

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačové grafiky a interakce

Čtečka dokumentů (zdrojů) pro portál nový Vokabulář webový

Bakalářská práce

Jan Vitha

Vedoucí: Ing. Vladimír Pokorný

Obor: Web a multimedia

Studijní program: Softwarové technologie a management

Květen 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vitha** Jméno: **Jan** Osobní číslo: **439596**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačové grafiky a interakce**
Studijní program: **Softwarové technologie a management**
Studijní obor: **Web a multimedia**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Čtečka dokumentů (zdrojů) pro portál nový Vokabulář webový

Název bakalářské práce anglicky:

Document (resource) reader for new Vokabulář webový portal

Pokyny pro vypracování:

Provedte analýzu potřeb uživatelů portálu Vokabulář webový. Zaměřte se zejména na moduly s dokumenty, tj. na Ediční modul, Mluvnice, Odborná literatura a Audiokničky. Navrhněte uživatelské rozhraní, které bude reagovat na různé velikosti obrazovek (mobil vs. desktop). Na základě analýzy proveďte implementaci čtečky. Implementaci proveďte v souladu s technologiemi použitými v portálu Vokabulář webový, tj. jazyk TypeScript, styly Less a integrace se serverovou částí v ASP.NET Core. Výslednou čtečku průběžně testujte se všemi typy dokumentů - obrázky, text, audio - a také v rámci uživatelských testů s alespoň 3 uživateli.

Seznam doporučené literatury:

S. Fenton. Pro TypeScript: Application-Scale JavaScript Development. Apress, 2014.
Vokabulář webový, dostupný na <https://vokabular.ujc.cas.cz/>
T. Lowdermilk, User-Centered Design, O'Reilly Media, 2013.
C. Peterson. Learning responsive web design: a beginner's guide. O'Reilly Media, Inc., 2014.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vladimír Pokorný, katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **25.01.2018**

Termín odevzdání bakalářské práce: **25.05.2018**

Platnost zadání bakalářské práce: **30.09.2019**

Ing. Vladimír Pokorný
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce, Ing. Vladimíru Pokornému, za cenné rady, které pomohly k jejímu dokončení. Rovněž bych rád poděkoval uživatelům za zpětnou vazbu při testování.

Dále děkuji rodině za podporu v průběhu studia i při vypracování této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 25. května 2018

Abstrakt

Bakalářská práce se zabývá návrhem a implementací čtečky literárních zdrojů na portálu Vokabulář webový. Návrh byl řešený s cílem co největší uživatelské přívětivosti a intuitivního ovládání. Výsledkem této práce je nová podoba čtečky literárních děl na portálu Vokabulář webový.

Klíčová slova: layout manager, Vokabulář webový, čtečka zdrojů, responzivní web design

Vedoucí: Ing. Vladimír Pokorný

Abstract

Bachelor thesis deals with design and implementation of literal resource reader for Vokabular webovy portal. Design was created to maximise user friendliness and intuitive control. The result of this thesis is new form of literary resource reader at Vokabular webovy portal.

Keywords: layout manager, Vokabulář webový, resource reader, responsive web design

Title translation: Document (resource) reader for new Vokabulář webový portal — Bachelor thesis

Obsah

1 Úvod	1		
O Vokabuláři webovém	1		
2 Analýza	3	4 Realizace	23
2.1 Struktura Vokabuláře webového .	3	4.1 Vyskytnuté problémy v průběhu implementace	23
2.2 Použité technologie v současném Vokabuláři	4	4.2 Výsledná architektura kódu	24
2.2.1 ASP.NET Core	4	5 Testování	27
2.2.2 TypeScript	5	5.1 Návrh podoby testu	27
2.2.3 LESS	5	5.1.1 Pre-test dotazník	27
2.3 Současná podoba čtečky literárních děl	6	5.1.2 Testování uživatelského rozhraní	27
2.3.1 Nedostatky současné čtečky . .	6	5.1.3 Post-test dotazník	28
2.3.2 Využitelné části současné čtečky	7	5.2 Testování s uživateli	28
2.4 Požadavky	8	5.2.1 Participant 1	28
2.4.1 Funkční požadavky	8	5.2.2 Participant 2	29
2.4.2 Nefunkční požadavky	9	5.2.3 Participant 3	29
2.5 Analýza použitelných knihoven .	10	5.2.4 Participant 4	30
2.5.1 GoldenLayout	10	5.2.5 Participant 5	30
2.5.2 wcDocker	11	5.2.6 Participant 6	31
2.5.3 DockSpawn	11	5.3 Vyhodnocení výsledků testování	31
2.5.4 PhosphorJS DockPanel	12	6 Závěr	33
2.5.5 Shrnutí porovnání jednotlivých knihoven	13	A Literatura	35
3 Návrh	15	B Seznam použitých zkratk	37
3.1 Návrh na zlepšení současné čtečky	15	C Pre-test dotazník	39
3.2 Návrh GUI	16	C.1 Otázky dotazníku	39
		3.2.1 Varianta pro desktop	16
		3.2.2 Varianta pro mobilní zařízení	18
		3.3 Návrh architektury pro kód čtečky	20

C.2 Odpovědi na dotazník.....	39
D Testované úkoly pro participanty testování	41
Desktopová varianta čtečky	41
Mobilní varianta čtečky	42
E Post-test dotazník	43
E.1 Otázky dotazníku	43
E.2 Odpovědi na dotazník.....	43

Obrázky

2.1	Struktura Vokabuláře webového .	3
2.2	Podoba čtečky při otevření všech panelů.....	7
2.3	Ukázka formuláře pro pokročilé vyhledávání	8
3.1	Zobrazení informací o díle	17
3.2	Finální podoba návrhu desktopové verze	18
3.3	Návrh řešení umístění nástrojových ikon.....	19
3.4	Finální návrh čtečky pro mobilní zařízení.....	20
3.5	Návrh struktury kódu	21
4.1	Výsledná struktura kódu	25

Tabulky

2.1	Výhody a nevýhody GoldenLayout	11
2.2	Výhody a nevýhody wcDrawer ..	12
2.3	Výhody a nevýhody DockSpawn	12
2.4	Výhody a nevýhody PhosphorJS DockPanel	13
2.5	Porovnání jednotlivých knihoven	13

Kapitola 1

Úvod

Tato bakalářská práce má za cíl na základě potřeb uživatelů Vokabuláře webového navrhnout uživatelské rozhraní pro moduly sloužící k práci s dokumenty tak, aby práce s jednotlivými historickými zdroji byla co nejvíce intuitivní a uživatelsky přívětivá. Součástí návrhu bude i příprava čtečky tak, aby byla následně schopna reagovat na různé velikosti obrazovky a stala se tak ovladatelná na přenosných zařízeních. Na základě provedené analýzy a vytvořených návrhů bude provedena implementace v souladu s technologiemi používanými na portálu Vokabulář webový. Výsledek této implementace bude na závěr otestován s koncovými uživateli.

O Vokabuláři webovém

Vokabulář webový jsou internetové stránky, které od listopadu 2006 postupně zpřístupňují textové, obrazové a zvukové zdroje k poznání historické češtiny. Tvůrcem a provozovatelem Vokabuláře webového je oddělení vývoje jazyka Ústavu pro jazyk český AV ČR, v. v. i. (dále ÚJČ). Úkolem Vokabuláře webového je poskytnout zájemcům o historickou češtinu především informace o její slovní zásobě. [19]

Roku 2010 zahájil ÚJČ spolupráci s Českým vysokým učením technickým s cílem vytvoření modernější podoby Vokabuláře webového. Projekt s názvem ITJakub měl vytvořit portál s využitím modernějších technologií, který by byl dostupný na různých typech zařízení, ať už se jedná o stolní počítače, mobilní telefony, tablety či čtečky elektronických knih. Součástí projektu bylo vytvoření nástrojů pro zapojení informací uchovaných na portálu do výuky na různých typech škol. Tento projekt byl dokončen roku 2015.[18]

Cílem navazujícího projektu s názvem RIDICS je vytvoření platformy pro uchování jazykového kulturního dědictví. První částí tohoto projektu je

vytvoření badatelského portálu, sloužícího k výzkumu dějin českého jazyka, kultury a umění zpřístupněním různorodých odborně zpracovaných zdrojů doplněné o metadata. Účelem badatelského portálu je vytváření nástrojů pro práci s daty za účelem umožnění jazykového bádání v mnoha oblastech, stejně tak jako zpřístupnění děl literárním vědcům, historikům a dalším specialistům z oblasti humanitních oborů.[4]

Druhou částí Vokabuláře webového je komunitní portál zpřístupněný odborníkům, studentům i široké veřejnosti, který umožní sdílet výsledky výzkumu badatelů a informovat o dění v jednotlivých oborech tuto cílovou skupinu. Zároveň umožní uživatelům portálu vést nad odbornými tématy debatu. [4]

