



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Pokročilý třídní jazykový slovníček

Diplomová práce

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: 1802T007 – Informační technologie

Autor práce: **Daniel Maděra**

Vedoucí práce: Ing. Jana Vítvarová, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Advanced Class Language Dictionary

Master thesis

Study programme: N2612 – Engineering and Informatics

Study branch: 1802T007 – Information Technology

Author: **Daniel Maděra**

Supervisor: Ing. Jana Vitvarová, Ph.D.



Tento list nahrad'te
originálem zadání.

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé diplomové práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum: 3. 1. 2017

Podpis:



Abstrakt

Klíčová slova

Abstract

Keywords

Obsah

1 Úvod	8
2 Analýza	9
2.1 Hlavní cíle	9
2.2 Existující řešení	10
2.3 Učení slovíček	12
2.4 Testování slovíček	14
2.5 Typy testů	14
2.6 Metoda rozloženého opakování	15
2.7 Specifikace požadavků	16
3 Návrh aplikace	18
3.1 Uživatelské role	18
3.2 Správa učebnic	19
3.3 Správa slovíček	19
3.4 Obtížnost slovíček	21
3.5 Procvičování slovíček	23
3.6 Generování slov pro procvičování	23
3.7 Připomínání slov	24
3.8 Motivace	26
3.9 Model databáze	26
3.10 Architektura aplikace	26
4 Klientská aplikace	27
4.1 Návrh aplikace	27
4.2 Vývojové prostředí	28
4.3 Knihovna React	28
4.4 Implementace technologií	28
4.5 Testování	28
5 Serverová aplikace	29
5.1 Technologie	29
5.2 Django a REST framework	29
5.3 Zabezpečení	29
5.4 PostgreSQL	29
5.5 Testování	29
6 Závěr	30
Použitá literatura	31

Seznam obrázků

1	Naznačení rozloženého opakování na základě křivky zapomínání	16
2	USE-CASE diagram aplikace	18

Seznam tabulek

1	Používání slov v britském anglickém jazyce	12
2	Matice koeficientů pro výpočet optimálních intervalů	26

Seznam zdrojových kódů

1 Úvod

2 Analýza

Výuka cizích jazyků je pro aktuální společnost jedna z nejzásadnějších otázek, ať se jedná o pracovní příležitosti v zahraničí, tak sociální problematika světa. Žáci a studenti se často účastní různých kroužků nebo později studenti využívají Erasmus programů, kde vyhledávají právě zlepšení komunikace v cizím jazyce.

2.1 Hlavní cíle

Hlavním cílem aplikace je připravit žáky na hodinu cizího jazyka a zároveň přirozeně rozvíjet slovní zásobu, tak aby studenti neztratili motivaci a chuť k poznávání nových výrazů. Dále také umožnit žákům se protestovat a ověřit, zda naučenou sadu slov ovládají. Aplikace by se měla adaptovat na zdatnost a úroveň každého ze žáků. Vyučujícím by aplikace měla usnadnit správu a import slovíček, které třída má umět v rámci dané učebnice a následně předkládat žákům vhodnou slovní zásobu například pro následující lekci.

2.1.1 Personalizace

Důležitým cílem aplikace by měla být personalizace podle potřeby jednotlivých žáků a celých tříd. Personalizace na úrovni žáka znamená přistupovat individuálně na základě jeho úrovně znalostí a dovedností. V případě konkrétní třídy jde o individuální přístup a to zejména v sadě slov, které je třeba v přípravě procvičovat.

2.1.2 Motivace

Důležitou součástí vzdělávání obecně je motivace. Tedy přimět žáky, aby z vlastní iniciativy chtěli rozvíjet své vědomosti. Problém ale je, že se děti přirozeně neučí z vlastní iniciativy, ale aby uspokojili okolí. Motivace se rozděluje do skupin - vnitřní a vnější nebo pozitivní a negativní [3]. V analýze bude zaměřeno pouze na motivační prostředky, které lze zařadit do aplikace pro procvičování slovíček.

Motivace založená na základě vlastních úspěchů je důležitá pro utvrzení sebevědomí žáků. Ocenění v případě zvládnutí sady slov nebo gramatického bloku lze v případě aplikace implementovat například hláškami s projevem pochvaly nebo jiným upozorněním na dosažený výsledek. Zajímavým prvkem v aplikaci by mohlo být i herní prostředí. Žáci si rádi hrají a již J. A. Komenský poukázal na důležitost her ve vzdělávacím procesu.

Jedním z dalších stěžejních faktorů motivace je kolektiv. Právě díky kolektivu, ve kterém funguje přirozená rivalita, jsou schopni žáci dosáhnout mnohem vyšších výsledků než kdyby se vzdělávali odděleně a samostatně. Rivalita a soutěživost může projevovat i velmi

negativním způsobem. Místo kamarádských vztahů mezi dětmi mohou vznikat nepřátelské, kde může docházet i k posměchu těch, kteří nemusí mít pro výuku až takové nadání. Zajímavější motivací tedy pro kolektiv je například vidina společně dosažených výsledků. V případě učení slovíček by to byl počet slovíček naučených za rok jako celá třída. Dochází zde k utužování kolektivu a děti by mohlo těšit to, že nějakým způsobem přispívají k úspěchům celé třídy.

2.1.3 Využití IT pro výuku

Posledních několik let se společnost ubírá trendem informačních technologií. Každý ze žáků má už od útlého věku přístup k počítači nebo k chytrému telefonu. Orientace a schopnost tyto zařízení používat není pro ně žádný problém. Přirozeně se tedy tyto zařízení postupně stávají součástí každodenní přípravy žáka na následující školní den. V některých případech tyto zařízení plně nahrazují klasické učebnice a jsou přímo začleněny do výuky. Použití informačních technologií má za následek i zlepšení motivace při výuce. Obecně je známo, že žáci raději studují slovíčka interaktivní formou hádanek, křížovek nebo her, než nekonečných seznamů slov.

Důležitost cizích jazyků se projevuje na míře používání například chytrých tabulí nebo tabletů při výuce. Tyto zařízení umožňují interaktivní výuku, kde lze využít nejen textových, ale také obrázkových a zvukových prostředků pro lepší zasazení nově nabytých vědomostí do kontextu.

2.2 Existující řešení

Na trhu lze nalézt nepřeberné množství aplikací pro výuku cizích jazyků. Řada z nich jsou téměř komplexní systémy, které provází studenta od základních frází a slovíček až po gramatické standardy cizího jazyka. Analýza existujících řešení byla zaměřena na aplikace, které se zabývají především testováním slovíček a frází.

Do analýzy existujících řešení byly zahrnuty tři desktopové, jedna webová a jedna mobilní aplikace.

