

Sem vložte zadání Vaší práce.

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

**Software pro podporu studia cizích jazyků
pracující s frekvencí výskytu slov
ve studijních materiálech**

Bc. Tomáš Bohuňovský

Vedoucí práce: Ing. Jiří Hunka

5. května 2015

Poděkování

Chtěl bych zde především poděkovat Ing. Jiřímu Hunkovi, mému vedoucímu diplomové práce za věcné připomínky a rady k jejímu zpracování. Dále bych chtěl poděkovat všem dobrovolníkům, kteří se mnou sdíleli své připomínky ve fázi testování i mimo ni.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 5. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Tomáš Bohuňovský. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Bohuňovský, Tomáš. *Software pro podporu studia cizích jazyků pracující s frekvencí výskytu slov ve studijních materiálech*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce řeší aplikaci doslovných překladů textu pro účely výuky cizích jazyků. Jejím cílem je tvorba prototypu, který postupně skrývá překlady jednotlivých slov. Algoritmus vychází z principu prokládaného opakování.

Klíčová slova Závěrečná práce, systém prokládaného opakování, srs, překlad textu, Scala, MongoDB, Play Framework.

Abstract

This thesis deals with application of literal translation of texts for the purpose of learning foreign languages. The objective is a creation of a prototype which gradually hides the translations of individual words. The algorithm is based on the principle of spaced repetition system.

Keywords Thesis, space repetition system, srs, text translation, Scala, MongoDB, Play Framework.

Obsah

Odkaz na tuto práci	viii
Úvod	1
1 Analýza	3
1.1 Psychologie učení	3
1.1.1 Dalův kón	8
1.2 Nástroje pro výuku jazyků	10
1.2.1 Samostudium bez elektroniky	11
1.2.2 Samostudium s elektronikou	14
1.3 Systém prokládaného opakování	16
1.4 Rešerše existujících řešení	18
1.4.1 Anki	18
1.4.2 Memrise	19
1.4.3 Lingua.ly	22
1.4.4 Duolingo	24
1.4.5 Busuu	27
1.4.6 LingQ	30
1.4.7 Lang-8	32
1.4.8 CentralEnglish	33
1.4.9 Tell Me More	34
2 Návrh	37
2.1 Popis aplikace	38
2.1.1 Elektronické formáty	38
2.1.2 Automatický překlad	39
2.1.3 Manuální korekce	40
2.1.4 Přizpůsobení uživateli	40
2.1.5 Export	41
2.2 Návrh aplikace	41

2.2.1	Případy použití	41
2.3	Volba technologií	42
2.3.1	Imperativní vs. Funkcionální přístup	44
2.3.2	Webový framework	45
2.3.3	Databáze	46
2.3.4	Klientská část	48
2.4	Návrh grafického zobrazení výstupu	49
3	Implementace	51
3.1	Databáze	51
3.1.1	Kolekce	53
3.1.2	Dictionary	53
3.1.3	UsersSecure	55
3.1.4	Projects	56
3.1.5	Uložení obsahu	57
3.2	SRS algoritmus	60
3.2.1	SuperMemo	60
3.2.2	Návrh algoritmu	62
3.3	Back-end	64
3.3.1	Model	65
3.3.2	Parsování a uložení EPUB	66
3.3.3	Export a SRS	69
3.4	Front-end	71
3.5	EPUB s překlady	73
4	Testování	75
5	Nástin budoucího směřování projektu	77
	Závěr	79
	Literatura	81
A	Rozhraní pro korektory	85
B	Seznam použitých zkratk a termínů	87
C	Obsah příloženého CD	89

Seznam obrázků

1.1	Bloomova taxonomie	5
1.2	Bloomova taxonomie – fáze	5
1.3	Imitum	8
1.4	Dalevo kón	9
1.5	Nepodložený kón	10
1.6	Ebbinghaus Forgetting Curve	17
1.7	Anki	19
1.8	Memrise metafora	20
1.9	Memrise - ukázka lekce	21
1.10	Lingua.ly	23
1.11	Duolingo – strom znalostí	26
1.12	Busuu – hlavní rozhraní	28
1.13	LingQ	31
2.1	Princip aplikace	39
2.2	Role	42
2.3	UseCase	43
2.4	Návrh výstupu EPUBu	49
2.5	Wireframe - User	50
3.1	Diagram vazeb obsahu	57
3.2	Simulace SM-2	62
3.3	EPUB - ukázka výstupu	74
A.1	Wireframe vystihující intuitivní rozhraní pro práci korektora	85

Seznam tabulek

1.1	Porovnání preferencí studentů CJ dle úspěšnosti	6
1.2	Porovnání času věnovaného studiu CJ ve volném čase dle úspěšnosti	7
1.3	Preferred learning methods of North Carolina	7
1.4	Preferred combined learning methods of North Carolina	8
4.1	Dotazník	75

Úvod

Výuka v posledních dvaceti letech podstoupila řadu změn, kterou zapříčinil především technický rozvoj této doby, a to konkrétně v oblasti elektroniky a multimédií. To nám dovoluje se od některých tradičních způsobů výuky oprotit, a jiné zase obohatit. Přínosem může být časová úspora, snížení finanční náročnosti studia, či individuální a interaktivní přístup.

Inovace v oblasti vzdělání jsou vyžadovány tempem dnešní doby. Jedinci již nestačí institucionálně získané vzdělání na celé životní období. Ani průběžné udržování znalostí již není dostačující. Do dvacátého století bylo vzdělání směřováno, až na výjimečné případy, jednotně. Lidé byli vybaveni základními znalostmi a konkrétní specializace zatím neměla ukotveno své místo. To se ovšem spolu s rozmachem téměř všech myslitelných vědních oborů měnilo.

Dnes, v době narůstajícího počtu různých směrů a specializací, již tento koncept jednotného vzdělání není udržitelný. Alespoň ne v jeho původní podobě. Jednotlivec již ani není schopen pojmout vše, natož v celé šíři. Navíc se již ani nelze spolehnout na vybraný směr učiva, který se může za několik let projevit jako slepý, či být nahrazen něčím zcela odlišným. Tím ovšem nezavrhují určitou nutnost společných dovedností a základní vybavenosti lidstva. A to především do té fáze, od které se již jedinec může na pouť studiem vydat sám.

Jedna z nejoceňovanějších schopností současného člověka je znalostní flexibilita. Takto lze označit člověka schopného se přizpůsobit téměř jakýmkoliv znalostním požadavkům. Netřeba zdůrazňovat, že se jedná o požadavek značně náročný.

Naštěstí existuje široká škála přístupů a nástrojů na zjednodušení učebního procesu. Rozšíření již zmiňované techniky a multimédií přináší do tohoto procesu zcela nový rozměr. Tím asi nejvíce viditelným přínosem je rychlost přenosu informací, a to především za pomoci internetu. Literatura se postupně digitalizuje a stává se tak masově dostupnou. To v konečném důsledku uzavírá onu vývojovou spirálu.

Ovšem existují i jiné, v mnohém pokročilejší, platformy jako internetové

přednášky, školení na dálku, materiál připravený s ohledem na jedince, možnost si kdykoliv cokoliv přehrát či zopakovat. Lze využívat audio, video, text, virtuální komunikace s lidmi s podobným zájmem a plno jiných směrů. Blíže se budu věnovat přínosům pro oblast výuky světových jazyků.

Obecně se v textu, konkrétně v kapitole Analýza 1, zabývám procesy učení a některými z jejich klasifikací. Poznatky dále uplatňuji při popisu metod sloužících k výuce jazyků. Konkrétně výuka jazyků se vyznačuje širokou škálou specifik, které lze s výhodou využít. Internet v této oblasti hraje roli diametrálně významnější než v jakémkoliv jiném oboru.

Tato diplomová práce se dále specializuje na zefektivnění práce s textem. Blíže představím výukovou metodu s efektem dlouhodobého uložení informace do paměti, což je v případě výuky jazyků klíčový bod. Prozkoumám existující metody a produkty využívající inovativních prvků. Na základě nashromážděných informací provedu analýzu a návrh aplikace se zaměřením na výuku jazyků, která se bude snažit využít potenciálu moderních přístupů.

Analýza

Účelem této kapitoly je příprava podkladů pro budoucí návrh aplikace. Zaměřím se na problematiku učení z hlediska způsobu, přístupu a vnímání člověka, a to především z psychologického pohledu. Také prozkoumám efektivitu různých způsobů výuky.

Existuje široká škála metod a nástrojů sloužící k výuce jazyků. Dle zaměření práce se budu zabývat převážně těmi, které pracují s texty. Rozboru podrobím jak metody klasické, mezi něž lze zařadit například papírové slovníky a texty, tak metody využívající elektronická zařízení. U těch klasických též popíši způsob elektronizace.

V poslední fázi provedu rešerši existujících aplikací, které *lze použít* k výuce jazyků. Zaměřím se především na takové, které inovativně využívají potenciálu nástrojů této digitální doby.

1.1 Psychologie učení

Pro naplnění podstaty této práce, tedy navržení a vytvoření prototypu aplikace sloužící k výuce cizích jazyků, je třeba osvětlit funkci samotného procesu učení. Tato problematika je úzce spojena s medicínskými obory. Ty se zaměřují na průzkum mozku jako *fyzického* objektu, kde zkoumají procesy a chemické reakce, přes stránku *psychologickou*, až po pojetí *filozofické*, u kterého se zabývají pojmy jako „vědomí“ či „myšlení“. Je důležité si uvědomit, že výše uvedené dělení je pouze orientační a slouží pro lepší uchopení problematiky. Takzvaný psychologický pohled by bez fyzické části pochopitelně nemohl existovat.

Fyzická stránka se zabývá stavbou, procesy a chemickými reakcemi v mozku. Na toto téma existuje nepřehledné množství publikací. Jednou z mnoha zajímavých prací je „Nervová soustava a funkce mozku“ (Neural Networks and Brain Function)[1], která popisuje vazby a komunikaci jednotlivých částí mozku. Dále například článek zkoumající mechanismy nervové soustavy a

tvorbu reakce na okolní vjemy[2], či práce zabývající se aplikací modelů na nervovou soustavu, modely popisující vnímání a vzájemné provázání [3] nebo článek, který se zabývá grafickým znázorněním vazeb neuronů[4]. Avšak bližší zkoumání této problematiky je za hranicemi tohoto textu.

Psychologický pohled je pro potřeby této práce zcela postačující. Touto problematikou se zabývá znatelně více publikací, výzkumů a průzkumů. To je dáno v podstatně řadě případů tím, že jsou kladeny nižší nároky na vybavení, které je k výzkumu nutné, a v některých případech i na vybavení znalostí. To ovšem neimplikuje nižší důležitost takové činnosti. Výsledky tohoto oboru se promítají do všech lidských oblastí. Od pracovních, přes osobní duševní vlastnictví, až po vzdělávání se. Vše je se vším propojeno. Tato oblast též bude sloužit jako podklad pro analýzu a návrh.

V této zajímavé oblasti je možné nalézt studie zkoumající: vliv okolí na vývoj jedince, vztahy mezi podněty a reakcemi u člověka, znám pod pojmem Pavlovův efekt, mezilidské vztahy, psychologi i řeči, pocity, chutě, deprese, psychologii učení, vnímání barev, vazby mezi barvou a tónem, ego a nepředstavitelné množství dalších věcí. Zaměřím se na psychologii učení.

Jedna z technických definic pojmu „učení“ v oblasti psychologie dle publikace *The Science of Psychology*[5] zní: „Relativně permanentní změna chování získaná zkušeností.“ či „Nabývání znalostí a dovedností samostudiem, zkušenostmi nebo zprostředkovanou naukou.“ dle oxfordského slovníku[6].

Jakožto teoretický vědní obor se snaží o pochopení procesu učení. Z historického vývoje vychází mnoho různých dělení. Vedle *nativistické teorie* – člověk své vrozené dovednosti a schopnosti pouze rozvíjí, *empirické teorie* – učení se smyslovým pozorováním, výsledky experimentů a zkušeností, *skepticismu*, a jiných, patří *behaviorální* či *kognitivní teorie* mezi ty známější.

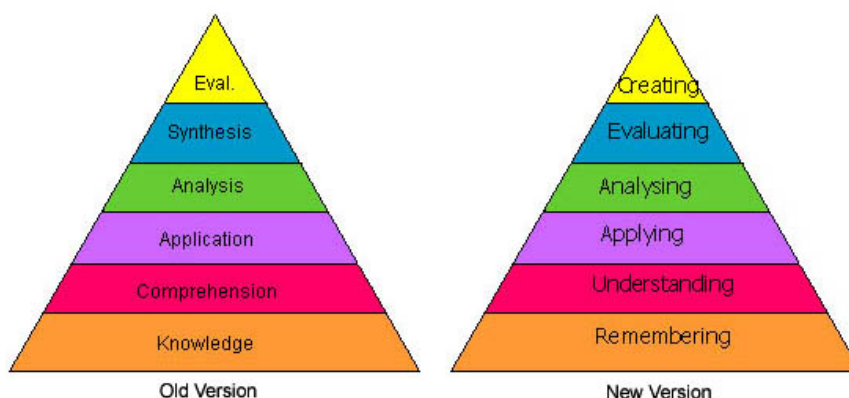
Příkladem behaviorálních je metoda „klasického podmiňování“¹, které je spojováno se jménem Ivan P. Pavlov. Experimenty na psech definoval schopnost učení reflexům při konkrétním podnětu. Zatímco kognitivní teorie si představuje člověka jako systém, který si informace začleňuje a třídí do systému. Též se zabývá způsobem, jakým se s informacemi pracuje.

Kognitivní psychologie je dnes považována za nejvýznamnější proud v psychologii. Je na ní postaveno mnoho teorií. Mezi nejrozšířenější a dodnes uznávanou je tzv. taxonomie fází učení – tzv. Bloomova taxonomie[7]. Rozděluje proces učení na šest částí viz obr. 1.1, přičemž jednotlivé fáze podrobněji vystihuje obrázek 1.2.

V této oblasti existuje široká škála dalších dělení a teorií. Je natolik široká, že lze najít mnoho studií zabývající se přímo „učením jazyků“ či přímo „e-learningem“. Z těchto studií budu dále vycházet.

Navzdory velkému množství zajímavých prací není tato oblast přehledná. Tuto skutečnost odhaluje disertační práce „Strategie učení cizímu jazyku“[8]

¹angl. Classical Conditioning



Obrázek 1.1: Bloomova taxonomie
zdroj: Bloom's Taxonomy[7]

lower order thinking skills			higher order thinking skills		
remember	understand	apply	analyze	evaluate	create
recognizing <ul style="list-style-type: none"> identifying recalling <ul style="list-style-type: none"> retrieving 	interpreting <ul style="list-style-type: none"> clarifying paraphrasing representing translating exemplifying <ul style="list-style-type: none"> illustrating instantiating classifying <ul style="list-style-type: none"> categorizing subsuming summarizing <ul style="list-style-type: none"> abstracting generalizing inferring <ul style="list-style-type: none"> concluding extrapolating interpolating predicting comparing <ul style="list-style-type: none"> contrasting mapping matching explaining <ul style="list-style-type: none"> constructing models 	executing <ul style="list-style-type: none"> carrying out implementing <ul style="list-style-type: none"> using 	differentiating <ul style="list-style-type: none"> discriminating distinguishing focusing selecting organizing <ul style="list-style-type: none"> finding coherence integrating outlining parsing structuring attributing <ul style="list-style-type: none"> deconstructing 	checking <ul style="list-style-type: none"> coordinating detecting monitoring testing critiquing <ul style="list-style-type: none"> judging 	generating <ul style="list-style-type: none"> hypothesizing planning <ul style="list-style-type: none"> designing producing <ul style="list-style-type: none"> constructing

(Table 2 adapted from Anderson and Krathwohl, 2001, pp. 67–68.)

Obrázek 1.2: Bloomova taxonomie - fáze

zdroj: <http://www.celt.iastate.edu/pdfs-docs/teaching/RevisedBloomsHandout.pdf>

paní Mgr. Vlčkové, která osvětluje fakt, že samotné šetření a kategorizování zatím nikdo jednoznačně nedefinoval.

„Nepanuje shoda ani na tom, jak strategie definovat, jakou mají povahu, jak se vyvíjí, zda lze vytvořit validní taxonomii strategií, zda jsou spíše mediátorem nebo prediktorem výsledku vzdělávání, zda jsou skutečně efektivní, nebo které z nich jsou efektivní, za jakých podmínek či zda vůbec je možný transfer strategií atd.“ Mgr. Vlčková[8], strana 6.

Navzdory komplexnosti tohoto problému se ve své obsáhlé práci věnuje mimo jiné popisu a třídění metod a postupů. Blíže je rozebírá v habilitační práci „Žákovské strategie učení cizímu jazyku ve všeobecném vzdělávání (průřezový výzkum)“ [9].

1. ANALÝZA

Mezi popisovanými strategiemi existuje dělení podle způsobu použití znalostí dle Kohena a Weavera (2006) na 4 části,[9] strana 39.

- Vybavování – mnemotechniky, grafické představování v mysli (seznamy, diagramy), vybavením si jazykového cvičení ze školy..
- Opakování a nacvičování
- Komunikační – snaha udržet komunikaci
- Zakrývací – kamufláž slabších stránek pro prevenci diskreditace

Dalším dělením je dle řečových dovedností (strana 39-40)

- poslech
- čtení
- psaní
- mluvení
- slovní zásoba – některé rozdělení zahrnují i gramatiku a překlady

Text také zmiňuje existenci mnoha jiných kategorizací, které se věnují rozšíření některé dovednosti a nepopírá ani existenci překryvu.

Značně jednodušší přístup používá výzkum The learning styles and strategies of effective language learners[10]. Ten přináší srovnání úspěšných a méně úspěšných studentů a jejich metod a přístupů k učení.

Práce odhaluje, že úspěšnější studenti preferují komunikativní styl oproti studentům méně úspěšným, kteří preferují dle terminologie práce autoritativní styl – myšleno výuku zprostředkovanou a předávanou učitelem. Odhaluje, že vlastní zájem je pro výuku téměř nezbytný. Oproti slabším studentům investují mnohem více svého času výuce a zdokonalování, a to především ve svém volném čase. Cíleně vyhledávají a obklopují se cizím jazykem. V tabulce 1.1 je srovnání 110 studentů a jejich preference a investice 1.2 do samovzdělávání.

	Communicative	Authority-oriented	Analytical	Concrete
More effective	41	10	13	1
Less effective	11	12	4	3

Tabulka 1.1: Porovnání preferencí studentů CJ dle úspěšnosti
zdroj: Table 1 : The learning styles and strategies of effective language learners[10]

	Less than 1 h	1–5 h	6–10 h	More than 10 h
More effective	14	31	10	22
Less effective	23	6	4	0

Tabulka 1.2: Porovnání času věnovaného studiu CJ ve volném čase dle úspěšnosti

zdroj: Table 2: The learning styles and strategies of effective language learners[10]

Jak text zdůrazňuje: „Do more effective students spend more time on English out of class because they are good at it, or are they good at it because they spend more time practicing and using it out of class?“

Dále práce odhaluje význam čtení u studentů. Pochopitelně má větší oblíbenost čtení dobrovolných článků doma než ve škole.

Zajímavé výsledky přináší článek Learning Best Through Experience[11], který mezi lidmi provedl výzkum preference způsobu učení^{1.3} a jeho kombinací. Je nutné upozornit, že výsledky jsou založeny na subjektivním vnímání účastníků, a tudíž nemusí odrážet skutečnou efektivitu.

Cooperative Extension clientele		
Learning Method	n	%
Doing	54	70.1
Seeing	14	18.2
Discussing	5	6.5
Hearing	3	3.9
Touching-feeling	1	1.3
Tasting	0	0.0
Smelling	0	0.0
Total	77	100.0

Tabulka 1.3: Preferred learning methods of North Carolina

zdroj: Table 1. Preferred learning methods of North Carolina[11]

Z tabulek 1.3 a 1.4 jasně jednoznačně vyplývá, že samotný akt vykonávání je nejvíce žádaný. To je následováno názornou ukázkou a diskuzí. Zajímavý je výsledek názorů na ústní výklad, který je v pořadí až za diskuzí, což nepřímo potvrzuje výsledky průzkumu práce The learning styles and strategies of effective language learners[10].

Na základě výsledků této kapitoly lze usoudit, že má smysl vytvořit nástroj, který bude sloužit zájemcům o jazyky především při dobrovolné činnosti. Texty je potřeba cílit na konkrétní uživatele s ohledem na individuální zájmy. Zároveň je vhodné se alespoň rámcově držet postupů popsaných Bloomovo ta-

1. ANALÝZA

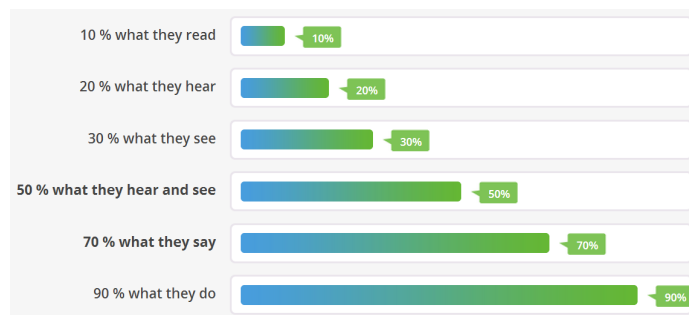
Learning by new Cooperative Extension agents in North		
Preferred Combination	n	%
Seeing/Doing/Discussing	10	37.0
Hearing/Seeing/Doing/Discussing	6	22.2
Seeing/Doing	5	18.6
Hearing/Seeing/Feeling/Smelling/Tasting/Discussing/Doing	2	7.4
Hearing/Seeing/Doing	2	7.4
Doing/Discussing	1	3.7
Doing/Seeing/Discussing/Touching-feeling	1	3.7
Total	27	100.0

Tabulka 1.4: Preferred combined learning methods of North Carolina zdroj: Table 4. Most preferred combinations of methods of learning by new Cooperative Extension agents in North Carolina[11]

xonomií. Jelikož tématem této práce je text, bylo by zajímavé zjistit, nakolik efektivně dokáže člověk text vnímat. Tím se zabývám v následující kapitole.

1.1.1 Dalův kón

V průběhu hledání podkladů jsem několikrát narazil na graf 1.3 (v různých úpravách):



Obrázek 1.3: Imitum - vnímání

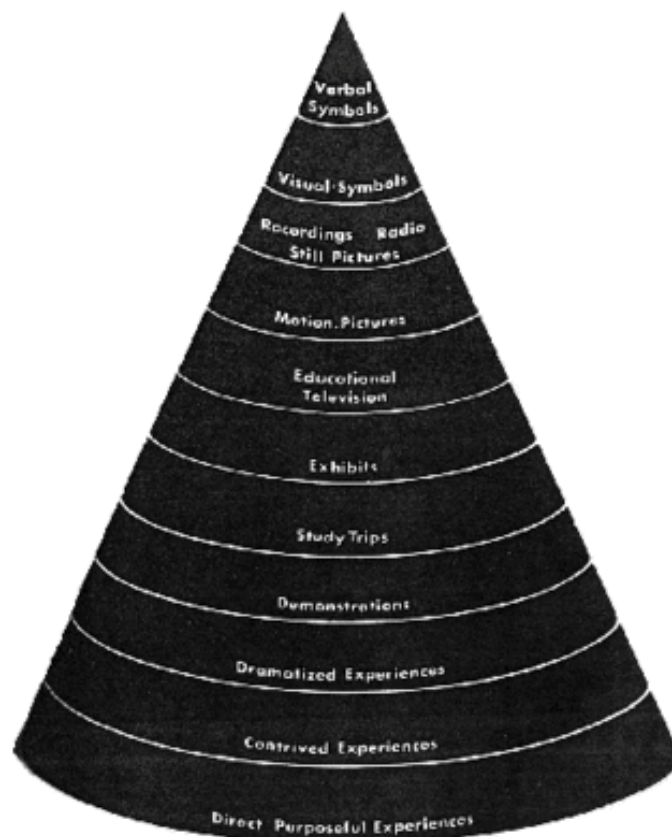
zdroj: <http://www.imitum.com/effective-method-to-learn-language>

Studie odhalující a potvrzující podobné výsledky by byla velice silným podpurným nástrojem při návrhu mé aplikace. Nezbyvá nic, než se na to podívat zblízka. Výše uvedený obrázek jsem našel na stránkách metody Imitum bez uvedení relevantního zdroje. To mě podnítilo k prozkoumání těchto čísel blíže. Našel jsem celou řadu podobných grafů. Ty vychází z tzv. Daleho kónu (Dale's Cone of Experience)1.4. Zajímavým poznatkem je fakt, že původní kónus neobsahuje žádné číselné vyjádření. Jak uvádí článek People remember 10%, 20%...Oh Really[12], který se též zabývá hledáním původu oněch čísel: „Ono vyčíslení není věrohodné. A i kdyby dávalo smysl, pořád může být

nebezpečné“².