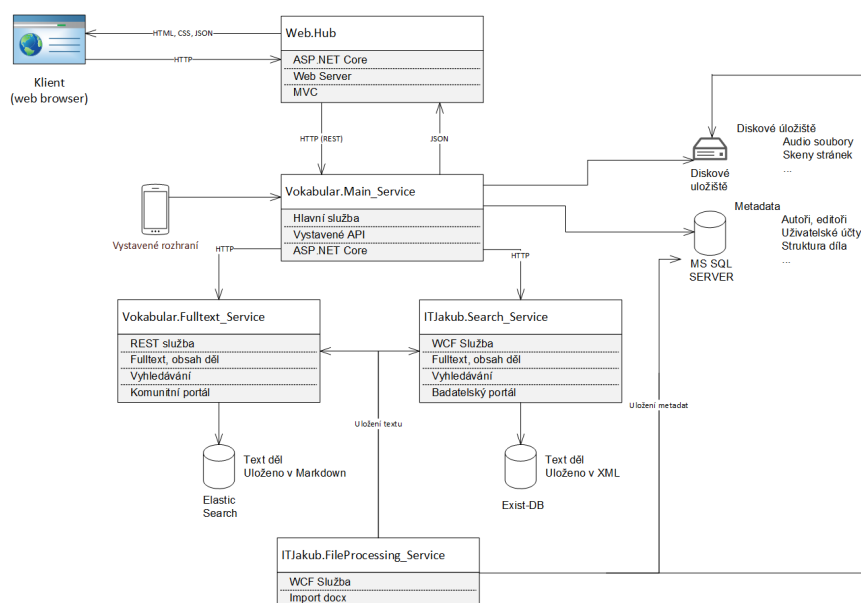
Kapitola 2

Analýza

V této kapitole je popsána struktura a použité technologie v projektu Vokabulář webový. Dále je cílem této kapitoly provést analýzu současné podoby čtečky literárních děl. Na základě této analýzy a analýzy potřeb uživatelů portálu Vokabulář webový jsou vytvořeny funkční požadavky, s jejichž využitím jsou následně porovnávány knihovny nabízející potřebné funkcionality.

2.1 Struktura Vokabuláře webového

Vokabulář webový je webová aplikace využívající technologii .NET. Celá tato aplikace je rozdělena do několika služeb, jak je zobrazeno na diagramu 2.1.



Obrázek 2.1: Struktura Vokabuláře webového

Klient posílá požadavek na Web.Hub, odkud dostává odpověď ve formě vygenerované HTML stránky a jejích stylů. Zároveň je tento požadavek zpracován Controllerem, odkud je předáván službě Vokabular.Main_Service, odkud je požadavek vyřízen, nebo v případě, že služba není schopna požadavek zpracovat, předán jedné z přidružených specializovaných služeb. Každá z jednotlivých služeb dále z dostupných databází získává požadovaná data, která jsou následně pomocí Vokabular.Main_Service vrácena jako JSON na Web.Hub, kde Controller zpracuje odpověď, vrátí příslušná data, která následně AJAX na straně klienta zpracuje a předá do dříve poslané stránky.

Služba ITJakub.Search_Service využívá databázi eXist-DB, která slouží jako úložiště textu. Tato služba je využívána pro fulltextové vyhledávání v díle pro badatelský portál. Obdobou této služby pro komunitní portál je Vokabular.Fulltext_Service se svou vlastní databází Elastic Search.

API Vokabuláře je vystaveno na službě Vokabular.Main_Service, což umožňuje následně vytvořeným rozhraním komunikovat přímo s touto službou. Příkladem takového rozhraní může být mobilní aplikace.

Pro nahrávání děl do Vokabuláře slouží služba ITJakub.FileProcessing_Service, která nahraný soubor zpracuje, metadata o díle uloží do databáze MS SQL Server. Text díla následně předá příslušné službě, která následně tento text uloží do své vlastní databáze.

2.2 Použité technologie v současném Vokabuláři

Implementace nové podoby čtečky literárních děl bude probíhat nad již existujícím projektem. Analyzované technologie jsou používány v rámci Web.Hub, kde bude probíhat následná realizace nové podoby čtečky literárních děl.

2.2.1 ASP.NET Core

Serverová část Vokabuláře webového využívá framework ASP.NET Core, který je součástí .NET frameworku sloužícího pro tvorbu webových aplikací a služeb. Tato sada knihoven nabízí mnoho řešení základních problémů, jako je například bezpečnost, práce s databází a mnoho dalších. [21]

Projekt Vokabuláře je rozdělen strukturou MVC, tedy Model-View-Controller, která je nabízena tímto frameworkem k řešení webových aplikací. V této struktuře Model zastupuje logiku na straně serveru, od které Controller přebere výstupní data, která následně předá do části View. Ta zastupuje v podstatě šablonu pro předaná data. V případě Vokabuláře je pak v těchto šablonách ve většině případů pro zpracování předaných dat použit TypeScript, nadstavba JavaScriptu. [20]

Pro vytváření View slouží v ASP.NET Core view-engine s názvem Razor¹, který má za cíl minimalizovat nutnou syntaxi vkládanou do View HTML šablon a poskytnout jednoduchý způsob zápisu. Pro psaní kódu je využita syntaxe jazyka C#, která je uvedena znakem zavináče². [13] [7]

■ 2.2.2 TypeScript

Vygenerovaný View na straně serveru je následně na straně klienta doplněn kódem JavaScriptu. V případě projektu Vokabulář webový je nad JavaScriptem využita nadstavba TypeScript.

TypeScript obsahuje veškeré funkce klasického JavaScriptu, které doplňuje o další vlastnosti, jako je typová kontrola proměnných nebo zavedení tříd a rozhraní s možností dědění mezi třídami nebo implementací rozhraní.[6] Jelikož TypeScript je nadstavbou nad funkcemi JavaScriptu, potom lze i běžný JavaScriptový kód považovat za validní kód v jazyce TypeScript³. Vytvořený kód v TypeScriptu je následně kompilován pro použití prohlížečem do JavaScriptu.[2]

Další výhodou TypeScriptu oproti klasickému JavaScriptu je typová definice. Jednotlivým proměnným je možno přiřadit typy, čímž při psaní kódu má programátor typovou kontrolu, která v JavaScriptu není dostupná. Výhodou tohoto přístupu je rovněž do jisté míry snadnější pochopitelnost kódu pro nově zapojeného programátora. JavaScript díky absenci typové kontroly dovolí do proměnné uložit libovolnou hodnotu, ať už se jedná o číslo, text, či jiný objekt, čímž je orientace v kódu oproti typovému TypeScriptu výrazně komplikovanější. Tyto datové typy může programátor vytvářet sám jako třídy či rozhraní. TypeScript rovněž nabízí využití generických datových typů, čímž umožňuje programátorovi napsat obecný algoritmus nezávislý například na typu parametru. [6] [17]

■ 2.2.3 LESS

Stejně jako v případě JavaScriptu, i nad klasickým jazykem CSS pro úpravu stylů jednotlivých stránek v případě Vokabuláře pracuje nadstavba, konkrétně s názvem LESS. Tato knihovna zavádí do klasického CSS vlastnosti známé z programovacích jazyků, za zmínku nejvíce stojí možnost vytvářet proměnné.

Již bylo zmíněno, že LESS nabízí podporu proměnných. V praxi pak využití této vlastnosti znamená, že oproti klasickému CSS, kde se opakující hodnota

¹Razor byl zaveden v ASP.NET MVC 3

²toto se týká pouze Razor souborů s příponou *.cshtml, která je používána v projektu Vokabuláře webového

³vyjímku tvoří příkaz *with* a rozšíření typické pro různé poskytovatele

musela hledat napříč dlouhým kódem a kolikrát i mnoha soubory, v případě LESS stačí zadefinovat proměnnou, která se následně využívá napříč celými styly a případná změna je pak mnohem méně komplikovaná. V případě úpravy takové proměnné na jednom místě pak LESS nabízí i úpravy výrazů za pomoci matematických operací nebo za pomoci různých funkcí. [16]

Velkou výhodou LESS je rovněž možnost zanořování. V CSS se v případě zaměření elementu zanořeného v jiném musí psát delší selektory oddělené mezerami. Oproti tomu LESS nabízí možnost zanořování v rámci kódu, tedy do stylů pro element vložíme selektor pro vnořený element. Tím se zvýší přehlednost kódu, zároveň je i omezeno opakování selektorů v kódu. [12]

2.3 Současná podoba čtečky literárních děl

V současné podobě čtečka literárních děl Vokabuláře webového⁴ nabízí uživatelské rozhraní, které využívá panely pro práci a zobrazení jednotlivých prvků čtečky literárních děl. Jednotlivé panely jsou následně rozděleny do dvou skupin podle jejich pozice v rámci stránky, tedy na Left-side a Right-side.

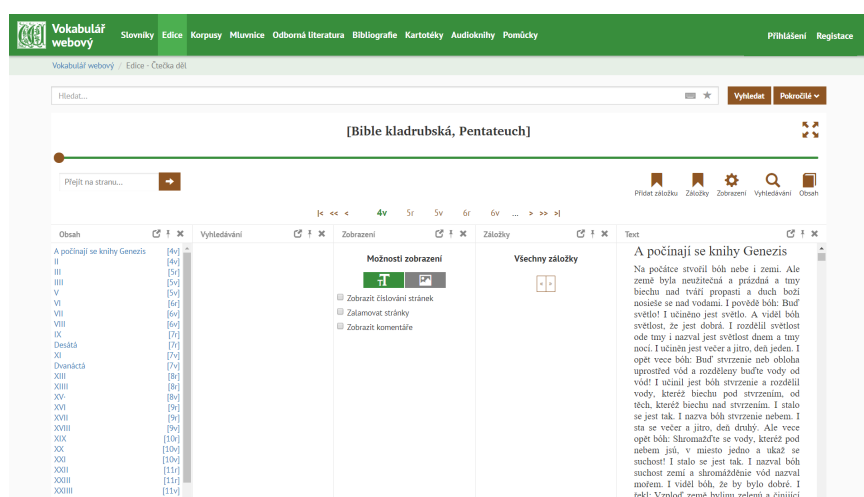
2.3.1 Nedostatky současné čtečky

V současné podobě má každé dílo svá metadata, informující o editorech díla, autorech, datu poslední úpravy a další. Rovněž je nabídnuta i ediční poznámka k jednotlivým dílům. Tyto informace jsou však dostupné pouze v seznamu jednotlivých děl a v případě zobrazení konkrétního díla a přechodu do čtečky jsou tyto informace nedostupné.

