sloupce. Významy jednotlivých sloupců jsou totožné jako u předchozích testů.

Na konci tabulky je uvedeno shrnutí napříč všemi slovy. Shrnutí opět obsahuje 3 sloupce. V sloupcích pro shrnutí je červeně zvýrazněna každá hodnota menší než 50%.

Pro variantu CREATIONN-PREF vyšly nejlepší výsledky pro hodnotu parametru LW 18 a to:

- 52.86% pro detekci správné výslovnosti,
- 60.53% pro detekci špatné výslovnosti,
- 56.69% pro průměr.

Pro variantu CREATIONN-5050 vyšly nejlepší průměrné výsledky 51.19% pro hodnotu parametru LW 18. Detekce správné výslovnosti je ale pouhých 2.38%, což je problém napříč všemi výsledky. Nejlepší detekci správné výslovnosti 6.25% je dosaženo při LW 9 a současně se jedná o druhý nejlepší průměrný výsledek 50.63%.

Jako výsledek z těchto testů byla pro:

- CREATIONN-5050 zvolena hodnota parametru LW 18 a to pro nejvyšší průměrnou detekci,
- CREATIONN-PREF zvolena hodnota parametru LW 9 a to pro nejvyšší detekci správné výslovnosti se zachováním jednoho z nejvyšších průměrů.

## 5.3 Testy modelů

Na závěr byly provedeny testy na všech 4 vygenerovaných fonetických slovnících:

- ALL2,
- ALLN,
- CREATION2,



- CREATIONN,

resp. na všech 12 vygenerovaných jazykových modelech:

- ALL2-NORM,
- ALLN-NORM,
- CREATION2-NORM,
- CREATIONN-NORM,
- ALL2-5050,
- ALLN-5050,
- CREATION2-5050,
- CREATIONN-5050,
- ALL2-PREF,
- ALLN-PREF,
- CREATION2-PREF,
- CREATIONN-PREF.

Jazykové modely měly nastaveny váhy JSGF dle navržených na základě předchozího testu, viz kap. 5.1. Současně parametr LW byl také nastaven na základě předchozího testu, viz kap. 5.2.

Výsledky testů pro všechny varianty 5050 a NORM jsou uvedeny v tab. C6, tab. C7, tab. C8 a tab. C9.

Pro každé slovo jsou uvedeny 3 sloupce. Významy jednotlivých sloupců jsou totožné jako u předchozích testů.

Každé trojici buněk, které odpovídají 1 slovu a 1 variantě modelu, je přiřazena červená barva, pokud správná a špatná výslovnost nebyla současně detekována s přesností větší než 50%. Díky tomu je patrné, že pro všechna slova bylo dosaženo nepoužitelných výsledků.

Výsledky testů pro všechny varianty PREF jsou uvedeny v tab. C4, tab. C5. Tabulka má dvě části, levou a pravou.

V levé části jsou uvedena všechna slova a jejich výsledky pro jednotlivé typy modelů. Pro každé slovo jsou uvedeny 3 sloupce. Významy jednotlivých sloupců jsou totožné jako u předchozích testů.

Pod tabulkou se nachází řádek se shrnutím výsledků napříč všemi slovy. Je zde červeně zvýrazněna každá hodnota menší než 50%.

Nejlepších průměrných hodnot 51.61% bylo dosaženo při variantě modelu CREATION2-PREF. Detekce správné výslovnosti je však pod 50% a to 41.58%.

Druhého nejlepšího průměrného výsledku 51.21% bylo dosaženo při variantě modelu CREATIONN-PREF. Zde je o trochu vyšší detekce správné výslovnosti 44.27%.

Z těchto výsledků je patrné, že žádný z modelů není pro validaci výslovnosti přijatelný.

V porovnání mezi modely dosahuje nejlepších výsledků způsob generování výslovnosti ve variantě CREATION.

Do pravé části popisované tabulky vstupují pouze ty výsledky, u kterých je validace správné i špatné výslovnosti alespoň 50%. Je patrné, že takovýchto slov je minimum. Na posledním řádku tabulky je uveden jejich počet pro každý typ modelu zvlášť. Jedná se o čísla 6, 6, 8, 9.

Z tohoto vyplývá, že u většiny slov je detekce správné resp. špatné výslovnosti s navrženým postupem nemožná. U zbylých 6 - 9 slov se dosahuje průměrné úspěšnosti 67.14% až 72.97%, což je stále pro logopedickou aplikaci nepoužitelný výsledek.

## 5.4 Shrnutí výsledků

Celkem byly provedeny 3 skupiny testů.

V rámci 1. z nich byly navrženy hodnoty vah JSGF modelů, při kterých bylo dosaženo nejlepších výsledků, a to:

- $1e - 15$  pro CREATIONN-5050,
- $1e20$  a  $1e0$  pro CREATIONN-PREF.

V rámci 2. z nich byla navržena hodnota parametru LW, při které bylo dosaženo nejlepších výsledků, a to:

- LW 18 pro CREATIONN-5050,
- LW 9 CREATIONN-PREF.

S těmito hodnotami byla provedena poslední skupina testů za účelem zjištění, zdali je navržený validátor výslovnosti dostatečný. V rámci těchto testů byly současně porovnání výsledků dosažené s jednotlivými typy modelů.

Výsledkem tohoto testu je, že validátor výslovnosti není dostatečný a není vhodné ho používat pro logopedické aplikace. Nejlepších průměrných výsledků v rozmezí úspěšnosti 67.14% až 72.97% bylo dosaženo při výběru pouhých 6 - 9 slov.

V rámci kap. 1 byla popsána všechna hlavní úskalí navrženého validátoru a případné varianty odlišného přístupu. V následující kapitole jsou tedy shrnuty možná rozšíření této práce, která by mohla vést k lepším výsledkům validace výslovnosti.

## 5.5 Možnosti rozšíření

Důvodem proč tato práce není vhodná pro logopedické aplikace je nepřesná validace výslovnosti. V této kapitole jsou shrnuty možnosti, které by mohly vést k lepším výsledkům.