2.2.1 TS Angličtina

Z analyzovaných řešení se jevila desktopová aplikace TS Angličtina od firmy Terasoft ta nejlépe funkčně propracovaná. Tato firma se zabývá širokou škálou výukových nástrojů se zaměřením na základní školy. V případě cizích jazyků se zabývají výukou anglického a německého jazyka. Hlavní předností aplikace je podpora nejvíce používaných učebnic cizího jazyka. Aplikace umožňuje testování různými způsoby - psaný překlad slova, porozumění mluvenému slovu, vybírání správných významů nebo doplňování vynechaných slov [1].

Analýza byla tvořena pouze z informací vydavatele. Bohužel firma Terasoft neposkytuje DEMO nebo TRIAL verzi aplikace, která by hlubší analýzu umožňovala.

2.2.2 Langsoft Teacher

Dalším analyzovaným řešením byla aplikace Langsoft Teacher, která je dostupná v podání desktopové aplikace, ale také i mobilní aplikace pro platformy iOS a Android. Aplikace je velmi komplexní, obsahuje různé moduly pro testování například v obrazové formě pro předškolní děti. Zajímavá vlastnost, kterou aplikace disponuje, je pamatování problematických slov a nabízení těchto slov častěji než těch bezproblémových. Dále program umožňuje automaticky rozšiřovat slovní zásobu, která je v testování zahrnuta [2].

2.2.3 Duolingo

Dalším v pořadí byla aplikace Duolingo. Jedná se o mobilní aplikaci pro Android. Zahrnuje učivo cizího jazyka od základních komunikačních frází až po tvorbu gramaticky složitějších vět. V aplikaci je připravena dlouhá řada cizích jazyků - němčina, angličtina, španělština (kastilština), italština a další. Chybí ale více referenčních jazyků. Aktuálně lze využít pouze angličtinu. Aplikace kromě standardních funkcí zahrnuje i rozpoznávání mluvených odpovědí. Zajímavým poznatkem byl systém motivace uživatelů, kde si každý mohl pozvat své přátele, mezi kterými docházelo k sdílení dosažených výsledků. Dalším motivačním základem bylo nutkání udržení plánu pravidelného testování, jelikož v opačném případě docházelo k automatickému zvyšování objemu testovacích dat. Nevýhodou aplikace byla nutnost připojení k internetu. V případě požití mobilních dat, docházelo ke zpoždění zejména při rozpoznání slov.

2.2.4 Vocabulary Trainer

Vocabulary Trainer je webová aplikace zdarma napsaná v jazyce PHP, která naučí 5000 nejvíce používaných slov daného jazyka. Aplikace umožňuje hodně možného nastavení. K dispozici je i řada jazyků včetně češtiny a to jako referenční i jako učený jazyk. Testování spočívá nejdříve ve čtení slov a následně uživatel vybírá možnosti odpovědi. Aktuálně překládané slovo si lze kdykoliv přehrát v různé rychlosti. Jako motivační základ aplikace využívá jednoduchý bodový systém. Součástí je i kalendář s email upomínkou k dalšímu testování. Program je propracovaný, ale poměrně pomalý a dlouho trvá zejména úvodní načítání dat. Její výrobce LanguageCourse S.L. poskytuje i mobilní aplikaci pro Android k učení anglických slov a frází.

2.2.5 Supermemo aplikace

Za zmínku ještě stojí aplikace Supermemo. Není to aplikace s připravenými daty k testování slov cizího jazyka, ale slouží čistě jako šablona pro testování jakéhokoliv druhu otázek. Aplikace implementuje algoritmus Supermemo, který je založen na metodě postupného zvyšování intervalu dotazování na dané otázky. Při každé odpovědi program spočítá, kdy by si uživatel měl danou otázku zopakovat tak, aby odpověď byla správná a zároveň se co nejvíce zvyšoval interval mezi aktuální a předchozí odpovědí. Aplikace se adaptuje na schopnosti uživatele, v přiměřeném měřítku buď zvyšuje nebo snižuje interval dalšího připomenutí.

Z výše uvedených aplikací až na Supermemo žádná neumožňuje personalizovaný výběr učiva. Tedy nelze vložit vlastní slovíčko nebo si určit sadu slov pro testování. Proto jsou tyto aplikace především cílené pro uživatele, kteří vnímají výuku jazyka jako samostatné vzdělávání sami sebe. Pro studenty, kteří absolvují lekce z cizího jazyka ve škole je tento typ vzdělávání nevyhovující, jelikož se musí učit dvě nezávislé skupiny slov. Sice dochází k rozvinutí slovní zásoby studenta i do jiných okruhů než je jeho učebnice a málokterý student má ještě energii, časovou dotaci a vlastní iniciativu na to, aby se připravoval na školní test ze slovíček a ještě rozvíjel samostatně svoji cizojazyčnou slovní zásobu.

2.3 Učení slovíček

Standardní učení slovní zásoby cizího jazyka je založeno na častém opakování slov. Dle výzkumu Biemillera a Boote se lze ročně zvládnout až 400 slov u studentů 3.—6. tříd [4]. Což v případě cizího jazyka poměrně velké číslo, ale problém je, do jaké míry je slovo správně ukotveno v dlouhodobé paměti. Porovnání, kolik je potřeba slov pro ovládnutí anglického jazyka, usnadní následující tabulka 1, která zahrnuje procentuální využití nejvíce používaných slov v každém z odvětví [6].

	Konverzace	Noviny	Akademický text
1. 1000 slov	84,3%	75,6%	73,5%
2. 1000 slov	6%	4,7%	4,6%
Akademické výrazy	1,9%	3,9%	8,5%
Ostatní	7,8%	15,7%	13,3%

Tabulka 1: Používání slov v britském anglickém jazyce

2.3.1 Zastaralý způsob učení

Dle vlastního průzkumu žáci základních škol nevyužívají k učení slovíček nikterak pokročilé technologie. Většinou si udržují vlastní slovníček, do kterého nově nabytá slova. Při učení zakrývají část s cizími slovy a na základě českého ekvivalentu se snaží vybavit

překlad slova. Tato metoda neposkytuje prakticky žádnou zpětnou vazbu. Žáci většinou pouze do krátkodobé paměti uloží slova, později při testu rychle zodpoví a následně během pár hodin si na slovo už ani nevzpomenou. Nevýhod a námětů na zlepšení metody má tato metoda celou řadu, ale jedním z klíčových vad je, že žáci si nevytvoří dostatečně souvislostí, aby řádně ukotvilo slovo v paměti.