Ze současného rozdělení panelů podle jejich pozice na stránce plyne nedostatek v podobě nízké míry přizpůsobení rozložení jednotlivých panelů. Left-side panely se otevírají z levé strany bez možnosti přesunutí. Způsob otevírání od kraje vedle sebe pak způsobuje, že již otevřené panely se posouvají dále a může dojít v krajním případě otevření všech dostupných panelů, že na prohlížení textu zbyde velmi málo místa (jak je vidět na obrázku 2.2) nebo dokonce dojde v případě menší šířky okna prohlížeče k zalomení panelů do dvou řádků.

Jednou z možností, která je uživateli nabídnuta pro manipulaci s panely je otevření do samostatného okna. Tato idea je zde však pouze nastíněna, jelikož nelze takto otevřené okno navrátit zpět do původního okna. Zároveň v případě otevření nástrojových panelů není načten v novém okně jejich obsah, zatímco v případě otevření panelu s textem díla pak k načtení obsahu sice dojde, nicméně text nereaguje na události vyvolané v původním okně. Jako

⁴Porovnávaná podoba čtečky je dostupná na adrese <http://censeo.felk.cvut.cz>



Obrázek 2.2: Podoba čtečky při otevření všech panelů

příklad tohoto chování lze uvést přechod na konkrétní stranu, kdy v původním okně se zdá, že skutečně k přechodu na požadovanou stranu díla došlo, text díla v samostatném okně však zůstává na své původní pozici.

S umístěním panelů nastává problém i v případě zobrazení Vokabuláře webového na mobilních zařízeních, kdy se panely s nástroji otevírají nad zdrojem díla, čímž ho odsouvají do spodní části obrazovky, čímž nutí uživatele v případě více otevřených panelů k dlouhému scrollování, pokud chtějí tento prvek zobrazit. Jelikož se však jedná o samotný zdroj díla, měl by být snadno dostupný.

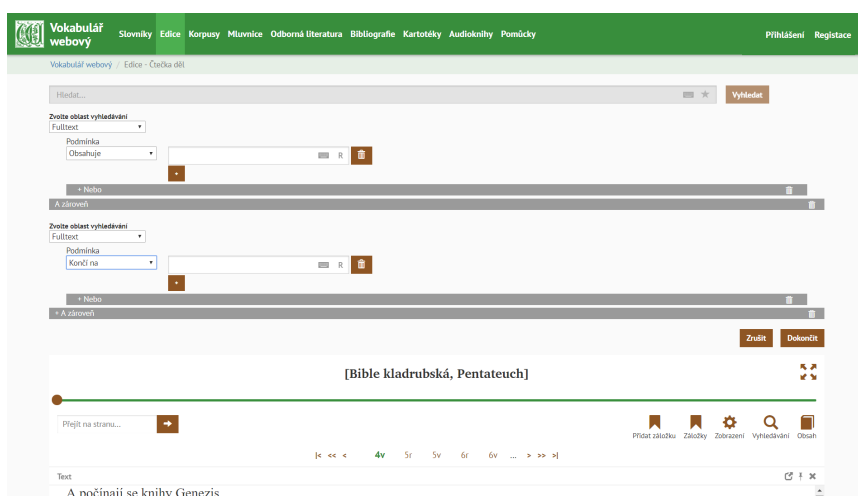
Posledním nedostatkem je pak nedostatečné pokrytí různých druhů zdrojů literárního díla. V současné podobě čtečka umožňuje práci s díly, které mají textový nebo obrazový zdroj, nicméně neexistuje nástroj pro zobrazení audio knih. Ty jsou v současné době řešeny zobrazením všech stop již v seznamu audio knih a nikoliv v samostatné čtečce.

2.3.2 Využitelné části současné čtečky

Současná podoba čtečky literárních děl nabízí velmi kvalitní způsob navigace v rámci díla. Uživateli je nabídnut posuvník, který slouží k rychlému pohybu napříč celým dílem. Navíc při posunu na tomto prvku je zobrazena nápověda s informací, na kterou stranu se uživatel právě snaží přesunout. Díky této vlastnosti může tedy tento nástroj sloužit i pro přesun na konkrétní stránku. Pro přesun na konkrétní číslo stránky je pod zmíněným posuvníkem k dispozici pole, které přijímá jako vstup konkrétní název stránky, který navíc při psaní doplňuje nápovědou. Posledním prvkem sloužícím k navigaci je pak klasické stránkování pro pohyb v okolí současné stránky. Díky kombinaci těchto tří prvků má každý uživatel možnost orientace v díle dle svých vlastních

preferencí.

K dispozici je uživatelům fulltextové vyhledávání napříč celým dílem. To je nabízeno ve dvou variantách. První možnost vyhledávání je klasické hledání konkrétního slova jeho vepsáním do formuláře a následným odesláním dotazu. Druhou možností je pak pokročilé vyhledávání. Zde je uživateli nabídnuto vytváření částečného vyhledávání, které lze doplnit o další podmínky. (ukázka formuláře pro pokročilé vyhledávání je znázorněna na obrázku 2.3) Na základě těchto podmínek je vygenerován do vyhledávacího řádku dotaz, který uživatel odešle. Výsledkem je vyznačení výsledků vyhledávání v textu spolu s odkazy na tyto výsledky.



Obrázek 2.3: Ukázka formuláře pro pokročilé vyhledávání

Pro práci s dílem čtečka nabízí několik nástrojů, ať už se jedná o obsah díla, fulltextové vyhledávání v díle, vytváření uživatelských záložek či přizpůsobení zobrazení zdroje díla. Tím je uživateli nabídnut pohodlný způsob pro práci s jednotlivými zdroji díla.

2.4 Požadavky

Na základě analýzy současné podoby čtečky byly sepsány požadavky pro její novou podobu. Při tvorbě požadavků bylo přihlédnuto i k potřebám uživatelů portálu Vokabulář webových.

2.4.1 Funkční požadavky

1. Uživateli je umožněno zobrazit v rámci otevřeného díla metadata
2. Uživatel může zobrazit ediční poznámku k danému dílu

3. Čtečka umožňuje zobrazit různé typy zdrojů (text, obraz, audio) s přihlédnutím k faktu že dílo může mít více zdrojů
4. Čtečka nabízí režim celého okna, sloužící především pro čtení textu díla
5. Uživateli je umožněna navigace v díle
 - V okolí současné stránky
 - Napříč celým dílem
6. Čtečka nabízí fulltextové vyhledávání napříč textem díla
 - Základní
 - Pokročilé
7. Vytváření záložek v rámci literárního díla
8. Uživatel má k dispozici nástroje pro práci s literárním dílem
 - Obsah
 - Záložky a jejich vytváření
9. Uživateli je umožněno upravovat zobrazení zdrojů literárního díla - čísla stránek, komentáře,...
10. Uživateli je umožněno zavřít a znovu otevřít každý existující panel
11. Uživatel má možnost přizpůsobit rozložení jednotlivých panelů
12. Čtečka umožňuje pracovat s nástroji v samostatných oknech
13. Na mobilním zařízení čtečka nabídne režim určený primárně ke čtení - tedy se skrytými nástroji a vyhledáváním; s maximálním využitím prostoru pro text díla

2.4.2 Nefunkční požadavky

1. Čtečka je schopna reagovat na různé velikosti obrazovky - konkrétně je připravena pracovat s mobilními zařízeními
2. Ke změně rozložení panelů je využit layoutovací framework
3. Obsah jednotlivých panelů čtečka zachovává při jejich zavření a následném znovuotevření
4. Pro realizaci jsou využity technologie používané současným Vokabulářem webvým
5. Čtečka je integrována jako součást Vokabuláře webového

■ 2.5 Analýza použitelných knihoven

Na základě předchozí analýzy portálu a funkčních požadavků čtečky literárních děl byly porovnány layout manager, který by svou funkcionalitou co nejvíce vyhovoval získaným požadavkům.

Při výběru vhodné knihovny bylo nejvíce přihlíženo k těmto kritériím

- Pozicování oken
 - Otevření panelu do samostatného okna
 - Změna pozice okna v rámci rozložení
 - Snadná změna rozměrů
- Možnost otevřít a zavřít panely v rámci rozložení
- Možnost nastavit, do jaké míry je uživateli dovolena interakce s jednotlivými panely a celým layoutem
- Podpora mobilních zařízení
 - Možnost přizpůsobení nabízených funkcí
- Volně dostupná licence

■ 2.5.1 GoldenLayout

Tento layout manager⁵ umožňuje vývojářům kontrolovat jaké části rozvržení jsou na stránce zobrazeny a jaké možnosti interakce mohou mít uživatelé s rozhraním. Tato interakce umožňuje uživateli například pohybovat prvky v rámci rozložení nebo vytvářet nové, přičemž všechny tyto prvky v celém rozvržení do sebe zapadají a vyplňují celou plochu. Další možností v rámci interakce s rozhraním je snadné vyjmutí prvku do samostatného okna a opětovné navrácení do rozložení. [8]

Nevýhodou tohoto layout manageru je problematická změna pozice okna na dotykových obrazovkách. Knihovna však nabízí možnost přizpůsobit, do jaké míry bude uživateli dovoleno manipulovat s rozložením, čímž by bylo možné tento nedostatek kompenzovat.