Celý Systém byl navržen tak, aby byl jednoduše rozšiřitelný. V Systému je tedy např. možné zaměnit výsledný validátor výslovnosti za jiný, aniž by se tím ovlivnil zbytek Systému (krom kvality validace výslovnosti). Pokud by se připravil nad rámec této práce nový validátor výslovnosti, mělo by být jednoduché jeho zapojení do Systému.

V kap. 1.2.1 jsou popsány 4 možné varianty validátoru výslovnosti. Jedna z těchto variant byla naprogramována v rámci této práce (tj. využití klasifikátoru využívajícího statistické metody s generováním špatné výslovnosti). Je tedy vhodné vyzkoušet i ostatní možnosti. Jako první v pořadí se nabízí využití stejného typu klasifikátoru využívajícího statistické metody s nastavením prahu. Tento typ má obdobné výhody jako aktuální řešení, navíc pro jeho vytvoření může posloužit aktuální implementace jako vhodný základ. Možným důvodem proč aktuální řešení nedosahuje správných výsledků je nepotvrzení jednoho z předpokladů a to že umíme vygenerovat zápis špatné výslovnosti v podobě fonémů. Varianta s prahem nemá tento předpoklad (tj. negeneruje žádnou špatnou výslovnost).

Není třeba rovnou přecházet na jiný typ řešení. Je možné ještě vyzkoušet několik změn v aktuálním přístupu. Prvním z nich je vytvoření kvalitnějšího akustického modelu. V rámci této práce byl pouze adaptován anglický akustický model (viz kap. 2.1.3). Je možné, že špatných výsledků bylo dosaženo pro velký rozdíl mezi anglickými a českými fonémy a současně použití nedostatku nahrávek pro adaptaci. Nabízejí se zde 2 varianty.

- Vytvořit nový akustický model na základě nasbíraných dat aplikací Speech Validator Admin.  
Zde ale není jisté jestli je nasbíraný počet nahrávek dostatečný.
- Vytvořit český akustický model na základě nějakého z dostupných korpusů (obsahující nahrávky a jejich transkripci) a jeho následnou adaptaci pomocí nahrávek nasbíraných aplikací Speech Validator Admin.  
Zde zůstává otázkou, zdali by se v korpusu vyskytovaly nahrávky i se špatnou výslovností a jaký to mělo případný dopad na výsledky validace.

Další možností je přizpůsobení validátoru hlasu řečníka za dozoru logopeda, který by kontroloval, zdali bylo slovo vysloveno správně (viz kap. 1.2.1). Vzhledem k míře špatných výsledků není očekávané, že by tento krok mohl mít za následek až tak velké zlepšení. Ve spojení s některou z výše navržených změn se ale jeví jako vhodné zlepšení.

Pokud by se podařilo zlepšit validaci výslovnosti, jako dalším vhodným krokem je zatraktivnit formu cvičení v Speech Validator aplikaci, tak aby dostatečně motivovala Uživatele k provedení cvičení (viz důvod atraktivního Systému).

# Závěr

Tato práce má následující 3 cíle.

- Navrhnout způsob validace výslovnosti na základě současných algoritmů pro rozpoznávání mluvené řeči.
- Realizovat navržený validátor výslovnosti.  
Validátor výslovnosti bude součástí Systému. Systém je určen jako podpůrný prostředek pro předávání potřebných materiálů a zefektivnění samostatného cvičení pomocí zpětné vazby při trénování výslovnosti. Ze zpětné vazby by měl Uživatel poznat, zdali trénované slovo vyslovil správně.
- Vyhodnotit použitelnost validátoru výslovnosti.

V rámci kap. 1 je navržen způsob validace výslovnosti na základě současných algoritmů pro rozpoznávání mluvené řeči. Dále byl návrh rozpracován v kap. 2. Navržený validátor byl následně spolu s celým Systémem realizován. Výsledkem jsou 3 aplikace Speech Validator Admin pro sběr dat, frontend Speech Validator a backend Speech Validator Server, který obsahuje validátor výslovnosti. V závěru práce byly provedeny testy, jejichž výsledky jsou popsány v kap. 5. Tímto byly všechny cíle práce splněny.

Jak ukazují výsledky testů (viz kap. 5.4), výsledný validátor není použitelný pro potřeby logopedických aplikací. Nejlepších průměrných výsledků rozpoznání správné resp. špatné výslovnosti 72.97% bylo dosaženo pouze pro několik vybraných slov (6). Nejlepší průměrné výsledky rozpoznání správné resp. špatné výslovnosti všech navržených slov se pohybovali mírně nad 50%. V rámci kap. 5.5 byla navržena různá rozšíření této práce, která by mohla vést k zlepšení výsledků validátoru výslovnosti.