2.3.2 Zvuková interpretace

Zejména v učení cizího jazyka je zvuková podoba a interpretace slov velice důležitá. Jelikož každý jazyk může hlásky různě zvukově interpretovat a pro nováčka v cizím jazyku nemusí textová výslovnost plně vyhovovat. Žák dále díky znění slova získá podvědomí o dialektu daného jazyka a zároveň dochází k lepšímu zapamatování slova. Právě díky zvukům dochází k propojení při učení i pravé mozkové hemisféry. A napomáhá tedy k vytvoření pevnější ukotvení v paměti. Dle studie bulharského vědce George Lozanova, který se zabýval studiem mozku a učebních metod, byl zjištěn obrovský přínos zvukových vjemů [5].

2.3.3 Problematika obtížnosti

Velkým problémem v učení slov je přizpůsobit obtížnost každému ze žáků individuálně. Jelikož ne všichni mají stejnou úroveň znalostí cizího jazyka a každý potřebuje jiné tempo pro zapamatování sady slov. Jsou žáci s výbornou pamětí, kterým stačí si slova pouze jednou projít a dokáží je používat, ale jsou žáci, kde nestačí je pětkrát zopakovat. Dalším problémem obtížnosti je z pohledu jednotlivých slov. Každé slovo má odlišnou obtížnost, které lze soudit například podle míry používanosti v jazyce, délky slova, zdali obsahuje přehlasování, dvojité písmena nebo podobnost s referenčním slovem.

2.3.4 Učení slov v kontextu

Pro učení slov cizího jazyka je rovněž důležité správné zasazení významu slova do kontextu. Dle průzkumu Biemillera a Bootea z roku 2006 bylo zjištěno, že u žáků od 10—13 let docházelo k nárůstu zapamatovaných slov o 4%, pokud byla slova předložena v kontextu vět. Důležitým poznatkem z toho průzkumu je, že žáci si nejen déle naučené slovo pamatovali, ale správně ho i interpretovali, když měli za úkol ho svými slovy vysvětlit [4]. Ve stejném průzkumu rovněž docházelo ještě k vyššímu zlepšení v učení, kdy si žáci zapisovali vlastními slovy definici a použití slova.

2.4 Testování slovíček

2.4.1 Aktivní a pasivní slovní zásoba

Slovní zásobu, kterou využíváme k tvorbě vět ať už v cizím nebo mateřském jazyce rozdělujeme na dvě skupiny - aktivní a pasivní. Pasivní zásoba je sada slov, která jsou pevně a spolehlivě uložena v naší paměti. Průměrný žák zná cca 50 000 výrazů. Její velikost je ovlivněna věkem, vzděláním a četbou. Slova ze této sady využíváme zejména při písemné formě, ať už se jedná o čtení nebo psaní. Aktivní zásoba je sada slov, ve které lze najít žádané slovo během desítek milisekund. U většiny lidí dosahuje velikosti 4 000 až 8 000 výrazů [7]. Slovo se díky používání dostává z pasivní do aktivní slovní zásoby.

2.4.2 Metody testování

Metod testování slovní zásoby je mnoho. Základním rozdělením je na pasivní a aktivní. V případě pasivního se jedná například o výběr z nabídnutých možností. Jde o případ, kdy student nemusí znát přesnou odpověď a dokáže otázku vyhodnotit správně vylučovací metodou. Při aktivním testování žáci musí odpověď znát, aby otázka byla vyhodnocena správně. Pasivní metoda má pozitivní vliv například na motivaci žáka, který u ní tolik netápe a za pomoci zdravého rozumu může test vyhodnotit správně. Problém ale vzniká při používání získaných a procvičených informací. V případě cizího jazyka lze pasivní slovní zásobu využít pro čtení a náslech, ale pro ovládnutí cizího jazyka je nedostatečná a tudíž nevhodná.

Dalším rozdělením pasivního a aktivního testování slovíček je rozpoznávání (*recognition*) a vzpomnutí (*recall*). Rozpoznávání spočívá v předložení cizího slova žákovi a pro správné zodpovězení musí najít český ekvivalent. Vzpomnutí je způsob testování opačným způsobem než rozpoznání. Tedy uživateli je předložen výraz v jeho mateřském jazyce a on musí nalézt správný výraz v cizím. Rozpoznání je z principu jednodušší pro uživatele než vzpomnutí. Lze ho tedy využít rovněž pro zvýšení motivace při procvičování slov. Ale pro ověření, zda je slovo ovládnuté či nikoliv je metoda rozpoznání také nedostatečná.

2.5 Typy testů

V analýze existujících řešení bylo procvičování a testování slov interpretováno různými způsoby. Obecně by se typ testů dal rozdělit na tři kategorie - textové testy, multimediální a herní testy.

2.5.1 Textové testy

Textové testy se vyskytovaly například jako výběr z více možností nebo spojováním spolu souvisejících významů, jak už ale bylo uvedeno v kapitole 2.4.2, jedná se o rozvíjení pasivní slovní zásoby. Zajímavějším typem testů je doplňování slov do vět a klasický překlad slova. Tyto typy rozvíjejí žádanou aktivní slovní zásobu. Standardně textové testy byly doplněny zvukovou interpretací cizího slova.

2.5.2 Multimediální a herní testy

Zajímavým využitím multimédií by mohlo být zahrnutí otestování výslovnosti. Tedy možnost nahrání odpovědi a následně by došlo k rozpoznání. Ale analýza a rozpoznání řeči není nikterak triviální záležitost. Dalším zajímavým řešením byly herní testy. Díky nimž docházelo k zvýšení motivace uživatelů. Problém je, že tento typ procvičování většinou není zas tolik efektivní, jelikož si uživatel procvičí například v případě doplňování písmen křížovky nebo známé hry *Hangman* pouze pár slov za poměrně dlouhý čas.