GoldenLayout je nabízen pod volně dostupnou licencí MIT [8].

⁵Layout manager dostupný na <https://golden-layout.com/>

Výhody	Nevýhody
Jednoduchá změna pozice panelů	Problematické ovládání na dotykových zařízeních
Více variant předem připravených CSS	
Přívětivé uživatelské rozhraní	
Dynamické přidávání panelů do rozložení	
Možnost přizpůsobení způsobu otevření nových panelů	

Tabulka 2.1: Výhody a nevýhody GoldenLayout

2.5.2 wcDocke

Stejně jako předchozí i zde se jedná o layout manager umožňující interakci s rozvržením na stránce⁶. Prostředí stránky je rozděleno na menší okna, která následně může uživatel libovolně dle svých preferencí rozmístit. Oproti GoldenLayout však postrádá možnost otevření okna do nového okna prohlížeče. Panel jako takový lze umístit kdekoli v rámci stránky, nicméně nikdy nevznikne nové okno prohlížeče. [10]

I v tomto layout manageru je velmi obtížný pohyb s jednotlivými okny v rámci rozvržení na mobilních zařízeních.

wcDocke je nabízen pod volně dostupnou licenci MIT [10].

2.5.3 DockSpaw

DockSpaw⁷ se od prvních dvou layout managerů liší způsobem interakce, jakým uživatel umísťuje jednotlivé panely. Oproti předchozím, kde panel byl „upuštěn“ na místě kam měl být vložen, zde probíhá dokování pomocí přetažení panelu na ikonku znázorňující nové umístění, v ostatních případech je panel umístěn volně v rámci okna prohlížeče. Opět však stejně jako wcDocke neumožní vytvoření nového okna prohlížeče ale pouze pohyb panelu v rámci již existujícího okna. Na dotykových obrazovkách mobilních zařízení nenabízí vůbec možnost změny pozice panelů.

Tato knihovna se svým uživatelským prostředím snaží přiblížit Visual

⁶Layout manager dostupný na <http://docker.webcabin.org/>

⁷Layout manager dostupný na <http://www.dockspawn.com/>

Výhody	Nevýhody
Jednoduchá změna pozice panelů	Problematické ovládání na dotykových zařízeních
Více variant předem připravených CSS	Nelze otevřít panel do samostatného okna
Možnost skrýt panely k okraji rozložení	Pomalejší odezva
Panel lze libovolně umístit jako plovoucí okno	
Dynamické přidávání nových panelů	

Tabulka 2.2: Výhody a nevýhody wcDocker

Studio IDE. [1] Díky této podobnosti se knihovna nejeví jako uživatelsky přívětivá pro uživatele na portálu Vokabulář webový.

DockSpawn je nabízen pod volně dostupnou licencí MIT [1].

Výhody	Nevýhody
Panel lze libovolně umístit jako plovoucí okno	Uživatelské prostředí podobné VisualStudio IDE
	Nelze otevřít panel do samostatného okna
	Umístění panelu pomocí ikon určující pozici
	Nelze měnit pozice panelů na dotykových zařízeních

Tabulka 2.3: Výhody a nevýhody DockSpawn

2.5.4 PhosphorJS DockPanel

Projekt PhosphorJS nabízí množství knihoven, mimo jiné také layout manager DockPanel⁸, sloužící k rozdělení webové aplikace do menších panelů. [3] Cílem tohoto layout manageru je vytvořit v okně prohlížeče panely, jejichž pozice lze následně uživatelem libovolně upravit. Zároveň umožňuje snadné a intuitivní změny rozměrů jednotlivých panelů pomocí posunu jejich okrajů. Tento layout

⁸Layout manager dostupný na <http://phosphorjs.github.io/examples/dockpanel/>

manager však neumožňuje otevření panelu v samostatném okně, což je jedním z hlavních požadavků při výběru knihovny a proto se tato knihovna jeví jako nevhodná pro potřeby mnou navrhované čtečky literárních děl.

PhosphorJS nabízí své knihovny pod licencí BSD 3-Clause.

Výhody	Nevýhody
Ve výchozí podobě pří- větivé uživatelské pro- středí (zapadající svým vzhledem do současné podoby Vokabuláře)	Nelze otevřít panel do samostatného okna
	Nelze umístit panel jako plovoucí okno
	Nelze měnit pozice pa- nelů na dotykových zařízeních

Tabulka 2.4: Výhody a nevýhody PhosphorJS DockPanel

2.5.5 Shrnutí porovnání jednotlivých knihoven

	GoldenLayout	wcDocker	DockSpawn	PhosphorJS
Licence	MIT[8]	MIT[10]	MIT[1]	BSD 3- Clause[5]
Pozicování panelů	Ano	Ano	Ano	Ano
Samostatná okna	Ano	Ano (pouze plovoucí okna)	Ano (pouze plovoucí okna)	Ne
Podpora mobilních zařízení	Ne	Ne	Ne	Ne

Tabulka 2.5: Porovnání jednotlivých knihoven

Žádná z porovnávaných knihoven nenabízela kvalitní podporu ovládání na mobilních zařízeních. Ačkoliv své funkcionality nabízely, jejich ovládání na dotykových obrazovkách v kombinaci s malým prostorem bylo velmi problematické. Jelikož se nepodařilo nalézt knihovnu, která by tento požadavek splnila, není absence tohoto požadavku při porovnávání považována za zásadní.

Každý layout manager je nabízen pod volně dostupnou licencí pro komerční i nekomerční užití, čímž všechny splňují požadavek pro použití ve Vokabuláři webovém. PhosphorJS nabízí svou knihovnu pod licencí BSD, která umožňuje volné použití a další šíření, včetně šíření s úpravami. Zbylé knihovny

jsou k dispozici s licencí MIT, která s BSD vychází s tím rozdílem, že více konkretizuje termíny *použití* a *šíření* [15]

Ačkoliv změnu pozice panelů uživatelem nabízí každá knihovna, práce s panely v samostatném okně umožňuje pouze jedna. Vedle knihovny GoldenLayout nabízí libovolné pozicování panelu i další dvě knihovny, wcDocker a DockSpawn, nicméně tyto knihovny nabízí pouze možnost umístit panel pouze v rámci aktuálního okna prohlížeče jako plovoucí okna. Nejedná se tedy o nová okna a není možné je umístit mimo okno webového prohlížeče a nelze tedy prohlásit, že by tyto knihovny zcela splnily požadavek na samostatná okna. Z porovnávaných knihoven je nejméně vhodná PhosphorJS z důvodu absence možnosti umístění panelu mimo layout do samostatného okna, navíc nenabízí ani možnost umístit libovolně panel jako plovoucí okno.

Při porovnání jednotlivých výhod a nevýhod zmíněných knihoven se jako nejlepší volba jeví GoldenLayout, jelikož nabízí přívětivé uživatelské rozhraní. Přizpůsobení pozice jednotlivých panelů je v tomto layout manageru velmi jednoduchá. Oproti jiným knihovnám nenabízí možnost umístění panelu do plovoucího okna, nicméně jako jediná plní požadavek na otevření panelu do samostatného nového okna prohlížeče.

Kapitola 3

Návrh

Cílem této kapitoly je vytvořit takový návrh, který uživateli zpříjemní a usnadní práci s literárním dílem. Toho lze dosáhnout kombinací přehledného uživatelského prostředí spolu se splněním uživatelských požadavků pro pozitivní zkušenost s používáním vytvořené čtečky. [11]

3.1 Návrh na zlepšení současné čtečky

Při analýze současné podoby čtečky literárních děl (viz kapitola 2.3) byly odhaleny některé nedostatky, jejichž řešením bude nutné se zabývat již během návrhu jednotlivých uživatelských rozhraní. V následujících několika odstavcích bude pouze nastíněno, jak dané problémy řešit, přičemž skutečný návrh řešení bude znázorněn v podkapitole 3.2.

Prvním zjištěným nedostatkem byla absence informací o díle spolu s ediční poznámkou po otevření čtečky. Jelikož se nejedná o součást díla jako takového, ale pouze o doplňující informace, jeví se jako nevhodné zakomponovat tyto údaje přímo do čtečky jako takové. Rovněž díky možnému rozsahu ediční poznámky bude vhodnější tyto údaje zobrazovat až v případě potřeby, ve výchozím stavu pak budou skryté.

Možný problém pro uživatele mohl vyvstat v momentě, kdy by bylo potřeba otevřít další zdroj daného díla. Aby se zamezilo zbytečnému zmatení uživatele, dojde ke sjednocení způsobu otevírání panelů s obsahem díla a panelů s nástroji pro práci s dílem.

Další problém pro uživatele může nastat v momentě, kdy se rozhodne otevřít více nástrojových panelů. Tento nedostatek bude možné vyřešit za pomoci seskupování panelů do jedné skupiny, ve které bude umožněno s jednotlivými panely pracovat, přičemž tato skupina zabere pouze menší část v celém

rozložení. Toto řešení je navíc podporováno zvoleným layout managerem. (viz podkapitola 2.5.1) Tento layout manager rovněž vyřeší i nedostatek s otevřením panelu do samostatného okna a jeho následným navrácením zpět do rozložení čtečky.