# Seznam použité literatury

- Arpabet - wikipedia. URL <https://en.wikipedia.org/wiki/Arpabet>. [Online; poslední úpravy 1.6.2017; cit. 2017-07-17].
- CARNEGIE SPEECH COMPANY (2001-2017). Nativeaccent® | products | carnegie speech. URL <http://www.carnegiespeech.com/products/nativeaccent.php>. [Online; cit. 2017-07-16].
- ECLI (2017). Learn english | pronunciation power software. URL <http://www.englishlearning.com/>. [Online; cit. 2017-07-16].
- GITHUB, INC. (2017a). Ufal-dsg/alex: Alex dialogue systems framework. URL <https://github.com/UFAL-DSG/alex>. [Online; cit. 2017-07-17].
- GITHUB, INC. (2017b). alex/phonetic\_transcription\_cs.pl at 72fd963c16e00adea6b8fb6c45441b33fc725f3c · ufal-dsg/alex. URL [https://github.com/UFAL-DSG/alex/blob/72fd963c16e00adea6b8fb6c45441b33fc725f3c/alex/tools/kaldi/local/phonetic\\_transcription\\_cs.pl](https://github.com/UFAL-DSG/alex/blob/72fd963c16e00adea6b8fb6c45441b33fc725f3c/alex/tools/kaldi/local/phonetic_transcription_cs.pl). [Online; cit. 2017-07-17].
- GITHUB, INC. (2017c). pocketsphinx/readme at master · cmusphinx/pocketsphinx. URL <https://github.com/cmusphinx/pocketsphinx/blob/master/README>. [Online; cit. 2017-07-18].
- GITHUB, INC. (2017d). sphinxbase/readme at master · cmusphinx/sphinxbase. URL <https://github.com/cmusphinx/sphinxbase/blob/master/README>. [Online; cit. 2017-07-18].
- GITHUB, INC. (2017e). sphinxtrain/readme at master · cmusphinx/sphinxtrain. URL <https://github.com/cmusphinx/sphinxtrain/blob/master/README>. [Online; cit. 2017-07-18].
- GÚTHOVÁ, M. a ŠEBIANOVÁ, D. (2005). *Terapie narušené komunikační schopnosti*. 1. vydání. Portál, Praha. ISBN 80-7178-961-5.
- HUNT, A. (2000). Jspeech grammar format. URL <https://www.w3.org/TR/jsgf/>. [Online; cit. 2017-07-16].
- LECHTA, V. (2003). *Klinická logopedie*. 1. vydání. Portál, Praha. ISBN 80-7178-546-6.
- NEUBAUER, K. (2010). *Logopedie učební text pro bakalářské studium speciální pedagogiky*. 3. vydání. GAUDEAMUS, Hradec Králové. ISBN 978-80-7435-053-5.
- PAVEL, L. (2012). Fonetická transkripce českého jazyka. URL [https://www.vutbr.cz/www\\_base/zav\\_prace\\_soubor\\_verejne.php?file\\_id=55781](https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=55781).
- PMQ SOFTWARE (2013-2016). Informace pro logopedy | pmq česky. URL <http://www.pmq-software.com/sw/cz/informace-pro-logopedy/>. [Online; cit. 2017-07-16].

- PSUTKA, J. (1995). *Komunikace s počítačem mluvenou řečí*. 1. vydání. Academia, Praha. ISBN 80-200-0203-0.
- PSUTKA, J. E. A. (2006). *Mluvíme s počítačem česky*. 1. vydání. Academia, Praha. ISBN 80-200-1309-1.
- SALOMONOVÁ, A. (2003). *Klinická logopedie*. 1. vydání. Portál, Praha. ISBN 80-7178-546-6.
- SHERIF, M. A. E. A. (2006). Computer aided pronunciation learning system using speech recognition techniques. URL <http://www.cs.toronto.edu/~asamir/papers/is06.pdf>. [Online; poslední úpravy 28.8.2006; cit. 2017-07-16].
- SHMYREV, N. Cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/>. [Online; cit. 2017-07-15].
- SHMYREV, N. Adaptation of the acoustic model – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/tutorialadapt/>. [Online; cit. 2017-07-16].
- SHMYREV, N. Training acoustic model – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/tutorialam/>. [Online; cit. 2017-07-16].
- SHMYREV, N. Adaptation of the acoustic model – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/tutorialadapt/>. [Online; cit. 2017-07-16].
- SHMYREV, N. Basic concepts of speech recognition – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/tutorialconcepts/>. [Online; cit. 2017-07-19].
- SHMYREV, N. Downloads – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/download/>. [Online; cit. 2017-07-18].
- SHMYREV, N. Pocketsphinx tutorial – cmusphinx open source speech recognition. URL <https://cmusphinx.github.io/wiki/tutorialpocketsphinx/#installation>. [Online; cit. 2017-07-18].
- SLASHDOT MEDIA (2017). Cmu sphinx - browse /acoustic and language models/us english at sourceforge.net. URL <https://sourceforge.net/projects/cmusphinx/files/Acoustic%20and%20Language%20Models/US%20English/>. [Online; cit. 2017-07-16].
- SPEECH-LANGUAGEAPPS.COM (2017). articulation << speech-languageapps.com. URL <http://speech-languageapps.com/articulation/>. [Online; cit. 2017-07-16].
- SRI INTERNATIONAL (2016). Eduspeak® speech recognition toolkit | sri international. URL <http://www.sri.com/engage/products-solutions/eduspeak-speech-recognition-toolkit>. [Online; cit. 2017-07-16].

- SRIKANTH, R., BO, L. a SALSMAN, J. (2012). Automatic pronunciation evaluation and mispronunciation detection using cmusphinx. URL <http://www.aclweb.org/anthology/W12-5808>. [Online; poslední úpravy 17.2.2013; cit. 2017-07-16].
- TALKIEBEAR S.R.O. Talkiebear.com. URL <http://talkiebear.com/>. [Online; cit. 2017-07-16].
- TEAM, H. Htk speech recognition toolkit. URL <http://htk.eng.cam.ac.uk/>. [Online; cit. 2017-07-15].
- UNCLE BOB (2017). The clean architecture | 8th light. URL <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>. [Online; poslední úpravy 13.8.2012; cit. 2017-07-16].
- VIRTUAL SPEECH CENTER, INC. (2017). Apps for speech and language therapy. URL <https://www.virtualspeechcenter.com/MobileApps.aspx>. [Online; cit. 2017-07-16].
- ÚZIS ČR (2010-2017). Logopedie | ÚZIS ČR. URL <http://www.uzis.cz/category/tematicke-rady/zdravotnicka-statistika/logopedie>. [Online; cit. 2017-07-14].
- ÚZIS ČR (2012). Aktuální informace č. 18/2012. URL [http://www.uzis.cz/system/files/ai\\_2014\\_13\\_0.pdf](http://www.uzis.cz/system/files/ai_2014_13_0.pdf).
- ÚZIS ČR (2013). Aktuální informace č. 17/2013. URL [http://www.uzis.cz/system/files/ai\\_2014\\_13\\_0.pdf](http://www.uzis.cz/system/files/ai_2014_13_0.pdf).
- ÚZIS ČR (2014). Aktuální informace č. 13/2014. URL [http://www.uzis.cz/system/files/ai\\_2014\\_13\\_0.pdf](http://www.uzis.cz/system/files/ai_2014_13_0.pdf).