„If we look at the numbers a little more closely, they are highly unconvincing. How did someone compare *reading* and *seeing*? Don't you have to *see* to *read*? What does *collaboration* mean anyway? Were two people talking about the information they were learning? If so, weren't they *hearing* what the other person had to say? What does *doing* mean?“

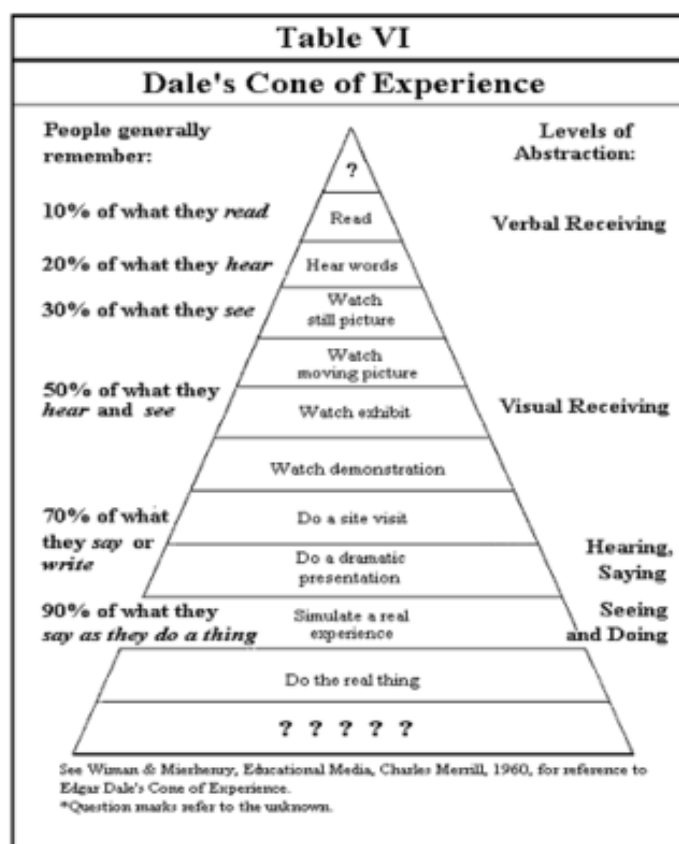
Dle článku se postupem času ony „výsledky“ rozšířily do ostatních prací, díky pouhému neověřenému přejímání z jiných publikací. Zajímavý je i fakt, že samotný Dale varoval své čtenáře, aby jeho model nebrali doslovně.



Obrázek 1.4: Dalevo kón
zdroj: Dale's Cone of Experience (Dale, 1969, p. 107)

Snaha zjistit nakolik efektivně vnímá jedinec text se objevila jako neprobádaná. Nenašel jsem jediný relevantní zdroj, na kterém bych mohl postavit základy. Budu tedy předpokládat, že je to věc značně individuální.

²*nebezpečné – na ta čísla spoléhat. pozn. autora



Obrázek 1.5: Nepodložený kón
zdroj: willatworklearning.com[12]

1.2 Nástroje pro výuku jazyků

Existuje široká škála postupů a nástrojů sloužící k výuce. Jazyky mají samy o sobě specifika, které možnosti ještě rozšiřují. Od zprostředkované výuky agenturou či učiteli, přes poslech hudby či mluveného slova, sledování filmů, přímou komunikaci s rodilým mluvčím, hry, aktivním zapojením se do zahraničních komunit, až po samostudium knih. A jak vyplývá z kapitoly zabývající se psychologií učení 1.1, nejlepší výsledky dosahovali studenti, kteří o látku měli zájem, jelikož se dobrovolně obklopovali jazykem ve svém volném čase. To je jistou formou samostudia.

Obklopit se jazykem lze mnoha způsoby. Avšak pro potřeby této práce se budu zabývat pouze těmi, které pracují především s textem. Opodstatnění tohoto vymezení vychází kromě zadání i z průniku řečových dovedností [9] popisované v kapitole psychologie učení 1.1 a textem. Jedná se o kategorie – čtení, psaní a slovní zásoba.

V následující kapitole záměrně používám slovo *metody*. Ačkoliv se vlastně

jedná o různé formy prací s textem, budu je považovat za atomární dělení, ke kterému se v samotných rešerších nástrojů budu odkazovat.

1.2.1 Samostudium bez elektroniky

V dobách neznalých dnešních možností byl papír, neuvažují-li rytiny, stavby či monumenty, téměř jediným spolehlivým nositelem informace. Není proto divu, že existuje široká škála materiálů v knižní podobě. Většina těch zaměřených na výuku cizích jazyků je určena pro samouky. Jsou jimi slovníky, mluvnice, cizojazyčné texty a jiné.

1.2.1.1 Slovníky

Nezbytným nástrojem při samostudiu jsou slovníky. Ty se dají dělit na mnoho různých kategorií. Od klasických překladových, přes slovníky nářečí a krajů, hovorové, až po slovníky odborné terminologie. Zaměřím se na tři hlavní skupiny – výkladové, překladové a obrázkové.

Výkladové slovníky obsahují *výklad* čili slovní popis daného slova v původním jazyce. Některé slovníky, tzv. thesaurus, jsou obohaceny o synonyma či nastiňují použití slova v určitých vazbách a kontextu. Navíc by měl čtenář být schopen vyčíst drobné nuance mezi rozdíly synonym. Vhodné jsou především při tvorbě překladu textu do cizího jazyka nebo rozšiřování slovní zásoby. Onen výklad zabraňuje, oproti překladovým slovníkům, špatnému pochopení významu. Bohužel jsou nevhodné pro začátečníky, protože výklad je v cílovém jazyce.

Překladové slovníky slouží především k přímému vyhledání překladu. Vzhledem k tomu, že neobsahují výklad slova, jsou vhodné především pro nalezení konkrétních slov při překladu do rodného jazyka nebo připomenutí slov, které již čtenář zná. Jako nevhodné se dají klasifikovat při překladu do cílového jazyka právě díky chybějícímu popisu použití a kontextu. Příkladem může být věta „Oni jsou chytrí“, kterou může dle slovníku *chytrý - deep* přeložit jako „They are deep“, což neodpovídá zamýšlenému sdělení.

Obrázkové slovníky či učebnice jsou jednou z forem publikací pro výuku jazyka. Bývají zpravidla zaměřeny pro začátečníky, ačkoliv dle mého názoru bývají nedocenenou součástí studia. Velkou výhodou těchto materiálů je vizuální názornost, kterou většina jiných nástrojů postrádá. Dnes lze nejčastěji narazit na materiály typu „Mých prvních 1000 slov“. Takové knihy jsou dělané formou znázornění každého slova, které bývá doprovázeno slovním označením, překladem a výslovností. Tato forma výuky slovní zásoby přispívá k lepší asociaci a zapamatování.

1.2.1.2 Gramatika

Knihy o gramatice pro samouky jsou zpravidla děleny dle úrovní. Jejich forma bývá postavena na cyklu vysvětlení určité části gramatiky spolu s nezbytně

nutnou slovní zásobou. Poté se získané znalosti demonstrují na příkladech či jednoduchém textu. A nakonec završeno ověřovacím testem. V následujících cyklech se nabalují postupně získané znalosti. Tento typ knihy je možné doporučit i začínajícím studentům.

Středně a výše pokročilí studenti, kteří se již někdy seznámili se složitějšími konstrukcemi jazyka si mohou pořídit knihu shrnující gramatiku jako celek. V našem prostředí jsou tyto publikace známy pod pojmem *mluvnice*. Takové knihy jsou vhodné spíše pro doplnění znalostí a cílené vyhledávání konkrétních pravidel a jejich výjimek. Bývají strukturované dle slovního určení od podstatných jmen přes slovesa, jež zahrnují často hledané časování, až po citoslovce. Jednou takovou, již století starou, zajímavou knihou zabývající se anglickým jazykem je „Manual of English Grammar and Composition“ [13].

1.2.1.3 Dvojjazyčné texty

Dvojjazyčné texty, označované též jako texty bilingvální, patří v oblasti literatury pro samouky za nejrozšířenější. Jedná se o knihy, které mají na jedné straně text v původním jazyce a na straně druhé překlad. Za důležitý považuji zmínit fakt, že se jedná o překlad do značné míry volný, což komplikuje samotné učení. Ona volnost významně ztěžuje orientaci při hledání významu konkrétního slova. Často se zde vyskytuje silná slovní nekoheze originálu a překladu. Tento problém je zvláště znát při překladech tzv. idiomů. Příkladem může být slovní spojení „Sharp as a tack“ - „Ostrý jako břitva“. Zde ono slovo *tack* neodpovídá slovu *břitva*, ale *cvoček/připínáček*. Navíc čtenář nemusí odhalit tuto nesrovnalost, a může slovo *tack* asociovat se slovem *břitva*. To je pochopitelně nežádoucí.

Další nevýhodou je nutnost se oprostít od čteného textu, a dané spojení či větu hledat na druhém listu/straně. Tento přeložený text již z podstaty nebývá stejně rozmístěn, což je při hledání konkrétního slova další matoucí faktor. Ono předposlední slovo na třetím řádku může být v překladu jako první slovo na řádku čtvrtém. Kromě toho je takový způsob přecházení mezi stránkami časově náročný, což v konečné fázi svádí ke čtení pouze překladu, a to není účelem.

Tuto formu tedy nepovažuji pro samouku za vhodnou. Doporučit se dá snad jen čtenářům pokročilým, kteří se mohou přiučit zvyklostem v překladu, a rozšířit si tak rozmanitost vnímání textu.

1.2.1.4 Redukované texty

Další metodou jsou takzvané *abridged* texty. Jedná se o zkrácenou podobu původního textu, kde se navíc může korigovat rozsah a použití slov. Jde tedy o knihu, která vystihuje stejný příběh, stejný děj, ale je přepsána určitou slovní zásobou. To umožňuje studentovi přesun od teorie k textu za kratší časové

období. Navíc se dříve dostane do praktického kontaktu s jazykem, kde může pozorovat další jevy, například gramatiku v praxi.

Příkladem může být kniha, jejíž slovní zásoba se skládá z pouze 1500 slov, které tvoří veškerý text. Toto dává oněm slovům prostor pro demonstraci použitelnosti v různých tvarech a významech. Čtenář si slova dobře procvičí a opakuje. Po dočtení může přejít ke knihám s více slovy, a postupně se tak zlepšovat. Ve finální fázi čtenář končí u původního nezkráceného textu.

Mezi negativa se dá zařadit jistá strohost textu. Slovní omezení eliminuje širokou škálu synonym, což způsobuje, že čtený text postrádá jistou rozmanitost a bohatost. Nicméně toho si povšimne spíše čtenář pokročilý.

Dále bývá problémem rozlišit různá vydání různých nakladatelství. A to především kvůli neunifikované definici slov v jednotlivých úrovních. Další obtíží je dostupnost takto upravené literatury.

1.2.1.5 Cizojazyčné texty

Poslední možností se jeví práce s originálním textem. Tento způsob je nevhodný pro úplné začátečníky. Neznalost základních slov značně zvyšuje nutnost hledání ve slovníku. Začátečník by se v takovém případě pohyboval více ve slovníku než v textu, ze kterého by neměl žádný požitok, ba dokonce by ho to mohlo odradit od studia. Někdy je možné z kontextu odvodit význam slova, ale nikoliv vždy. Zde je již nezbytnou potřebnou mít po ruce slovník. V určité fázi zdatnosti již máte natolik dobrou zásobu, že můžete nechat slovník doma nebo si vyhledat slova po návratu.

1.2.1.6 Doslovný překlad

Otázkou tedy zůstává způsob zabudování překladu významu jednotlivých slov přímo do textu, tak aby to nebylo rušivé a zároveň bylo rychle dostupné. I na toto již existuje řešení a to napsat překlad onoho slova rovnou pod text. Můžeme k tomu použít dva přístupy. Prvním je překládání každého slova. Druhým je překlad pouze neznámých či dostatečně neprocvičených slov.

Nejblíže se tomuto řešení přibližuje metoda *ORIGINAL*, která se vyskytuje například v sérii knih *White Fang* od *Jacka Londona*. Tyto knihy se vyznačují tím, že čtenář začíná knihu číst s kompletním překladem, a jak se postupně pročítá příběhem, čímž si zároveň opakuje některá slova, tím překlad řídne. Respektive slova často opakovaná se již více nevyskytují. Cílem je dočtení knihy v téměř originálním jazyce. Na tuto knihu následuje další pokračování s převzatou slovní zásobou z knihy první.

Ovšem i tato metoda má své nedostatky. Prvním a tím nejhlavnějším je značně omezená škála knih. Druhým je nemožnost přizpůsobit náročnost a intervaly opakování pro různé čtenáře.

Z výše uvedených metod dvojjazyčného textu vychází nejlépe řešení s překladem pod slovem. Toto řešení má v případě elektronizace velký potenciál.

Umožňuje automaticky předpřipravovat různé verze knih s různou složitostí. Dále je možné vytvořit určité série knih jako celky.

1.2.1.7 Specializované časopisy

Na trhu existují specializované časopisy různých jazykových zaměření. Časopisy zaměřené především na větší odběry (školy) jako například Bridge[14] a časopisy pro širokou veřejnost, které jsou dostupné i u drobných distributorů, typu EasyEnglish.

Tyto produkty mají zajímavou koncepci. Forma je uzpůsobena především pro mladší ročníky, čemuž je přizpůsoben jak grafický styl, tak i obsahová náplň článků a značné množství zábavných kvízů a her.

Asi nejpodstatnější částí jsou samotné články, které jsou tematicky zaměřené na aktuální dění. Podané bývají ve zjednodušené formě 1.2.1.4. Od politiky, přes sport a kulturu, až po vědu. Tématům se hranice rozhodně nekladou. Tyto články mívají na konci shrnutí, které obsahuje zajímavosti, zvláštní gramatické jevy či slovní spojení. Tuto formu mají např. časopisy typu Bridge.

Časopis typu EasyEnglish byl založen na postupném vydávání jednotlivých navazujících dílů, a to nejen z hlediska správných odpovědí, ale spíše se stupňující obtížností náplně. Pochopitelně se začínalo od úplných základů a tematických jevů. Lze to přirovnat způsobu čtení klasické knihy.

1.2.1.8 Metody slovní zásoby

Zajisté je na místě zmínit metody věnující se slovní zásobě. Avšak z důvodu rozsáhlosti tohoto tématu věnuji problematice vlastní kapitole *Systém prokládaného opakování* 1.3.

1.2.2 Samostudium s elektronikou

V této chvíli jsem již představil většinu *klasických* nástrojů samostudia. Nyní se podívám na metody moderní. Ze zaměření této práce vyplývá, že to budou metody využívající elektroniky. Současná doba nám dává ohromný prostor k inovacím. Je to dáno především prudkým rozvojem techniky za období posledních 20 let. Pochopitelně plno nápadů již vzniklo dávno v minulosti, nicméně až dnes jsou technologie k efektivní realizaci. A co víc, plno dříve nedostupných technologií je dnes již rozšířeno masám a staly se každodenní samozřejmostí.

Abych se vyhnul znovupopisování každé z metod uvedených v kapitole Samostudium bez elektroniky 1.2.1, implicitně lze předpokládat, že veškerý text lze převést do digitální podoby. V případě slovníků uvažuji i interaktivní zpracování, kdy je možné filtrovat a přímo vyhledávat.

Není již tedy třeba přenášet a listovat v tlustých slovnících, zařízení již dokáží rozpoznat hlas nebo ho rovnou převádět na text – Speech-to-Text. Také je možné si nechat text přečíst – Text-to-Speech. Navíc máme mocný nástroj zvaný Internet. Ten nám exponenciálně rozšiřuje možnosti aplikací.

Počínaje cloudovými úložišti, kde lze uschovat všechny soubory a tudíž není třeba před cestou rozmýšlet, kterou z oblíbených knih zabalit.

1.2.2.1 Elektronická čtečka

Elektronizace se v současné době dostává do popředí a s ní i digitalizace textu. Spolu s elektronickými čtečkami, jinak též e-book či e-reader, jež jsou zařízení schopná zobrazovat digitální textové záznamy. Vzniká zde prostor pro knihy nové generace. Nezáleží, zdali máte knihu o stovce stran či velká díla o tisíce stranách, protože váha a rozměry budou vždy stejné. Jejich hlavní výhoda tkví v téměř neomezené kapacitě³. Displej se vlastnostmi blíží skutečnému papíru (včetně barev) a energetická náročnost je takřka zanedbatelná⁴. Navíc dnešní čtečky toho umí samozřejmě mnohem víc. Standardní výbavou bývají offline slovníky, především výkladové. V případě napojení na internet jsou schopny dohledávat širší informace, umožňují nákup nových knih na dálku, nebo být jen základní prohlížení internetu.⁵ Dále jsou schopny přehrávat audio soubory, a to buď přes externí (sluchátka) nebo vestavěný zvukový aparát. Ovládání je zajištěno externími tlačítky nebo gesty na dotykové obrazovce.

Z pohledu výuky jazyků se toto zařízení perfektně hodí pro čtení zjednodušených a původních textů a děl. V případě potřeby, se jednoduchým označením vybere slovo, které se automaticky vyhledá ve slovníku a poskytne výklad jeho významu. Je to velice rychlá a nerušivá operace, která odbourává nutnost psát či hlasově vyhledávat (S2T). Navíc to umožňuje eliminovat externí zdroje (např. slovníky). Dále lze toto zařízení využít například k přehrávání audioknih.

Avšak méně vhodné jsou pro četbu dvojjazyčné literatury, a to především díky výchozímu nastavení zobrazení jedné strany přes celý displej. Nastavení zobrazení dvou stran na displej nebývá schůdným řešením. Text je v takovém režimu vzhledem k rozměru zobrazovací plochy příliš malý. Další slabší stránkou je obtížnější a pomalejší listování, což naznačuje, že se toto zařízení nehodí pro publikace, které se nečtou po stránkách. Encyklopedie, receptáře či seznamy.

1.2.2.2 Tablety a jiné zobrazovací prostředky

Běžné zobrazovací prostředky mívají téměř okamžitou překreslovací dobu a znatelně vyšší výpočetní výkon. To umožňuje rychlou manipulaci a přepínání dokumentů, či přehrávání videa. Pochopitelně na takových zařízeních je též možné číst knihy v digitální podobě. Rychlost odezvy umožňuje rozmanitější

³rok 2014 – řádově jednotky tisíc knih v offline zařízení, s přístupem na internet neomezeně

⁴v případě e-ink a podobných technologií

⁵již ze své podstaty nejsou vhodné k zobrazování pohyblivých či graficky náročných multimédií

aplikace, listování, hledání v nestrukturovaných textech či zobrazování barevného obsahu. Avšak i toto zařízení má negativum a to především v nutnosti podsvícení, což značně zkracuje dobou výdrže baterie⁶. Výše zmíněné elektronické čtečky lze považovat za speciální případ zobrazovacích prostředků.

1.2.2.3 Online komunikace

Asi nejvýznamnějším polem, kde se moderní technologie dostávají ke slovu je Internet. S tím souvisí okamžité propojení se světem, z čehož vyplývá možnost bezmezné komunikace. Není těžké se obklopit rodilými mluvčími či studenty z celého světa. Takzvaná online komunikace zahrnuje jak psané konverzace (elektronickou poštu, instantní messangery, . . .), mluvené (voice komunikátory) či grafické (video)komunikátory. Všechny tyto prostředky dnes fungují v takzvaném real-time režimu.

To umožňuje odbourat tzv. simulované rozhovory, které jsou časté ve školách, kdy spolu žáci rozmlouvají nad daným tématem. Výhodou je možnost hovořit o tématech blízkých a také to, že se jedná o skutečný nenucený obyčejný rozhovor. Hovoří-li student dobrovolně přes internet, lze předpokládat, že ho to i naplňuje a baví. Což vede ke snaze se zlepšovat na jazykovou úroveň protistrany. Mnohdy k tomu dochází podvědomě bez většího přičinění.

Dokonce existují servery, které mají za cíl zprostředkovat kontakty studentům mezi sebou (viz rešerše 1.4). Lze si vybrat, ze které země má protistrana pocházet, či si předem vybrat téma diskuze. Navíc relativní anonymita internetu umožňuje odhodit stud.

1.3 Systém prokládaného opakování

V oblasti výuky slovní zásoby a memorování existuje celá škála rozmanitých podpůrných nástrojů. Nicméně mezi nejúčinnější patří ty, které jsou založeny na takzvaném principu prokládaného opakování. Anglické znění této metody je více vypovídající – Spaced Repetition System, dále jen SRS.

Metoda vychází z nikterak nápaditého způsobu učení se – opakování. To je s lidstvem od nepaměti. Samotné metodiky opakování mají nepřehledné množství reprezentací. Vědomost je opakována staršími či zkušenějšími (např. děti se takto nevědomě učí od rodičů). Další častou metodou je účelné zapisování vědomostí na kartičky či audio záznam. Student si je pak vědomě opakovaně prohlíží, a tím si znalost opakuje. Je to metoda, která lze použít na libovolného jednotlivce. Cílem opakování je zesílení vjemu dané informace.

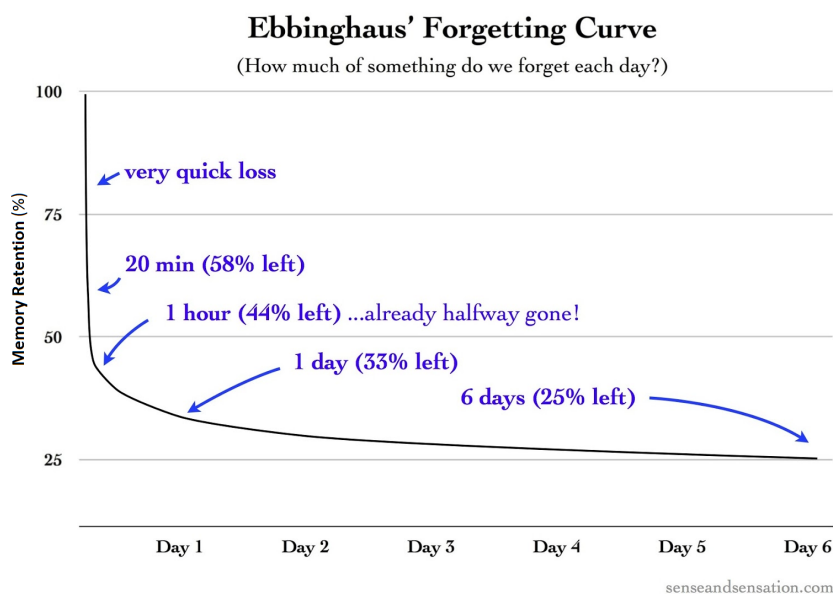
Na podobném principu funguje i metoda vypisování informací či tvoření si „taháků“. Úsilí při tvoření těchto prepisů má dvojitý efekt. Jednak se student musí aktivně zamýšlet nad obsahem, a také ho pochopit. Poté je schopen

⁶nejnovější e-readry již mají možnost volitelného podsvícení, především pro čtení v šeru - avšak s menší intezitou

vybrat ty nejdůležitější informace. Jejich následným zápisem dochází k zesílení vjemu dalším opakováním.

Na zcela opačné straně stojí metoda studenty nazývaná jako „biflování“ (angl. cramming). Jedná se o jeden z nejefektivnějších způsobů pro naučení se velkého kvanta znalostí během velice krátké doby. Nejčastěji slouží při nutnosti složení jednorázové zkoušky či testu. Ovšem tato metoda má své největší úskalí v retenci získaných informací v dlouhodobé paměti. Z dlouhodobého praktického hlediska se jedná o metodu značně neefektivní.

Vyvstává tedy otázka, zdali je opakování efektivní a jestli ho ovlivňuje četnost vystavování se informací. Čas v dnešní době patří mezi nejdražší komoditu, a tak není divu, že se koncem 19. tého století začal Hermann Ebbinghaus[15] zajímat o vztah četnosti opakování a pamatování si.



Obrázek 1.6: Ebbinghausova křivka znázorňující retenci informace v paměti
zdroj: <http://www.senseandsensation.com/2013/03/retrieval-getting-and-forgetting-part-8.html>

Roku 1885 definoval pan Ebbinghaus takzvanou „Ebbinghausovu křivku zapomínání“⁷, jež je znázorněna na obrázku 1.6. Ta odhaluje skutečnost, že pokud není znalost obnovována, dochází ke ztrátě znalostí z paměti exponenciálním tempem. Také definoval, že znovu-naučení trvá mnohem méně času a úsilí.

Výsledkem bylo zjištění, že nemá smysl opakovat vědomosti se stejnou frekvencí. Obecně lze říci, že informaci je vhodné opakovat těsně před zapomenutím. Tedy znalost „oživit“. Opakování později by zničilo efekt, a opakování

⁷ v anglické terminologii známa jako Ebbinghaus Forgetting Curve

dříve by zase bylo plýtváním času – neefektivní. Odtud pramení onen název – systém prokládaného opakování – Spaced Repetition System.

Na základy těchto formálních objevů později navázaly další práce odborné veřejnosti. Cílem některých prací bývala snaha sestavit *křivku zapomínání* mnohem přesněji. Jinými slovy se snažili o konkrétnější parametrizaci. Jiné práce se snažily aplikovat znalost do praktického řešení. Vznikla tak některá doporučení, postupy a návody využívající SRS.

Největším úskalím SRS v praktickém použití byly příliš vysoké požadavky na režii. Požadovalo to od studenta jistou dávku sebekázně a odhodlanosti. Bylo potřeba si ručně vyrábět znalostní karty, sám si stanovovat kdy bude další opakování, sebehodnocení, atd... Tyto činnosti se dnes delegují na techniku. Tudíž jediným úkolem studenta je věnovat čas samotnému učení a opakování. Takovými nástroji jsou například Anki1.4.1 či Memrise1.4.2.

1.4 Rešerše existujících řešení

Pro ucelení náhledu na problematiku se v této kapitole zabývám některými z existujících produktů sloužící k výuce jazyků. Jedná se o produkty, které z převážné většiny využívají principu SRS, nebo přináší zajímavý inovativní aspekt. Takovým prvkem může být netradiční přístup k tvoření obsahu či komunita v roli korektorů. Pro úplnost a zvýraznění kontrastu uvedu na konci této kapitoly jeden z komplexních, avšak klasických multimediálních, nástrojů1.4.9.

1.4.1 Anki

Tato volně dostupná aplikace slouží jako univerzální rozhraní pro cyklické opakování znalostí, jež plně využívá principů SRS.