3.2 Návrh GUI

Jelikož je cílem této práce nová podoba čtečky literárních děl, před samotnou implementací bude nutné vytvořit návrh, jakým způsobem rozložit jednotlivé komponenty v nové podobě čtečky.

Čtečka literárních děl musí reagovat na různé rozměry obrazovek tak, aby byla zobrazitelná na každém zařízení se zachováním svých funkcionalit. Bude tedy třeba vytvořit více návrhů, jeden pro desktopovou variantu a druhý pro mobilní zařízení. [14]

Tyto návrhy budou vytvořeny s využitím software Axure RP8, který umožní vytvořit představu o následné podobě čtečky literárních děl.

3.2.1 Varianta pro desktop

Prvním vytvořeným návrhem bude podoba pro desktop. V této variantě nebude příliš problém s nedostatkem prostoru.

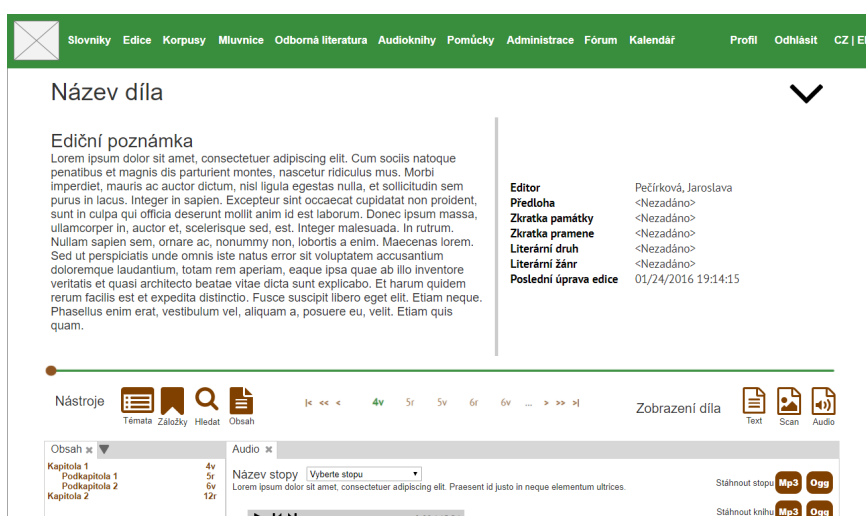
Rozložení ovládacích prvků

Při návrhu desktopové podoby čtečky literárních děl bylo nutné především změnit rozložení jednotlivých prvků v rámci hlavičky díla.

V současné podobě čtečky není k dispozici přístup k ediční poznámce nebo metadatům literárního díla. Tyto dvě informace budou v návrhu umístěny jako skrytý prvek, který bude možné zobrazit pomocí ikony na úrovni nadpisu. Po zobrazení vznikne prostor pro zobrazené informace odsunutím samotné čtečky do nižší části obrazovky. (viz obrázek 3.1)

Pro orientaci a pohyb v rámci literárního díla slouží posuvník, který bude sloužit především k přesunu na libovolnou stránku v rámci díla. Tento ovládací prvek doplní stránkování, které slouží pro pohyb v okolí aktuální stránky, přičemž uživateli bude dána zpětná vazba o tom, na jaké stránce se právě nachází díky zvýraznění aktuálního čísla stránky.

Jelikož funkce formuláře, který v původní čtečce sloužil k přesunu na li-



Obrázek 3.1: Zobrazení informací o díle

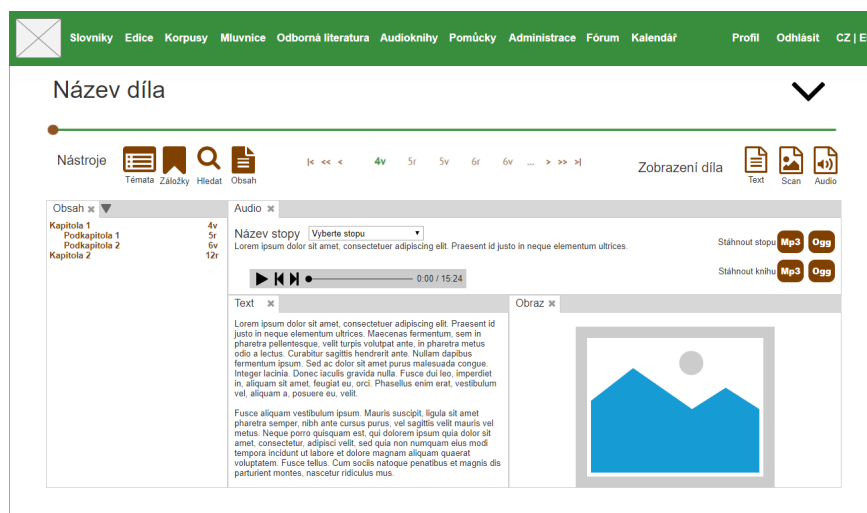
bovolnou stránku dle jejího čísla, bude zastoupena dostatečně zbývajícími dvěma prvky pro orientaci v díle, tento nástroj se jeví nadbytečným. Odstraněním tohoto prvku vznikne prostor pro nové umístění ikon otevírajících nástrojové panely. Tyto ikony budou přesunuty na levou stranu obrazovky s tím, že ve výchozím nastavení se budou nástrojové panely otevírat v levé části rozložení samotného díla. Pro uživatele tedy může být přívětivější krátká vzdálenost mezi ikonou a nově vzniklým prvkem. Na pravé straně obrazovky pak vznikne prostor pro ikony ovládající zobrazení jednotlivých zdrojů literárního díla.

■ Rozložení panelů

Při návrhu, jakým způsobem budou panely rozmístěny v rámci rozložení se jedná pouze o návrh výchozího zobrazení. Zvolený layout manager následně umožňuje uživateli jednotlivé panely rozmístit libovolně dle jeho vlastních preferencí.

Nástroje budou v rámci rozložení umístěny pomocí karet. Mezi jednotlivými otevřenými nástroji bude pak možné snadno přepínat kliknutím na příslušnou kartu, přičemž v případě většího množství panelů jsou následně panely zobrazitelné za použití rozbalovací nabídky

Oproti tomu panely zobrazující jednotlivé zdroje díla jsou zobrazeny tak, aby bylo možné zobrazit všechny zdroje zároveň. Toho bude docíleno především díky umístění nástrojových panelů formou karet, čímž vznikl ve zbývající části rozložení dostatečný prostor. Ačkoliv v současné době nemají díla zároveň všechny tři způsoby zobrazení zdrojů, návrh bude pro tuto situaci do budoucna již přizpůsoben.



Obrázek 3.2: Finální podoba návrhu desktopové verze

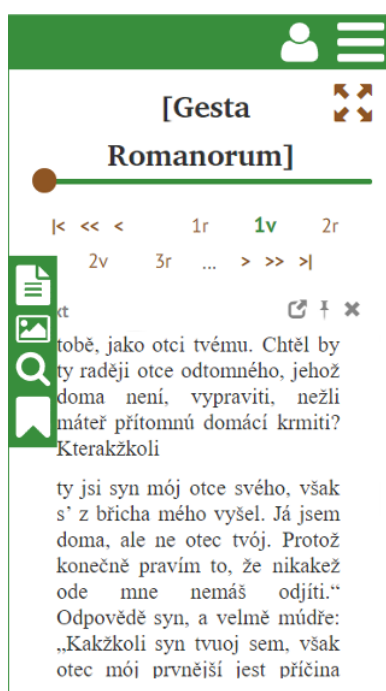
3.2.2 Varianta pro mobilní zařízení

Návrh varianty pro mobilní zařízení z velké části vychází z desktopové podoby s tím, že rozložení jednotlivých prvků bude muset být uzpůsobeno omezenému prostoru, způsobeného menšími rozměry obrazovek takových zařízení.

Hlavička díla s ovládacími prvky

Prvním problémem, který vzhledem k rozměrům dostupné plochy nastane je umístění ikon otevírajících jednotlivé nástrojové panely, stejně tak jako umístění ikon pro zobrazení jednotlivých zdrojů daného díla. Možným řešením se zdá být seskupení nástrojových ikon do postranního panelu s fixní pozicí vzhledem k obrazovce. (viz obrázek 3.3) Takový prvek by však velmi zasahoval do samotné čtečky, proto ve finálním návrhu bude tento panel skrytý, zobrazitelný ikonou při kraji obrazovky pouze v případě potřeby.

Takovýto návrh řešení (viz obrázek 3.3) ukazuje, že jednotlivé prvky stránkování budou díky omezenému prostoru zalamovány, čímž celé stránkování nebude působit jako jeden celistvý prvek. Rovněž bude třeba zvážit, zda posuvný prvek pro orientaci v rámci díla není příliš nepřesným způsobem ovládání na dotykové ploše, neboť při větším počtu stránek v rámci díla by mohlo být pro uživatele velmi obtížné přejít na konkrétní stránku. Nejvhodnějším řešením se jeví odstranění stránkování i posuvníku a nahrazení těchto prvků polem pro zadání stránky s tím, že toto pole nabídne uživateli automatické doplňování jak pro název stránky, tak pro její pořadí v rámci jednotlivého díla. Náповěda pro pořadí stránky v rámci díla bude přidána z důvodu různých způsobů číslování, kdy číslo stránky nemusí nezbytně odpovídat jejímu skutečnému pořadí. Tento prvek doplní informace pro uživatele,



Obrázek 3.3: Návrh řešení umístění nástrojových ikon

na jaké stránce se aktuálně nachází.