# Seznam obrázků

B1	Speech Validator Admin Uvítací obrazovka . . . . .	63
B2	Speech Validator Admin obrazovka Výběr hlásky . . . . .	64
B3	Speech Validator Admin obrazovka Výběr slova . . . . .	64
B4	Speech Validator Admin obrazovka Nahrávky . . . . .	65
B5	Speech Validator Admin obrazovka Nahrávka . . . . .	65
B6	Speech Validator Uvítací obrazovka . . . . .	66
B7	Speech Validator obrazovka Výběr hlásky . . . . .	66
B8	Speech Validator obrazovka Výběr slova . . . . .	67
B9	Speech Validator obrazovka Cvičení . . . . .	67



# Seznam tabulek

1.1	Rozdělení českých souhlásek . . . . .	12
2.1	Seznam slov . . . . .	30
2.2	Počet nahrávek . . . . .	31
2.3	Počet nahrávek pro adaptaci . . . . .	32
2.4	Transformace ČFA na Arpabet . . . . .	33
2.5	Počet nahrávek pro testování . . . . .	34
A1	Zastoupení dyslálie . . . . .	62
C1	Výsledky testů pro výběr vah JSGF CREATIONN-5050 . . . . .	68
C2	Výsledky testů pro výběr vah JSGF CREATIONN-PREF . . . . .	69
C3	Výsledky testů pro výběr parametru LW . . . . .	70
C4	Výsledky testů pro PREF 1/2 . . . . .	71
C5	Výsledky testů pro PREF 2/2 . . . . .	72
C6	Výsledky testů pro 5050 1/2 . . . . .	73
C7	Výsledky testů pro 5050 2/2 . . . . .	74
C8	Výsledky testů pro NORM 1/2 . . . . .	75
C9	Výsledky testů pro NORM 2/2 . . . . .	76

# Přílohy

**Příloha A: Procentuální zastoupení dyslálie ve statistických výkazech o činnosti logopedických pracovišť u osob do 18 let**

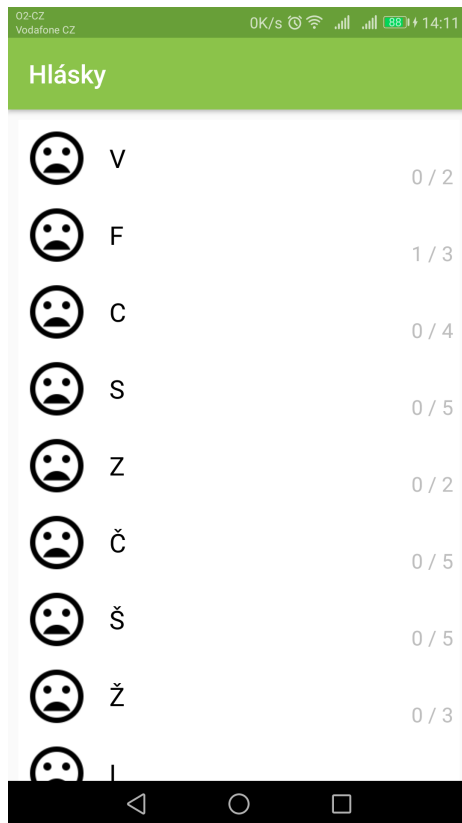
<b>Rok</b>	<b>Procentuální zastoupení [%]</b>
2013	48,4%
2012	48,8%
2011	49,1%

Tabulka A1: Zastoupení dyslálie

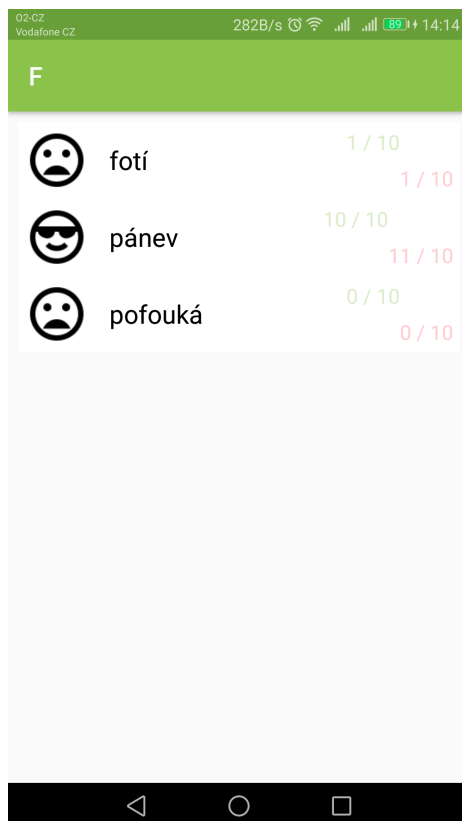
## Příloha B: Obrazovky



Obrázek B1: Speech Validator Admin Uvítací obrazovka



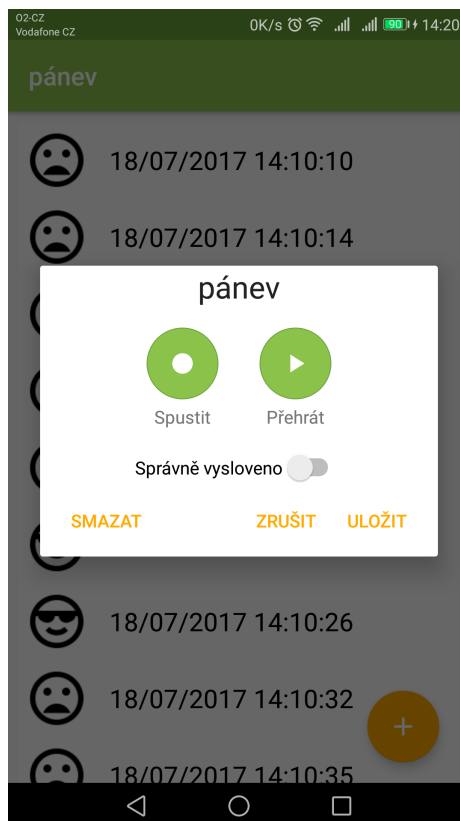
Obrázek B2: Speech Validator Admin obrazovka Výběr hlásky