2.6 Metoda rozloženého opakování

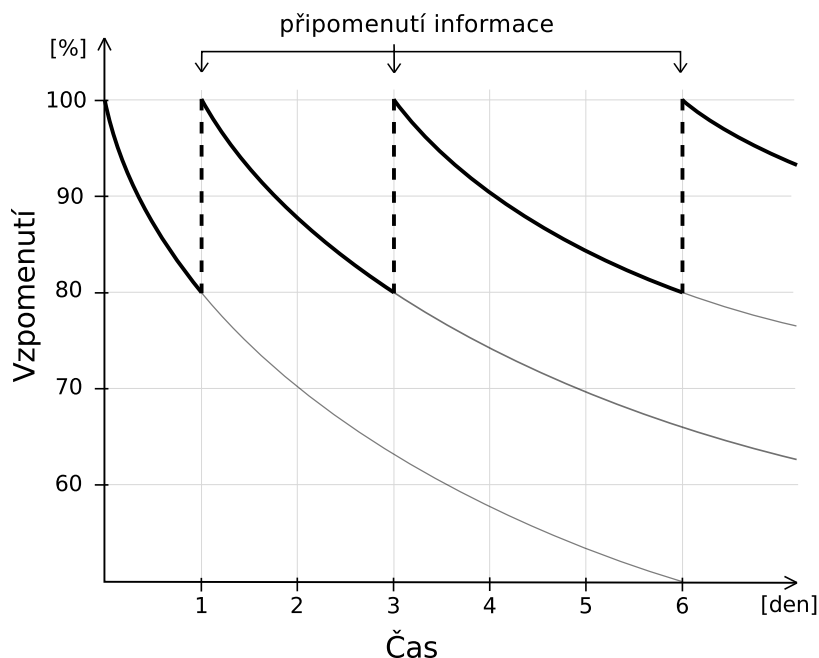
Až na výjimečné paměti je obecně známo, že pokud chceme informaci v paměti uchovat dlouhodobě, je nutné opakovaně připomínat. Pokud informace není vyžívána s největší pravděpodobností dojde ke ztrátě této informace. V případě, že uživatel s touto informací pracuje častěji, výrazně navyšuje šance pro její zapamatování. Metoda rozloženého opakování (*spaced repetition*) je učební technika založená na opakovaném připomínání informace s navyšujícím intervalem. Na začátku učebního procesu, jsou intervaly krátké například na hodinu, 4 hodiny nebo celý den. S postupem se tyto intervaly mohou zvyšují až na týdny a měsíce. Ideální systém rozloženého opakování nabídne zopakování informace před tím, než dojde k jejímu eventuálnímu nevzpomenutí.

2.6.1 Křivka zapomínání

Křivka zapomínání je graf exponenciálního charakteru, který znázorňuje za jak dlouho a s jakou pravděpodobností je možné si na nově nabytou informaci vzpomenout. První tuto křivku specifikoval už v roce 1885 Hermann Ebbinghaus. Na ose Y lze nalézt procentuální pravděpodobnost vzpomenutí na informaci, na ose X je čas. V čase 0 dojde k nabytí informace a s postupem času dochází k zapomínání. U průměrných pamětí dojde k potenciálnímu zapomenutí nové nesouvisející informace bez kontextu během pěti dní [10].

Následující obrázek 1 znázorňuje metodu rozloženého opakování na křivce zapomínání. V čase s 80% pravděpodobností vzpomenutí na danou informaci dojde k znovu připome-

nutí informace. Křivka zapomínání se posune na ose Y a znovu dochází k postupnému zapomínání. Díky exponenciálnímu charakteru interval mezi jednotlivými opakováními narůstá také exponenciálně. V ideálním případě se posune exponenciála tak, že se bude limitně blížit 80% hranici.



Obrázek 1: Naznačení rozloženého opakování na základě křivky zapomínání

2.6.2 Leitnerův algoritmus

Na Leitnerův algoritmus lze nahlížet jako na známý učební postup v podobě karet s otázkami na informace. Tyto karty jsou rozděleny do skupin, které jsou označeny například 1 až 3, kategorií ale obecně může být více. Nová karta je zařazena do první kategorie. Karty v první kategorii jsou opakovány častěji, v poslední kategorii jsou karty opakovány méně často. Karty jsou přemisťovány podle toho, jak uživatel zvládá odpovídat na jednotlivé otázky. V případě, že uživateli otázka na kartě nedělá problémy, přesune kartu do vyšší kategorie. V opačném případě, kdy uživatel nemůže opovědět správně na danou otázku, karta se přesune do první kategorie a cyklus pro danou kartu začíná znovu.

2.7 Specifikace požadavků

Na základě analýzy byla provedena specifikace požadavků, které budou implementovány v aplikaci pro testování slovíček z cizího jazyka.

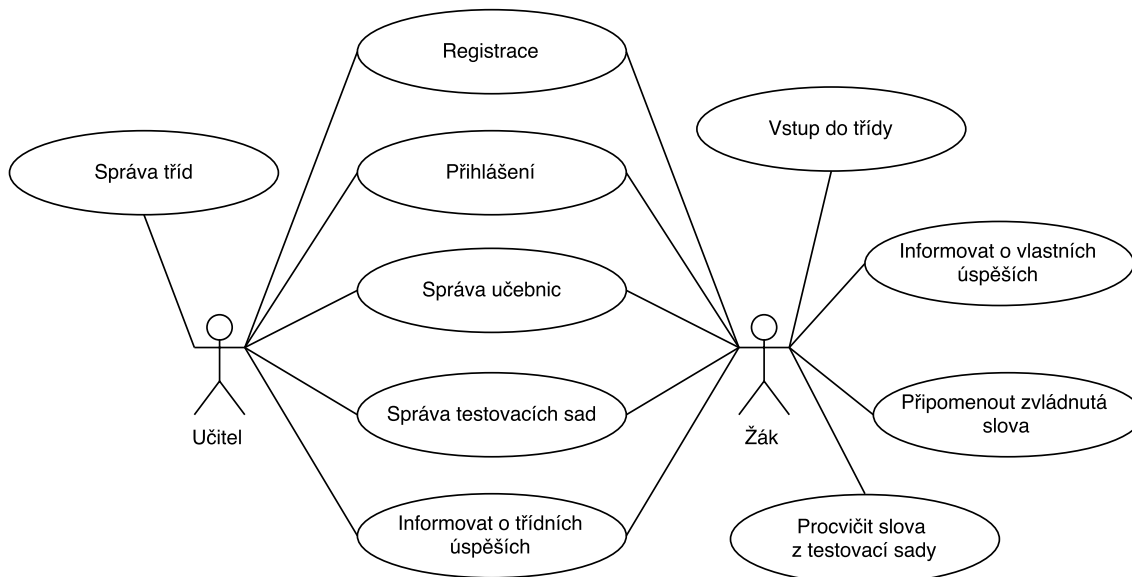
1. správa učebnic

- tvořit, editovat a mazat vlastní učebnice

- umožnit publikovat učebnici pro veřejnost
2. správa slovíček v učebnici
 - tvořit, editovat a mazat slovíčka
 - hromadně slovíčka do učebnice importovat
 - částečně automatizovat zvukovou a obrazovou interpretaci slovíčka
 3. správa modulů a tématických okruhů v učebnici
 - tvořit, editovat a mazat moduly a tématické okruhy učebnice
 - přiřazovat slovíčka do daných modulů a okruhů
 4. správa uživatelských skupin (tříd)
 - tvořit, editovat a mazat uživatelské skupiny
 - umožnit uživatelům se přihlásit do skupiny
 5. správa testovacích sad
 - tvořit, editovat a mazat testovací sady
 - umožnit vybírat slova pro testovací sadu z vlastních i veřejných učebnic
 - přiřazovat testovací sady ke skupinám uživatelů
 6. procvičování slovíček
 - vybrat testovací sadu slovíček
 - generovat slova na základě úrovně uživatele
 - zahrnout obrazovou a zvukovou interpretaci do procvičování
 - možnost uložit stav testování a umožnit pozdější navázání
 7. připomínat a procvičovat ovládnutou slovní zásobu
 8. motivace
 - motivovat v rámci uživatelské skupiny
 - motivovat vlastní iniciativu k procvičování