Odstraněním prvků stránkování vznikne nový prostor, kde bude možné umístit ikony pro zobrazování jednotlivých typů zdrojů spolu s ikonami pro úpravu zobrazení v rámci zdroje. Finální návrh zobrazuje šest ikon vedle sebe, což je předpokládaný krajní případ v současné době, kdy dílo má zdroje jak textové, obrazové tak i audio. Takový počet ikon může být následně při realizaci poněkud problematicky ovladatelný na dotykové obrazovce, proto je zde počítáno i s případným rozdělením těchto ikon na dvě skupiny, které by byly zalomeny pod sebou.

Dalším problémem k vyřešení je umístění informací o díle a ediční poznámky daného díla. Způsob řešení z desktopové varianty návrhu se zde zdál nevhodným, jelikož při velmi dlouhé ediční poznámce by byl zbytek čtečky odsunut velmi nízkou. To by v praxi mohlo znamenat, že uživatel dočte tuto poznámku, vrátí se zpět nahoru, zavřít poznámku a až pak přejít k zobrazení samotného díla. Způsob zobrazení ediční poznámky i detailů o díle bude tedy vyřešen formou oken otevíraných přes obrazovku, přičemž ikony k zobrazení těchto oken byly umístěny na úroveň názvu literárního díla.



Obrázek 3.4: Finální návrh čtečky pro mobilní zařízení

■ Využití layout manageru v rámci mobilního zařízení

Poslední otázkou, kterou bylo nutné si položit při návrhu mobilního rozložení čtečky literárních děl, byla ohledně využití zvoleného layout manageru. Jelikož manipulace s jednotlivými panely na dotykové obrazovce byla obtížná a zároveň mobilní zařízení nenabízí dostatečný prostor aby uživatel mohl mít například dva panely v řadě vedle sebe, bude tato funkcionalita vypnuta s tím, že na mobilních zařízeních budou tyto možnosti uživateli odepřeny.

Finální návrh (viz obrázek 3.4) pak pracuje tak, že při otevření nového panelu bude aktuální panel zavřen a nahrazen novým.

■ 3.3 Návrh architektury pro kód čtečky

Před samotnou implementací bylo třeba promyslet, jakým způsobem strukturovat samotný kód.

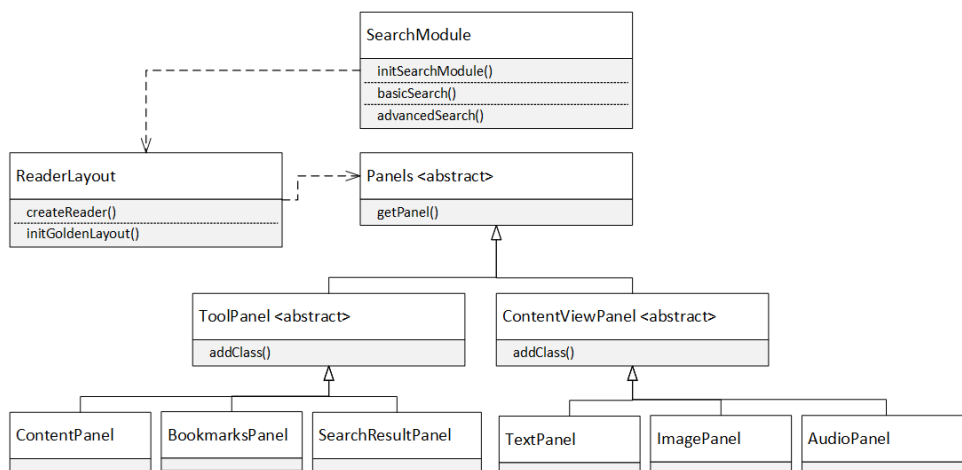
Základní rozdělení kódu bude rozloženo do tří tříd. První třída s názvem ReaderLayout obsáhne kód, který vytvoří celé prostředí čtečky, tedy hlavičku s ovládacími prvky čtečky a zároveň i inicializaci zvoleného layout manageru. Rovněž se v této třídě budou nacházet proměnné, konkrétně pole, která budou obsahovat již vytvořené panely aby bylo možné s nimi na jednom místě

pracovat, pokud by nastala situace při které bude třeba provést jistou funkci nad všemi panely.

Panely tvoří stromovou strukturu, přičemž listy této struktury zastupují jednotlivé konkrétní typy panelů. Abstraktní třída Panels tedy bude obsahovat pouze metody společné pro všechny panely, přičemž některé tyto metody zde nebudou mít svou vlastní implementaci, která se liší podle typu panelu. Tuto třídu následně rozšíří dvě další třídy, rovněž abstraktní. Tyto třídy, ToolPanel a ContentViewPanel, pouze přiřadí vytvořeným panelům parametr *class* aby bylo možné následně všem nástrojovým panelům přiřadit vlastní styly stejně tak jako všem panelům zobrazující zdroje díla. Od těchto dvou tříd následně budou dědit již konkrétní panely, kde se v každé třídě vytváří obsah pro daný typ panelu.

Poslední třída bude mít za úkol vytvořit a obstarávat vyhledávání. Tato třída vytvoří vyhledávací pole pro čtečku, přičemž následně i bude obstarávat vyhledávání napříč celým dílem.

Přibližný návrh struktury kódu je zobrazen na obrázku 3.5



Obrázek 3.5: Návrh struktury kódu

Kapitola 4

Realizace

Po vytvoření návrhů pro desktopovou i mobilní verzi čtečky literárních děl bylo nutné tyto návrhy realizovat. Implementace podle těchto návrhů probíhala, nicméně v průběhu byly tyto návrhy lehce upraveny.

4.1 Vyskytnuté problémy v průběhu implementace

V průběhu implementace čtečky zdrojů se vyskytl problém neexistující metody pro stažení celé audio knihy. Nástroj k prohlížení audio knih v původní verzi čtečky tuto možnost sice nabízel, nicméně přístup k řešení tohoto úkolu byl v dřívější podobě pro novou podobu čtečky nevyhovující. Bylo tedy nutné vytvořit metodu na úrovni backendu, která vrátila potřebné údaje ke stažení balíčku všech stop vybrané audio knihy,

Jedním z požadavků bylo umožnit uživateli otevření libovolného panelu do samostatného okna. Tuto funkcionalitu ve výchozí konfiguraci nabízí zvolený layout manager. Jelikož však na události v jednom panelu musí reagovat i zbývající panely, bylo nutné využít `EventEmitter`, objekt nabízený pomocí `GoldenLayout`. Využitím tohoto objektu je umožněno vyvolávat a naslouchat událostem napříč více panely i v případě jejich otevření do samostatného okna prohlížeče. [9]

S využitím responzivního designu stránky bylo umožněno realizovanému návrhu reagovat a přizpůsobit se různým rozměrům obrazovky. Toho bylo docíleno použitím relativních jednotek při určování rozměrů jednotlivých prvků a s využitím `@media` syntaxe pro odlišení stylů desktopové a mobilní verze. [14]

Dále bylo třeba připravit na změnu rozměrů i JavaScriptový kód, jelikož konfigurace `GoldenLayout` se na mobilním zařízení od desktopové verze liší. Původním záměrem bylo implementace dvou podob třídy `ReaderLayout`, jedna

pro desktopovou a druhá pro mobilní verzi čtečky, přičemž příslušná třída by se vybírala na základě rozměrů stránky při načítání. Toto řešení se ukázalo být jako nevyhovující, jelikož v případě zmenšení okna prohlížeče by bylo přepínání konfigurací příliš náročné. Vhodným řešením se ukázala implementace dvou konfigurací GoldenLayout, přičemž při překročení nastavené hranice rozměrů okna prohlížeče dojde k opakovanému vytvoření layout manageru s příslušnou konfigurací. Tím bylo umožněno čtečce dostatečně reagovat na změnu rozměrů obrazovky.

4.2 Výsledná architektura kódu

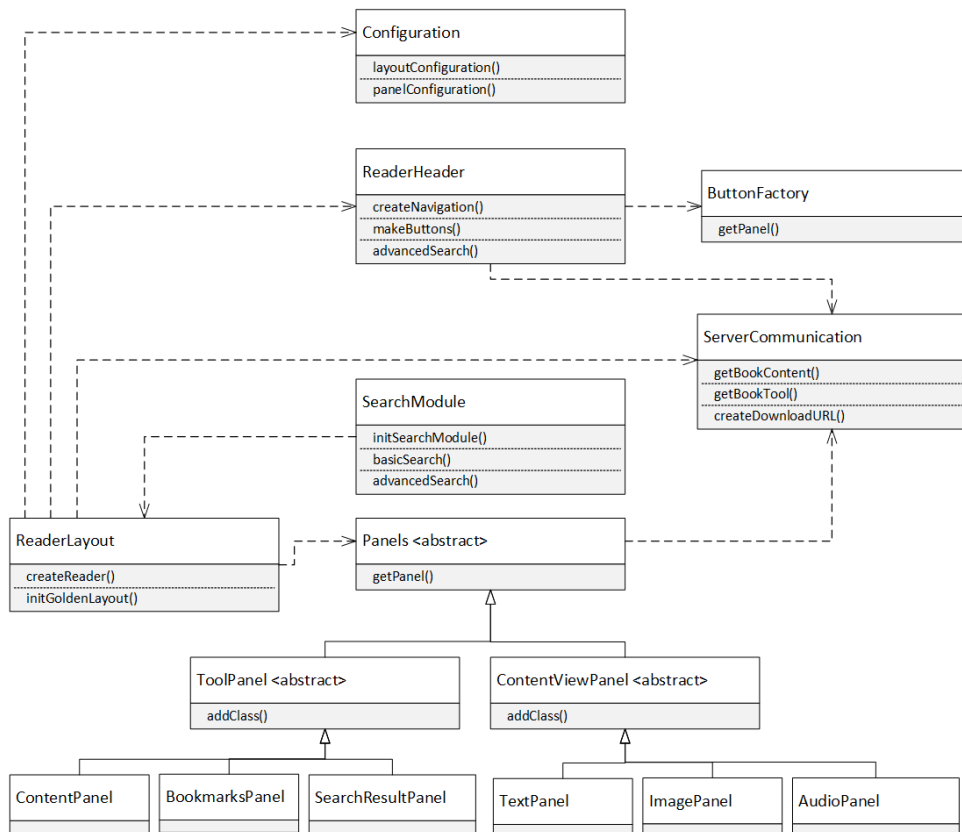
Před samotnou implementací proběhl návrh, jakým způsobem strukturovat kód do tříd. (viz kapitola 3.3) Ačkoliv tato architektura zůstala zachována, při samotné implementaci byla třída ReaderLayout rozdělena do několika dalších tříd pro větší přehlednost kódu.