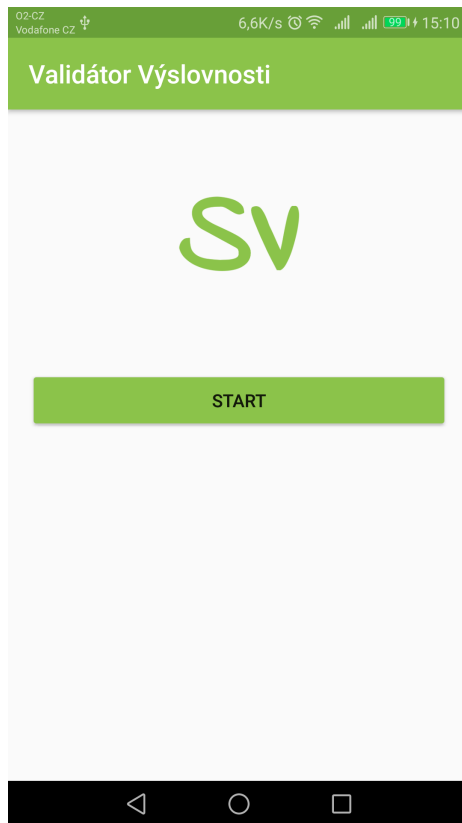
Obrázek B3: Speech Validator Admin obrazovka Výběr slova



Obrázek B4: Speech Validator Admin obrazovka Nahrávky



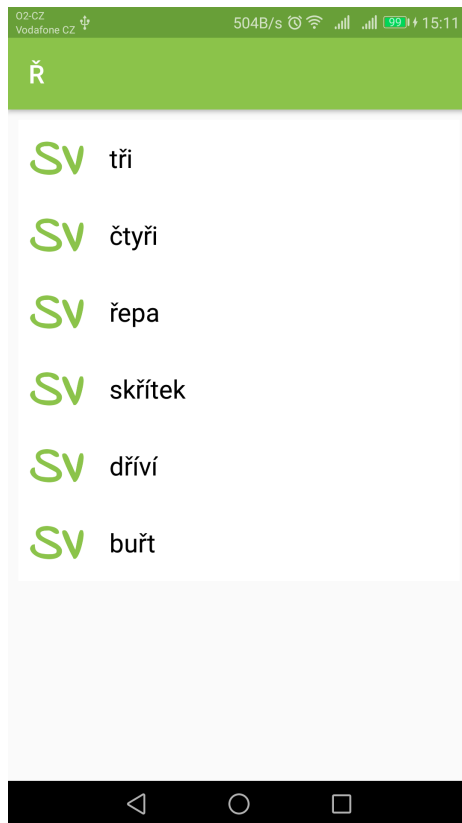
Obrázek B5: Speech Validator Admin obrazovka Nahrávka



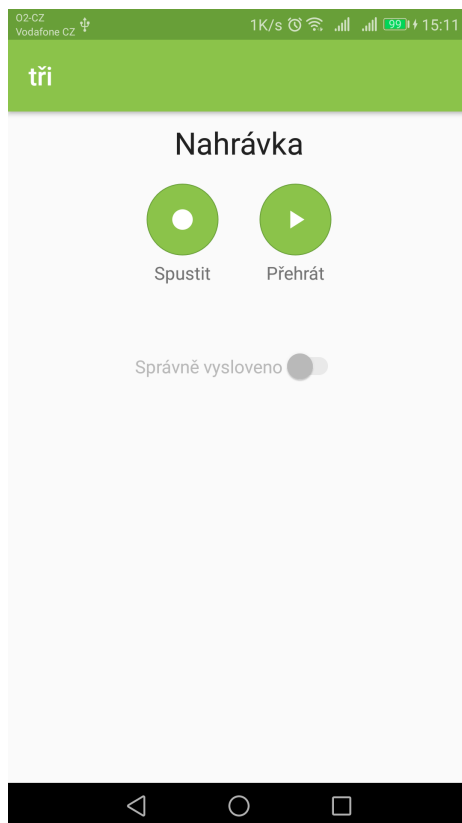
Obrázek B6: Speech Validator Uvítací obrazovka



Obrázek B7: Speech Validator obrazovka Výběr hlásky



Obrázek B8: Speech Validator obrazovka Výběr slova



Obrázek B9: Speech Validator obrazovka Cvičení

## Příloha C: Výsledky testů

	VANA		CINKA		LISKA		DRIVI		VONI		MAGI	
	Správné	Špatně	Správné	Špatně	Správné	Špatně	Správné	Špatně	Správné	Špatně	Správné	Špatně
1e+20-1e+20	0%	80%	0%	100%	0%	100%	0%	100%	0%	100%	0%	100%
1e+15-1e+15	13%	70%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+10-1e+10	12.50%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+5-1e+5	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-5-1e-5	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-10-1e-10	12.50%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-15-1e-15	37.50%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-20-1e-20	25%	80%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+4-1e+4	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+3-1e+3	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+2-1e+2	25%	70%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+1-1e+1	12.50%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e+0-1e+0	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-1-1e-1	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-2-1e-2	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-3-1e-3	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%
1e-4-1e-4	25%	60%	0%	100%	50%	50%	0%	100%	0%	100%	50%	50%