3 Návrh aplikace

Na základě specifikace požadavků byl vytvořen USE-CASE diagram aplikace. Diagram na obrázku 2 znázorňuje pouze obecné bloky funkčnosti aplikace. V následujících kapitolách bude každý z jednotlivých USE-CASE bloků popsán detailněji.



Obrázek 2: USE-CASE diagram aplikace

3.1 Uživatelské role

Do aplikace by měly vstupovat dva typy uživatelů - učitel a žák. Učitel z principu má za úkol spravovat testovací sady, učebnice a třídy. Žáci oproti nim mají za úkol vstupovat do svých tříd, procvičovat slova z testovacích dat, které jsou připravené od učitele a připomínat si zvládnutá slova.

Po konzultacích s vedoucím práce a vyučujícím cizího jazyka na základní škole došlo k několika změnám v návrhu právě v oblasti uživatelských rolí a i v principu aplikace. Vznikl požadavek, že aplikace by mohla fungovat spíše jako portál pro vzdělávání a ne jako výukový nástroj pro učitele. V návrhu tudíž vzniklo rovnocenné partnerství mezi učitelem a žákem.

3.1.1 Sjedení uživatelských skupin

Podnětem pro sjednocení uživatelských skupin vedlo umožnění žákům procvičovat si slova i v případě, že nejsou součástí žádné třídy. Tak aby mohl kdokoli se do aplikace přihlásit, vybrat si testovací sadu a vyzkoušet svoje znalosti slovíček. Tato funkce umožňuje žákům volnější přístup k výuce. Nebylo totiž v požadavcích to, aby učitel měl přehled o znalostech žáka - právě naopak. Účelem aplikace je poskytnout žákům možnost se samostatně

vzdělávat a procvičovat slovní zásobu a ne vyučujícím poskytnout nástroj pro otestování, zda žák disponuje požadovanými znalostmi.

3.2 Správa učebnic

Z důvodu personalizace slovíček bylo nutné v návrhu aplikace zařadit učebnice, díky nimž si žáci budou procvičovat pouze slovíčka, která jsou pro ně aktuální ve výuce cizího jazyka. Učebnice obsahují moduly a v modulech se nacházejí jednotlivá slova. Při tvorbě učebnice bude zvolen cizí jazyk, pro který je učebnice připravená. V rámci jedné učebnice jsou slova unikátní. Tzn. do konkrétní učebnice nemohou patřit dvě stejná slova. Majitel učebnice je může editovat, vytvářet, skrýt pro veřejnost a mazat i v případě, že je využívána některým z ostatních uživatelů.

V aplikaci se může vyskytovat i více stejných učebnic. Například dva různí vyučující vedou své hodiny cizího jazyka dle stejné učebnice, ale každý si chce přizpůsobit sadu slov po svém. Proto se v aplikaci musí učebnice identifikovat majitelem a názvem učebnice.

3.2.1 Sdílení učebnic

Jelikož aplikace slouží jako portál pro vzdělávání, ve výchozím stavu jsou učebnice a slovíčka v nich veřejně přístupné jakémukoliv uživateli. Příchozí uživatel si tedy bude moci buď vytvořit vlastní učebnici se svými slovy, které si chce procvičit nebo může vyhledat už z vytvořených učebnic. Díky sdílení nový uživatel může prakticky okamžitě po přihlášení začít procvičovat slova a nemusí složité žádná slova importovat. Tento přístup je cílený právě především pro žáky, kteří si chtějí samostatně procvičovat slova a nepatřit do žádné konkrétní třídy.

3.3 Správa slovíček

Jednotlivá slovíčka se budou ukládat do daným modulů učebnice. Slovíčko se bude charakterizovat - překladem v cizím jazyce, významem v mateřském jazyce, definice slova v cizím i mateřském jazyce a použití ve větách z učebnice. Dalšími atributy, kterými slovíčko bude disponovat, jsou obtížnost a slovní druh. Slovíčko bude možné doplnit o multimediální interpretaci a to v podobě zvuku a obrázku.

3.3.1 Textová forma

Textová forma slovíčka spočívá v překladu slova a jeho významu v mateřském jazyce. Pro zjištění významu slova se může využít již hotové aplikace. Například Google Translate poskytuje aplikační rozhraní pro překlady slov i celých vět. Výhodou by bylo usnadnění

zadávaní a import slovíček do aplikace. Jelikož se aplikace bude zabývat aktivním rozpoznáním a vzpomínáním, je vhodné neautomatizovat významy slov. Učebnice nebo učitelé se mohou s Google Translate lišit v definování významů slov. Dalším problémem je cena využívání překladů z Google Translate, která je účtovaná měsíčně a to \$20 za 1 milion znaků [9].

3.3.2 Zvuková forma

Do aplikace by měla být implementována i zvuková interpretace slovíčka v cizím jazyce. Zvukové nahrávky mohou být zařazeny do aplikace dvěma způsoby - manuálně vlastními nahrávkami nebo využít Google Text to Speech API, které poskytuje zdarma, pokus se měsíční využití zvukových nahrávek obsáhne do 60 minut [9]. V případě jedno nebo dvou-slovných spojení, které zaberou přibližně 4–8 vteřin, je dostačující pro cca 600 slovíček za měsíc. Aplikace bude záznam lokálně ukládat, aby bylo se co nejvíce omezilo využívání API. Zvuková nahrávka bude získána při importu slovíček do učebnice. Zadávající uživatel si bude moci zvolit, k jakému slovu chce zvukovou interpretaci.

3.3.3 Obrazová forma

Obrazová interpretace slovíčka bude rovněž získaná z Google Custom Search API. Toto aplikační rozhraní není omezené a lze v parametrech omezit vyhledávání pouze na obrázky. Při importu si uživatel zvolí pro jaká slovíčka chce obrazovou formu. Automatizace v tomto případě není vhodná, jelikož ne všechny slovíčka lze interpretovat jako obrázek. Uživatel si také bude moci zvolit obrázek z několika možností. Po vybrání obrázku bude vytvořena komprimovaná kopie souboru a uloží se rovněž zdroj, odkud je obrázek získán. Tato forma bude v aplikaci sloužit pouze jako nápověda pro žáky, díky níž si mohou ke slovíčku vytvořit hlubší asociaci.