Do samostatné třídy s názvem ServerCommunication byly přesunuty části kódu, obsluhující komunikaci se serverem. V případě potřeby dat ze serveru jsou volány příslušné metody této třídy, které vracejí AJAX request, který je následně v samotném místě volání zpracován. Do této třídy byly rovněž přidány metody sestavující URL adresu pro stažení audio stopy. Tato metoda byla vytvořena z důvodu opakovaného využití vytvořených adres, které se lišily pouze v jednotlivých parametrech id stopy a typ audio souboru.

Oproti původnímu návrhu vznikla třída ReaderHeader, která je inicializována ve třídě ReaderLayout. Tato třída má za úkol vytvořit hlavičku čtečky s prvky, které slouží k práci s literárním dílem. Při vytvoření této třídy následně vznikla další, s názvem ButtonFactory, jelikož tvorba jednotlivých tlačítek a odkazů v hlavičce byla velmi často opakována pouze s některými drobnými změnami. Tímto krokem se zbytečnému opakování zamezilo.

Konfigurace jednotlivých panelů, stejně tak jako počáteční konfigurace celého GoldenLayout byla vytvořena v samostatné třídě LayoutConfiguration. Tím bylo zjednodušeno vrácení příslušné výchozí konfigurace pro GoldenLayout na základě toho, zda se jedná o mobilní zařízení nebo desktop.

Výsledná struktura kódu je znázorněna na obrázku 4.1.



Obrázek 4.1: Výsledná struktura kódu

Kapitola 5

Testování

Ačkoliv během implementace byla funkčnost čtečky literárních děl průběžně testována, pro zlepšení uživatelského rozhraní proběhlo testování s uživateli.

5.1 Návrh podoby testu

Testování uživatelského rozhraní si klade za cíl odhalit nedostatky v návrhu uživatelského rozhraní. Během tohoto testování plní uživatel zadané úkoly, které mají za úkol simulovat běžné použití aplikace. Úkony takového účastníka jsou během testování sledovány moderátorem testování.

5.1.1 Pre-test dotazník

Před samotným testováním účastník vyplní takzvaný pre-test dotazník. Cílem tohoto dotazníku je zjistit informace o tom, zda má daný uživatel zkušenosti s dřívější podobou čtečky literárních děl na portálu Vokabulář webový nebo s jiným nástrojem sloužícím k zobrazení elektronických knih. Otázka ohledně zkušeností s jinými nástroji pro zobrazení elektronických knih je položena pro případ, že by účastníkovi chyběla nějaká funkce a existující nástroje by mohli sloužit pro porovnání. Tento dotazník probíhá v podobě vyplnění formuláře vytvořeného pomocí Google Formuláře. (Příloha C.1, výsledky pre-test dotazníku se nachází v příloze C.2)

5.1.2 Testování uživatelského rozhraní

Testování uživatelského rozhraní probíhá tak, že účastník dostává seznam úkolů, které má splnit. (Příloha D) Při plnění těchto úkolů uživatel své kroky

nahlas komentuje, říká co je v daný moment jeho cílem a jak toho cíle chce dosáhnout. Stejně tak má zmínit, pokud mu z nějakého důvodu nejde cíl splnit nebo neví jak jej dosáhnout. Během tohoto testování je pořizován hlasový záznam pro následnou analýzu průběhu, který je nahráván s využitím aplikace Hlasový záznam dostupné v operačním systému Windows 10.

Při samotném testování je přítomen moderátor testování, nicméně nesnaží se do průběhu zasahovat. Může poradit v případě že se uživatel evidentně odkloní od zadaného cíle nebo pokud při plnění zadaného cíle z nějakého důvodu delší dobu neví, jakým způsobem pokračovat.

Plnění zadaných úkolů probíhá na zapůjčeném notebooku a zapůjčeném telefonu, čímž bylo zajištěno že každý uživatel měl stejné podmínky při testování.

■ 5.1.3 Post-test dotazník

Na závěr testování je každému uživateli předložen dotazník, ve kterém zhodnotil průběh testování. Cílem tohoto dotazníku je zjistit, do jaké míry se jeví uživatelům ovládání intuitivní. Rovněž v dotazníku měli zhodnotit, zda nastala situace, kdy nebylo zřejmé, jakým způsobem pokračovat v plnění úkolu. Pro uživatele se zkušenostmi s předchozí podobou čtečky Vokabuláře webového pak je připravena otázka navíc, do jaké míry se uživatelům jeví provedené změny přínosné. Stejně jako Pre-test dotazník i tento dotazník probíhá vyplněním formuláře vytvořeného službou Google Formuláře. (Otázky v dotazníku jsou v příloze E.1)

■ 5.2 Testování s uživateli

Výběr uživatelů pro testování probíhal mezi zaměstnanci ÚJČ pro pokrytí badatelské skupiny uživatelů, kterou v testování zastupují participanti čtyři až šest. Rovněž byli vybráni i uživatelé mimo ÚJČ pro pokrytí komunitní skupiny, do které spadají první tři uživatelé.

■ 5.2.1 Participant 1

První participant na testování v dotazníku před testem uvedl, že má již zkušenosti s testováním uživatelského rozhraní z jeho zaměstnání. Dá se tedy předpokládat, že díky těmto zkušenostem se u participanta nedostavila nervozita, která by mohla v průběhu testování způsobit vyšší míru zadržování nad jednotlivými úkoly. Tento participant však nemá zkušenosti s předchozí

podobou čtečky literárních děl Vokabuláře webového, zkušenosti má pouze s čtením elektronických knih s využitím služby Google Books.

V dotazníku po provedení testování uživatel uvedl, že při hledání ediční poznámky a detailů o díle (v rámci úkolu číslo 2 v příloze D) bylo pro něj problematické zjistit, kterým tlačítkem tyto informace zobrazit, jelikož hledaná ikona postrádá popisek. Kromě tohoto problému hodnotil novou podobu čtečky vesměs kladně, spokojen byl především s jednoduchostí změny rozložení panelů v rámci čtečky. V průběhu samotného testování se uživatel v některých případech musel zamyslet více nad dalším krokem, což bylo způsobeno pravděpodobně tím, že tento nástroj jakožto i některé použité výrazy pro něj byly nové.

■ 5.2.2 Participant 2

Stejně jako první participant, i tento měl již zkušenosti s testováním uživatelského rozhraní. Z dotazníku vyplněného před testem vyplývá, že ačkoliv se samotným Vokabulářem webovým neměl zkušenosti, s prohlížením elektronických děl se setkal jak v textové podobě, tak v audio podobě.

Po testování uživatel neuvedl žádné nedostatky v dotazníku, nicméně v průběhu testování se setkal s podle něj nevhodným formátováním čísel stránek v textu, jelikož číslo stránky je více odsazeno od stránky, ke které patří, než od té následující. Tento fakt vedl v průběhu testování k mírnému zmatení. Další odhalený nedostatek byl odhalen v souvislosti s úkolem 2, kdy bylo cílem zobrazit ediční poznámku a informace o díle. Ačkoliv se u tohoto úkolu příliš nepozastavil, k odhalení vhodné ikony vedoucí k zobrazení informací o díle došel spíše vylučovací metodou než intuitivním určením. Rovněž participant zmínil i možné řešení volbou vhodnější ikony.

■ 5.2.3 Participant 3

Třetí participant uvedl, že se s Vokabulářem webovým nemá zkušenosti, nicméně setkal se již s jinými portály podobného zaměření, konkrétněji uvedl nástroje ČNK a francouzský portál Frantext. Díky těmto zkušenostem by mohla být práce s Vokabulářem pro tohoto uživatele intuitivnější.

V celkovém dojmu měl uživatel vesměs kladný pocit z testované čtečky, nicméně průchod jednotlivými úkoly nebyl vždy zcela intuitivní a bezproblémový. Při vytváření záložek se participant setkal s nemožností vytvořit záložku bez přiřazeného štítku, jelikož povinnost alespoň jednoho štítku k záložce není nijak naznačena. Dalším problémem bylo složité hledání detailních informací, jelikož samotná ikona šipky, která pro zobrazení těchto informací slouží, je bez jakéhokoliv popisku a uživatel tedy nevidí na první pohled,

co pod ní hledat. Pro tohoto uživatele bylo problematické přesunout panely na jiné místo v rámci rozložení. Ačkoliv první postup uživatele byl správný, došlo k nechtěnému posunu panelu na jiné místo, čímž byl uživatel od tohoto postupu nadále odrazen.