Jak již bylo zmíněno, aplikace je pouhým rozhraním, které spoléhá na činnost komunity v oblasti tvorby obsahu. Ten je prezentován ve formě znalostních balíků. Vzhledem k univerzálnímu zaměření aplikace jsou k dostání balíky z různých oborů. Mezi nimi nechybí biologie, chemie, zeměpis, zákony, fyzika a jiné. Znalost je předávána libovolnou formou. Může se jednat o obrázek, audio či video nahrávku, text nebo jejich kombinace. Komunita kolem tohoto nástroje je poměrně silná a zahrnuje i univerzitní sféru, kdy někteří univerzitní učitelé tvoří pro své studenty vlastní balíky, např. pro výuku japonštiny.

Hlavní činností aplikace je přidělování správných prodlev k jednotlivým znalostem, určení nové znalosti k naučení a zprostředkování časových odhadů opakování. To vše provádí v závislosti na parametrizaci a spolupráci studenta.

Učení i opakování má stejný průběh. Nejprve je zobrazena otázka či úkol. Student ji vypracuje a následně si nechá odkrýt odpověď. Po svém vlastním zhodnocení zvolí obtížnost dané otázky (jednoduché, střední, těžké). Také má



Obrázek 1.7: Procvičování pojmu s manuální evaluací obtížnosti – ankisrs.net

možnost požádat o zopakování. Zvolená odpověď přímo odpovídá časové prodlevě mezi dalším opakováním. Čím těžší se studentovi znalost zdá, tím dříve přijde na řadu. Nástroj přímo zobrazuje konkrétní časovou prodlevu, která bude nastavena, což se v porovnání s ostatními produkty jeví jako netypické. Toto zachycuje screenshot 1.7 pořízený přímo z aplikace.

Produkt nabízí širokou nabídku parametrizace na míru. Od vlastnoručního rozdělení časových rozestupů, určení kolik opakování za den, maximální rozestupy, časový limit, až po reakci na chybu. Pochopitelně lze vybírat mezi režimy opakování – pouze zapomenuté, s předstihem či se věnovat pouze znalostem novým.

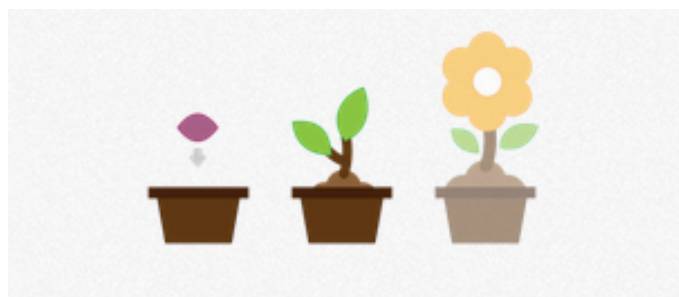
1.4.2 Memrise

Tato webová aplikace staví na podobných principech jako Anki 1.4.1. Avšak zpracováním se výrazně liší. Cílí na vytvoření platformy pro výuku nejen jazyků, ale i jiných předmětů učení. Jak název napovídá, aplikace se snaží zpříjemnit ukládání informací do lidské paměti. Memrise staví na principu memorování, které probíhá ve třech fázích. První fází je naučení se nové znalosti následovaného postupným opakováním, dokud se nedostane do dlouhodobé paměti. Její autoři používají příznačné přirovnání procesu učení ke květinám (viz obr. 1.8).

Jedná se o cloudovou službu, která je v základní verzi zdarma. Aplikace poskytuje spolu s webovým rozhraním i klienty pro mobilní zařízení. Registraci lze provést i přes Facebook účet, což rozšíří aplikaci o soupeřivý aspekt, který je popsán dále.

Na výběr je ze široké škály různých kurzů. Ty jsou rozřazeny dle jazyků a kategorií. Mezi kategoriemi se vyskytují například jazyky, umění a literatura, matematika a věda, historie, zábava, atd. . . Každá kategorie má pochopitelně ještě podkategorie. Účastnit se lze vícera kurzů.

Kurzy jsou vnitřně uzpůsobeny do jednotlivých celků, které ovšem nemusí



Obrázek 1.8: „We use a light gardening metaphor to help you understand the ways your learning grows and withers in the brain depending on how you look after them. So, when you learn on Memrise, A “flower” must first be planted in the mind, and then you water (review) these memories that have been planted to ensure that they do not eventually wilt and disappear from your long-term memory.“ – memrise.com

odpovídat náročnosti. Začíná se procvičováním nových znalostí, které se následně opakují. Samotné učení ve standardním nastavení přidává pět nových znalostí, které jsou opakovány v 25 úkolech. Takto se postupuje tak dlouho, až je probrána celá lekce. Kurzy i lekce lze různě přeskakovat. Důvodem je nevýznamnost učení se věcem, které již student ovládá. Úkoly se skládají z disciplín typu asociace obrázek-slovo, slovo-slovo, překlad, výběr, skládání věty z předpracovaných částí a jiné.

Výuka je založena na memorování. To ovšem neznamená dril. Memrise používá inteligentní způsob učení, kdy vyučovaného vystavuje pojům v různých časových odstupech. Rozestup mezi dny, kdy je potřeba slovo zopakovat se zvětšuje s každým úspěšným zopakováním. Memrise uvádí, že opakovat vše je zbytečné. Klíčem je identifikovat co je třeba procvičit a kdy. Tím se šetří čas. Jak to vypadá v praxi lze vidět na screenshotu z aplikace 1.9

Memorování je možné upravovat dle situace a časových možností. Na výběr jsou tři přednastavené módy (nových slov/úkolů k procvičování).

- Standardní (5/25)
- Pokročilé (10/50)
- SuperPokročilé (20/100)
- Vlastní

U každého kurzu se sleduje dosažené hodnocení, úroveň a porovnání s ostatními účastníky kurzu. Díky propojení se sociální sítí FB lze sledovat pokrok v plnění úkolů v porovnání s přáteli, což slouží jako další motivace.

Služba je postavena na modelu, který spoléhá na tvorbu obsahu komunitou. Každý uživatel si může vytvářet kurz pro své soukromé účely, například

The screenshot shows the Memrise interface for Level 1, items 1-15. A progress bar is at the top, followed by 'Learning Options' and an 'Overwater' button. Below, there are 15 items, each with an English word, its Czech translation, and a water drop icon indicating the time to learn. The items are: quagmire (bažina, in about a day), mite (roztoč, in 6 days), rigid (přísný, tuhý, in 6 days), atrocious (ukrutný, krutý, in 12 hours), exquisite (nádherný, vynikající, in 6 days), immense (obrovský, ohromný, in about a day), accomplished (vynikající, kvalifikovaný, in 12 hours), spotless (bez poskvrny, čistý, in 12 hours), squalid (špinavý, bidný, in 12 hours), parched (vyprahlý, okoralý, in 12 hours), keen (bystrý, dychtivý, in 12 hours), jubilant (rozjásaný, radostný, in 12 hours), scalding (vařící, in 11 hours), ravenous (žravý, hladový, in about a day), and vivacious (temperamentní, čiperný, in 12 hours).

English	Czech	Time to learn
quagmire	bažina	in about a day
mite	roztoč	in 6 days
rigid	přísný, tuhý	in 6 days
atrocious	ukrutný, krutý	in 12 hours
exquisite	nádherný, vynikající	in 6 days
immense	obrovský, ohromný	in about a day
accomplished	vynikající, kvalifikovaný	in 12 hours
spotless	bez poskvrny, čistý	in 12 hours
squalid	špinavý, bidný	in 12 hours
parched	vyprahlý, okoralý	in 12 hours
keen	bystrý, dychtivý	in 12 hours
jubilant	rozjásaný, radostný	in 12 hours
scalding	vařící	in 11 hours
ravenous	žravý, hladový	in about a day
vivacious	temperamentní, čiperný	in 12 hours

Obrázek 1.9: V pravé části se nachází časový odhad, kdy bude znalost připravena k opakování (zavlažení) – memrise.com

pro naučení se správnému přiřazování hlavních měst ke státům. Takový kurz lze následně zveřejnit, a tím ho zpřístupnit ostatním uživatelům Memrise.

Asi největší potenciál má nástroj pro sblížení žáků a učitelů. Vyučující se může rozhodnout připravovat kurzy za pomoci této služby, a následně je zpřístupnit svým žákům. Ti se mohou učit touto zábavnější formou, která se inteligentně stará o každého individuálního účastníka kurzu. Učitel na oplátku dostává výsledky a statistiky jednotlivých žáků. Na těchto podkladech může vyučující stavět výuku a učinit jí hodnotnější.

Nevýhodou se s přibývajícím počtem kurzů může stát jistá redundantnost či překryv náplní kurzů. Je to dáno samozřejmě onou otevřeností k vytváření kurzů bez schvalovacího procesu (vyjma kurzů tvořených Memrise). Tento přístup otvírá prostor pro kurzy poskytující nepravdivé informace či rovnou kurzy falešné. Lze se jen těžko vyvarovat vtipálkům, kteří vytvoří kurz vyu-

čující „fakt“, že „hlavním městem České republiky je Bratislava“. Do určité míry lze nekvalitní kurzy rozpoznat dle počtu účastníků a hodnocení kurzu. Nicméně v současné době se jedná o poměrně novou službu, která si své zájemce neustále získává.

Nabízí se jisté srovnání s aplikací Anki1.4.1. Oba produkty přenechávají tvorbu obsahu na komunitě, a mají zakomponován SRS systém. Liší se převážně v ovládacích prvcích a možnostech parametrizace. Anki nabízí mnohem širší možnosti nastavení, avšak s každou novou možností se snižuje intuitivnost produktu. Oba nabízejí grafické zobrazení pokroku a jinou retrospektivu. Nicméně zásadní výhodou Memrise je centrální, kvalitně a přehledně zpracovaný přístup přes web⁸, bez nutnosti lokální instalace produktu a následného ručního přidávání balíků.

1.4.3 Lingua.ly

Další z uváděných služeb, oceněnou věhlasnými jmény jako NewYorkTime, The Guardian, TechCrunch a jinými, je Lingua.ly. Aplikace je zaměřena na výuku slovní zásoby a memorování. Nicméně její přístup je razantně odlišný od již zmíněných. Některými charakteristikami a myšlenkami se vzdáleně blíží Memrise1.4.2.

Služba je zdarma dostupná široké veřejnosti. Ba dokonce přímo cílí na jednotlivce. Přístup do aplikace probíhá skrze internetový prohlížeč, kde se spolu s nainstalovaným add-onem mění v překvapivě mocný nástroj. Další možností je přístup přes rozhraní pro mobilní zařízení.

Unikátnost této aplikace tkví v jejím přístupu, tvorbě a nabídce obsahu. Lingua.ly totiž nenabízí obsah vůbec žádný. Ano, je to nástroj pro učení slovní zásoby, ale nenabízí nic jako kurzy či lekce, zanedbám-li volitelnou ukázkovou sadu. Na otázku „Z čeho se tedy bude student učit?“ lze odpovědět poměrně jednoduše „Z toho, co si uživatel sám přidá do svého slovníku v aplikaci“. Je to do značné míry pochopitelný krok. Lze jen těžko tvořit lekce dle nějakého hodnocení úrovní. Každý jedinec se nachází v jiné fázi učení jazyka, čemuž odpovídá i různá slovní zásoba. Dokonce i certifikovaní studenti na stejné úrovni doznávají značných rozdílů, co se týká rozptylu slovní zásoby. Student techniky má znalost jiných slov než student medicíny.

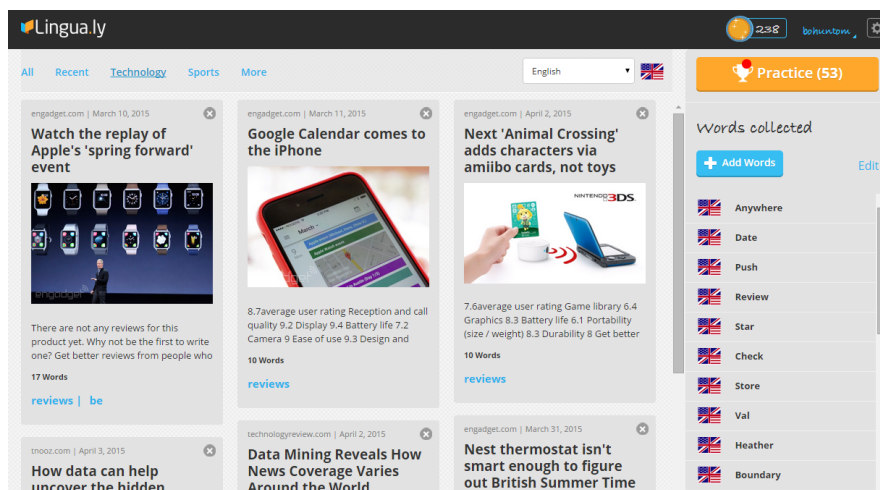
Řekne-li se „ruční přidávání slov“ či „vlastní tvorba slovníku“, na mysl okamžitě nastupuje představa rutiny. Nejprve přidání slova, jeho významu, výslovnost, případně asociační obrázek. Takových aplikací je nepřeberné množství a nepřinášejí nic nového. Navíc by takový program mohl být nahrazen libovolným tabulkovým procesorem. Z praxe je navíc známo, že největším nepřítelem takových řešení je lenost samotných uživatelů, kteří by po velice krátké době přestali nová slova přidávat.

⁸Anki též nabízí webové rozhraní, avšak jedná se o naprosto neporovnatelný produkt

Tato rutina je však do značné míry zjednodušena. Nyní stačí zadat pouze slovo, které se toužíte naučit, a aplikace vše ostatní udělá za vás. Význam je dohledán ze slovníků, které má aplikace sama k dispozici. Zároveň tak popis. Lingua.ly dokonce obohatí slovo o fonetický zápis a audio záznam jeho výslovnosti. Nakonec, pro podporu učení, je ke slovu přiřazen obrázek, který je s významem slova spjat. Přidat lze pochopitelně více slov najednou. Ačkoliv je toto usnadnění v časovém porovnání se starými metodami přínosem, stále to nedostačuje tempu dnešní doby.

Lingua.ly to vyřešilo velice elegantně, a zároveň efektivně. Běžný uživatel stráví nejvíce času pohybem na internetu. Ať již zjišťováním nějakých informací, dopisováním na komunikačních serverech či jen pročítáním zajímavostí. Tato každodenní činnost konfrontuje studenta s tisíci slovy, aniž by si to uvědomoval. A toho služba využívá.

Obyčejným označením slova⁹ na libovolné webové stránce se na obrazovce okamžitě objeví nerušivý překlad slova i s výslovností a zároveň je slovo přidáno do slovníku na webu Lingua.ly. Tato procedura nikterak nezdržuje ani neodvádí pozornost od pročítané stránky. Toto je ten pravý způsob jak téměř automaticky plnit slovník. Tímto nenápadným způsobem se přidá dostatečný počet slov, která si lze následně procvičovat. Při následném brouzdání internetem se ve webových stránkách zvýrazňují slova, která jsou obsažena ve slovníku uživatele, což nepřímo nabádá se snažit vzpomenout na jeho význam.



Obrázek 1.10: Rozhraní aplikace Lingua.ly. Vpravo se nachází nejnověji přidaná slovní zásoba a vlevo se nacházejí články, které obsahují daná slova. – lingua.ly

Po přihlášení se uživatel dostane do velice jednoduchého rozhraní (viz obr.1.10). To je rozděleno na dvě části. Ta první, dominantní, představuje rozhraní zobrazující články z internetu, které obsahují slova ze slovníku. To

⁹ dvojklik na browseru, tap na smartphonech

slouží jako demonstrace použití slov v reálném kontextu. Články jsou velice zjednodušené. Obsahují pouze název, ilustrační obrázek a text, obvykle v rozsahu jednoho odstavce. Na výběr jsou články z bohatého spektra kategorií. Aktuality, technologie, byznys, vzdělání, kultura, vaření, zdraví až po náboženství.

A ve druhé části se nachází seznam slov ve slovníku s možností editace a přezkoušení. Cvičení oněch slovíček probíhá vesměs podobně jako u konkurenčních nástrojů. Prvním krokem je volba jednoho ze tří módů cvičení. Ten neznačí nic jiného, než počet slov k procvičení. Poté se spustí samotné testování, kdy aplikace zobrazí slovo s otázkou, zda-li ho uživatel zná. Pakliže nezná, je odhaleno. V opačném případě ho z něho vyzkouší – dojde k zobrazení pojmu spolu s přehráním audiozáznamu výslovnosti. V časovém intervalu šesti vteřin musíte vybrat z osmi možností správný překlad či významové vysvětlení. S postupně se snižujícím časovým limitem se snižují možnosti na čtyři.

Automatický výběr slov k procvičování probíhá inteligentně. Problematická slova se zobrazují častěji než slova bezproblémová. Stejně tak se aplikace snaží dodržovat určité „zaležení“ slov v paměti. To je systém jako v Memrise 1.4.2 kdy mezi procvičováním slova drží určité časové mezery.

Tato jednoduchost rozhraní je bohužel vykoupena omezenou informativní hodnotou. Nelze sledovat, jaká je dosažena znalost u konkrétních slov. Navíc nelze vybrat, které slovo chci do testu zahrnout či alespoň vynutit opakování. Zároveň se vyskytuje problém s hovorovými výrazy, které slovníky většinou neobsahují. Takové slovo nelze přidat. Dokonce ani ručním vepsáním. To vyplývá z podstaty automatického vyhledávání ve slovníku, o němž píše výše v této kapitole.

Nesporná konkurenční výhoda tohoto řešení tkví v její unikátnosti a efektivnosti tvorby vlastní slovní zásoby. Není problém použít aplikaci na cestách.

1.4.4 Duolingo

Duolingo je aplikace, která se platformou řadí mezi cloudové aplikace (viz slovník pojmů). Přístupovat k této službě lze přes klasický webový prohlížeč či speciální aplikace pro smartphony a tablety.

Za zmínku stojí poměrně zajímavá obchodní strategie této služby. Aplikace si klade za cíl naučit co největší počet lidí novým jazykům. A to zcela zadarmo. Vybírat lze z velkého počtu jazyků (viz dále). Poté, co se studenti naučí novým znalostem, od určité úrovně lze přejít do tzv. Immersion módu (doslova přeloženo jako „ponořit se“). V tomto módu je studentům zprostředkován reálný text v cílovém jazyce, který mají za úkol přeložit. Onen přeložený text je odkoupen zadavatelem.

Překlad textu prochází několika fázemi. Nejdříve je překládán, na čemž pochopitelně pracuje libovolný počet lidí. Dále dochází k několikanásobnému zkontrolování a pravděpodobně ještě k nějakým interním úpravám.

Jak již bylo zmíněno výše, aplikace je zadarmo. Stačí se tedy zaregistrovat pomocí emailu či některé z hlavních sociálních sítí (FB,G+). Následuje volba cizího jazyka a může se ihned začít. Aplikace umožňuje zapsání libovolného počtu kurzů, tudíž se není třeba volby obávat.

V českém prostředí aplikace neposkytuje momentálně žádné kurzy. V přípravě je kurz angličtiny, který se zanedlouho dostane do fáze testování. Jelikož obsah kurzů vyrábějí a rozšiřují dobrovolníci z řad uživatelů, dá se po přílivu lidí na anglický kurz očekávat značné rozšíření nabídky i jiných jazyků. Z tohoto důvodu testuji tuto aplikaci z pozice anglicky mluvícího uživatele (rodný jazyk).

Díky tomu, že anglické kurzy doplňují lidé z celého světa, mám zde na výběr z velkého množství jazyků. Mohu volit francouzštinu, němčinu, itaštinu, portugalštinu, holandštinu či irštinu, turečtinu, ruštinu a jiné. V blízké době lze pochopitelně očekávat i jazyky jako čínština, japonština či korejština. Z této široké nabídky jsem si zvolil německý jazyk (cizí jazyk).

Po výběru kurzu přechází uživatel na hlavní stránku, které dominuje potencionální strom¹¹ zobrazující jednotlivé úrovně jazyka, kterými je třeba projít. Samozřejmě s každým kurzem se liší i strom znalostí, což je dáno jednak složitostí učeného jazyka, a také výsledkem časové investice komunity. Němčina disponuje značně propracovaným a rozsáhlým počtem úrovní.

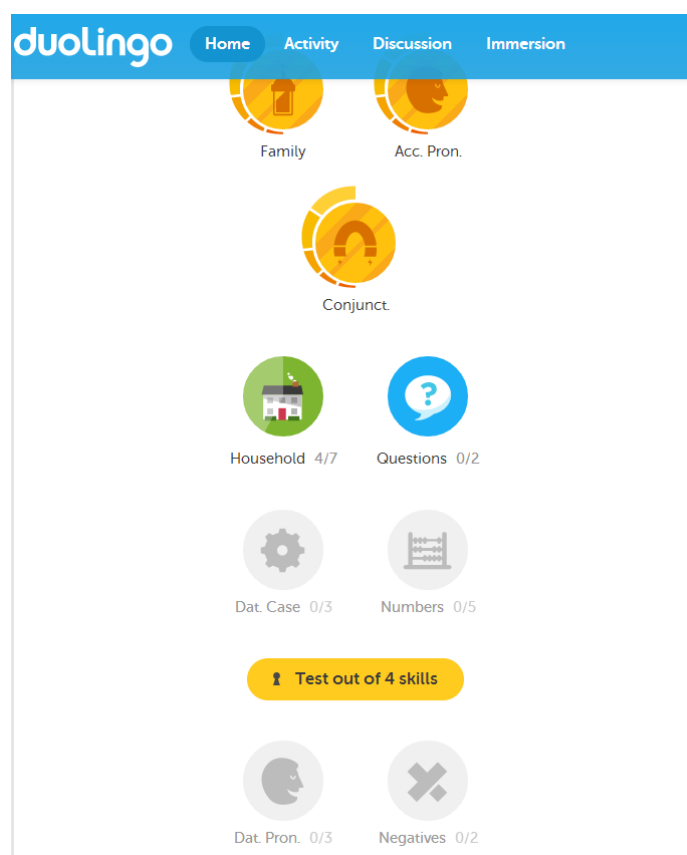
Cílem je se od základu propracovat až k vrcholu stromu. Průchodem se postupně nabalují znalosti a ztěžuje úroveň. Nakonec se nám podaří strom zdolat. Po zvládnutí bychom měli být schopni základní konverzace na ulici, restauraci či letišti. Dle názorů studentů německého kurzu, kteří kurz dokončili, odhaduji zdatnost na úrovni B1¹⁰.

Na úplném začátku studia jsou probírána základní slovesa – být a mít, slovní zásoba – potraviny, zvířata, fráze (dobrý den, na shledanou, . . .), množná čísla, poté následuje přirozený přechod ke složitějším časům a tvorbě otázek, atd. . .

Aby se student dostal na další patro, musí splnit vše na patře předcházejícím. Za splnění každého úkolu jsou přiděleny tzv. zkušenostní body (EXP), jež dále určují jeho úroveň. Za splnění větších tematických celků jsou vypláceny odměny v podobě cihliček (lingots), o nichž píše dále. Aplikace sleduje pokročilost slovní zásoby. K dispozici jsou též grafy pokroku. Dále aplikace sleduje tzv. „streaky“, které udávají, kolik dní se student věnuje učení bez přerušování. Minimální požadovaná aktivita za den odpovídá splnění jednoho úkolu. Celý model je podpořen soutěživým faktorem, kdy se lze porovnávat s ostatními uživateli, ale hlavně s přáteli – propojení přes sociální síť. Pro takzvané falešné začátečníky, to jsou ti, co se jazyk již někdy v minulosti učili, ale vždy začínají od začátku, aplikace nabízí řešení. Aby předešla otravnému

¹⁰Úrovně definované organizací Council of Europe (COE), která si klade za cíl sjednotit označování úrovní napříč evropskými jazyky. Tzv. CEFR - The Common European Framework of Reference for Languages

1. ANALÝZA



Obrázek 1.11: Strom znalostí angličtiny v Duolingo – duolingo.com

vnučování úplných základů těmto lidem, nabízí v určitých úrovních „vyrovnávací test“. Ten obsahuje souhrn znalostí z pater pod touto úrovní. Po jeho splnění jsou všechna předchozí patra otevřena a uznána jako splněná.

Výše jsem popsal strom a význam pater. Patro neboli úroveň se skládá z různých témat. Tato témata se skládají z různého počtu konkrétních testů. Lze si představit, že na patře máme tři témata. Množná čísla, zvířata, oblečení. Oblečení se však může skládat z například šesti testů. Každý z testů seznamuje s jinými slovy z kategorie oblečení. Až po splnění všech šesti testů se stává téma „oblečení“ splněným.

Test se skládá z 20 úkolů, což časově odpovídá přibližně pěti minutám. Tudíž se nejedná o časově náročnou činnost. Úkoly jsou složeny z různých disciplín. Překlad z němčiny do angličtiny a naopak. Každá věta v cizím jazyce je přečtena do sluchátek, tudíž se student učí i poslechu a výslovnosti. Další disciplínou je diktát vět, který ověřuje poslech. Student tvoří doslovný přepis. Nahrávku je možné pustit vícekrát a to ve dvou režimech – normální a zpomalený. Další významnou součástí je praktické mluvení. Student má za úkol přečíst či zopakovat větu dle poslechu. Nebo dokonce přeložit a ústně sdělit

větu do aplikace. Záznam je následně vyhodnocen. Lze se setkat s doplňováním správného tvaru nebo výběrem všech správných možností. V úrovních určených k procvičování slovní zásoby se setkáváme s asociacemi obrázků \leftrightarrow *slovo*.