Díky Google Custom Search API lze vyhledávání parametrizovat. Výrazem pro vyhledávání bude referenční slovo v cizím jazyce, jelikož cílový jazyk bude nejspíš anglický, německý nebo španělský a všechny tyto jazyky jsou více rozšířené než ten český. Dalším parametrem vyhledávání je omezení pouze na obrázky s licencí a „Volně užívat nebo sdílet“, která umožňuje obrázek zkopírovat a nekomerčně publikovat s uvedením zdroje. Bohužel společnost Google Inc. nezaručuje pod jakou licencí obrázky aktuálně se na stránkách prezentuje. Proto bude uživatel při výběru obrázku vyzván, aby zkontroloval licenci použití.

3.4 Obtížnost slovíček

Při zadávání slova do učebnice bude automaticky vyhodnocena jeho obtížnost, která je rozdělena do čtyř kategorií - snadné, střední, těžké a nemožné. Vyhodnocení obtížnosti je založeno na kombinaci několika parametrů slova - jeho délky, podobnosti s českým ekvivalentem a další atributy jako počet přehlasovaných písmen nebo počet výskytů dvojitých písmen. V případě nevhodně vygenerované obtížnosti slova, uživatel ji bude moci opravit dle vlastního uvážení. Délka slova je prahově rozdělena do kategorií obtížnosti, následně bude vyhodnocena podobnost s významem pomocí Levensteinovy vzdálenosti popsané v podkapitole 3.4.2. Poté následuje vyhodnocení přehlasování a dvojitých hlásek, které ale ovlivňují obtížnost už minimálně.

3.4.1 Výběr algoritmu

Levenshteinova vzdálenost je velmi podobná známe Hammingově vzdálenosti s tím rozdílem, že Hammingova vzdálenost uvádí počet pozic se stejným symbolem v obou řetězcích. Kdežto Levenshteinova uvádí počet jednoznakových editací. Pro určení míry podobnosti mezi slovem v cizím jazyce a jeho překladem je vhodnější Levenshteinova vzdálenost, jelikož akceptuje řetězce s různou délkou a v případě, že písmeno chybí, je to vyhodnoceno pouze jako odstraněné písmeno a algoritmus pokračuje dál. V případě Hammingovy vzdálenosti, vynechané písmeno naruší porovnávání zbytku řetězce.

3.4.2 Levenshteinova vzdálenost

Levenshteinova vzdálenost v informatice je míra rozdílu mezi dvěma řetězci. Pro výpočet vzdálenosti se využívá matice s rozměry velikostí délek řetězců. V podstatě vzdálenost je minimální počet jednoznakových úprav ve slově, aby vzniklo slovo referenční. Za jednoznakovou úpravu se považuje smazání, záměna nebo vložení písmene na dané místo.

Vzorec 1 je matematický zápis Levenshteinovy vzdálenosti, kde a a b jsou řetězce $a(i)$ a $b(j)$ jsou indexované znaky v řetězcích. Výsledná vzdálenost je rovna hodnotě v matici $dist_{a,b}$ na pozici $dist_{a,b}(|a|, |b|)$, kde $|a|$ a $|b|$ jsou délky řetězců [8].

$$dist_{a,b}(i, j) = \begin{cases} max(i, j) & min(i, j) = 0 \\ min \begin{cases} dist_{a,b}(i-1, j) + 1 \\ dist_{a,b}(i, j-1) + 1 \\ dist_{a,b}(i-1, j-1) + \begin{cases} 0 & a(i) = b(j) \\ 1 & a(i) \neq b(j) \end{cases} \end{cases} & min(i, j) \neq 0 \end{cases} \quad (1)$$

V případě, že se pohybujeme v matici v prvním sloupci a prvním řádku přiřazujeme hodnoty 1 až délka řetězce a do sloupce a 1 až délka řetězce b do řádku. Tedy první sloupec a první řádek v matici slouží jako referenční inicializace. V generování matice se následně pokračuje a porovnávají se jednotlivé znaky. První položka v minimu je případ, kdy písmeno bylo odstraněno z a . Druhá položka je případ, kdy došlo k vložení znaku do a na základě znaku v b řetězci. A poslední položka je případ porovnání znaků. V okamžiku, kdy znaky se rovnají, hodnota vzdálenosti na diagonále zůstává, v opačném případě se přičítá jednička.

Výpočet Levenshteinovy vzdálenosti dvou slov znázorňuje matice 2, kde dochází k porovnání českého slova *cyklus* a anglického výrazu *cycle*. Tučně je znázorněna cesta výpočtu. Prováděné jednoznakové operace jsou žádná, žádná, žádná, záměna, záměna a vložení.

	<i>c</i>	<i>y</i>	<i>c</i>	<i>l</i>	<i>e</i>	
0	1	2	3	4	5	
<i>c</i>	1	0	1	2	3	4
<i>y</i>	2	1	0	1	2	3
<i>k</i>	3	2	1	1	2	3
<i>l</i>	4	3	2	2	1	2
<i>u</i>	5	4	3	3	2	2
<i>s</i>	2	5	4	4	3	3

(2)

3.4.3 Určení globální obtížnosti

Základním parametrem globální obtížnosti slova je jeho délka, která má váhu 4 v celkovém průměru jednotlivých hodnot obtížnosti. Na základě prahových hodnot je délka rozdělena do čtyř kategorií - $\langle 1, 6 \rangle$ je snadné, $\langle 6, 11 \rangle$ je střední a $\langle 11, \infty \rangle$ je slovo těžké. Dalším parametrem k určení obtížnosti je Levenshteinova vzdálenost mezi referenčním slovem a jeho překladem, jejíž váha je 2. Znovu zde figurují nastavitelné prahové hodnoty - v případě, že se procentuální podobnost (tj. $pp = (1 - a/l) * 100$; kde l je délka slova a a je počet jednoznakových editací) nachází v intervalu $\langle 70, 100 \rangle$, je obtížnost nejlehčí; v intervalu $\langle 30, 70 \rangle$ je střední a v intervalu $\langle 0, 30 \rangle$ je slovo těžké. Přehlasování znaku a dvojitý výskyt stejného znaku mají oba stejnou váhu 1. Bez výskytu přehlasování a dvojitého znaku je obtížnost kvalifikována jako snadná, jeden výskyt je střední obtížnost a více výskytů je obtížnost vyhodnocena jako těžká.