Tabulka C1: Výsledky testů pro výběr vah JSGF CREATIONN-5050



	VMA		CIWA		LISKA		DRUV		VONI		MAGT		SUM	
	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné
1e+20-1e+15	25%	80%	53%	100%	57%	100%	0%	100%	0%	100%	0%	100%	6.55%	96.67%
1e+20-1e+10	12.50%	40%	26%	60%	44%	60%	0%	94.74%	0%	70%	35%	50%	15.18%	65.79%
1e+20-1e+5	62.50%	20%	41%	30%	36%	30%	0%	94.74%	0%	20%	40%	100%	42.32%	41.12%
1e+20-1e+4	62.50%	20%	41%	20%	29%	20%	31%	40%	100%	40%	50%	100%	52.32%	46.99%
1e+20-1e+3	75%	20%	48%	30%	36%	30%	40%	100%	60%	30%	45%	100%	60.12%	38.33%
1e+20-1e+2	75%	20%	48%	30%	36%	30%	40%	94.74%	80%	20%	50%	100%	63.45%	34.12%
1e+20-1e+1	75%	20%	48%	40%	41%	40%	40%	89.47%	65%	80%	50%	100%	63.45%	36.58%
1e+20-1e+0	75%	20%	48%	40%	49%	30%	51%	40%	78.95%	80%	50%	100%	70.60%	33.16%
1e+20-1e-10	87.50%	0%	44%	30%	51%	20%	53%	100%	36.84%	68%	100%	100%	90.77%	16.14%
1e+20-1e-15	87.50%	0%	44%	30%	58%	20%	53%	100%	31.58%	66%	100%	100%	93.15%	15.28%
1e+20-1e-20	100%	0%	50%	20%	60%	20%	53%	100%	26.32%	63%	100%	100%	97.62%	12.72%
1e+10-1e+5	37.50%	50%	44%	100%	50%	100%	57%	0%	100%	50%	0%	90%	8.63%	88.33%
1e+10-1e+3	37.50%	50%	44%	100%	50%	100%	57%	0%	100%	50%	0%	90%	11.41%	81.67%
1e+10-1e+2	37.50%	50%	44%	100%	50%	100%	49%	0%	100%	50%	0%	90%	19.35%	80.00%
1e+10-1e+0	50%	20%	35%	60%	48%	70%	49%	0%	100%	50%	0%	70%	21.43%	65.00%
1e+10-1e-5	75%	20%	48%	40%	27%	50%	46%	20%	100%	60%	40%	60%	43.14%	50.00%
1e+10-1e-10	87.50%	10%	49%	40%	34%	20%	31%	60%	68.42%	64%	100%	40%	64.27%	31.40%
1e+10-1e-15	87.50%	10%	49%	40%	34%	20%	31%	60%	63.16%	72%	100%	10%	70.38%	25.53%
1e+10-1e-20	87.50%	0%	44%	30%	36%	20%	31%	100%	63.16%	82%	100%	0%	76.09%	20.53%
1e+5-1e+0	25%	40%	33%	90%	45%	90%	45%	0%	100%	50%	0%	100%	4.17%	85.00%
1e+5-1e-10	75%	30%	53%	40%	27%	60%	44%	0%	94.74%	20%	70%	45%	28.53%	58.12%
1e+5-1e-15	87.50%	10%	49%	40%	27%	50%	39%	40%	68.42%	54%	60%	40%	46.73%	38.07%
1e+5-1e-20	87.50%	10%	49%	40%	34%	50%	39%	60%	63.16%	62%	60%	20%	55.22%	33.86%
1e+0-1e-5	37.50%	50%	44%	100%	50%	100%	50%	0%	100%	50%	0%	90%	6.25%	88.33%
1e+0-1e-10	62.50%	40%	51%	80%	40%	90%	52%	0%	100%	50%	0%	90%	15.58%	81.67%
1e+0-1e-15	75%	40%	58%	70%	35%	80%	47%	0%	94.74%	20%	70%	45%	20.99%	70.79%
1e+0-1e-20	75%	30%	53%	60%	35%	80%	47%	40%	84.21%	62%	20%	40%	30.44%	65.70%
1e-5-1e-10	37.50%	50%	44%	100%	45%	0%	50%	0%	100%	50%	0%	100%	6.25%	88.33%
1e-5-1e-15	62.50%	50%	56%	80%	40%	100%	57%	0%	100%	50%	0%	100%	18.91%	86.67%
1e-5-1e-20	75%	40%	58%	80%	40%	100%	57%	20%	94.74%	57%	20%	80%	24.33%	82.46%
1e-10-1e-15	62.50%	50%	56%	100%	50%	100%	50%	0%	100%	50%	0%	100%	10.42%	90.00%
1e-10-1e-20	62.50%	50%	56%	100%	50%	100%	50%	0%	100%	50%	0%	100%	13.20%	90.00%
1e-15-1e-20	37.50%	50%	44%	100%	50%	100%	50%	0%	100%	50%	0%	100%	6.25%	91.67%
1e-1-1e-1	12.50%	50%	31%	100%	50%	100%	50%	0%	100%	50%	0%	100%	2.08%	91.67%
1e-3-1e-3	37.50%	50%	44%	100%	45%	0%	45%	0%	100%	50%	0%	100%	6.25%	88.33%
1e-4-1e-4	50%	40%	45%	90%	45%	90%	52%	0%	100%	50%	0%	100%	9.03%	85.00%
1e-5-1e-5	75%	30%	53%	80%	35%	90%	45%	0%	100%	50%	0%	100%	16.27%	80.00%
1e-6-1e-6	75%	30%	53%	80%	38%	80%	54%	0%	100%	50%	0%	100%	22.82%	73.33%
1e-7-1e-7	75%	30%	53%	60%	32%	80%	44%	20%	100%	60%	20%	80%	37.42%	56.33%