Aplikace se snaží neustále na studenta mluvit, a zároveň ho nutí též hovořit. Nevědomě se tak učí (driluje) automatické odpovědi či tvary sloves, aniž by si toho aktivně všímal. V pokročilejších úrovních si student začne uvědomovat, že ačkoliv se neučil konkrétní skloňování a tvary, umí je používat. Celý postup působí velice přirozeně.

Prvkem, který zjednodušuje učení, je bezesporu okamžité zobrazení překladu při označení daného slova. Slova v textech se odlišují barvami. Červeně jsou zvýrazněna slova nová a černě známá. Je-li tedy úkolem přeložit či přečíst nějaký text, existuje možnost zobrazení překladu každého slova, synonyma či bližší popis. To eliminuje nutnost používat slovníků či nástrojů třetích stran, kterými by si student v opačném případě dopomáhal. Celou výuku tak urychluje.

Na splnění testu jsou vymezeny tři povolené chyby. V případě neúspěchu je třeba opakovat test znovu. Aplikace nesleduje jen dosavadní pokroky, ale hlídá, které z témat nebylo dlouho procvičováno. Podle toho doporučí některé z již zvládnutých témat zopakovat. Naštěstí není třeba procházet každý test v tématu, ale jen zažádat o souhrnný opakovací tematický test.

Tolerance k překlepům u drobností je též milým překvapením. Chybu detekuje, upozorní na ni, ale uzná. Tudíž neodradí v případech, kdy došlo k drobnému překlepu v rychlosti a zápalu při odpovídání, nikoliv neznalosti samotné. Vlastně i tento systém promíjení chyb přispívá k větší svižnosti a učení se chybami.

Lingoty, jak jsem na začátku zmínil, jsou jakousi měnou, za kterou lze koupit různé drobné bonusy. Těmi mohou být třeba i nestandardní lekce, jako jsou „flirtování“ či „hovorové idiomy“. Dále je možné přikoupit toleranci jedné chyby pro jeden test nebo zmrazit jeden neaktivní den, kdy se vám „streak“ při neaktivitě nepřerušuje. Možností je samozřejmě mnohem více.

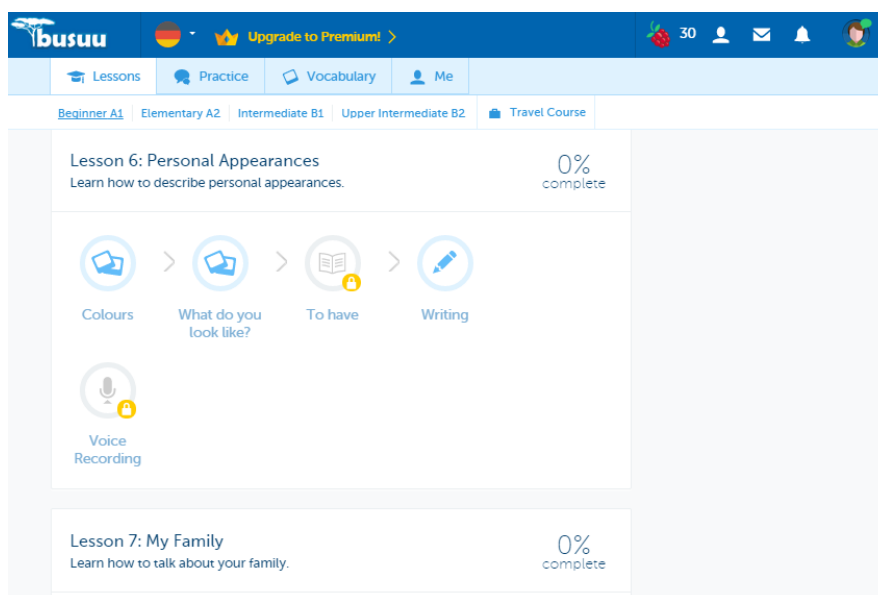
Duolingo získalo několik ocenění a bylo označeno za startup roku v oblasti vzdělávání 2013[16].

1.4.5 Busuu

Ve výše popsaných řešeních se občas vyskytují slova typu sociální síť a komunita. Ve většině případů byla tato slova přítomna v popisu podpůrných nástrojů daného produktu. Jednalo se spíše o propojení lidí za účelem soutěživosti. V Busuu, jehož exotický název pocházející z Kamerunčiny znamená *jazyk*, je tomu zcela naopak. Tato služba by bez komunity jen stěží obstála.

Služba staví na zajímavé myšlence. Tato aplikace se dá do určité míry považovat za Facebook, který ovšem nepropojuje přátele, ale zpravidla spojuje neznáme lidi.

1. ANALÝZA



Obrázek 1.12: Za povšimnutí stojí i fakt, že Busuu skutečně vychází z FB – busuu.com

Na výběr je z přibližně 12 jazyků, mezi nimiž se čeština nevyskytuje. Test tedy probíhá ve variantě angličtina (rodný jazyk) a němčina (cizí jazyk). Po přihlášení se uživatel nachází na hlavní stránce¹¹, kde se nacházejí příspěvky, které ovšem nejsou jeho či jeho „neznámých“ přátel, ale představují jednotlivé kurzy a cvičení^{1.12}. Tematicky jsou velice bohaté – volby a politika, sblížení s přírodou, vesmír, kdybych vyhrál loterii,...

Kurzy jsou rozděleny dle úrovně A1-B2 (standardní klasifikace). Dále poskytují tematicky zaměřený kurz na cestování či byznys. Mezi kurzy i cvičeními je možné libovolně přecházet, nicméně je potřeba mít na paměti jejich případnou návaznost. To především v nižších úrovních, kde jsou rozdíly výraznější.

Cvičení mohou mít čtyři různé části. První je procvičování a výuka slovní zásoby za pomoci tzv. flashcards¹². Zde je k dispozici obrázek s textovým překladem a zvukovým doprovodem. K tomu je zároveň poskytnuta ukázková věta, což napomáhá pochopit kontext použití. Další je práce s textem. Zobrazený originální text je doprovázen zvukovou nahrávkou. Zároveň se vedle textu nachází volný překlad ve stylu bilingválních knih^{1.2.1.3}. Tento text je zprůhledněn, aby neodváděl pozornost. Poté následuje cvičení na přiřazování správných slov do vět právě přečteného článku. Nakonec následuje opakování. Dalším typem cvičení je skládání slov do správného pořadí.

Busuu též myslí na zdokonalování dovednosti psaní. Principiálně funguje

¹¹konotace termínem *zed* v prostředí Facebooku

¹²V původním významu se jedná o papírové kartičky, které obsahují na jedné straně pojem a na druhé straně vysvětlení.

na individuálním popisu tří, často tematicky souvisejících, obrázků a odpovědí na doplňující podpůrné otázky, například: „Považujete rozhodnutí soudního systému vždy za správné?“. Napsaný text je odeslán komunitě ke kontrole. Jedná se o velice zajímavý princip, kdy uživatel není omezen určitou podmožinou správných odpovědí, což umožňuje různé experimenty a hříčky s jazykem. Nicméně je třeba zdůraznit, že složitost a rozmanitost odpovědi si volí každý student individuálně. Záleží pouze na jeho svědomí.

Na stejném principu fungují takzvaná média. Ty představují další podnět ke psaní. Podkladem může být opět předložený obrázek či reálné video, většinou umístěno na YouTube. Úkolem je napsat nějaký komentář, popis či názory.

Naprosto stejný průběh má voice recording. V něm se uživatel stává aktérem v simulovaném dialogu. Průběh dialogu je předem vytyčen konkrétními texty. Úkolem je pouze zopakovat či přečíst onen text. Výsledek je opět odeslán ke kontrole komunity.

Nápomocné souhrny gramatiky a vestavěné slovníky jsou samozřejmostí.

Prvkem, se kterým jsem se v ostatních řešení nesetkal, je vytyčení konkrétního cíle. Ten si volí uživatel sám. Aplikace v návaznosti na nastavení připraví výukový plán tak, aby cíle bylo možné dosáhnout. Jedná se především o plánování časové, tedy ve smyslu „Kolik toho chcete zvládnout za týden?“ či „Kolik času denně věnujete výuce?“.

Hlavní síla aplikace ovšem nestojí na výše popisovaných způsobech výuky, ale na komunitním ekosystému. Ten se nejvíce poodhalí v módu korekce.

Na přehledové stránce korekcí je k dispozici bilance mezi *nám* a *námi* provedenými korekcemi. Uživatel, stejně jako ostatní účastníci sítě, představuje komunitu. Základní myšlenka je vskutku jednoduchá. Studentova cvičení kontrolují rodilí mluvčí. A na oplátku student kontroluje cvičení těm, kteří se učí jeho rodnému jazyku. Poměr mezi korekcemi snižuje či zvyšuje prioritu pro kontrolu vašich úkolů. V mém uvažovaném případě kontrolují cvičení psaná či mluvená anglickým jazykem. Aplikace však umožňuje přístup ke všem jazykům, tudíž mohou opravovat i jiný jazyk.

Oprava probíhá podobně jako odpověď k úkolu. Korektor si prohlédne zadání a odpověď. Poté napíšete do textového pole svůj názor, zmíníte a zdůvodníte chyby, případně poradíte jak dál. Komplexnější a názornější úpravy zde umožňuje do jisté míry rich-editor. Výkon či snaha tazatele o opravu je hodnocena hvězdičkami.

Když jsem výše zmiňoval, že se jedná svým způsobem o sociální síť, tak jsem neměl na mysli jen vzájemné opravování. Po vzoru zaběhlých sítí tu je k dispozici funkce přidávání přátel, posílání zpráv i instant chat, profily atd. To pochopitelně implikuje možnost zdokonalování se běžnou komunikací, která je dle mého názoru hodnotnější. Umožňuje se totiž oprostít od zaběhlých výukových konstruktů a spisovných frází, a tak se dostat i do prostředí skutečně mluveného jazyka.

Co aplikaci chybí je zabudovaná hlasová či video komunikace. Vnitřní chat se tedy často stává nástrojem pro předání si kontaktů na jiné komunikační prostředky typu Skype, kde účastníci pokračují. Dalším, mnohem složitějším oříškem k rozlousknutí je kvalita, korektnost a objektivnost hodnocení ostatními uživateli. Vše funguje na dobrovolnosti a důvěře. Opravy s komentářem jsou do značné míry vypovídající a studentovi nápomocné. Hvězdy ovšem nemají žádnou vypovídající hodnotu. Hodnocení kromě faktu, že není nikterak sjednoceno, může odrážet i momentální rozpoložení korektora. Zároveň není nikterak zaručeno, že opravující je opravdu rodilý mluvčí, či jen o trochu pokročilejší student, než jste vy sám. To samé ovšem platí i pro studenty. Náročnost si určují sami. Pokud chtějí úkol odbýt, není problém napsat jednu větu základní slovní zásobou. Pravdou jest fakt, že škodí pouze sami sobě. Dalším takovým problémem je početní vyváženost jednotlivých mluvčích. Anglická cvičení jsou opravena do pár minut, kdežto na němčinu se může čekat i pár hodin či dnů.

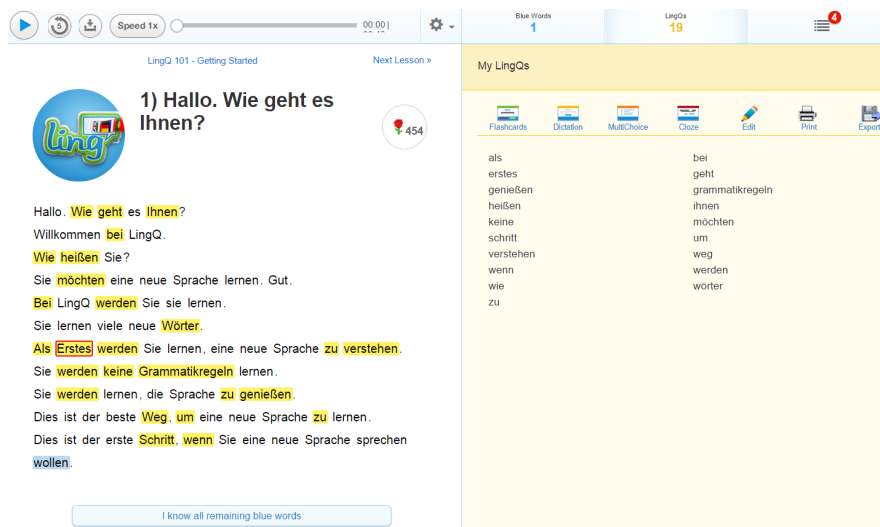
Závěrem snad jen říci, že Busuu staví na velice zajímavém konceptu. Ačkoliv samotné výukové materiály nepoužívají nějakých inovativních přístupů, podařilo se jim zabudovat do cvičení individuální kreativitu a přiblížily studenty s podobnými zájmy k sobě.

1.4.6 LingQ

Následující službu lze opět označit za sociální síť. Dokonce je jí v širším pojetí než Busuu1.4.5. Výuka jazyků je založena na poslechu, který je doprovázen textem.

Kurzy jsou rozděleny do úrovní. Kurz znamená tematicky ucelený soubor menších cvičení. Před vstupem do cvičení lze sledovat, kolik nových slov obsahuje. Typicky jsou témata rozdělena na menší úseky tak, aby cvičení obsahovalo několik desítek nových výrazů. Tedy takový počet, aby cvičení neodradilo složitostí.

Samotné cvičení, jehož podobu znázorňuje obr.1.13, probíhá formou poslechu s doprovodným textem v cílovém jazyce. Takovému textu zpočátku čtenář nerozumí, což vůbec nevádí. Každé slovo lze označit a tím si nechat vypsát informace o slovu, jeho překladu, výslovnosti či použití ve větě. Úkolem je se naučit nová slova. Tedy přemístit je ze skupiny neznámých slov mezi slova známá. K tomu slouží možnosti jako přidat slovo mezi slova k opakování, u nichž lze ještě nastavovat prioritu (nakolik slovo znáte). Tato slova jsou přidána do slovníku, kde je lze procvičovat za pomoci SRS metody1.3. Slova, která jsou již dostatečně procvičena nebo je čtenář zná z dřívějších lekcí, lze ignorovat. Tento postup se opakuje, až je čtenář seznámen s celým textem. Pochopitelně je možné nechat zobrazit kompletní překlad v rodném jazyce, nicméně se doporučuje spíše pro kontrolu a ověření, že výraz či sdělení je korektně pochopeno.



Obrázek 1.13: Průběh výuky - audio nahrávce sekunduje text (vlevo), a pomocné nástroje (vpravo). – lingq.com

Text doprovází nahrávka, kterou Lingq doporučuje opakovaně přehrávat nejen při aktivním učení s Lingq, ale i při cestování, sportu, atd... K tomu nabádá nejen tutoriál, ale i tlačítko s možností přímého stáhnutí nahrávky pro offline použití.

Slovník poskytuje k procvičování několik módů. Prohlížení flashcards, diktát, kdy je slovo přehráno a úkolem je ho napsat, multiple-choice test, „cloze“ test, což znamená doplňování chybějících slov do kontextu. Dále je možné slovník různě editovat.

Stejně jako v Busuu lze přepínat jazykový kontext. Není problém přepnout do režimu výuky jiného jazyka. Možnosti lze využít jak pro výuku, tak pro korektora.

Sociální stránka aplikace vychází z konceptu Facebooku, kdy jsou studentovi na hlavní stránce zobrazeny příspěvky jak jeho, tak ostatních účastníků sítě. Ty je možné komentovat, sdílet či hodnotit. Hodnocení probíhá předáváním virtuálních růží.¹³ Komentáře či příspěvky lze psát v libovolném jazyce. Téma není předurčené, tudíž lze psát o všem co přijde na mysl.

Ještě před odesláním příspěvku je k dispozici několik šikovných nástrojů. Prvním je automatická kontrola pravopisu. Ta zkontroluje, zdali text neobsahuje chyby způsobené překlepem, nikoliv však slovosled. Dále je možné k příspěvku přiložit audio nahrávku, která zpravidla obsahuje sdělení namluvené hlasem studenta. Dále lze připojit standardní multimediální soubory jako videa, obrázky či běžné odkazy. Takové jsou možnosti obsahu. Není to ovšem zdaleka vše, co před odesláním lze udělat. Důležitou součástí je žádost

¹³ Analogie „to se mi líbí“ z Facebooku

o konkrétní z oprav: obsah psaní, ohodnocení výslovnosti (z přiložené audio nahrávky), překlad, atd. . . Po odeslání již čekáte na odpověď či opravy.

Lingq též propojuje lidi. Propojuje lektory, kterými jsou jak běžní lidé z komunity nebo opravdoví certifikovaní lektori. Výuka lektory je řízena nabídkou. Lektor si libovolně určí kdy má čas, jak dlouho věnuje lekci a co bude náplní. To je zaneseno do diáře, který je všem dostupný. Student si vybere a kontaktuje lektora. S ním si může ještě doladit náplň. Vlastně se lze domluvit na čemkoliv. Celá procedura je dobrovolná. Lektorem se pochopitelně můžete stát i student. Může vyučovat například svůj rodný jazyk. A to se nemusí jednat o výuku gramatiky, ale třeba „jen“ neformální rozhovor. Výuka je zprostředkována pomocí videohovoru či pouze hlasově pomocí technologie Skype. Lingq zde slouží jako přímý zprostředkovatel kontaktů.

Zajímavým přístupem je způsob opravování textu, kdy korektor má nástroje pro zvýraznění chyb (škrtnání, zvýrazňování,..), může rozdávat body a zároveň hodnotí procenty jednotlivé disciplíny. Těmi jsou například předložky, slovesa, volbu slov, atd. Celé toho hodnocení je poté vykresleno přehledným grafem. To umožňuje studentovi snadněji identifikovat slabé stránky a poskytnout tak návod na zdokonalení.

1.4.7 Lang-8

Tento tokijský projekt byl založen dvěma osobami. Ze všech popisovaných produktů se jedná o čistou sociální síť. Znamená to, že neposkytuje výukové materiály k žádnému jazyku. Respektive veškerý materiál a podklady tvoří příspěvky a komentáře napsané komunitou. Na komentáře lze reagovat jako na kterékoliv jiné sociální síti. Avšak s jedním podstatným rozdílem. Očekává se, že jednotlivé projevy budou podrobeny korektuře.

Pojetí této služby koresponduje s funkčností webu. Při registraci se volí pouze rodný jazyk. To bude, není-li zvoleno jinak, jazyk, který bude zobrazován ke kontrole. Poté následuje volba cizího jazyka. Následně aplikace provede uživatele všemi (třemi) hlavními akcemi, které lze v aplikaci provádět. Správcem se, odeslat příspěvek a opravit (reagovat) cizí příspěvek.

K opravě slouží vcelku pohodlné rozhraní, kdy je text rozdělen na jednotlivé věty. Pro zajištění oprav zde funguje mechanismus sbírání bodů za každou provedenou korekturu a zároveň poplatek za každý opravený kus uživatelského textu. Podle toho, jaký je poměr mezi provedenými/obdrženými korekcemi, takovou prioritu budou mít příspěvky při zobrazování na hlavní stránce ostatním členům sítě. Ti, co poskytl více oprav, jsou představeni na první pozici, a tudíž mají větší šanci, že na jejich příspěvky co nejdříve někdo zareaguje.

Toto je vlastně veškerá nabídka této služby. Jednoduchá, přehledná aplikace, která dělá přesně to, co uživatel očekává. Spojuje lidi, nechává prostor pro kreativní tvorbu a zajišťuje získání zpětné vazby. Někteří účastníci pojali zasílání příspěvků jako svůj blog, jiní zasílají nápady a nálady, další překládají básně. Fantazii se meze nekladou.

Sít umožňuje vytvářet skupiny, ve kterých se mohou uzavřeně bavit přátelé či tematicky zaměřené lidi. Vnitřní zaslání zpráv je samozřejmostí.

Jazyky s menším počtem rodilých mluvčích jsou zvláštní skupinou. Na jedné straně je pravda, že na menšinové jazyky, mezi kterými je i čeština, může žadatel na opravu čekat i nějaký ten den. Na druhou stranu, když má rodilý mluvčí takového jazyku příležitost získat nějaký bod za poskytnutí opravy, neváhá a opravuje. Chci říci, že má-li jazyk několik desítek na síti aktivních mluvčích, s velkou pravděpodobností se dočká žadatel opravy do několika hodin.

Jako nevýhodu bych zmínil neexistenci zabudovaného audio/video komunikátoru. Účastníci jsou v takovém případě odkázáni na předání si čísel a přesunutí se na jiné komunikační médium.

1.4.8 CentralEnglish

Pojetím opět velmi originální internetová aplikace. Tentokrát zaměřená na výuku jazyka anglického. Služba staví na sledování videí z reálného života, učení se slovní zásoby a opakování po hlavním aktérovi videa. Pro učení je na výběr ze dvou hlavních kategorií – kurzy a samostatná videa.

Pozitivem jest fakt, že videa jsou zpravidla autentickými záběry, nikoliv zinscenované scény pro potřeby CentralEnglish. Bývají převzata z televizních pořadů, zpráv, projevů slavných lidí, včetně prezidentů, až po videa vytvořena běžnými uživateli. Tematicky se zabývají historií, technikou, sportem, aktuálním děním, literaturou, hrami, a podobně. Bývají v rozsahu jedné, až dvou minut. Jsou rozdělena dle náročnosti do tří úrovní. Po výběru videa se přechází do výukového módu. Ten obsahuje čtyři úkoly.

Prvním je zhlédnutí celého videa. Spolu s přehráváním jsou k dispozici interaktivní titulky v originálním znění. Jednotlivá slova v titulcích je možno označovat. Tím se pozastaví video a otevře se text s vysvětlením k danému slovu a zároveň je přehrána výslovnost. Po přečtení se slovo automaticky přidá na seznam později procvičovaných slov a také do uživatelského centrálního slovníku, přičemž je zvlášť upozorněn na zajímavé jevy.

Po zhlédnutí videa následuje přechod k fázi učení slov. Ta probíhá formou přehrávání krátkých sekvencí, ve kterých se vyskytuje dané slovo. Úkolem je doplnit titulky o ono slovo. Když jsou všechna slova procvičena, přechází se na část třetí.

Tuto část považuji za nejzajímavější. Úkolem je opakovat po aktérovi na videu. Lze se tak vžít do slavných prezidentů či si zahrát na reportéra. Nejprve je přehrána krátká pasáž. Po poslechu má student za úkol zopakovat či přečíst z titulků projev. Nakonec je jeho projev podroben hodnocení, kde se sleduje správnost výslovnosti. Jednotlivé nedostatky jsou znázorněny v titulcích změnou barvy u slov, která byla vyslovena hůře či vyloženě špatně. Postupně, po těchto krátkých pasážích, dojde ke zhlédnutí celého videa.

Posledním úkolem je kvíz dotazující se na naučená slova. Nejedná se o nic jiného, než o další testování a memorování.

I zde funguje tzv. soutěžní aspekt, kde se lze porovnávat dle získaných bodů s ostatními návštěvníky.

Uživatel má k dispozici vlastní slovník, který si tvoří postupným zaškrtnutím jednotlivých slov ve videích. Ten umožňuje spustit kvíz, který spustí memorovací test. Slovo je přečteno, pro dokreslení kontextu je zobrazena věta obsahující slovo, a přehraje se úryvek videa. Úkolem je vybrat ze čtyř možností správnou definici slova. Dalším typem je dopisování hodícího se slova dle kontextu a poslechu videa.

Memorování zde funguje inteligentně, podobně jako u Memrise 1.4.2. Slovník zobrazuje indikátory *pamatování*. Tudiž existuje přehled, které slovo je jak procvičené.

Služba slibuje přes 9000 videí a přes 100 kurzů. Kladem je silná konfrontace s rozmanitými intonacemi hlasu, rychlostmi mluvení a jiným přízvukem. Nehrozí přivyknutí na jeden hlas. Navíc možnost napodobovat aktéry je nesmírně zábavná a přínosná.

1.4.9 Tell Me More

Toto desktopové řešení představuje jeden z komplexních produktů nabízející klasickou výuku cizích jazyků. Zaměřuje se na rozličné jazyky. Počínaje angličtinou, němčinou, španělštinou a jinými evropskými jazyky, až po japonštinu a čínštinu. Balíky obsahují náplň rozdělenou do několika úrovní. Angličtina nabízí čtyři úrovně, čemuž jsou přizpůsobeny i různá témata. Forma výuky na vybrané úrovni nezávisí.

Aplikace nabízí tři módy výuky. Volnou výuku, řízenou výuku a dynamickou výuku. Volná výuka nechává studenta svobodně volit pořadí, témata, disciplíny atp. . . Tato možnost je vhodná především pro studenta, který již disponuje určitou znalostí jazyka na dané úrovni a chce si například procházet gramatiku či si něco zopakovat.

Řízená výuka znázorňuje předem připravený plán výuky, který je třeba postupně projít až do konce. Za cíl si zpravidla klade vybavit a připravit¹⁴ studenta znalostmi na mezinárodně uznávané úrovni¹⁵. V případě anglického jazyka – TELC, TOEFL. . .

Dynamická výuka, jak již název napovídá, automaticky uzpůsobuje průběh výuky. Upravuje ho v závislosti na úrovni studenta. Zjednodušeně řečeno, když student nedělá chyby a postupuje rychleji k cíli, tak se uzpůsobí též program. Do výuky pak zahrnuje spíše problematické věci a jednoduché vynechává. Program předefinuje výuku na základě výsledků po každé uzavřené kapitole, kterou student projde. Navíc umožňuje volbu takzvaného cíle – porozumění, vyjadřování, slovní zásobu či gramatiku. Dokonce lze určit pořá-

¹⁴Native English speakers sometimes score lower grades than people who have only been studying English for a few years. This can happen if you take the test unprepared. - IELTS[17]

¹⁵Dle jazyka existují různé certifikační autority a úrovně. Mezi většinou evropských jazyků se za standard považuje TELC - The European Language Certificates

dovanou hloubku znalosti v konkrétních oblastech. Mohu si tedy například stanovit, že se chci naučit poslech dokonale a gramatikou se nechci příliš zabývat.