3.5 Procvičování slovíček

3.5.1 Vyhodnocování odpovědí

3.5.2 Metody testování

3.5.3 Náповědy

3.5.4 Kontrola podvádění

3.6 Generování slov pro procvičování

Způsob generování slov je důležité pro správné ukotvení slova v dlouhodobé a ne krátkodobé paměti. Při testování bude docházet i k několika násobnému opakování slova. Základem generování je častěji a vícekrát předkládat slova, která jsou pro žáka problematická a slova, která jsou zvládnuta žákem s přehledem, jsou méně častěji testována. Pro generování byla zvolena metoda rozloženého opakování v podobě přizpůsobeného Leitnerova algoritmu, který je popsán v kapitole 2.6.2.

Algoritmus pro procvičování slov se dělí na dvě části. V první části se zjišťuje, jaká slova jsou pro zkoušeného uživatele jednoduchá a naopak s jakými slovy má problém - tato fáze se bude dále nazývat inicializační. Po této fázi následuje fáze s použitím Leitnerova algoritmu - tato fáze bude nazývána testovací.

3.6.1 Inicializační fáze

Na začátku testování jsou k dispozici slova s přiřazenou obtížností. Tato obtížnost je buď globální v případě, že se uživatel nesetkal ještě s daným slovem v aplikaci nebo uživatelská v případě, že uživatel dané slovo již procvičoval. Dle této obtížnosti se slovíčka seřadí tak, aby se střídala nejnižší a nejvyšší obtížnost. Na prvním místě tedy je slovo s nejnižší obtížností, následuje slovo s nejvyšší, dále slovo s druhou nejnižší atd. Důvodem tohoto seřazení je zvýšení motivace dětí. V případě, že by žák dostal na začátku nejtěžší slova, mohlo by si vytvořit odpor k procvičování v domněnání, že slova jsou příliš složitá pro něj.

Uživatel tedy postupně odpovídá na slovíčka. Jeho odpověď je vyhodnocena buď správně, špatně nebo neúplně. Na základě vyhodnocené odpovědi a globální nebo uživatelské obtížnosti je ke slovu přiřazen počet správných opakování pro dokončení slova.

3.6.2 Testovací fáze

Po inicializační fázi následuje fáze testovací. V této fázi již je známo, jaká slova dělají uživateli problém. Oproti standardnímu Leitnerovu systému popsaném v kapitole 2.6.2,

kde na začátku jsou všechny otázky zařazeny do první kategorie, zde se otázky rozdělují do třech kategorií na základě inicializační fáze. Následuje první fáze, kde dochází postupnému procházení otázek v první Leitnerově kategorii od nejtěžší pro uživatele po nejlehčí. V případě správně nebo neúplně zodpovězené otázky, slovo je přesunuto do vyšší kategorie. Následuje druhá fáze, kde se prochází otázky z první a druhé Leitnerovy kategorie. I zde platí, na základě odpovědi dochází k přesouvání otázek mezi kategoriemi, jak už tomu je ve standardním provedení Leitnerova systému. Ve třetí fázi jsou nabízeny otázky z první, druhé a třetí Leitnerovy kategorie. Po průchodu třetí fáze následuje znovu fáze první. Takto algoritmus cykluje do doby, dokud uživatel nepřeruší test nebo nezůstanou žádná slova, které by uživatel nedokončil.

Za dokončené slovo se považuje takové, kde uživatel odpověděl správně nebo neúplně tolikrát, kolik má slovo přiřazených nutných správných opakování, a zároveň poslední odpověď nesmí být neúplná, ale pouze správná. Je to z důvodu, aby bylo ověřeno, že uživatel má správně zapamatovanou informaci.

3.6.3 Adaptivní a globální obtížnost

V případě přerušení testu dochází k ukládání aktuální Leitnerovy kategorie k danému uživateli. Díky tomu lze znovu navázat v případě, že uživatel se bude chtít k testování dané skupiny slov vrátit. Leitnerova kategorie vypovídá, jak problematické je slovo pro uživatele. V případě, že se slovo nachází v první kategorii je problematické a v případě, že se nachází ve třetí, uživatel ho zvládá bez obtíží. Díky této informaci dochází k adaptivnímu generování obtížnosti. V případě, že není u daného slova a uživatele uložena Leitnerova kategorie, je v inicializační části brána globální obtížnost.

Například se může stát, že globální obtížnost slova je v kategorii těžkých, ale uživatel se s tímto slovem už několikrát setkal a dobře se mu pamatuje. Je proto zbytečné slovo mu opakovat vícekrát, jelikož na základě obtížnosti a správnosti odpovědi se generuje počet opakování.

3.7 Připomínání slov

Připomínání slov je důležitou součástí aplikace. Funkce slouží k tomu, aby studenti nezapomněli na již naučená a zvládnutá slova z procvičování. Jelikož se jedná o webovou aplikaci, není úplně možné připomínat slova v přesných intervalech přímo v aplikaci. Způsob implementace připomínání slov je závislý na tom, aby se uživatel sám přihlásil a spustil funkci připomínání slov. Je možné na základě softwarového démonu Cron v určitých intervalech ze serveru odesílat emailová upozornění se slovy, které by si měl uživatel připomenout.

3.7.1 Supermemo algoritmus

Supermemo algoritmus je rovněž další implementace metody rozloženého opakování popsané v kapitole 2.6. První implementace Supermemo algoritmu v papírové podobě byla vytvořena již v roce 1985. V aplikaci je využívání aktuální verze 11, poprvé implementována v roce 2005.

Algoritmus má dvě možnosti implementace - optimální interval a pokročilé opakování [11]. Verze s optimálním interval je pro účely aplikace dostačující, jelikož pro implementaci s pokročilým opakováním by bylo nutné ukládat ke každému uživateli nepřehledné množství dat. Pro správnou funkci je nutné ukládat každou odpověď ke každému ze slovíček, na základě nichž se aplikace adaptuje na schopnosti uživatele. Oba algoritmy je možné si vyzkoušet v podobě desktopové aplikace popsané v kapitole 2.2.5. V aplikaci lze pozorovat změny na uživateli v podobě křivky zapomínání a dalších statistických informací.