Tabulka C2: Výsledky testů pro výběr vah JSJF CREATIONN-PREF

	VANA		CIVKA		LISKA		DRIVY		VONI		MAGI		SUM							
	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné	Správne	Špatné						
PREF -lw 6.0	75%	30%	52.50%	42.86%	41.43%	71.43%	30%	50.72%	40%	78.95%	59.48%	80%	20%	50.00%	100%	10%	55.00%	66.22%	34.83%	51.52%
PREF -lw 6.5	75%	20%	47.50%	57.14%	48.57%	71.43%	30%	50.72%	40%	78.95%	59.48%	80%	20%	50.00%	100%	10%	55.00%	70.60%	33.16%	51.88%
PREF -lw 7.0	75%	20%	47.50%	42.86%	41.43%	71.43%	30%	50.72%	40%	78.95%	59.48%	80%	20%	45.00%	100%	10%	55.00%	66.22%	31.49%	49.85%
PREF -lw 8.0	75%	20%	47.50%	57.14%	48.57%	85.71%	30%	57.86%	60%	68.42%	64.21%	80%	20%	45.00%	100%	20%	60.00%	76.31%	31.40%	55.86%
PREF -lw 9.0	87.50%	10%	48.75%	57.14%	38.57%	71.43%	30%	50.72%	60%	73.68%	66.84%	80%	20%	45.00%	100%	20%	60.00%	76.01%	27.28%	51.65%
PREF -lw 10.0	87.50%	10%	48.75%	71.43%	45.72%	71.43%	20%	45.72%	60%	63.16%	61.58%	80%	20%	45.00%	100%	20%	60.00%	76.39%	23.86%	51.13%
PREF -lw 11.0	87.50%	10%	48.75%	71.43%	50.72%	71.43%	20%	45.72%	60%	68.42%	64.21%	100%	30%	65.00%	100%	20%	60.00%	81.73%	29.74%	55.73%
PREF -lw 12.0	87.50%	20%	53.75%	57.14%	43.57%	71.43%	20%	45.72%	60%	68.42%	64.21%	100%	30%	65.00%	100%	30%	65.00%	79.35%	33.07%	56.21%
PREF -lw 13.0	87.50%	30%	58.75%	85.71%	57.86%	71.43%	20%	45.72%	60%	78.95%	69.48%	100%	20%	60.00%	100%	40%	70.00%	84.11%	36.49%	60.30%
PREF -lw 14.0	0%	100%	50.00%	57.14%	43.57%	71.43%	30%	50.72%	60%	73.68%	66.84%	100%	20%	60.00%	100%	40%	70.00%	64.76%	48.95%	56.85%
PREF -lw 15.0	0%	100%	50.00%	42.86%	36.43%	71.43%	30%	50.72%	60%	73.68%	66.84%	100%	10%	55.00%	100%	40%	70.00%	62.38%	47.28%	54.83%
PREF -lw 16.0	0%	100%	50.00%	57.14%	38.57%	71.43%	50%	60.72%	60%	73.68%	66.84%	100%	0%	50.00%	100%	40%	70.00%	64.76%	47.28%	56.02%
PREF -lw 17.0	0%	100%	50.00%	71.43%	45.72%	71.43%	40%	55.72%	60%	68.42%	64.21%	100%	0%	50.00%	100%	40%	70.00%	67.14%	44.74%	55.94%
PREF -lw 18.0	0%	100%	50.00%	0%	50.00%	57.14%	40%	48.57%	60%	63.16%	61.58%	100%	10%	55.00%	100%	50%	75.00%	52.86%	60.53%	56.69%
PREF -lw 19.0	0%	100%	50.00%	0%	100%	50.00%	0%	50.00%	60%	63.16%	61.58%	60%	20%	40.00%	83.33%	40%	61.67%	33.89%	70.53%	52.21%
PREF -lw 20.0	0%	100%	50.00%	0%	100%	50.00%	0%	50.00%	60%	68.42%	64.21%	100%	10%	55.00%	0%	100%	50.00%	26.67%	79.74%	53.20%
5050 -lw 6.0	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 6.5	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 7.0	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 8.0	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 9.0	37.50%	70%	53.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	95.00%	50.63%
5050 -lw 10.0	37.50%	70%	53.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	95.00%	50.63%
5050 -lw 11.0	37.50%	70%	53.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	95.00%	50.63%
5050 -lw 12.0	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 13.0	37.50%	60%	48.75%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	6.25%	93.33%	49.79%
5050 -lw 14.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%
5050 -lw 15.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%
5050 -lw 16.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%
5050 -lw 17.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%
5050 -lw 18.0	0%	100%	50.00%	0%	100%	14.29%	100%	57.15%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	2.38%	100.00%	51.19%
5050 -lw 19.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%
5050 -lw 20.0	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0.00%	100.00%	50.00%

Tabulka C3: Výsledky testů pro výběr parametru LW

	ALL2_PREF-1w18.0		ALLN_PREF-1w18.0		CREATION2_PREF-1w18.0		CREATIONN_PREF-1w18.0		ALLN_PREF-1w18.0		CREATION2_PREF-1w18.0		CREATIONN_PREF-1w18.0		
	Správne Špatné	Avg	Správne Špatné	Avg	Správne Špatné	Avg	Správne Špatné	Avg	Správne Špatné	Avg	Správne Špatné	Avg	Správne Špatné	Avg	
VANA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MAVA	62.50%	51.25%	62.50%	51.25%	30%	46.25%	30%	46.25%	62.50%	46.25%	30%	46.25%	62.50%	46.25%	
FOTI	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PANEV	80%	50.00%	80%	45.00%	80%	40.00%	80%	40.00%	80%	40.00%	80%	40.00%	80%	40.00%	
FOFOUKA	100%	50.00%	100%	50.00%	100%	50.00%	100%	50.00%	100%	50.00%	100%	50.00%	100%	50.00%	
COP	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PEC	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PECKA	33.33%	66.67%	33.33%	66.67%	80%	40.00%	80%	40.00%	83.33%	56.67%	30%	56.67%	83.33%	56.67%	
SOVA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PES	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MISKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
KOST	20%	60.00%	20%	60.00%	40%	60.00%	40%	60.00%	40%	60.00%	40%	60.00%	40%	60.00%	
PUSA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
ZUBY	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
KOZA	16.67%	58.33%	16.67%	58.33%	83.33%	20%	51.67%	100%	60.00%	83.33%	20%	51.67%	100%	60.00%	
CINKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
OCI	60%	75.00%	60%	75.00%	100%	20%	60.00%	100%	20%	60.00%	100%	20%	60.00%	100%	75.00%
MIC	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
KOCKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MICEK	0%	80%	40.00%	40.00%	100%	0%	50.00%	100%	0%	50.00%	100%	0%	50.00%	100%	0%
SALA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MYS	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MASINKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
TUZKA	0%	90%	45.00%	45.00%	0%	90%	45.00%	0%	90%	45.00%	0%	90%	45.00%	0%	90%
MYSKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
ZABA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
JEZEK	50%	60.00%	66.67%	70%	68.33%	66.67%	70%	68.33%	66.67%	70%	68.33%	66.67%	70%	68.33%	
LOUPEZNIK	20%	55.00%	20%	55.00%	20%	55.00%	20%	55.00%	20%	55.00%	20%	55.00%	20%	55.00%	
LUSKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
DELO	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PALKA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
PIL	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
RYC	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
ROHY	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
METR	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
DRAK	16.67%	58.33%	16.67%	58.33%	100%	20%	51.67%	100%	20%	51.67%	100%	20%	51.67%	100%	20%
PERE	14.29%	52.60%	14.29%	52.60%	100%	20%	51.67%	100%	20%	51.67%	100%	20%	51.67%	100%	20%
DARKY	14.29%	57.14%	14.29%	57.14%	100%	20%	51.67%	100%	20%	51.67%	100%	20%	51.67%	100%	20%
BRAMBORA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
REPA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
DRIVI	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
SKRTEK	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
BURT	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
TRI	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
CTYRI	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
TATA	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
TUDY	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
MGST	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	0%	50.00%	
BOTY	42.86%	51.43%	42.86%	51.43%	100%	20%	60.00%	100%	10%	55.00%	100%	10%	55.00%	100%	55.00%