Software nabízí širokou škálu disciplín. Rozdělené je má do sekcí (učeben) – kultury, slovní zásoby, gramatiky, ústního vyjadřování a písemného vyjadřování. Pokrývá tedy veškeré sledované aspekty.

V učebně kultury se setkáme například s úryvky z filmů. Nemusíme ovšem zůstat jen u přehrávání ukázek. Můžeme se vžít do role z některých postav a propůjčit jim náš hlas. Vybrané texty namluvíme naším hlasem (dabujeme), a poté si je opět přehrajeme. Hlasové nahrávání je pod drobnohledem zabudovaného analyzátoru hlasu, jenž umožňuje sledovat rozdíly v intonaci našeho hlasu oproti roditelým mluvčím. Encyklopedie je jakýmsi ztvárněním nauky o naturáliích. Náplň je tematicky bohatá. Od popisu měst, přes různé památky, až po umění.

Procvičování zajišťují různé disciplíny. Kvízy zaměřené na kulturu, přiřazování názvů měst a států na slepou mapu, asociace slov, obrázků a jiné.

Učebna slovní zásoby staví na zajímavě zpracovaném slovníku. Asi nepřekvapí filtrování dle sledované úrovně. Začátečník tak má pod písmenem A jen několik málo desítek slov, kdežto pokročilý jich tam má stovky. Pozitivem je, že úroveň lze měnit, takže se nestane, že by se začátečník nedostal k některému z hledaných pokročilých slov. Překvapením je filtrování dle kategorií. Jako například nákupy, móda, příroda, města, letiště, doprava, geografie, měřidla a míry, čas atd. Ani abstraktní kategorie nechybí – úsudky, názory, osobnost, pocity. . . Slovník je velice obsáhlý, ale díky rozmanité filtraci a kategorizaci je dobrou pomůckou.

Zajímavým úkolem pro ověření znalostí je přiřazování jednotlivých slov do kategorií. Například ve výběru máme „arm, apple, pear, leg“ a kategorie „human body“ a „fruit“. Jinak lze procvičovat opět za pomoci různých asociací, hledání vhodných výrazů do vět, luštěním křížovek a osmisměrek, šibenici, zde označenou jako tajemné sdělení, doplňování do kontextu a jiné.

Jako zdroj znalosti zde slouží klasický textový popis gramatiky s (video)ukázkami řazený dle úrovně či kategorie, kde jsou na výběr různé typy vět, jako záporné, tázací, časování, atd. . .

Pro procvičování se hodí zpravidla doplňovací cvičení nebo cvičení zaměřená na úpravu slov do správných tvarů. Učebna ústního vyjadřování umožňuje dabing již zmíněných úryvků z filmů či simulované dialogy, nácvik výslovnosti slov a vět.

Písemné vyjadřování poskytuje procvičování v podobě diktátu či kreativního popisu. Veškerý náš pokrok můžeme sledovat v bohatých statistikách. Ty mimo jiné využívá i aplikace samotná a to především v dynamickém módu.

Simulované rozhovory jsou zajímavým zpestřením. Jsou podpořené video záznamem, který zprostředkovává osobu a situaci, ve které komunikujeme. Atmosféru umocňuje pohled na situaci z vlastní perspektivy. Posloucháte rozhovory osob, s nimiž komunikujete, přičemž máte k dispozici aktuální přepis

právě mluveného slova. Ve chvíli, kdy vstupujete do interakce, především položením otázky na vás, máte za úkol odvětit na některou z nabízených odpovědí. To lze provést dvěma způsoby. Buď pouhým označením odpovědi, anebo hlasem, kdy musíte předpřipravenou odpověď vyslovit, čímž dojde k výběru. Odpovědi mají různou charakteristiku. Mohou být všechny smysluplné, pak jimi určujeme směr rozhovoru. Ale mohou se vyskytovat odpovědi nesmyslné a zde již je třeba volit správnou odpověď zapadající do kontextu. Výhodou tohoto řešení je naučení se standardní podoby a zvyklostem daného jazyka při rozhovoru.

Tento produkt se snaží o převedení klasické lektorované výuky do elektronické podoby. Bohužel postrádá jakýkoliv sociální aspekt či propojení s okolním světem. Je ale nutné zmínit, že se jedná o produkt staršího data, kdy internet nebyl tak masově rozšířen.

Návrh

Poznatky získané v kapitole analýza1 poslouží jako zdroj informací pro návrh a sestavení aplikace. Nyní je třeba vymezit a jasně specifikovat podstatu aplikace2.1. Avšak, než se pustím do konkrétního návrhu, shrnu získané informace, provedu pár úvah a zavedu nějaké předpoklady.

Z psychologie učení1.1 vyplývá, že významným faktorem pro zvládnutí učiva je vlastní nadšení pro látku. S tím úzce souvisí *rychlejší* zvládnutí dané oblasti, a probuzení zvědavosti a chuti se dále zlepšovat. To, dříve či později, přivádí studenta k různým formám samostudia, kde si student oproti zprostředkovaným metodám určuje tempo, hloubku a směr výuky sám.

Je tedy vhodné zavést předpoklad, že typickým uživatelem této aplikace bude samouk. Tato úvaha do značné míry směřuje podobu produktu. A to tak, že v ideálním případě bude obsah přímo provázaný se zájmy konkrétního uživatele ve všech zmiňovaných aspektech – tempo, hloubka, směr. Řešením tedy nemůže být monolitický univerzální obsah. Na druhou stranu, tento předpoklad s sebou nese i řadu přínosů. Jelikož bude aplikace využívána nadšenci a dobrovolníky, předpokládám, že bude mít přesnější vypovídající hodnoty ohledně efektivity a přínosu metody nebo produktu samotného. Toho se dá využít pro další zlepšování.

Opomenuta nesmí zůstat kapitola popisující nástroje pro práci s textem1.2. Ta odhaluje propastné rozdíly mezi klasickými přístupy, a těmi využívající elektroniky. Zajisté je potřeba zmínit, že většina moderních nástrojů vychází z principu těch klasických. V některých případech se dokonce jedná o pouhý převod na novou platformu.

Elektronizace klasických přístupů rozšiřuje možnosti použití. Snižuje časovou náročnost úkonů nesouvisejících se samou podstatou konkrétních metod. Od pouhého vyhledávání pojmů ve slovnících, až po mnohem náročnější režii při používání SRS v papírové podobě. Toto odstínění uživatelů od rutinních činností vede ke značnému ušetření času a zvýšení efektivity. Toho se pokusím využít při návrhu aplikace.

Ačkoliv je kapitola Analýza1 značně rozsáhlá, neobsahuje veškeré tech-

nické detaily jako algoritmy, frameworky či knihovny. V oblastech souvisejících s konkrétním vymezením aplikace budu tyto detaily popisovat právě v této kapitole.

2.1 Popis aplikace

Do návrhu aplikace vstupuji s několika předpoklady. Prvním je zaměření na samouky. Dalším, přímo vycházejícím ze zadání, je orientace na text a software.

Pro zaujetí samouka je třeba tvořit obsah dle jeho zájmu. Obecně největší zásoba textů je zaměřena na veřejnost – beletrie, drama, horory, sci-fi, a jiné. E-knihy svým množstvím dohánějí knihy tištěné. To přináší řadu výhod. Pro čtenáře širokou nabídku dle jejich přání a pro účely této práce odbourává nutnost řešit převod textu do elektronické podoby¹⁶. Za zdroj textu od této chvíle považuji obecně knihy. Nezmíním-li jinak, bude se jednat o formu elektronickou.

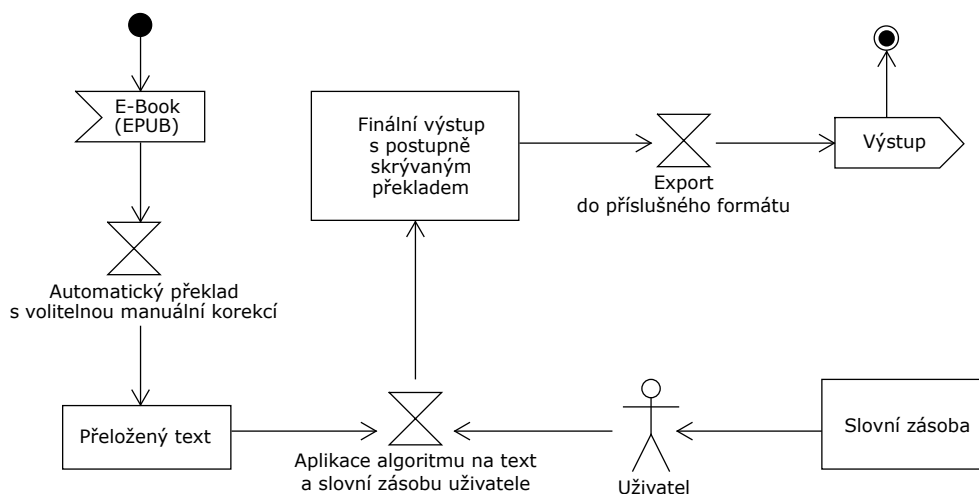
Z nástrojů pracujících s textem 1.2, se mi jeví forma doslovného překladu jako nejzajímavější. Přináší hned několik výhod. Za prvé čtenář může číst knihu v původní podobě, není tedy třeba zkracovat text či slovní rozsah. Dále obsahuje překlad pod slovem, což šetří čas a neodvádí pozornost. Není třeba tolik řešit skloňování a různé úpravy, což značně zlehčuje významovou úpravu textu. A v neposlední řadě se zdá, že metoda SRS lze individuálně škálovat na uživatele.

Na obrázku 2.1 znázorňuji obecný princip tvorby takové knihy i s přizpůsobením na uživatele. Vstupem jest kniha v elektronickém formátu. Takový text je dále převeden do příslušných struktur a databází nástroje. Následně je každé slovo zcela automaticky přeloženo dle nejčastějšího překladu. Takový text může volitelně projít manuální významovou korekcí. V této fázi je text kompletně přeložen a nezbyvá než aplikovat metody SRS. Po jejich aplikaci se z textu odstraní překlady již známých slov. Sekvenčně se dále dle specifického algoritmu vyfiltrují slova, která se již čtenář naučil. Navíc se u každého slova tvoří a ukládají statistiky pro další použití. Poslední fází je export do libovolného jednoduše čitelného formátu – EPUB, webový prohlížeč, PDF, atd. . .

2.1.1 Elektronické formáty

Existuje široká škála formátů, avšak těmi nejznámějšími jsou *epub*, *mobi*, *azw* či *pdf*. Každý z nich má svoje výhody a nevýhody, které často vychází z historických konotací. Nejvíce rozšířený a podporovaný napříč různými zařízeními je formát EPUB, se kterým budu dále pracovat. Jelikož volba formátu není

¹⁶Nástrojům převádějící text do elektronické podoby se říká OCR - Optical Character Recognition



Obrázek 2.1: Principiální znázornění funkce aplikace

zásadní pro tuto práci, dovoluji se odkázat na vcelku přehlednou *orientační* tabulku rozdílů jednotlivých formátů¹⁷, avšak pro ověření informací nabádám vyhledat i jiné zdroje.

EPUB je oficiálně spravován organizací IDPF - International Digital Publishing Forum[18]. „EPUB defines a means of representing, packaging and encoding structured and semantically enhanced Web content — including XHTML, CSS, SVG, images, and other resources — for distribution in a single-file format.“[18]

Principiálně se jedná o komprimované soubory metodou *ZIP*. Jejich hierarchie má určitou strukturu, která je vymezena přímo umístěním nebo jednotlivými konfiguracemi klíčových souborů. Každý EPUB musí obsahovat složku *META-INF* se souborem *container.xml*. Ten ve své struktuře ukládá adresu na kořenový soubor pojmenovaný *content.opf*.

Tento kořenový soubor volitelně uchovává bližší informace o obsahu – autor, název díla, rok vydání, atd. . . Kromě těchto informací odkazuje na jednotlivé soubory nesoucí samotný obsah – HTML soubory, obrázky, či soubory určující pořadí. Soubory XHTML již nemají předepsaný formát, avšak musí být validní dle W3C - The World Wide Web Consortium[19].

2.1.2 Automatický překlad

Další fází po načtení textu je samotný překlad. Ten bude využívat volně dostupné slovníky. Využit lze přímo API¹⁸ online či offline služeb. To je vhodné například pro získávání konkrétních překladů jednotlivých slov, avšak značně

¹⁷Rozdíly mezi formáty : http://en.wikipedia.org/wiki/Comparison_of_e-book_formats

¹⁸Application Program Interface

nepraktické pro další zpracování. Pro aplikaci SRS je třeba provádět různé statistiky, podle kterých budu slovník upravovat. Navíc v případě online slovníků nemůžu zaručit, že nebude nějak zásadně manipulováno s daty. Stavět tedy něco většího nad takovými službami není rozumné.

Jelikož tato aplikace bude se slovníkem pracovat do mnohem větší hloubky, využiji dostupných slovníků v textové podobě, které následně převedu do vlastní databáze. Databázové struktury budou pochopitelně mnohem bohatší než jen párové slovo a překlad.

Samotný překlad bude fungovat na principu 1:1, kdy se pro překlad použije nejčastější význam slova. Příkladem může být slovo *long*, které mimo jiné může mít významy *toužit*, *dlouhý*, *daleký*, avšak nejčastěji odpovídá překladu *dlouhý*. Priority těchto překladů se budou měnit v závislosti na práci korektorů (viz dále 2.1.3, jež budou opravovat právě tyto případné nesrovnalosti. Filozofií takového překladu je přeložit text s co největší přesností.

Při tvorbě překladů z angličtiny do ostatních jazyků lze s výhodou využít specifika jazyka. Odpadá tak například nutnost řešit skloňování. Avšak tento zádrhel by se dal vyřešit detailnějším slovníkem obsahující veškeré přípustné tvary.

Během tohoto procesu se budou pochopitelně provádět i přidružené procesy, které budou zaměřeny především na statistiky.

2.1.3 Manuální korekce

Pro upřesnění významových nesrovnalostí, vzácných jazykových obrátů a jiných anomálií, vstupuje do procesu korektor. Tím může být libovolná osoba schopná porozumět textu v obou jazycích. Jelikož se jedná o práci s textem, můžou jimi být i méně zkušení studenti cizího jazyka, kteří mají možnost v případě nejasností dohledat informace z dalších zdrojů. Zefektivnění manuální korekce lze dosáhnout vhodným a přehledným rozhraním, které bude poskytovat doplňující informace v rámci aplikace. Ušetří se tak čas při hledání jiných zdrojů. Podobu takového rozhraní blíže znázorňuji v příloze A,

Jak již bylo zmíněno v sekci Automatický překlad 2.1.2, na pozadí každé korekce probíhá ukládání změn do databáze, která dále aplikuje jisté heuristiky vylepšující každý další překlad.

Text upravený korektorem je již připraven pro veřejné publikování. Avšak, aby cílil na konkrétní uživatele je nutné vypustit překlady některých slov. V samotném procesu se jedná o doplňující krok, který pro implementaci není nezbytně nutný.

2.1.4 Přizpůsobení uživateli

V této fázi je třeba z již přeloženého a případně opraveného textu odstranit překlady. Přichází na řadu metoda prokládaného opakování - SRS{sec:srs.

Samotná metoda má širokou historii a také různé implementace. Z tohoto důvodu věnuji tvorbě algoritmu a jiným detailům vlastní kapitolu 3.2.

Pro lepší popis zde tento proces rozdělují na dva průběhy. Nejdříve dojde k odstranění všech slov, která čtenář zná v konkrétním daném významu. To je důležitý aspekt. Může se stát, že slovo *long* ve významu *dlouhý* již zná. To se odstraní. Avšak ve významu *toužit* se musí přítomnost překladu zachovat.

Druhý průchod již řeší prokládání a případné přeřazování výrazů mezi slova naučená. To bude mít na starosti algoritmus, který bude sledovat celkový výskyt slov viz 3.2.

2.1.5 Export

Poslední fází je takový text vyexportovat do požadovaného formátu. V mém případě opět do EPUBu. Za zmínku určitě stojí zdůraznit, že jednotlivé soubory EPUBu jsou vlastně *xhtml* validní výstupy (viz 2.1.1).

2.2 Návrh aplikace

V popisu aplikace jsem definoval funkci a rámcově popsal činnost jednotlivých fází. Co jsem záměrně opomenul je podoba či volba platformy. Toto rozhodnutí bude klíčové a následně ovlivní volbu technologií 2.3. Nejprve definuji činnosti, které bude moci uživatel provádět, a v návaznosti vyberu vhodnou platformu.

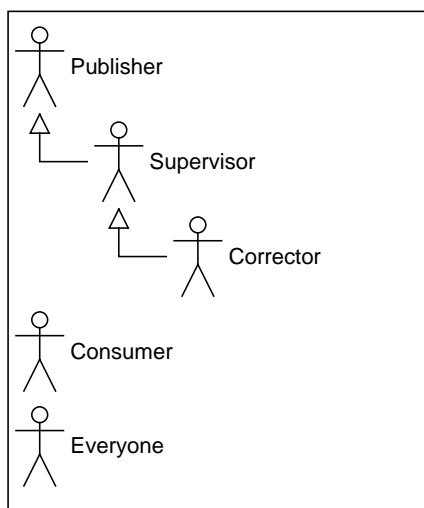
2.2.1 Případy použití

Princip aplikace je víceméně přímočarý a spoléhá na vlastní interní mechanismy. Nedává tedy uživateli mnoho prostoru k manipulaci se samotným procesem. Klíčovou úlohou čtenáře je volba knihy, kterou by si přál přečíst. Případně postupný výběr knih již přečtených. Tato možnost má odstranit dle aplikace teoretickou nulovou znalost jazyka, a přitáhnout tak i čtenáře zkušené. Export tak bude poslední akcí, která ukončí jednu iteraci práce s programem.

Výše zmíněné činnosti uvažovali *klasického* uživatele – čtenáře. Avšak uvnitř aplikace může existovat jistá hierarchie rolí, které by mohli mít na starosti činnosti jako jsou veřejná publikace či korekce překladů. Jsou jimi:

- Everyone - všichni, bez ohledu na práva
- Consumer - spotřebitel obsahu, čtenář, uživatel
- Corrector - vykonavatel významových oprav textu
- Supervisor - provádí kontrolu již opravených textů
- Publisher - zpřístupňuje spotřebitelům přeložený a zkontrolovaný obsah, též má možnost stáhnout obsah z nabídky

Tuto hierarchii rolí nejlépe vystihuje obr.2.2. Znázornění jednotlivých akcí jim přidělených ukazuje diagram případu užití viz 2.3



Obrázek 2.2: Virtuální hierarchie rolí

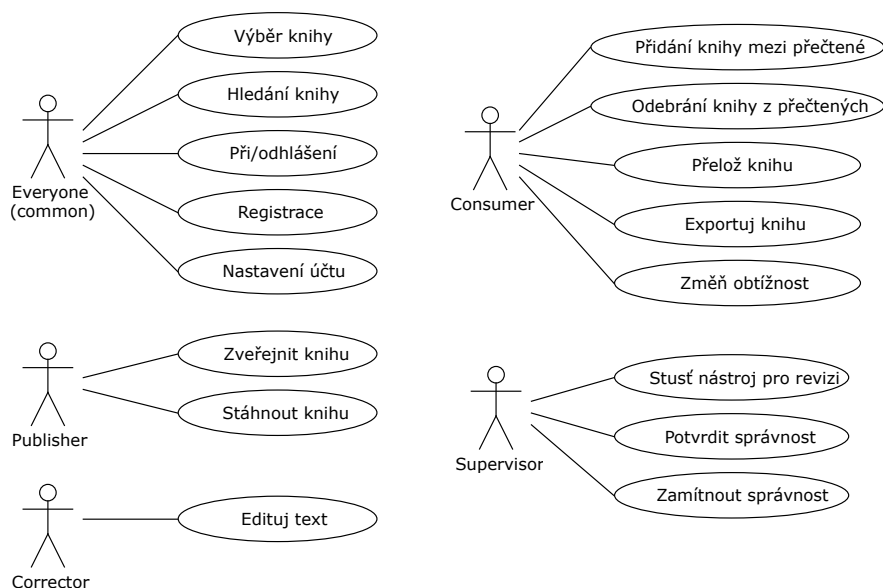
Avšak náplní této práce je tvorba doslovných překladů. Z tohoto důvodu se zaměřuji pouze na samotného uživatele, do něhož agreguji akce přidání a schválení knih. Uvedený soupis akcí a rolí je jen ilustrační. Implementace této funkčnosti by přímo nesouvisela s prací.

2.3 Volba technologií

Trendem dnešní doby je transformace aplikací z lokálních instalací na webové služby. To přináší výhody jak vývojářům, tak uživatelům. Odpadají problémy s aktualizacemi, obnovováním obsahu nebo přístupem z jiných zařízení. Z těchto důvodů budu aplikaci směřovat na web.

Principiálně lze takovou aplikaci rozdělit na serverovou a klientskou část. Serverová část zajišťuje poskytování a manipulaci s daty. Klientská část se stará převážně o zobrazení poskytovaného obsahu. Výběr vhodných technologií zakládám na právě uvedeném rozdělení.

Serverem, v této chvíli, nazývám stroj, který provádí klíčové výpočty a transformace textu. Takové prostředí nemusí být nutně součástí webového serveru. Jako výhodou takového řešení je možnost použití dostatečně robustních nástrojů a knihoven, které nejsou zamýšleny pro webové účely. Dále to umožní z bezpečnostního hlediska odstínit některá rizika. S takovým strojem se může komunikovat externě a předávat mu pouze konkrétní požadované úkoly přes vlastní rozhraní.



Obrázek 2.3: Model případů užití aplikace. Uživatel *everyone* zahrnuje akce, které mohou vykonávat všichni účastníci.

Nicméně potřeba, alespoň nějakého jednoduchého, webového serveru je nevyhnutelná. Zde se nabízí použití volně dostupného Apache, a například jazyka PHP nebo Ruby¹⁹.

Toto rozdělení je prakticky možné a celkem časté. Avšak tyto kombinace vnáší do projektu požadavek na znalost více technologií. To může do budoucnosti představovat zbytečnou složitost a náročnost pro správu takového kódu. Zároveň bude třeba řešit propojení jednotlivých technologií, což nemusí být vždy efektní a bezpečné.

Za ideální řešení považuji takovou technologii, která obratně využívá výhod z obou směrů. Na jedné straně požadují robustnost jazyka a pokročilých nástrojů, a na straně druhé, schopnost a vhodnost technologie i pro použití ve webovém prostředí.

Vzhledem k tomu, že lze logicky rozdělit úkoly na webové, které budu implicitně považovat za rychlé, a výpočetní, které mohou být výpočetně časově náročné, uvaluji požadavek na schopnost technologie spravovat úkoly asynchronně a paralelně. Tento požadavek je již značným filtrem technologií. Avšak zdaleka ne všechna řešení mají tyto schopnosti rozumně implementovány, a to především z hlediska použití, které musí být intuitivní.

¹⁹Zmíněný server a jazyky nejsou pochopitelně jediné možné

2.3.1 Imperativní vs. Funkcionální přístup

Nabízí se srovnání různých přístupů programování. Nejčastějším typem v komerčním prostředí je imperativní styl. Je založen na popisu řešení problému. Tedy například pro součet všech čísel v poli je nutné explicitně založit proměnnou *result* s výchozí hodnotou nula. Poté přes smyčku, která má další proměnné, projít každý prvek, který dále přičítám k proměnné *result*. Typickými představiteli tohoto přístupu jsou jazyky C, C++, Java, a většina ostatních.

Oproti tomu funkcionální přístup definuje množinu funkcí, které transformují vstup na odpovídající výstup. Jinými slovy nepopisuje postup, ale předmět, nebo-li CO chceme dostat. Pro předchozí příklad sčítání prvků v poli předáme funkci, která manipuluje s prvky pole další funkci, která definuje operaci sčítání dvou prvků. Ačkoliv se to na první pohled nemusí zdát, je toto řešení mnohem univerzálnější a bezpečnější. Tímto přístupem se lze vyhnout chybám způsobených např. ukazováním na data mimo rozsah pole nebo počet cyklů ve smyčkách. Jelikož čisté funkcionální jazyky nemají proměnné, veškeré smyčky se dělají rekurzemi. Funkce nemodifikují staré hodnoty, ale generují hodnoty nové. Pro konkrétní vstup vrací vždy stejný výstup.

Funkcionální programování se z výše uvedených důvodů často právem považuje za výpočetně náročné. Je to jeden z důvodů, proč se ještě do nedávna používaly tyto jazyky téměř výhradně v univerzitních a výzkumných kruzích. Nicméně dostupnost výpočetního výkonu se natolik rozšířila, že toto negativum začíná ztrácet na významnosti. Ke slovu se tak začínají dostávat výhody. Díky tomu, že neexistují proměnné, nemají funkce žádné vedlejší efekty. To je klíčové pro paralelizaci výpočtů. Odpadá totiž starost se synchronizací společných proměnných a dalších problémů. Představiteli jsou např. jazyky Lisp či Haskell.

Není tedy divu, že se tento přístup uplatňuje při tvorbě rozsáhlých aplikací, které jsou škálovány přes široké spektrum různých výpočetních jednotek. Typickým příkladem jsou systémy nadnárodních bankovních institucí či platformy pro obchodování s akciemi.

Nicméně se najdou takové matematické problémy, kde by zavedení byť jedné proměnné ušetřilo značnou část výpočetního výkonu. Z tohoto důvodu vznikají též jazyky hybridní, které kombinují výhody z obou přístupů.