Definice spočtení optimálního interval [11]:

$$I(1) = OF[1, L + 1] \tag{3}$$

$$I(n) = I(n - 1) * OF[n, AF] \tag{4}$$

kde:

- n počet již provedených připomenutí
- $I(n)$ je n -tý interval v řadě intervalů pro opakování
- AF je absolutní obtížnost
- L počet, kolikrát si uživatel nemohl vzpomenout na slovo
- OF matice optimálních koeficientů pro zvyšování intervalu
- $OF[1, L + 1]$ koeficient z matice na prvním řádku a $L+1$ sloupci
- $OF[n, AF]$ koeficient z matice, který koresponduje n -tému opakování a obtížnosti slova

Matice koeficientů k výpočtu optimálních intervalů je získána z freeware aplikace SuperMemo 15. Tato matice je statisticky nejvhodnější pro žáky základních tříd. Ukázkou matice znázorňuje následující tabulka 2.

3.7.2 Příklad řady optimálních intervalů

V případě, že si uživatel pokaždé správně vzpomene na naučenou informaci spočtené intervaly pro středně těžké slovo budou přibližně - 2 dny, 6 dní, 14 dní, 27 dní, 49 dní, 82 dní, 131 dní, 7 měsíců, 10 měsíců, 15 měsíců, 21 měsíců, 2,5 roku, 3,5 roku atd. V

	1	2	...	8	9	10
1	2.48	2.22		1.12	1.00	0.89
2	1.20	1.80		5.40	6.00	6.60
3	1.20	1.50		3.30	3.60	3.90
4	1.20	1.40		2.60	2.80	3.00
...						
17	1.20	1.24		1.46	1.50	1.54
18	1.20	1.24		1.45	1.48	1.52
19	1.20	1.23		1.43	1.47	1.50
20	1.20	1.23		1.42	1.45	1.48

Tabulka 2: Matice koeficientů pro výpočet optimálních intervalů

případě, že dojde k zapomenutí slova, tj. uživatel při připomenutí nebude moci slovo vybavit, interval se ze začátku zkrátí a začíná se v matici koeficientů na dalším sloupci - což znázorňuje parametr L v rovnici 3.

3.8 Motivace

3.9 Model databáze

3.10 Architektura aplikace

4 Klientská aplikace

4.1 Návrh aplikace

4.1.1 Single-page

Single page application (SPA) je jedna webová stránka, která poskytuje celou aplikaci. Při prvním načtení je přenesen veškerý kód pro aplikační běh, tj. HTML šablony, JavaScript i kaskádové styly (CSS). Na základě akcí prováděných uživatelem jsou poté dynamicky načítány ostatní zdroje jako data, obrázky apod. Celá stránka se nikdy neobnovuje jako tomu je v klasické webové aplikaci. Veškerá aplikační logika je přenesena ze serveru na klienta za účelem server odlehčit. V případě SPA server většinou po načtení aplikace poskytuje pouze data a to nejčastěji ve formátech JavaScript Object Notation (JSON) nebo Extensible Markup Language (XML).

SPA komunikuje se serverem obvykle dvěma způsoby - web sokety nebo AJAX (asynchronous JavaScript and XML) dotazy. Další technologií pro komunikaci, která by mohla být využita je Server-sent events (SSEs), která umožňuje serveru zasílat data klientovi přes HTTP protokol. Pro účely aplikace bude omezenou pouze na AJAX komunikaci, jelikož pro web sokety a tudíž obousměrnou komunikaci v aplikaci využití není.

4.1.2 Model aplikace

4.1.3 Editovatelné seznamy

4.1.4 Design

4.1.5 Implementace algoritmů

4.1.6 Adaptovaná Levenshteinova vzdálenost

4.1.7 Google API

4.1.8 Zabezpečení

4.2 Vývojové prostředí

4.2.1 Webpack

4.2.2 Babel

4.2.3 JSX

4.3 Knihovna React

4.3.1 Abstraktní DOM

4.3.2 Mobx

4.3.3 Imutabilita

4.3.4 Směrování

4.3.5 Komponenty

4.4 Implementace technologií

4.4.1 Adresářová struktura

4.5 Testování

5 Serverová aplikace

5.1 Technologie

5.1.1 Webová aplikace

5.1.2 Architektura

5.1.3 HTTP a REST API

5.1.4 WSGI

5.1.5 Implementace algoritmů

5.2 Django a REST framework

5.2.1 Autorizace

5.2.2 Autentizace

5.2.3 Optimalizace API

5.3 Zabezpečení

5.3.1 HTTP/2

5.3.2 OAuth2

5.3.3 JWT

5.3.4 CORS

5.4 PostgreSQL

5.5 Testování

6 Závěr

Použitá literatura

- [1] Terasoft a.s. *Terasoft - Výukové programy* [online] 2002-10-07. [cit. 2016-12-15]. Dostupné z: http://www.terasoft.cz/czpages/cd_aj15.htm
- [2] LangSoft s.r.o. *Language Teacher* [online]. [cit. 2016-12-07]. Dostupné z: <http://www.langsoft.cz/teacher.htm>
- [3] KREJČOVÁ Lenka. *Psychologické aspekty vzdělávání dospívajících*. 1. vyd. Praha: Grada Publishing, 2011 [cit 2016-12-18]. ISBN 978-80-247-3474-3.
- [4] BIEMILLER Andrew, BOOTE Catherine. *An effective method for building meaning vocabulary in primary grades*. Vol 98(1), Journal of Educational Psychology, 2006 [cit 2016-12-18].
- [5] LOZANOV Georgi. *Suggestology and Outlines of Suggestopedy*. 1. vyd. Gordon and Breach, 1978 [cit 2016-12-18]. ISBN 0-203-39282-5.
- [6] I. S. P. Nation *Learning Vocabulary in Another Language*. Cambridge University Press, 2001 [cit 2016-12-18]. ISBN 0-521-800927.
- [7] SVOBODOVÁ Jana, SVOBODOVÁ Diana, KULDANOVÁ Pavlína, GEJGUŠOVÁ Ivana, ROSOVÁ Milena, NOVÁK Radomil. *Lexikologie* [online] 2003. [cit. 2016-12-19]. Dostupné z: <http://www.osu.cz/fpd/kcd/dokumenty/cestinapositi/lexikologie.htm>
- [8] Wikimedia Foundation, Inc. *Levenshtein distance* [online] 2016-12-20. [cit. 2016-12-21]. Dostupné z: https://en.wikipedia.org/wiki/Levenshtein_distance
- [9] Google, Inc. *Google Cloud Platform* [online]. [cit 2016-12-21]. Dostupné z: <https://cloud.google.com/>
- [10] Hermann Ebbinghaus *Memory: A Contribution to Experimental Psychology*. New York, 1913 [cit 25-12-2016]. ISBN 978-16-142-7166-6.
- [11] WOZNIAK Piotr A. *Repetition spacing algorithm used in SuperMemo 2002 through SuperMemo 2006* [online] 2006-04-02. [cit. 2016-12-26]. Dostupné z: <https://www.supermemo.com/english/algsm11.htm>