Tabulka C4: Výsledky testů pro PREF 1/2



	ALL2_5050 -lw 9.0		ALLN_5050 -lw 9.0		CREATION2_5050 -lw 9.0		CREATIONN_5050 -lw 9.0					
	Správne	Špatné	Avg	Správne	Špatné	Avg	Správne	Špatné	Avg			
VANA	0%	100%	50.00%	0%	100%	50.00%	25%	70%	47.50%	37.50%	70%	53.75%
MAVA	0%	90%	45.00%	12.50%	90%	51.25%	0%	90%	45.00%	0%	90%	45.00%
FOTI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PANEV	20%	50%	35.00%	60%	30%	45.00%	80%	30%	55.00%	80%	30%	55.00%
POFOUKA	40%	18.18%	29.09%	40%	18.18%	29.09%	60%	9.09%	34.55%	60%	9.09%	34.55%
COP	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PEC	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PECKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
OPICE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
SOVA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PES	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MISKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
KOST	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PUSA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
ZUBY	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
KOZA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
CINKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
OCI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MIC	0%	100%	50.00%	0%	100%	50.00%	0%	91.67%	45.83%	0%	91.67%	45.83%
KOCKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	14.29%	100%	57.14%
MICEK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
SALA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MYS	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MASINKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
TUZKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MYSKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
ZABA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
JEZEK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
LOUPEZNIK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
LISKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DELO	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PALKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PIL	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
RYC	0%	100%	50.00%	0%	100%	50.00%	33.33%	75%	54.17%	33.33%	75%	54.17%
ROHY	0%	100%	50.00%	0%	100%	50.00%	0%	90.91%	45.45%	0%	90.91%	45.45%

Tabulka C6: Výsledky testů pro 5050 1/2

METR	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DRAK	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
PERE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DARKY	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
BRAMBORA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
REPA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DRIVI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
SKRITEK	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
BURT	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
TRI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
CTYRI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
TATA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
TUDY	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MOST	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
BOTY	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
SUD	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	90%	45.00%
JEDE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DUM	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
NOHY	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
BRANA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
BUBEN	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	90%	45.00%
DITE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
VIDI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
TELO	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
KOTE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
SED	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
NITE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
VONI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
KOS	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MUSKA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MASKA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
LUK	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	90.91%	45.45%
GOL	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
MAGI	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
HOUPE	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%
DUHA	0%	100%	50.00%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%	0%	100%	50.00%

Tabulka C7: Výsledky testů pro 5050 2/2

	ALL2_NORM		ALLN_NORM		CREATION2_NORM		CREATIONN_NORM	
	Správné	Špatné	Správné	Špatné	Správné	Špatné	Správné	Špatné
		Avg		Avg		Avg		Avg
VANA	0%	100%	50.00%	0%	100%	50.00%	25%	60%
MAVA	0%	90%	45.00%	0%	90%	45.00%	0%	90%
FOTI	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PANEV	20%	60%	40.00%	60%	10%	35.00%	80%	100%
POFOUKA	40%	27.27%	33.64%	40%	27.27%	33.64%	80%	18.18%
COP	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PEC	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PECKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
OPICE	0%	100%	50.00%	0%	100%	50.00%	0%	100%
SOVA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PES	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MISKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
KOST	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PUSA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
ZUBY	0%	100%	50.00%	0%	100%	50.00%	0%	100%
KOZA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
CINKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
OCI	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MIC	0%	100%	50.00%	0%	100%	50.00%	0%	91.67%
KOCKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MICEK	0%	100%	50.00%	0%	100%	50.00%	0%	100%
SALA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MYS	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MASINKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
TUZKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
MYSKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
ZABA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
JEZEK	0%	100%	50.00%	0%	100%	50.00%	0%	100%
LOUPEZNIK	0%	100%	50.00%	0%	100%	50.00%	0%	100%
LISKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
DELO	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PALKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%
PIL	0%	100%	50.00%	0%	100%	50.00%	0%	100%
RYC	0%	100%	50.00%	0%	100%	50.00%	83.33%	41.67%
ROHY	0%	100%	50.00%	0%	100%	50.00%	90.91%	45.45%

Tabulka C8: Výsledky testů pro NORM 1/2

METR	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
DRAK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
PERE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
DARKY	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
BRAMBORA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
REPA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
DRIVI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
SKRITEK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
BURT	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
TRI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
CTYRI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
TATA	0%	100%	50.00%	0%	100%	50.00%	14.29%	100%	14.29%	100%	57.14%	14.29%	100%	57.14%
TUDY	0%	100%	50.00%	0%	100%	50.00%	14.29%	100%	14.29%	100%	57.14%	14.29%	100%	57.14%
MOST	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
BOTY	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
SUD	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	90%	45.00%
JEDE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
DUM	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
NOHY	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
BRANA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
BUBEN	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	90%	45.00%
DITE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
VIDI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
TELO	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
KOTE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
SED	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
NITE	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
VONI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
KOS	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
MUSKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
MASKA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
LUK	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
GOL	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
MAGI	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%
HOUPE	0%	100%	50.00%	0%	100%	50.00%	11.11%	90%	11.11%	90%	50.56%	11.11%	90%	50.56%
DUHA	0%	100%	50.00%	0%	100%	50.00%	0%	100%	0%	100%	50.00%	0%	100%	50.00%

Tabulka C9: Výsledky testů pro NORM 2/2