Takovým moderním hybridním jazykem je Scala. Spustitelný kód běží na strojích JVM. Je silně staticky typovaný, což odbourává typové kolize a zajišťuje primární kontrolu správnosti kódu. Budeme-li se striktně držet obecného doporučení a vyhýbat se proměnným a funkcím se side-efekty, dále se sníží chybovost kódu či se tím alespoň zjednoduší hledání chyb. Takový přístup má ovšem význam i pro následnou paralelizaci, kde v případě neexistence závislých proměnných lze výpočty provádět zcela odděleně od zbytku architektury.

Aby toho nebylo málo, je potřeba pohlídat, aby veškeré navazující technologie jako webový výstup či databáze též umožňovaly asynchronní práci.

Kdyby tomu tak nebylo, výrazně by se snížil efekt této technologie.

2.3.2 Webový framework

Scala je dostatečně robustní jazyk, který pomocí knihoven implementuje komunikaci s internetem. Avšak v základu neobsahuje, nebo jen v omezené míře, implementaci různých webových přístupů a standardů. To ovšem řeší webové frameworky, které přinášejí vlastní knihovny zjednodušující práci s asynchronním zpracováním, manipulaci s datovými formáty jako jsou *JSON*, *XML*, *xHTML*, nebo implementaci bezpečnostních protokolů a standardů. Jedná se ovšem jen o nástin možností.

V prostředí Scaly existuje, navzdory mladosti jazyka, široká škála frameworků. Některé z nich jsou globálně používány významnými hráči na internetovém trhu. V následujících podsekcích představím právě ty nejrozšířenější, protože již prokázaly svoji použitelnost, a ve většině případů přichází i s dostatečnou dokumentací.

2.3.2.1 Scalantra

Tento minimalistický framework staví svou filozofii na jednoduchosti. V základu poskytuje jen nezbytné minimum nástrojů se soustředěním na serverovou část. O to větší volný prostor nechává vývojářům, kteří si mohou poskládat knihovny a technologie dle potřeb konkrétního projektu. Dále si zakládá na škálovatelnosti, kterou zajišťuje dnes čím dál více významnějším *Akka* nástroji²⁰.

Scalantra obsahuje dobře dokumentovanou API a určitou sadu návodů. A je doprovázena knihou *Scalantra in Action*. Tento framework používá řada známých organizací, a to především jako interní back-end REST API. Jedná se například o *BBC Future Media*, *IGN*, *LinkedIn Signal*, *Netflix* a jiné.

2.3.2.2 Lift

Framework prohlašující se za nejmocnější a nejbezpečnější dneška. Toto tvrzení podkládá schopnostmi jako lazy-loading, což znamená, že server nečeká před odesláním webové stránky na dokončení náročných úkolů. Místo nich pošle šablonu a data posílá dodatečně, aniž by o ně browser explicitně žádal. To je zajištěno pomocí metodologie Comet²¹ s použitím AJAXu. Též je k němu dostupná široká nabídka návodů a knih. Používají ho například *FourSquare*, *InnovationGames* či *Sketchboard.io*.

²⁰Akka[20] představuje sadu nástrojů a principiálních postupů pro tvorbu webů splňujících tzv. *Reactive Manifesto*[21]

²¹Comet je obecný název pro různé techniky, které umožňují serveru zasílat (push) data do browseru, aniž by o ně bylo explicitně požádáno. Bližší popis technik v IBM developer sheetu[22]

2.3.2.3 Spray

Tvůrci prohlašují tento projekt za odlehčenou sadu knihoven, které poskytují podporu REST/HTTP postavenou na Akka[20]. Jedná se spíše o platformu, kterou si má každý rozpracovat směrem, který potřebuje. Implementace používá pouze nástroje z jazyka Scala a Akka. Pro různé účely ho používá *VMware*, *Ebay* nebo třeba *Movio*.

2.3.2.4 Play Framework

Jedná se o vysokorychlostní Java a Scala framework, který je též postavený na Akka[20]. Vniřně je řešený zcela asynchronně a nepoužívá žádné bloku-
jící operace. Ve výchozím nastavení je RESTful. Obsahuje širokou škálu nástrojů, knihoven a technologií. Zastane práci s Websockety, Comet, Event-Source, usnadňuje manipulaci s JSONem a Big Data. Mezi nástroji je například kompilér pro CoffeeScript či LESS.

Avšak na programátorovi zůstává odpovědnost zvolit technologie odpovídající této filozofii. Například volbou blokující databáze se nevyužije celý potenciál frameworku. Díky své popularitě má podporu mnoha vývojových prostředí, což značně usnadňuje práci. Též pro něj existuje široká škála pluginů rozšiřující možnosti. Používají ho *LinkedIn*, *The Guardian*, *Coursera*, *Klout* a jiní.

Z důvodu široké podpory, množství návodů a integraci různých nástrojů volím pro další práci Play Framework. Volím tak z důvodu mé neznalosti těchto řešení. Minimalizuji tak nutnost hledat vhodné moduly a způsoby jak propojit různé technologie. Výhody Scalantry nebo Spray by se tak v mém případě staly nevýhodami.

2.3.3 Databáze

Relační databáze mají za sebou historii několika desetiletí. Prošly značným vývojem. Během tohoto procesu se definovaly teorie návrhů, nejlepší metodiky a přístupy²², a vznikla široká komunita. Na těchto databázích je implementována převážná většina projektů.

S masivním nástupem internetu vznikala potřeba zpracovávat do té doby nepředstavitelné množství dat různého a často předem nedefinovaného formátu. To je oblast, kde klasické relační řešení požadující explicitní popis dat nenabízí jednoduché řešení. Odpovědí na tyto požadavky jsou databáze zvané NoSQL. To v původním záměru mělo znamenat *No SQL*, avšak jak následný vývoj ukázal, že není rozumné zahodit veškeré zkušenosti z relačních databází, a tak se toto označení transformovalo na význam *Not Only SQL*. Existuje celá řada různých typů NoSQL databází.

²²Příkladem nechť je definice normálních forem(normalizace)

2.3.3.1 Klíč-Hodnota

Tento nejjednodušší typ NoSQL databáze, jak název napovídá, ukládá pouze klíč a hodnotu. Implementačně se jedná o hash tabulku s unikátním klíčem a odkazem na blíže nespecifikovaná data. Její výhodou je extrémní rychlost.

2.3.3.2 Wide-column

Principiálně vycházejí z typu klíč-hodnota, s tím rozdílem, že mohou obsahovat dynamicky se měnící množství sloupců. Zároveň neexistují žádná omezení na jejich pojmenování.

2.3.3.3 Dokumentové

Databáze určené k částečně strukturovaným datům se nazývají dokumentové. Jednotlivé záznamy jsou dokumenty. Ty mohou obsahovat libovolnou strukturu, avšak musí být v korektním předem určeném formátu jako jsou například JSON nebo XML. Oproti předchozím zmiňovaným typům, dokáží tyto databáze provádět dotazy nad jednotlivými položkami.

2.3.3.4 Grafové

Jsou vhodné pro ukládání dat, která mají mezi sebou nepředvídatelné vztahy. Vnitřně jsou reprezentovány grafovou strukturou – uzly a hrany. Lze je využívat například pro sledování toků dat či sociálním propojení uživatelů různých služeb.

2.3.3.5 Ostatní

Avšak uvedené kategorie nejsou kompletním výčtem. Lze se setkat třeba s XML databázemi, objektovými či různými hybridy. Rozsáhle zpracovaný soupis různých existujících produktů ze všech různých kategorií poskytuje web NoSQL-Database²³.

Z výše uvedeného přehledu nejvíce vyhovují databáze dokumentové. A to z následujících důvodů. Celá aplikace je zaměřena na práci s texty. Navíc odpadá nutnost fixního návrhu, jelikož tento typ databáze nemá problém přidat nová data v jiné podobě přímo za běhu. To do budoucna usnadní případné přidání nové sledované kategorie do DB přímo za běhu bez nutnosti předělávat kód na mnoha místech. Takto volná možnost působnosti ovšem přenáší větší zodpovědnost na programátora.

V oblasti dokumentových databází existuje celé spektrum různých řešení. MongoDB, Cassandra, CouchDB, HBase, Redis a jiné. Srozumitelný přehled a srovnání lze najít na webu Kristofa Kovace²⁴.

²³<http://nosql-database.org/>

²⁴<http://kkovacs.eu/cassandra-vs-mongodb-vs-couchdb-vs-redis/>

Vhodná volba databáze je velice důležitým krokem. Jak jsem již naznačil, musí zapadat do konceptu neblokovaného zpracování požadavků. Kdyby jej nespĺňovala, aplikace by musela čekat na dokončení činnosti databáze. Tím by docházelo k blokování a tudíž by se nevyužil veškerý potenciál řešení. Zároveň je třeba brát ohledy na různou podporu driverů pro jazyk Scala.

Vzhledem k jednoduše definované dotazovací sadě nástrojů, jednoduchém ukládání dokumentů různé struktury v lidmi dobře čitelném JSONu jsem se přiklonil k MongoDB. Této volbě přizpívá i existence dobře dokumentovaných knihoven pro jazyk Scala, respektive Play Framework. Konkrétně se jedná o ReactiveMongo, které podporuje plně neblokující a asynchronní I/O operace.

2.3.4 Klientská část

Ač by se mohlo zdát, že část komunikující s uživateli je nezávislá na volbě technologií serverové části, úplně tomu tak není. Vzhledem k tomu, že jsem vhodně poskládal back-endovou část, mohu nyní zcela využít všech předností konkrétních řešení.

Nabízí se možnost implementace tenkého klienta pro jednotlivé platformy - Windows, Android, IOS, a jiné. Nicméně univerzálním řešením je klasické webové rozhraní, se kterým umí pracovat všechny zmíněné systémy, potažmo zařízení. Tato volba umožní použití jednotných technologií napříč celou aplikací.

Play Framework přímo obsahuje nástroje pro generování statického XHTML. Těch budu také využívat. Avšak moderní responsivní web se skládá i z dynamické části. Ta je většinou tvořena pomocí JavaScriptu či frameworku a knihoven na něm postavených. Nyní přichází na řadu ukázka vhodného poskládání jednotlivých technologií.

Jak jsem již zmiňoval, MongoDB vnitřně ukládá dokumenty ve formátu JSON²⁵, jehož zápis je pro člověka snadno čitelný. Dynamickou část bude řešit přímo JavaScript. Z toho vyplývá, že při asynchroním požadavku o data na server, označovaném jako AJAX²⁶, se výsledná data bez další nutné transformace rovnou odesílají na klienta. Vzhledem k tomu, že zpracování na všech fázích je neblokované, nepředstavuje ani značné množství dotazů pro server problém, a navíc celou komunikaci značně zrychluje. Takto přijatá data jsou v JavaScriptu reprezentována jako objekty. Tomu odpovídá i následná práce s nimi.

Podobně jako v případě volby databáze či scala web frameworků existuje nespočet nástrojů postavených na JavaScriptu. Hledám co nejjednodušší způsob, kterým lze provádět AJAX požadavky na server a snadno manipulovat a generovat obsah podle získaných dat.

²⁵JavaScript Object Notation

²⁶Asynchronous JavaScript and XML

2.3.4.1 Angular a ReactJS

Během předimplementační fáze jsem experimentoval s Angularem a Reactem. První zmíněný pochází od Googlu, existuje delší dobu a má podporu široké komunity. Jedná se o robustní nástroj, který jednotlivé akce a části kódu propojuje přes různé vnitřní mechanismy. To umožňuje velice přehlednou tvorbu generovaných šablon. Na druhou stranu to vyžaduje větší přehled o umístění a souslednosti kódu.

Oproti tomu druhý nástroj, který vznikl pro účely Facebooku, neobsahuje téměř žádné skryté funkce, tzv. magic²⁷. Celá filozofie ReactJS tkví v přímočaré jednoduchosti. Staví na rozdělení jednotlivých prvků na podkomponenty. Ty se starají jen a pouze o zobrazení dodaných dat.

Právě z důvodu přímočarosti jsem zvolil pro další práci ReactJS. Pěkné srovnání i s funkčními ukázkami lze najít na codementor.io[23].

2.4 Návrh grafického zobrazení výstupu

Výstupem programu bude kniha ve formátu epub. Je třeba si uvědomit, že funkcí aplikace je rozšíření původní knihy o překlady. To znamená, že ačkoliv existuje možnost kompletně předělat formátování např. do jednotného stylu, kterým by reprezentoval knihu pocházející z této aplikace, budu se snažit zachovat původní vzhled.

Dávám si za cíl udělat překlady co nejméně rušivé. K tomu by mělo přispět zmenšení textu překladu oproti originálu. Pro snadnost hledání překladu budu překlady centrovat. Výsledek by měl být podobný ukázce na obrázku 2.4

Lorem ipsum world sit amet, consectetur adipiscing elit.
Quisque a vesmír odio nunc ultricies ipsum placerat
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque a vulputate odio nunc ultricies ipsum placerat
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque a vulputate odio nunc ultricies ipsum placerat
Lorem ipsum world sit amet, consectetur adipiscing elit.
Quisque a vesmír odio nunc ultricies ipsum placerat
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque a vulputate odio nunc ultricies ipsum placerat
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Quisque a vulputate odio nunc ultricies ipsum placerat

Obrázek 2.4: Návrh podoby elektronické knihy po finální úpravě

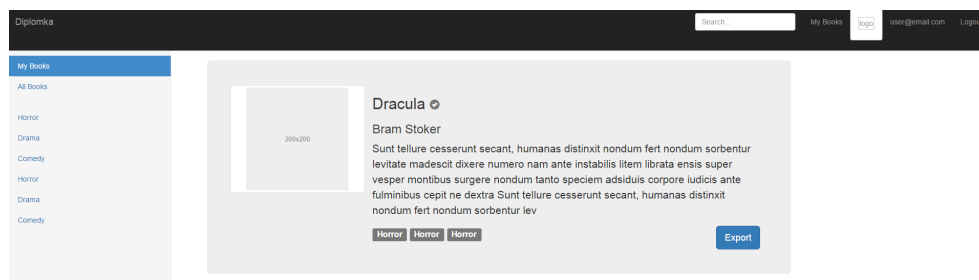
Z návrhu vyplývá, že aplikace s uživatelem bude komunikovat prostřednictvím webového rozhraní. Tomu je nutné přizpůsobit i uživatelské rozhraní. Nejdůležitějším uživatelem je *Consumer*, který představuje čtenáře/zákazníka.

²⁷Magic, vychází ze slova kouzlo. Označuje skrytou funkčnost, která manipuluje běh akcí nad rámeč explicitního vyjádření. Např. proměnná s názvem „ID“ je automaticky hashována.

2. NÁVRH

Ten má možnost přidání nové knihy do systému a poslat požadavek na přeložení knihy dle své slovní zásoby – export. Opět zdůrazňuji, že se aktuálním pojetím jedná o pouhé rozhraní pro posílání příkazů na server.

Způsobů jak zprostředkovat uživateli tuto funkcionalitu je vícero, avšak vzhledem k tomu, že aplikace má uživateli pouze zprostředkovat manipulaci s vybranou knihou, pokusím se funkcionalitu soustředit do jednoho okna. Představu podoby takového uživatelského rozhraní zachycuje lo-fi model viz 2.5.



Obrázek 2.5: Wireframe zachycující uživatelské rozhraní se soustředěním funkcionality do jednoho okna

Rozhraní ostatních uživatelů bude do značné míry stejné. Lišit se bude pouze v různé nabídce titulů knih a funkčních tlačítek provádějící manipulaci. Consumer nemá mít přístup k nepublikovaným knihám, kdežto například Corrector ano. Ukázkou rozhraní korektora znázorňuji v příloze A. Jelikož se jednotlivá okna dalších uživatelů mění jen v malých detailech, nepovažuji za přínosné se věnovat jejich dalšímu popisu.

Implementace

V návrhu jsem provedl vymezení z hlediska technologií. Pro přehled je ještě jednou shrnu. Server je postaven na jazyku Scala s využitím Play frameworku. Databáze bude typu dokumentové NoSQL, konkrétně MongoDB. Statické části, jako šablony, css bude poskytovat framework s tím, že práci s dynamickými daty zajistí ReactJS.

3.1 Databáze

NoSQL databáze obecně nepožadují striktní definici struktury či významu dat. Též mají různé dotazovací mechanismy, které se liší dle typu databáze. Oproti SQL mají většinou značně skromnější možnosti formulace dotazů. To je dáno buď tím, že data nejsou významově rozlišována, nebo tím, že je databáze optimalizována pro rychlost ukládání či čtení.

MongoDB ukládá dokumenty do kolekcí. Kolekci si lze představit v terminologii SQL jako tabulku a dokument jako záznam v tabulce. Dokumenty jsou uloženy ve struktuře JSON, na kterou lze nahlížet jako na objekt. Ten má své položky označeny konkrétním jménem. Jim jsou posléze přidělené hodnoty. Takovou strukturu znázorňuje obrázek 3.3. Vzhledem k této formě uložení podporuje MongoDB širší sadu dotazování. Avšak, stále nedosahuje možností SQL. Hledání lze provádět nad konkrétními položkami, avšak defaultně se vrací každý dokument v celém svém rozsahu. Lze pomocí rozšířených funkcí do určité míry transformovat výsledek, avšak na úkor výkonnosti. Pochopitelně lze i měnit jednotlivé položky u konkrétních záznamů.

Na druhou stranu, složitější dotazy typu JOIN jsou neproveditelné nebo se značným omezením jak do výkonu, tak rozsahu možností. Takové dotazy jsou řešeny většinou na straně aplikace ve formě několika dotazů nad různými kolekcemi. Získaná data je pak nutné ručně agregovat do potřebné podoby. To pochopitelně motivuje designera aplikace a databáze k návrhu celé koncepce tak, aby se na všechna aktuálně potřebná data dalo zeptat jedním dotazem.

To ovšem přináší několik, pro relační databáze zcela nemyslitelných, konceptů. Je třeba se do určité míry vzdát normálních forem. Uvedu příklad, kde je vhodné řešit daný problém trochu odlišně od relačního přístupu.

Požadavkem aplikace je zobrazit seznam projektů i se jménem zakládající osoby. Mám kolekci uživatelů se strukturou znázorněnou na obr.3.5. Obsahuje, kromě jiných, položky ID, celé jméno a email. Dále mám kolekci s projekty, které obsahují informace o projektech, včetně meta informací s odkazem na konkrétního uživatele, který prováděl danou operaci. Pochopitelně u takové informace uvádím ID odkazující do kolekce uživatelů. Pro splnění požadavků je třeba provést dvě operace. Načíst projekty, pak uživatele dle ID, a získat pouze jejich celé jméno a email. Zobrazení projektů je tak značně neefektivní. Kvůli získání jedné či dvou informací je nutné provádět další dotaz. Řešením je tedy do struktury meta informací u projektu přidat položku se jménem a emailem. Nyní stačí jeden dotaz.

Teď se nabízí zamyšlení. Tím, že dochází k duplicitě informací se snižuje a zjednodušuje dotaz, což vede ke zvýšení rychlosti. Na druhou stranu došlo k porušení normální formy, a tím vznikla i náchylnost k inkonzistenci dat. Jejich integritu je třeba ohlídat na straně aplikace. Při změně jména uživatele, je třeba zvláště aktualizovat všechny meta informace. To může být časově náročné. Tímto jsem chtěl zdůraznit, že je vždy nutné důsledně promyslet zdali aplikace opravdu potřebuje konkrétní informaci. Jestliže ano, pak zohlednit, jak často se zobrazuje. Pro občasné zobrazení není důvod tvořit duplicitu, protože se nejedná o rychlostně kritickou část. Jen by tím zbytečně vznikl prostor pro narušení integrity dat. Ovšem jsou případy, kdy je takové řešení vhodné.

Jelikož používám napříč celou aplikací jednu databázi, snažil jsem se vyvarovat jakýkoliv duplicit dat týkajících se obsahu samotných knih. Obsah knih se bude zpracovávat převážně na straně serveru, tak není nutná optimalizace pro rychlost. Duplicity se tak vyskytují pouze u dat související přímo s webovým zobrazením. Na takové případy vždy upozorním.

Databáze NoSQL jsou ze své podstaty schema-less. Neexistuje tedy standardizovaný způsob logického zobrazení návrhu uspořádání. Nic jako relační schéma neexistuje. Přesto lze v některých způsobech definovat formální popis uspořádání, alespoň nad určitou podmnožinou dat. V mém případě s výhodou využiji přímo formátu JSON. Jelikož se jedná o strukturovaný, člověkem snadno čitelný formát, kde je hodnotou většinou text, zavedu implicitní derivaci typů dle postupu popsaného též jsonem3.1. To dále usnadní prezentaci struktury jednotlivých kolekcí, kdy z ukázkových dat vyplývá typ. Z důvodu typové bezpečnosti se zavazuji používat stejný typ dané položky napříč všemi dokumenty.

Listing 3.1: Defaultní konverze

```
{
  "_id" : ObjectId("5515732de5e21e36679db6e0"),
  "text" : "text value",
```

```

    "boolean" : false,
    "int" : 42,
    "double" : 42.0,
    "date" : 1427469061237
}

```

3.1.1 Kolekce

V této sekci se věnuji již přímo popisu konkrétních kolekcí v databázi a jejich logickému provázání. Obecně lze provést neformální rozdělení na data určená k expedici na web a interní účely. Mezi ty webové data lze považovat kolekce UsersSecure a Projects.

3.1.2 Dictionary

Slovník je jednou z nejdůležitější částí aplikace. Slouží k několika odlišným účelům. Je výchozím bodem pro automatické překlady, interní slovníky a pro udržování statistik.

Tato kolekce transformuje do podoby znázorněné na diagramu 3.3 volně dostupný slovník GNU/FDL Anglicko-Český slovník[24]. Ten obsahuje informaci o slově, jeho překladu, jménu osoby, která slovo definovala, případně výslovnosti a slovním určením. V souhrnu se jedná o více než 220 000 záznamů.

Listing 3.2: Definice CSV formátu slovníku

```

obecn : <orig>,<pekklad>,<druh>,<vslovnost>,<kdo definoval>
pkklad :
fashion,styl,n:,,Michal Ambro
fashion,mda,n:,,
fashion,zpsob,n:,,Rostislav Svoboda

```

Nicméně při vyhledání záznamů dle cizího slova se vrátí několik výsledků. Vždy s jiným překladem. To je pochopitelně v pořádku. Avšak pro účely aplikace by byly třeba následné transformace. Příkladem je automatický překlad. Všechny výsledky musím seřadit vzít tu s nejčastějším překladem, aktualizovat statistiky a znovu uložit. To samé v případě interního slovníku, kdy při korekcích budeme provádět jednotlivé změny. Výhodou agregace dat do této struktury 3.3 je fakt, že se pracuje pouze s jedním dokumentem. V tomto případě může být slovo prezentováno jako unikátní index, což opět vede ke zrychlení. Transformaci má na starosti funkce `Dictionary.init(csvfile: String)`.

Listing 3.3: Dictionary

```

{
  "_id": ObjectId("5502c62b72624c3000e41a5a"),
  "key": "fashion",
  "value": "mda",

```

```
"stats": {
  "totalOccurrences": 186,
  "averageSpacing": 832
},
"additional": {
  "pronunciation": "",
  "who": ""
},
"meanings": [
  {
    "value": "mda",
    "stats": {
      "approved": 11,
      "disapproved": 0
    }
  },
  {
    "value": "zpsob",
    "stats": {
      "approved": 0,
      "disapproved": 5
    }
  },
  {
    "value": "styl",
    "stats": {
      "approved": 0,
      "disapproved": 0
    }
  }
]
}
```

3.1.2.1 Vocabulary

Jinými slovy se jedná o uživatelův slovník, respektive soupis jeho slovní zásoby. Na rozdíl od jiných činností se záznamy jednotlivých uživatelů neukládají do jedné kolekce. Její název je odvozen složeninou slova „vocab“ a unikátním ID uživatele. To zabraňuje při poruše ztrátě všech informací uživatelů a zároveň usnadňuje testování. Do budoucna by se toto rozdělení dalo využít pro paralelizaci.

Slovník bude využíván především během přizpůsobování textu na míru uživateli. V tom případě se bude dle určitého algoritmu posuzovat zdali cizí slovo v konkrétním daném významu má být skryto či nikoliv. Agregace dat zde tedy nemá žádný smysl. Pro urychlení vyhledávání stačí zavést index jak na položku *key*, tak *value*. Viz znázornění 3.4

Listing 3.4: UserVocab - slovní zásoba uživatele

```

{
  "_id": ObjectId("55479bba0cf3c5906e33bfe4"),
  "key": "fashion",
  "value": "mda",
  "stats": {
    "totalOccurrences": 6,
    "averageSpacing": 0,
    "repetitionDay": 1,
    "EFactor": 2.5
  }
}

```

3.1.3 UsersSecure

Kolekce obsahující kompletní informace o uživateli. Název obsahuje slovo Secure, protože v kódu aplikace využívám modulu SecureSocial, který zajišťuje celou problematiku přihlašování a registrování přes email či sociální síť. Tento modul pak pro svojí činnost vyžaduje ukládání některých specifických informací, které tudíž formují podobu dokumentů ukládaných do této kolekce. Struktura dokumentu je znázorněna na diagramu 3.5.

Listing 3.5: UsersSecure

```

{
  "_id": ObjectId("54ff83f506e612350b4b704c"),
  "providerId": "facebook",
  "userId": "104871568446846",
  "firstName": "Firstname",
  "lastName": "Lastname",
  "fullName": "Firstname Lastname",
  "email": "random@random.com",
  "avatarUrl":
    "https://fbcdn-profile-a.akamaihd.net/hprofile-ak-xap1/v/t1.0-1/s200x200/68836_16877921
  "roles": [
    "user"
  ],
  "authMethod": {
    "method": "oauth2"
  },
  "oAuth2Info": {
    "accessToken":
      "CAALmPJfzQVIBAOw9H0uGrevxeVuXZAikHKTGSfNtywlcPBFLeJNPLfPN7WZAlaSbc1HNPfgJlkVp3p71Wzi
    "expiresIn": 5160462
  }
}

```

3.1.4 Projects

Projekt svým zaměřením obsahuje univerzální informace jako název, URL adresu na zastupující obrázek, a jiné informace. Zároveň obsahuje informace o produktu. Tím může být kniha či vlastní texty. Tato informace má podčásti titul, autory, zástupný obrázek a odkaz na obsah.

Jak se na první pohled může zdát (viz obr. 3.6), dokument obsahuje duplicity. Ty mohou být vhodné mezi jednotlivými kolekcemi. Jenže tady jsou součástí jednoho dokumentu. Nutně tomu tak být nemusí.

Jelikož produkt může být dvojitého typu – kniha či vlastní text – je potřeba možnost ponechat vlastní volby popisků. Ze standardizovaných knih lze vyzískat kompletní informace o autorech, názvu díla a jiných informacích. To u vlastních textů nemusí být zajištěno. Navíc, lze tímto způsobem odstínit záměnu mezi stejnými knihami od jiných nakladatelství. Z tohoto důvodu jsou položky v kategorii *product* vyplňovány automaticky.

Listing 3.6: Projects

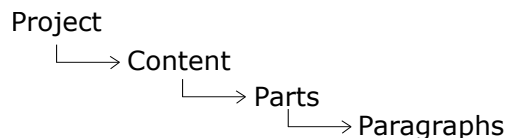
```
{
  "_id": ObjectId("5515732de5e21e36679db6e0"),
  "name": "Project Name",
  "description": "Description",
  "coverUrl": "url",
  "tags": [
    "horror",
    "drama"
  ],
  "isPublished": false,
  "meta": {
    "created": {
      "date": 1427469061237,
      "fullname": "T B",
      "userId": "hacker.tom@seznam.cz"
    },
    "updated": [

  ]
},
  "product": {
    "title": "The Ghosts of Sleath",
    "authors": [
      "Herbert, James"
    ],
    "coverUrl": "Images/00003.jpeg",
    "content": ObjectId("551573051fd7de5204153abf")
  }
}
```


3.1.5 Uložení obsahu

Při výběru databáze jsem se rozhodl pro dokumentové s představou, že celý obsah knihy uložím tak jak je. Ono je to fakticky možné a také jsem to tak na začátku řešil. Avšak brzy vyvstaly první problémy při manipulaci s takto rozsáhlým dokumentem. Za prvé je při manipulaci nutné načíst kompletní dokument. Trochu lze optimalizovat rozsah projekcí pouze nezbytných částí, ale nejedná se o významné rozdíly²⁸. Za druhé takto rozsáhlé dokumenty díky limitům prohlížečů není možné rozumně zobrazovat. To je na škodu při ladění. Zároveň jsou nepřehledné.

Postupně jsem tedy rozděloval strukturu na menší logické celky. Vzájemné provázání jednotlivých kolekcí/dokumentů znázorňuje diagram 3.1. Je třeba zdůraznit, že rozdělení je inspirováno strukturou EPUB, který obsah volitelně rozděluje na menší části.



Obrázek 3.1: Diagram logických vazeb mezi kolekcemi ukládající obsah knih

3.1.5.1 Content

Kolekce *projects* odkazuje na kolekci *content*^{3.7}. Ta obsahuje již konkrétní informace o uložení obsahu textu. Ukládá místo uložení původního textu a pak odkazy na jednotlivé soubory obsahující konkrétní části textu.

Záznam jednotlivé části obsahuje odkaz do kolekce *parts* a relativní adresu označující umístění v archivu EPUB, která zahrnuje i název souboru.

Listing 3.7: Content

```

{
  "_id": ObjectId("5515d45f1fd7dea6061547c1"),
  "filename": "data/epub/kniha.epub",
  "parts": [
    {
      "partId": ObjectId("5515d45f1fd7dea6061547c2"),
      "partHref": "OEBPS/Text/titlepage.xhtml"
    },
    {
      "partId": ObjectId("5515d45f1fd7dea6061547c6"),
      "partHref": "OEBPS/Text/part0000.html"
    },
  ],
}
  
```

²⁸Konkrétně u MongoDB záleží na konkrétních verzích a optimalizacích

3. IMPLEMENTACE

```
{
  "partId": ObjectId("5515d45f1fd7dea6061547c8"),
  "partHref": "OEBPS/Text/part0001.html"
},
{
  "partId": ObjectId("5515d4851fd7dea6061554c2"),
  "partHref": "OEBPS/Text/part0048.html"
}
],
"meta": {
  "updated": [

]
}
}
```

3.1.5.2 Parts

Jeden dokument 3.8 v kolekci parts představuje jeden soubor v archivu EPUB. Dokument obsahuje odkazy na jednotlivé odstavce do kolekce paragraphs a duplicitu popisující relativní uložení souboru.

Listing 3.8: Parts

```
{
  "_id": ObjectId("5515d4851fd7dea6061554c2"),
  "partHref": "OEBPS/Text/part0048.html",
  "paragraphs": [
    {
      "order": 0,
      "paragraphId": ObjectId("5515d4851fd7dea6061554a9"),
      "updated": [

]
    },
    {
      "order": 1,
      "paragraphId": ObjectId("5515d4851fd7dea6061554aa"),
      "updated": [

]
    },
    {
      "order": 2,
      "paragraphId": ObjectId("5515d4851fd7dea6061554ab"),
      "updated": [

]
    }
  ]
}
```

```
}
```

3.1.5.3 Paragraphs

Tato kolekce je již finální a obsahuje konkrétní text3.9. Jeden dokument odpovídá jednomu odstavci. Text je zde rozdělen na jednotlivá slova, která jsou doplněna překladem.

Listing 3.9: Paragraphs

```
{
  "_id": ObjectId("5515d4821fd7dea60615539a"),
  "text": [
    {
      "orig": "He",
      "translation": "on"
    },
    {
      "orig": "raised",
      "translation": " zvil "
    },
    {
      "orig": "the",
      "translation": "ten"
    },
    {
      "orig": "shotgun",
      "translation": "brokovnice"
    },
    {
      "orig": "to",
      "translation": "k"
    },
    {
      "orig": "his",
      "translation": "svmu"
    },
    {
      "orig": "shoulder.",
      "translation": "rameno"
    }
  ]
}
```

3.2 SRS algoritmus

Hned v úvodu je vhodné zmínit, že samotná snaha vytvořit univerzální algoritmus hledající optimální rozestupy je již z principu velice ambiciózní. Největším problémem je neschopnost generalizace činnosti mozku. Mozek každého člověka má různou schopnost retence informací. Nabízejí se dvě možnosti řešení této situace.

Nastavit algoritmus tak, aby vyhovoval co nejširší skupině lidí. Další možností je sestavení samo se učícího algoritmu. Ten by ideálně vycházel z hodnot vyhovující většině, a posléze by se adaptoval na konkrétního uživatele.

3.2.1 SuperMemo

Algoritmů pro stanovení optimálních časových rozestupů existuje celá škála. Tím nejpoužívanějším jsou algoritmy P.A.Wozniaka, který se zabývá tvorbou algoritmů od osmdesátých let, kdy zveřejnil algoritmus SM-0 (SuperMemo). Tehdejší algoritmus ještě nevyužíval počítačů, ale spoléhal se na papír a tužku. Postupem času se algoritmus vyvíjel až do dnešní verze SM-15.

Nejčastěji převzatým algoritmem je SM-2. Ten je implementován či z něj vychází produkty jako Anki1.4.1, který jsem popsal v analýze, Mnemosyne a jiných. Hlavním důvodem pro používání této rané verze algoritmu je jeho robustnost a přitom dle autora jednoduchost. Obecně je považován za dostatečně dobrý a je volně dostupný.

Další verze vylepšují algoritmus. Kritika občas zpochybňuje efekt některých nově zahrnutých aspektů. Ty neúměrně zvyšují složitost kódu, což může vést k náchylnosti k chybám. Nicméně jak se časem ukázalo, vylepšení měla svůj smysl. Sám Wozniak tvrdí, že již mnoho prostoru pro vylepšování algoritmu není, respektive: „Interestingly, the greatest progress in the algorithm is likely to come from a better definition of the model of human long-term memory.“ v odpovědi na otázku, zdali může být algoritmus SuperMemo dále vylepšován[25].

3.2.1.1 Algoritmus SM-2

Jak jsem uvedl výše tento algoritmus vystihuje podstatu SRS a zároveň je snadno pochopitelný. Při popisu vycházím ze specifikace z oficiálního zdroje SuperMemo[26]. Ta vychází z diplomové práce P.A. Wozniaka, Optimization of Learning (1990).

Algoritmus je dostupný pod licencí Open Source za podmínek přiložení následujícího copyrightu a referencí na webové stránky :
Algorithm SM-2, © Copyright SuperMemo World, 1991
<http://supermemo.com> a <http://supermemo.eu>

Vstupem algoritmus jsou dvě hlavní informace – kolikáté opakování se provádí a hodnocení správnosti odpovědi. Výsledkem je pak informace, za kolik dnů je potřeba provést opakování.

Pro odhad rozsahu dní mezi jednotlivými opakováními byl vypočítán následující vztah:

```
I(1):=1
I(2):=6
for n>2 I(n):=I(n-1)*EF
```

where:
 I(n) - inter-repetition interval after the n-th repetition (in days)
 EF - easiness factor reflecting the easiness of memorizing and retaining a given item in memory (later called the E-Factor).

Hodnota E-faktoru se smí pohybovat mezi 1.3 pro nejsložitější a 2.5 pro nejsnadnější znalosti. Pokles pod tuto hranici způsobí příliš časté opakování. Pro získání nové hodnoty musí student ohodnotit složitost úkolu (proměnná q) na škále 0-5, kde 0 – výpadek paměti a 5 – perfektní odpověď. V obecném tvaru:

```
EF' :=f(EF,q)
```

where:
 EF' - new value of the E-Factor
 EF - old value of the E-Factor
 q - quality of the response
 f - function used in calculating EF'.

specifically:
 EF' :=EF+(0.1-(5-q)*(0.08+(5-q)*0.02))

Note, that for $q=4$ the E-Factor does not change.

Kroky algoritmu jsou následující:

1. rozděl znalost na nejmenší možné celky
2. všem novým položkám nastav E-faktor 2.5
3. opakuj položky dle intervalů $I(n)$
4. získej ohodnocení odpovědi od uživatele - q
5. spočítej nový EF'
6. jestliže kvalita odpovědi (q) bylo menší jak 3, začni položku opakovat od začátku $I(1)$, $I(2)$, ..., ale neměň EF

3. IMPLEMENTACE

7. po dokončení opakování položek, pokračuj v opakování těch, které nedosáhli hodnocení 4 tak dlouho, dokud hodnocení 4 nebo lepší nezískají.

Tabulky 3.2 zobrazují změnu proměnných EF a I(n) v závislosti na konstantním hodnocení q.

3.2.2 Návrh algoritmu

Podstata této práce má však některá specifika znemožňující převzít již stávající algoritmy. Text je statický, proklad slov je neměnný a náhodný, interakce od uživatele téměř nulová (pouze před překladem textu) a čas je zde pokriven do rychlosti a objemu čtení.

Tím nejzásadnějším problémem je samotný text. Ten je v atomární podobě složen ze slov, což představuje informaci. Při prvním přečtení dochází k zavedení informace a každý další výskyt je již opakováním. Díky statické povaze textu nelze tuto frekvenci plánovat. Informace se ve většině případů dostane na řadu předčasně nebo opožděně. To je ovšem závislé ještě na další proměnné a tou je rychlost čtení textu.

Předčasné opakování lze považovat za neefektivní ve smyslu plýtvání časem. Pozdní zopakování je však už problémem vážným. Podle principů SRS je třeba informaci oživit těsně před zapomenutím. V opačném případě dojde k zapomenutí. V případě extrémně velkých časových rozestupů

Dalším zádrhelem je interakce s uživatelem. Každá informace je uložena v paměti s určitou úspěšností, jinými slovy něco se učí snadněji něco hůře. K tomu v klasickém SRS slouží ohodnocení uživatelem. Jelikož tato možnost nepřichází v úvahu, je nutné se pokusit vytvořit odhad složitosti přímo z konkrétní informace. Nabízí se přidělení složitosti dle počtu písmen slova s rozlišením: 1-3 písmena => q = 5, 4-8 písmena => q = 4, 9-více písmena => q = 3. Chování ostatních proměnných je znázorněno v tabulce 3.2.

Hodnocení: 5 triviální			Hodnocení 4 pamatuji si			Hodnocení: 3 spíše si pamatuji		
Opakování	Další den	E-Faktor	Opakování	Další den	E-Faktor	Opakování	Další den	E-Faktor
n	I(n)	EF	n	I(n)	EF	n	I(n)	EF
1	1	2.5	1	1	2.5	1	1	2.5
2	6	2.6	2	6	2.5	2	6	2.36
3	16	2.7	3	15	2.5	3	14	2.22
4	43	2.8	4	38	2.5	4	31	2.08
5	120	2.9	5	95	2.5	5	64	1.94
6	348	3	6	238	2.5	6	124	1.8
7	1044	3.1	7	595	2.5	7	223	1.66
8	3236	3.2	8	1488	2.5	8	370	1.52
9	10355	3.3	9	3720	2.5	9	562	1.38
10	34172	3.4	10	9300	2.5	10	776	1.3
11	116185	3.5	11	23250	2.5	11	962	1.3
12	406648	3.6	12	58125	2.5	12	1058	1.3
13	1463933	3.7	13	145313	2.5	13	1016	1.3

Obrázek 3.2: Simulace algoritmu SM-2 při fixaci hodnot kvality odpovědi

Kód algoritmu uvažující toto odstínění uživatele by mohl mít následující podobu:

```
def SM2Algorithm(item: VocabEntry) : VocabEntry = {
  //next scheduled repetition in days
  val nextDayToRepeat = item.stats.repetitionDay match {
    case 1 => 1
    case 2 => 6
    case value => Math.round(value * item.stats.EFactor).toInt
  }

  //simulated estimation of repetition quality response
  val q = item.key.size match {
    case 1 | 2 | 3 => 5 //perfect response
    case 4 | 5 | 6 | 7 | 8 => 4 //correct response after a
      hesitation
    case _ => 3 //correct reponse recalled with serious difficulty
  }

  val oldEF = {
    val oef = item.stats.EFactor
    if ( oef < 1.3) 1.3
    else if (oef > 2.5) 2.5
    else oef
  }

  val EF = oldEF + (0.1-(5-q)*(0.08+(5-q)*0.02))

  item.copy(stats = VocabEntryStats(item.stats.totalOccurrences +
    1,item.stats.averageSpacing,nextDayToRepeat,EF))
}
```

Nicméně algoritmus vrací interval dalšího opakování. To by znamenalo, že se překlad slova zobrazí 1x a až po uplynutí doby (určitého počtu slov) je odkryt. Během této doby by si musel čtenář slovo pamatovat, jinak by nemohl číst dál. To je při objemu různých slov v textu nemyslitelné.

V původním případě se při dřívějším opakování otázka vždy odkryje a resetuje se a naplňuje se další opakování posunutý o původní interval. To by v případě textu znamenalo, že by se slovo při každém zobrazení vždy ukáže s překladem, přičemž se plánování dalšího opakování bude odsouvat dopředu.

Z důvodů výše uvedených jsem se rozhodl použít pro-pozitivní přístup s tím, že po určitém počtu opakování čtenář již dané slovo zná. Počet opakování je vázaný na složitost daného pojmu. Ta je v tomto případě vyhodnocena počtem znaků. Čím více znaků, tím těžší. Algoritmus je následující:

```
def isToRepeat(item: VocabEntry) : Boolean = {
  // early repetition preferred
```

3. IMPLEMENTACE

```
val defaultRepeat = 130
val weight = defaultRepeat * (item.key.length/10.0)

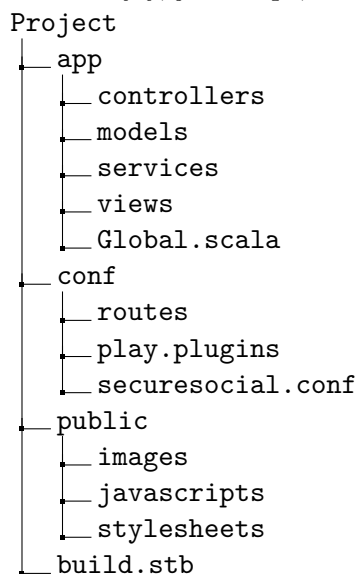
if(item.stats.totalOccurrences > weight ) false else true
}
```

Vstupem algoritmu je slovo z uživatelského slovníku. Výstupem je potvrzení nebo zamítnutí zobrazení překladu, a modifikovaný záznam o slově. Tato možnost je zvláště vhodná pro ukládání specifických dat. Pro implementaci jiného algoritmu stačí předefinovat tuto funkci.

3.3 Back-end

Jak jsem již zmínil v návrhu, server bude implementován v jazyce Scala s využitím frameworku Play. Ten vychází z filozofie MVC – Model-View-Controller. Jedná se o rozdělení kódu dle funkce. Controller řídí tok informací, model zajišťuje zpracování dat a view se stará o prezentaci dat. U tohoto rozdělení se často uplatňuje pravidlo „Fat model, thin controller“.

Následující souborová struktura, která nezahrnuje složky sloužící k interním účelům, znázorňuje uspořádání vně frameworku. Nejdůležitější je složka *app*, která je dále strukturovaná dle MVC. Pro potřeby implementací pluginů a jiných služeb jsem zavedl dodatečnou složku *services*. Dále při prvotním startu provádím přes objekt *Global* základní inicializaci. Složka *conf* obsahuje nastavení pro pluginy a routovací pravidla. *Public* dále obsahuje veřejný obsah, jako CSS styly, javascript, obrázky,...



3.3.1 Model

Nejdříve popíši způsob komunikace s databázovým serverem. Jak jsem v návrhu nastínil, pro jazyk Scala existuje ReactiveMongo driver²⁹ podporující neblokovanou komunikaci s MongoDB.

```
private val driver = new MongoDriver
private val connection = driver.connection(List("localhost"))

// Gets a reference to the database "diploma"
private val db = connection("diploma")

// Gets a reference to the collection "dictionary"
// By default, you get a BSONCollection
private val collection : JSONCollection = db("dictionary")
```

Popis kódu je dostatečně vystihující. Za zmínku stojí zdůraznit implicitní konverze BSONCollection do JSONCollection. BSON je zkratkou pro Binary JSON. Specifikace³⁰ definuje zápis JSONu do binární podoby, a zároveň přináší nové datové typy. Třídou BSONCollection definuje přímo driver databáze. JSONCollection je součástí Play Frameworku, který definuje vnitřní konverzi. Ačkoliv lze celkem podobně pracovat s oběma kolekcemi, Play poskytuje další nástroje pro pohodlnou manipulaci JSONu. Navíc, při odesílání JSONu na web by stejně muselo dojít ke konverzi. BSON kolekce tedy používám pouze v případě, kdy konverze není nutná, tedy především při ukládání informací do databáze.

Vkládání dokumentu do kolekce zajišťují funkce *insert* nebo *update*. Pro ukázkou použiji část kódu sloužícího k iniciální tvorbě databáze dictionary3.3. Ta vzniká transformací souboru sloužící jako zdroj databáze. Ten je ve formátu CSV.

```
// prearrange indexing on "key"
// NOTE: Extreme speed up !!
collection.indexesManager.ensure(Index(List("key" ->
    IndexType.Ascending),unique = true,background = true))

val selector = BSONDocument(
    "key" -> key
)

val modifier = BSONDocument(
    "$setOnInsert" -> BSONDocument(
        "key" -> key,
        "value" -> value,
        "stats.totalOccurrences" -> 0,
```

²⁹<http://reactivemongo.org/>

³⁰<http://bsonspec.org/>

3. IMPLEMENTACE

```
        "stats.averageSpacing" -> 0,
        "additional.pronunciation" -> pronunciation,
        "additional.who" -> who
    ),
    "$push" -> BSONDocument(
        "meanings" -> BSONDocument(
            "value" -> value,
            "stats" -> BSONDocument(
                "approved" -> 0,
                "disapproved" -> 0 )))
    )

collection.update(selector,modifier,upsert = true)
```

Jelikož se jedná o kód, který se bude provádět pouze při neexistenci slovníku, což se předpokládá pouze při prvotním zavedení aplikace, prvním řádkem deklaruji element *key* za unikátní index. To značně urychlí vyhledávání i vkládání.

Zdrojem dat je soubor, který na každém řádku obsahuje slovo a jeho významový překlad. Existuje-li více významů, původní slovo se opakuje. Z tohoto důvodu používám update s nastavením upsert – vložení v případě neexistence. Nejprve se udělá výběr dokumentů dle selektoru, a poté na nich provede modifikaci. Význam zápisu lze z kódu přirozeně vyčíst, tudíž ho dále nebudu popisovat.

Další ukázkou může být příklad inkrementace statistiky celkového výskytu slova. Za zmínku stojí především poslední řádek, který nepožaduje informaci o úspěšnosti. Defaultně MongoDB nedává odpověď, avšak lze toho docílit doplňujícím dotazem. Ten nám driver obstarává automaticky, protože většinou je třeba ošetřit nepříznivý stav. V tomto případě se při případné chybě nic vážného nestane, proto nepožaduji kontrolu. To pochopitelně snižuje počet požadavků na databázi.

```
// fidelity is not crucial
// absolute number of usage
def addOccurrence(key: String) : Unit = {
    val selector : BSONDocument = BSONDocument("key" -> key)
    val modifier = BSONDocument(
        "$inc" -> BSONDocument(
            "stats.totalOccurrences" -> 1 ))

    collection.uncheckedUpdate(selector,modifier)
}
```

3.3.2 Parsování a uložení EPUB

Výše jsem popisoval implicitní konverze mezi formáty JSON a BSON. Nyní přichází vhodná chvíle zmínit další z konverzí a to převod *case class* do JSONu.

Pro funkční převod je nutné napsat Read a Write pravidla převodu nebo využít automatických formátérů třídy `Json`. Následující ukázka znázorňuje operaci vkládání nového dokumentu do kolekce `content`. Argumentem pro `insert` je třída `ContentFile`, která je konvertována na json. Ten je další konverzí přeměněn na požadovaný bson dokument. Tento mechanismus funguje i naopak. Po úspěšném uložení se v kolekci nachází nový záznam v podobě zobrazené na diagramu 3.7.

```
object projectProductImplicitFormat {
  // format definition is required for each class in the scope
  (bottom-up)
  implicit val contentParagraphFormater = Json.format[ParagraphHeader]
  implicit val contentPartHeaderFormater = Json.format[PartHeader]
  implicit val contentPartFormater = Json.format[Part]
  implicit val contentFileFormater = Json.format[ContentFile]
}

def insertContentFile(content: ContentFile): Unit = {
  collection.insert(content)
}

case class ContentFile(_id: Option[BSONObjectID], filename: String,
  parts: Seq[PartHeader], meta: Meta)

case class PartHeader(partId: BSONObjectID, partHref : String)

case class Part(_id: BSONObjectID, partHref: String, paragraphs:
  Seq[ParagraphHeader])

case class ParagraphHeader(order: Int, paragraphId: BSONObjectID,
  updated: List[MetaEntry], approved: Option[MetaEntry] )
```

Naprostu stejné mechanismy jsou použity pro komunikaci s jakoukoliv jinou kolekcí. V některých případech jsou využity převody tříd, v jiných jsou operace `insert` zapsány explicitně, jako v ukázce updatu dictionary.

V kapitole Elektronické formáty 2.1.1 jsem popsal strukturu *epubu* s odkazem na specifikaci. Získávání konkrétních dat z tohoto formátu má za úkol třída *EPUBParser*. Ta získá informace o knize, jako je název díla, autory, titulní ilustraci, atd. . . Dále vytěží informace o uložení obsahu. Spolu s tím poskytuje funkce vyparsování všech odstavců. Nicméně není vhodné, aby sama zajišťovala či jiným způsobem byla vázána na databázovou strukturu.

Samotné parsování XML-like dokumentů připomíná XPath, a to především v navigaci jednotlivých souborů :

```
private val contentXML = loadXMLFileFromZip(contentFilePath)
val author = contentXML \ "metadata" \ "creator" text
val title = contentXML \ "metadata" \ "title" text
```

3. IMPLEMENTACE

```
val coverUrl = contentXML \ "manifest" \ "item" find{ node => (node \
"@id").text == "cover"} map(_ \@ "href") getOrElse("")
```

Poslední řádek má za úkol najít mezi elementy *item* takový, který má atribut nazvaný *id* s hodnotou *cover*. Po nalezení získá hodnotu atributu *href*, který odkazuje na místo uložení. V případě, že neexistuje je výsledkem prázdný řetězec. Stejným způsobem se zachází při parsování všech přidružených dokumentů.

Vzhledem k rozdělení obsahu na různé kolekce, což je popsáno v kapitole 3.1.5, a kaskádovitému rozdělení viz obr. 3.7 předávám funkcím pro vkládání záznamů jednotlivé části, které si třída dle svých potřeb dále upraví, a zavolá vkládání na nižší úrovni. Následující ukázka znázorňuje tvorbu obsahu ve třídě Content.

```
def fromEPUB(epub: EPUBParser) : BSONObjectID = {
  val id = BSONObjectID.generate

  def preparePart(partHref : String) : PartHeader = {
    val paragraphs = epub.getParagraphsFrom(partHref)

    def prepareParagraph(paragraph: (Node,Int)) : ParagraphHeader =
      {
        val id = BSONObjectID.generate
        ParagraphEntry.insertParagraphEntry(id,paragraph._1.text)
        ParagraphHeader(paragraph._2, id, List(),None)
      }

    val paragraphsObjects = paragraphs.map(prepareParagraph)
    val id = BSONObjectID.generate
    Part.insertPart(Part(id,partHref,paragraphsObjects))
    PartHeader(id,partHref)
  }

  val parts = epub.partHrefs.map(preparePart)
  val meta = Meta()
  val content = ContentFile(Some(id),epub.file,parts,meta)
  insertContentFile(content)

  id
}
```

Za zmínku stojí `epub.partHrefs.map(preparePart)`, který připraví sekvenci tříd `PartHeaders`, které dále budou odkazovat na existující části do kolekce `parts`. To zajistí funkce `preparePart`.

Asi nejkličovější část se nachází v nejnižší úrovni. Objekt `ParagraphEntry` rozděluje odstavce na jednotlivá slova, která opatřuje překlady. Děje se tak při vkládání.

```

def insertParagraphEntry(id: BSONObjectID, text: String) : Unit = {

    val words = text.split("\\s+") // split over whitespaces -
        newlines, tabs, spaces,..
    val wat = words.map{ word =>

        val textParser = "\\w+".r // regex parser definition
        // extracts only alphanumeric characters and lowercase
        // note: for lookup purpose only
        val pureWord =
            textParser.findFirstIn(word).getOrElse("").toLowerCase

        val transOpt = Await.result(
            Dictionary.getValueAndIncreaseOccurrence(pureWord) , 5
            seconds )
        val trans = transOpt.map(value => {(value \
            "value").as[String]}).getOrElse("")
        WordAndTranslation(word,trans) // the original is saved with
            punctuation intentionally
    }

    val paragraph = ParagraphEntry(id,wat)
    collection.insert(paragraph)
}

```

Nejdůležitější je část zpracování jednotlivých slov. Slovník mimo zkratky a názvy obsahuje slova napsaná malým písmem. Je tedy nutné odstranit čárky, tečky, velká písmena a jiné.

3.3.3 Export a SRS

Opačným úkolem ke vkládání je export. Získávání dat bude kvůli referencím opět probíhat vodopádovým způsobem, až do chvíle, kdy dojde k načtení každého jednotlivého odstavce. Každý odstavec je rozdělen na slova, která jsou doplněna překladem. V tu chvíli přijde na řadu ke slovu aplikace SRS algoritmu. Ten má za úkol dle slovní zásoby konkrétního uživatele a určitých vzorců odstranit (skrýt) ony překlady.

Databáze uživatele je popsána v kapitole 3.1.2.1. Aby nedocházelo ke zbytečným dotazům na databázi, tak jsou v průběhu jednotlivé záznamy cachovány. K tomu využívám třídy mutable.Map, kterou si lze představit jako databázi klíč-hodnota. Na konci exportu je uživatelův slovník opět aktualizován.

Listing 3.10: Vocabulary caching

```

def getVocab(key: String, value:String) : VocabEntry = {
    val cached = vocabMap.get((key,value))
    val item = cached.getOrElse{

```

3. IMPLEMENTACE

```
    val dbres = Await.result(userVocab.find(key, value), 10 seconds)
    val res = dbres.map(_.as[VocabEntry]).getOrElse{
      VocabEntry(key, value, VocabEntryStats(0, 0, 1, 2.5))
    }
    vocabMap((key, value)) = res
    res
  }
  //increase occurrence
  val newVocabEntry = item.copy(stats =
    item.stats.copy(totalOccurrences =
      item.stats.totalOccurrences + 1))

  vocabMap((key, value)) = newVocabEntry
  newVocabEntry
}
```

Následně je vygenerován html kód, kterým bude nahrazen původní odstavec. Podoba kódu je ovlivněna algoritmem, který má za úkol vyhodnotit, zda-li má dojít ke skrytí překladu. V následující ukázce si lze povšimnout volání algoritmu v podmínce.

Listing 3.11: Transformace paragrafu do výstupní podoby

```
def getString() : NodeSeq = {
  val p = parEntriesIterator.next()
  val words = p.text.map{wat =>

    def withTranslation(item: WordAndTranslation) =
      <span class="showtrans">
        <span>{item.orig}</span>
        <span class="trans">{item.translation}</span>
      </span>

    val resWithoutSpace =
      if(isToRepeat(getVocab(wordParser.findFirstIn(wat.orig).getOrElse("").toLowerCase, wat).
        withTranslation(wat)

        resWithoutSpace ++ xml.Text(" "))
  }
  words.foldLeft(Text("").toSeq)(_ ++ _)
}
```

Poté již zbývá nové xhtml dokumenty zabalit do epub. V této fázi je kniha připravena ke čtení.

3.4 Front-end

Poslední součástí aplikace je uživatelské rozhraní. V této práci je vnímáno spíše jako prostředník pro spouštění událostí na serveru. Z tohoto důvodu se nezabývám grafickou částí a rozhraními, které slouží k operacím pro doplňující role – schválení překladu, publikování, atd. Knihy jsou automaticky považované za schválené a publikované.

Odeslání konkrétního view mají na starosti různé funkce controlleru. Ten je zavolán při každém vstupu na konkrétní adresu. Definice provázání adresy a akce se nachází v souboru `conf/routes`.

```
POST /api/exportEPUB      @controllers.Application.exportEPUB
GET  /api/getProjectsList @controllers.Application.getProjectsList
POST /api/uploadEPUB     @controllers.Application.uploadEPUB
```

Před zpracováním každého požadavku je prováděno ověřování uživatele. O to se stará bezpečnostní modul `SecureSocial`. Vyhodnotí-li se úspěšně, provede se mnou definovaná akce. V opačném případě se automaticky zobrazí přihlašovací obrazovka s možností registrace.

První ukázka je akce z controlleru `Application`, která odpovídá na ajax žadající výpis všech projektů. Její volání zařídí GET požadavek na `/api/getProjectsList`, který je definovaný v routách (viz výše). Výsledek je odeslán v jsonu. Za povšimnutí stojí práce se `SecuredAction`, které je předaný můj kód pro exekuci v případě ověření uživatele.

```
def getProjectsList = SecuredAction.async {
  Project.getList.map(p => Ok(Json.toJson(p)))
}

def main = SecuredAction { implicit request =>
  Ok(views.html.main(request.user))
}
```

Druhá akce ukazuje jednoduché přesměrování na hlavní stránku v případě úspěchu při ověřování. Za povšimnutí stojí volání html templatu `main`, kterému jsou předány údaje o uživateli. Ty jsou v šablonovacím systému převzaty a vloženy na připravená místa do html – viz ukázka níže.

```
@(user: MySocialUser)
.... hidden ....
<ul class="nav navbar-nav navbar-right">
  <li><a href="#">My Books</a></li>
  <li><a href="#" class="thumbnail"><img src=@user.avatarUrl
    alt="logo" /></a></li>
  <li><a href="#">@user.email</a></li>
  <li><a href="/logout">Logout</a></li>
</ul>
```

3. IMPLEMENTACE

Výše uvedená ukázka lze považovat do určité míry za statickou. O seznam projektů a volání akcí, jakou je například export konkrétní knihy se již stará React.js, jQuery a jiné javascript technologie. Následující kód znázorňuje získání seznamu všech projektů. Doporučuji při čtení kódu začít atipicky ze zdola. React je požádán o render komponenty ProjectList. Této komponentě je pomocí tzv. property předána adresa, ze které získá ony projekty.

Nejdůležitější funkcí komponent je *render*, která generuje obsah. Komponenta ProjectList definuje také stavy, které jsou po načtení přes Ajax request přiřazeny získaná data. K těmto datům se přistupuje v proměnné *projectBoxes*. Ta pro každý projekt volá generování komponenty ProjectBox. V podobném duchu předávání zodpovědností a dat na nižší úrovně se docílí výsledku.

```
var ProjectList = React.createClass({
  getInitialState: function() {
    return {projects: []};
  },
  componentDidMount: function() {
    $.ajax({
      url: this.props.url,
      type: 'GET',
      dataType: 'json',
      success: function(data) {
        this.setState({projects: data});
      }.bind(this),
      error: function(xhr,status,err) {
        console.error(this.props.url, status, err.toString());
      }.bind(this)
    });
  },
  render: function() {
    var projectBoxes = this.state.projects.map(function(project) {
      return (
        <ProjectBox data={project} id={project._id.$oid} />
      );
    });

    return (
      <div id="content" className="content">
        {projectBoxes}
      </div>
    );
  }
});

React.render(
  <ProjectList url='/api/getProjectsList' />,
  document.getElementById('container')
);
```


Rozhraní pro korekturu textu v práci implementováno není. Podoba takového rozhraní se nachází v příloze A. Algoritmicky se jedná o nezajímavou část, kdy se ručně mění význam slov za jiný z nabízených vestavěným slovníkem. Ačkoliv to přispívá k uhlazení překladů a v komerční produkci se jedná o část nezbytnou, principiálně nemá s SRS nic společného.

3.5 EPUB s překlady

Abych docílil zobrazení překladů pod slovem, uzavřel jsem každé slovo do span tagů a doplnil CSS následovně:

```
<span class="showtrans">
  <span>{original}</span>
  <span class="trans">{preklad}</span>
</span>

.showtrans {display:inline-block; text-align: center; vertical-align:
  top; text-indent: 0}
.showtrans > span {display: block; white-space: nowrap}
.trans {font-size: 80%;}
```

Z ukázky vyplývá, že se jedná o sadu za sebou jdoucích bloků. To jsem se snažil vyřešit použitím speciálního tagu `<ruby>`, který slouží k zobrazování výslovnosti znaků u východo-asijských jazyků. Nicméně překlady zobrazuje nad slovem a bylo by třeba ho poupravit pomocí CSS. Dále je problém s podporou tohoto tagu prohlížeči.

Dalším pokusem o sémantičtější zápis bylo použití CCS3 a modifikátoru `:after` s vlastností `content`. To mi umožnilo každé původní slovo obklopit do jediného span tagu a překlad zaznamenat do atributu. Pominu-li limitovanou podporu e-readerů, narazil jsem na překryvy překladů v případě, že byly delší než překládané slovo. Vrátil jsem se tedy k výše uvedenému řešení.

Výsledek je vidět na obrázku 3.3

CHAPTER XIII

DR. SEWARD'S DIARY—*continued.*

THE funeral was arranged for the
pohřební průvod uspořádaný
next succeeding day, so that Lucy and her
následující
mother might be buried together. I
zahrabaný
attended to all the ghastly formalities,
navštěvovaný děsně formality

Obrázek 3.3: Ukázka výsledného úryvku z knihy Dracula od Brama Stokera

Testování

V této fázi již existuje funkční prototyp. Nyní je třeba otestovat funkcionalitu a použitelnost nástroje. Klíčovou částí této práce je finální výstup, tedy kniha s překladem ve formátu EPUB. Testování bude prověřovat algoritmus a vzhled překladu.

Z pohledu uživatele se lze dotazovat na optimální složitost (rychlost) skrývání překladů. Vzhledem k tomu, že významová kvalita překladu je závislá především na manuální korekci, záměrně opomínám dotazovat tuto vlastnost. Formátování a rozložení výsledného produktu vychází z originálu. Avšak dále je obohaceno o kód umožňující prezentaci překladů pod slovem. Ten je doplněn o styly zmenšující velikost fontu překladu a rozestup mezi slovy.

Pro získání relevantních informací jsem připravil dotazník 4.1 a export reálných textů na kterých byl průzkum prováděn. Jednalo se o volně dostupné knihy z *Project Gutenberg*³¹, který hostuje tituly např. *Dracula*, *Alice's Adventures in Wonderland* či *The Picture of Dorian Gray*.

Dotazník					
Jak rychle jsou skryta již naučená slova?	Příliš brzy	Brzy	Akorát	Pozdě	Příliš pozdě
Je zobrazení překladu rušivé?	Ano	Ne			
Velikost písma překladu bych uvítal/a	Zmenšit	Ponechat	Zvětšit		
Rezetup mezi slovy bych chtěl/a	Zmenšit	Ponechat	Zvětšit		
Připomínky k překladu	Doplňte prosím..				
Připomínky k formátování	Doplňte prosím..				

Tabulka 4.1: Dotazník pro uživatelské testování výstupu práce

Z výsledku průzkumu jsem zjistil, že po grafické stránce jsem zvolil parametry rozumně. Velikost překladu schválilo 7/10 lidí. Mezery mezi slovy vyhovují 8/10 lidí. Překlad považuje za nerušivý 9/10 osob. K formátování byla jen jedna textová připomínka: „Jak by to vypadalo s jiným fontem?“. Nejdříve se to zdálo jako dobrá připomínka, avšak při pročitání specifikace

³¹<https://www.gutenberg.org/>

4. TESTOVÁNÍ

jsem si uvědomil, že přednost většiny e-čteček spočívá v možnosti volby fontu uživatelem.

Podstatně více připomínek bylo k překladu:

- „Překlady členů jsou popisem, nikoliv překladem - The- určitý člen. Také jsem si všimla, že jsou přeloženy názvy např. měst a to není potřeba.“
- „Problém s překladem složených slov jako late-hour a jiné.“
- „Zajímavý nápad!! Ale chtělo by to vylepšit. Ze začátku jsou slova pro nováčka dost těžko k pochopení. Dále v textu, když už se objevují překlady jen u složitějších slov, tak už překlady odpovídají.“
- „Neumí to přivlastňovací slova - angels's“

Většinu z výše uvedených bodů by vyřešila manuální významová korektura. Během tohoto procesu by se automaticky upravil nejčastější významový překlad. Tím by budoucí překlady neměli zmíněné neduhy. Například uvedené slovo „the“ by odpovídalo překladu „ten“.

Slova, která jsou modifikována dle mluvnických pravidel daného jazyka nejsou, a v některých případech ani nemohou být, součástí slovníků. V angličtině se jedná třeba o přivlastňování „son's - synův“ nebo „sons' - synů“. Množné tvary tvořené pomocí přípony „-s/-es“ nebo minulé časy tvořené „-d/-ed“. Samozřejmě se nejedná o kompletní výčet možností. Nabízí se zde možnost upravovat slova do základních tvarů, použít bohatší slovník nebo v textu provádět kontextovou analýzu.

Poslední z možností přistupuje k textu tak, že analyzuje konkrétní věty. Slovům přiřadí slovní určení, časy, atd. Poté je převede do základní formy. Dle těchto charakteristik se vybere příslušné slovo ze slovníku. To by eliminovalo slovesné nekorektnosti ve větách: „The school door“ - „Ten vyučovat dveře“ namísto „Ten školní dveře“. Avšak syntetická analýza cizích jazyků je nad rámec zaměření této práce.

Jako kritické bych vyhodnotil výsledky průzkumu v oblasti vyhodnocení nabytých znalostí. 7/10 osob považuje rychlost skrývání nových slov za pomalé, 2/10 dokonce za příliš pomalé. Jen jednomu čtenáři se tempo zdálo přijatelné. V textovém popisu byla připomínka: „Některá slova se i přes časté opakování neskrývají nebo až za dlouhou dobu.“ Tyto výsledky už mají signifikantní váhu.

Z výše uvedených důvodů jsem se rozhodl snížit hladinu pro zobrazování překladu úpravou výchozího počtu opakování a zjemněním atributu `weight`. Změny jsou již zapracovány v kódu zobrazeném v návrhu algoritmu.

Konkrétní vyplněné dotazníky jsou dostupné na přiloženém CD. Ve většině případů se jedná o přepis odpovědí testerů během procesu zkoušení. V několika případech mi odpovědi byly zaslány emailem.

Nástin budoucího směřování projektu

V průběhu prací na projektu jsem našel mnoho cest, kterými by se dal nápad dále směřovat. A to jak na mikro tak i na makro úrovni.

Na úrovni nižší by zajisté šlo vylepšit filtrovací algoritmus. Jedna z možností by byla implementace pokročilejších SRS s kombinací již implementovaného algoritmu. Oprostím-li se od klasického tisknutého textu a požadavku čtení knihy na dnešních e-čtečkách, lze uvažovat tablety a jiná zařízení, která jsou schopna zprostředkovat přímou interakci s uživatelem. To by otevřelo možnost využít již existující SRS algoritmy téměř beze změny. Slova by se po zobrazovala jen ve stanovených intervalech, a v případě potřeby by se odkrývala na vyžádání.

Dalším směrem ke zkoumání je automatický překlad. Především aplikace kontextových algoritmů. Analýza slovních druhů či databáze častých spojení by mohla snížit nepřesnost překladů na přijatelnou úroveň. Poté by již nebylo potřeba manuální korekce.

Ze širšího pohledu by šlo kolem tohoto prototypu postavit sadu přidružených aplikací. Pro prodej těchto překládaných knih vlastní elektronický obchod. Pro uživatele cílené učení slovní zásoby. Funkcionalitou by se mohly podobat produktům Memrise nebo Anki. Nově nabytá slovní zásoba by byla zohledněna při tvorbě následně požadovaných knih. Nástroj pro korekci textů by též mohl být samostatnou aplikací určenou např. pro tablety. To by umožnilo využít k překladu dobrovolníky, kteří by mohli provádět korekce na cestách ve vlaku atp.

V neposlední řadě by se celý koncept dal podpořit soutěživým aspektem v rámci sociálních sítí. A to ve všech zmíněných přidružených aplikacích. Soupeření v počtu zkontrolovaných textů, přečtených knih či objemu slovní zásoby.

Závěr

V práci jsem detailně prozkoumal současné nástroje podporující výuku cizích jazyků. Vybral jsem především ty, které souvisely s principem prokládaného opakování. Zároveň jsem přiblížil psychologický aspekt učení, kde shrnuji fáze učení tzv. Bloomovu taxonomii a efektivitu učících stylů.

V rámci této práce jsem se zajímal o číselné vyjádření efektivitu čtení vzhledem k vnímání informace. Navzdory tomu, že se mnoho publikací odkazovalo na tzv. Dalův kón, nenašel jsem konkrétní průzkum. Následné hledání této informace vedlo ke zjištění, že samotný Dale nikdy čísla nezveřejnil. Čísla byla pravděpodobně výmyslem marketingu firem při prodeji produktů.

Poznatky získané v analýze určily zaměření aplikace na samouky a metodu SRS. V návrhu jsem vymezil funkci nástroje a popsal pracovní postup. Vybral jsem odpovídající technologie, popsal algoritmus a úspěšně implementoval. Aplikace umí do systému importovat knihy ve formátu EPUB a provést překlad v závislosti na algoritmu. Výsledek je zprostředkován ve stejném knižním formátu. Výstup jsem otestoval na uživatelích. Jejich poznámky jsem dále zpracoval.

Aplikace je funkční a dle reakcí uživatelů nápadem zajímavá. Prostor pro vylepšení se skrývá v nových výzkumech prohlubující znalosti v oblasti retence informace v lidském mozku a nalezení lepšího algoritmu, který bude umět efektivně transformovat časovou osu do čteného textu.

Literatura

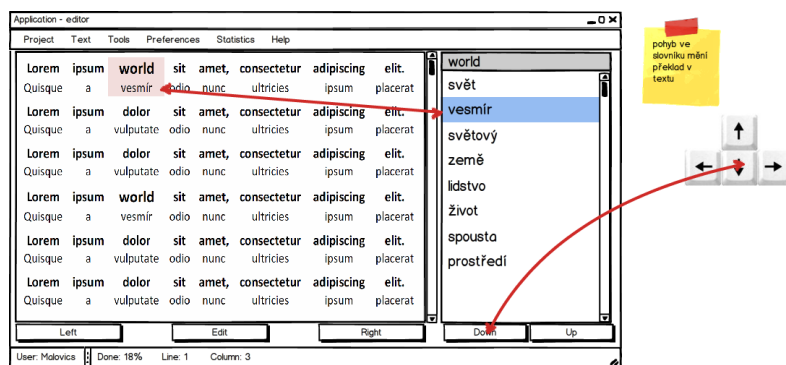
- [1] Rolls, E.; Treves, A.: *Neural Networks and Brain Function*. Oxford University Press, první vydání, 1998, ISBN 978-0-19-852432-8.
- [2] Salzman, C.; Newsome, W.: Neural mechanisms for forming a perceptual decision. *Science*, ročník 264, April 1994: s. 231–237, ISSN 0036-8075.
- [3] von der Malsburg, C.: Binding in models of perception and brain function. *Current Opinion in Neurobiology*, ročník 5, February 1995: s. 520–526.
- [4] Bullmore, E. T.; Basset, D. S.: Brain Graphs: Graphical Models of the Human Brain Connectome. *Annual Review of Clinical Psychology*, ročník 7, April 2011: s. 113–140.
- [5] King, L.: *The Science of Psychology*. McGraw-Hill, druhé vydání, 2010, ISBN 0073532061.
- [6] Oxford Dictionaries. Stav ze dne 9.12.2014. Dostupné z: <http://www.oxforddictionaries.com/definition/english/learning>
- [7] Forehand, P. M.: Bloom's Taxonomy: Original and revised. 2005, the University of Georgia.
- [8] Vlčková, M. M. K.: *Strategie učení cizímu jazyku*. Dizertační práce, Masarykova univerzita v Brně, Pedagogická fakulta, Katedra pedagogiky, 2005.
- [9] Vlčková, M. M. K.: *Žákovské strategie učení cizímu jazyku ve všeobecném vzdělávání (průřezový výzkum)*. Dizertační práce, Masarykova univerzita v Brně, Pedagogická fakulta, Katedra pedagogiky, 2010.
- [10] Wong, L. L.: The learning styles and strategies of effective language learners. *System*, ročník 36, June 2011: s. 144–163.

- [11] Richardson, J. G.: Learning Best Through Experience. *Journal of Extension*, ročník 32, č. 2, August 1994, ISSN ISSN 1077-5315.
- [12] Thalheimer, P. W.: People remember 10published on the Work-Learning Research website (www.work-learning.com) in 2002. Dostupné z: http://www.willatworklearning.com/2006/05/people_remember.html
- [13] Nesfield, J. C.: *Manual of English Grammar and Composition*. MACMILLAN AND CO.,London, 1908.
- [14] Bridge Magazine: . Dostupné z: <http://www.bridge-online.cz/>
- [15] International Encyclopedia of the Social Sciences: *Ebbinghaus, Hermann*. 1968, encyclopedia.com. (April 5, 2015). Dostupné z: <http://www.encyclopedia.com/doc/1G2-3045000327.html>
- [16] Empson, R.: *Fast-Growing Duolingo Caps Off A Big Year With The Crunchie For “Best Education Startup”*. TechCrunch.com, 2014, stav k 9.12.2014. Dostupné z: <http://techcrunch.com/2014/02/11/fast-growing-duolingo-caps-off-a-big-year-with-the-crunchie-for-best-education-startup/>
- [17] IELTS British Council: . Dostupné z: <http://takeielts.britishcouncil.org/choose-ielts/native-english-speakers>
- [18] International Digital Publishing Forum: . Dostupné z: <http://idpf.org/>
- [19] The World Wide Web Consortium (W3C): . Dostupné z: <http://www.w3.org/>
- [20] Akka: . Dostupné z: <http://akka.io/>
- [21] The Reactive manifesto: . Dostupné z: <http://www.reactivemanifesto.org/>
- [22] Carbou, M.: *Reverse Ajax, Part 1: Introduction to Comet*. IBM - developerWorks, 2015, stav k 14.4.2015. Dostupné z: <http://www.ibm.com/developerworks/library/wa-reverseajax1/wa-reverseajax1-pdf.pdf>
- [23] Harrington, C.: *React vs AngularJS – How the two Compare*. Codementor.io, 2014, stav k 29.12.2014. Dostupné z: <https://www.codementor.io/reactjs/tutorial/react-vs-angularjs>
- [24] GNU/FDL Anglicko-Český slovník: . Dostupné z: <http://slovník.zcu.cz/uvod.php>

- [25] Wozniak, D. P.: SuperMemo, 2005, questioned: Dawid Calinski, dec 10, 2004, 01:51:18. Web ke stavu 30.4.2015. Dostupné z: <http://www.supermemo.com/english/algs11.htm>
- [26] Wozniak, D. P.: Optimization of learning. 1990, publication adapted fo web (P.A.Wozniak, May 10, 1998) - stav ke dni 30.4.2015. Dostupné z: <http://www.supermemo.com/english/ol/sm2.htm>

Rozhraní pro korektory

Pro představu uvádím možnou podobu rozhraní pro korektory. Tento wireframe vznikl během vypracování zadání v předmětu Návrh uživatelského rozhraní.



Obrázek A.1: Wireframe vystihující intuitivní rozhraní pro práci korektora

Seznam použitých zkratk a termínů

Student, žák Představuje konzumenta obsahu

Lektor, učitel Představuje osobu provázející studenty procesem učení

Asistovaná výuka Výuka, při které je přítomen lektor

Habituace Organismus se učí ignorovat známé podněty, které nemají vážné následky

CJ Cílový jazyk, cizí jazyk

RJ Zdrojový jazyk, rodný jazyk

S2T, speech-to-text převod řeči na text

T2S, text-to-speech převod textu do řeči

API Application Program Interface

JSON JavaScript Object Notation

GUI Graphical user interface

XML Extensible markup language

AJAX Asynchronous JavaScript and XML

I/O Vstupně(input) výstupní(output) operace

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
│ └─ install.txt.....	informace o konfiguraci a spuštění
├─ thesis.....	zdrojová forma práce ve formátu L ^A T _E X
└─ pruzkum.pdf.....	výstup testování
text.....	text práce
├─ DP_BOHUNOVSKY_TOMAS.pdf.....	text práce ve formátu PDF
└─ Dracula.epub.....	kniha s doslovným překladem