■ 5.2.4 Participant 4

Tento participant byl prvním z ÚJČ, díky čemuž měl zkušenosti s původní čtečkou literárních děl Vokabuláře webového, kde rovněž používal i nástroj pro přehrávání audio knih.

Při testování uživateli nebylo zřejmé, jakou funkci plní štítek při vytváření záložky a nevěděl tedy, proč musí mít každá záložka svůj přiřazený štítek. Funkce těchto štítků musela být uživateli v průběhu testování vysvětlena. Participant se rovněž setkal s problémem při plnění úkolu číslo dva, kdy nevěděl kde hledat detailní informace o díle. Kromě těchto odhalených nedostatků byly změny přinesené do čtečky hodnoceny tímto uživatelem převážně pozitivně.

■ 5.2.5 Participant 5

Participant číslo pět měl rovněž zkušenosti s původní čtečkou Vokabuláře webového. Kromě Vokabuláře pak uvedl, že ke čtení elektronických knih používal hardwarovou čtečku a v případě počítače využívá pouze klasický prohlížeč pdf souborů. Rovněž i k přehrávání audio knih nepoužívá žádný specializovaný nástroj, ale pouze běžný přehrávač audio souborů.

Uživatel se při testování setkal se stejným problémem jako předcházející participant v podobě zmatení při vytváření záložek, kdy při otevření dialogu tvorby záložek je nutné přiřadit záložce štítek, jelikož v tomto okně dále slovo záložky není zmíněno. I při plnění úkolu číslo dva byl odhalen stejný problém, jaký měli předchozí uživatelé. K zobrazení detailních informací uživatel využil původní způsob přes seznam děl, jelikož ikony pro zobrazení v rámci čtečky si nevšiml, což s nejvyšší pravděpodobností způsobila nejasná ikona v kombinaci s absencí popisku. V průběhu testování uživatel několikrát nevěděl, jakým způsobem pokračovat v plnění daného úkolu, což bylo podle jeho slov způsobeno velkým množstvím informací na stránce. Pravděpodobně v kombinaci s mírnou nervozitou pak uživatel některé popisky přehlížel a díky tomu nevěděl jak dál. Na konci testování uživatel hodnotil pozitivně možnost a způsob přizpůsobení rozložení jednotlivých panelů. I přes občasné problémy se čtečkou uživatel hodnotil novou podobu pozitivně.

■ 5.2.6 Participant 6

Poslední participant před testováním uvedl, že má zkušenosti jak s předchozí podobou čtečky Vokabuláře webového, tak s nástroji pro čtení elektronických knih. Zde uvedl jako příklad nástroj Wooky, který běžně používá. K přehrávání audio knih používá aplikaci Librevox na zařízení s operačním systémem Android.

V průběhu testování participant postupoval s jistotou, ve většině případů s ovládním čtečky neměl problémy. Při plnění úkolu číslo dva se však setkal s problémem, kdy nevěděl pod jakou ikonou zobrazit detailní informace. V případě úkolu číslo pět pak nevěděl, jakým způsobem přesunout panel, jelikož se ho snažil uchopit na libovolném místě v jeho hlavičce, jelikož je na takovéto ovládní zvyknutý z průzkumníka ve Windows. Na závěr testování uživatel uvedl, že celkově se čtečka ovládá intuitivně a pohodlně, přičemž vyzdvihl intuitivnost ovládní mobilní podoby portálu.

■ 5.3 Vyhodnocení výsledků testování

V průběhu testování byly odhaleny opakující se problémy. Každý z uživatelů nemohl najít v nové čtečce detailní informace o otevřeném díle. To je způsobeno nejasnou ikonou šipky dolů bez popisku. Řešením tedy může být jasnější ikona nebo doplnění popisku ke stávající ikoně, případně kombinace obojího.

Dalším nedostatkem, který by bylo třeba opravit je lepší rozlišení mezi slovy záložka a štítek. Při tvorbě záložky je uživateli nabídnuto dialogové okno, kde musí vytvářené záložce přiřadit štítek. Tento požadavek pro přiřazení nikde není uveden jako povinný, což způsobilo že někteří uživatelé nemohli při testování vytvořit záložku, jelikož jí nepřiradili štítek. Rovněž při otevření okna pro tvorbu záložek již není nikde uvedeno slovo *záložka*, ale je využíváno slovo *položka*. Řešením těchto problémů by mohla být úprava názvosloví.

Jeden z participantů odhalil chybějící implementovaný tisk. Ačkoliv se jedná o ojedinělý nález, tisk díla byl jedním z požadavků a proto lze tento problém považovat za závažný a připomínku uživatele za relevantní.

Odhalené problémy uživatelé považovali spíše jako drobné nedostatky a přinesené změny i způsob ovládní hodnotili pozitivně. Většině uživatelů přišlo ovládní intuitivní a to včetně nově zavedené funkcionality v podobě změny rozmístění panelů. Při ovládní mobilní podoby čtečky pak nikdo z uživatelů nenarazil na nějaký zásadní problém. (uvedené odpovědi v post-test dotazníku jsou dostupné v příloze E.2)

Kapitola 6

Závěr

Cílem práce bylo navrhnout a vytvořit novou podobu čtečky literárních zdrojů pro portál Vokabulář webový na základě analýzy potřeb koncových uživatelů, která měla být implementována v souladu s používanými technologiemi tohoto portálu.

Na základě analýzy potřeb uživatelů portálu Vokabulář webový byl vytvořen návrh. Při vytváření návrhu bylo cílem vytvořit uživatelsky přívětivé prostředí, které splňuje požadavky získané v analýze. Bylo však potřeba vytvořit dvě podoby rozhraní, pro desktop podobu čtečky a pro mobilní zařízení, kdy je třeba počítat s menšími rozměry obrazovky.

Při analýze se nepodařilo nalézt layout manager ovladatelný na dotykové obrazovce. S tímto problémem bylo počítáno již při návrhu, kdy nebylo s možností přizpůsobení rozložení příliš počítáno. Následně při implementaci byl tento problém vyřešen změnou konfigurace v případě mobilních zařízení, čímž byly problematické prvky layout manageru odstraněny.

Navržené řešení čtečky bylo implementováno v souladu s technologiemi používanými Vokabulářem webovým a s využitím volně dostupné knihovny GoldenLayout, která byla vybrána při analýze dostupných knihoven jako nejvhodnější. Čtečka je schopna pracovat se všemi v současné době používanými zdroji díla, tedy textovým, obrazovým i audio zdrojem. Zároveň díky responzivnímu designu realizovaných návrhů je čtečka použitelná jak na počítači tak na mobilních zařízeních.

Výsledná čtečka prošla úspěšně testováním, kdy uživatelé hodnotili způsob ovládání i uživatelské rozhraní převážně pozitivně. Zároveň pak při testování odhalili některé drobné nedostatky, které se však podařilo na základě zpětné vazby uživatelů odstranit.

Do budoucna se počítá s několika způsoby rozšíření. Čtečka je připravena na případné přidávání dalších způsobů zobrazení zdrojů literárních děl. Toho

by mohlo být docíleno přidáním příslušného panelu do layoutu, což by v budoucnu nemělo být příliš problematické. V současné době design počítá pouze s mobilními zařízeními a desktopem. Do budoucna je však možné jednoduše přidat i další způsoby rozložení stránky pro další zařízení či rozměry okna.



Příloha A

Literatura

- [1] Dock Spawn. <http://www.dockspawn.com/>, 2012. [Online; cit. 10.05.2018].
- [2] TypeScript - JavaScript that scales. <http://www.typescriptlang.org/index.html>, 2014. [Online; cit. 18.04.2018].
- [3] PhosphorJS: About. <https://phosphorjs.github.io/about.html>, 2015. [Online; cit. 15.05.2018].
- [4] Ústav pro jazyk český AV ČR, v. v. i. Výzkumná infrastruktura pro diachronní bohemistiku . <http://vokabular.ujc.cas.cz/informace.aspx?t=ridics&o=ovokabulari>, 2016. [Online; cit. 17.05.2018].
- [5] C. Coblert. phosphor/LICENSE at master · phosphorjs/phosphor · GitHub. <https://github.com/phosphorjs/phosphor/blob/master/LICENSE>, 2017. [Online; cit. 14.05.2018].
- [6] S. Fenton. *Pro TypeScript: Application-Scale JavaScript Development*. The expert's voice in TypeScript. Apress, 2014.
- [7] J. Galloway, P. Haack, B. Wilson, K. Allen, and S. Hanselman. *Professional ASP.NET MVC 4*. ITPro collection. Wiley, 2012.
- [8] W. Hempel. GoldenLayout - a multi-window javascript layout manager for webapps. <http://golden-layout.com/faq/>. [Online; cit. 09.05.2018].
- [9] W. Hempel. GoldenLayout - a multi-window javascript layout manager for webapps. <http://golden-layout.com/docs/>. [Online; cit. 28.03.2018].
- [10] J. Houde. Web Cabin Docker Demo. <http://docker.webcabin.org/>, 2014. [Online; cit. 10.05.2018].



Příloha B

Seznam použitých zkratek

- UJČ Ústav pro jazyk český AV ČR, v. v. i.
- HTML HyperText Markup Language, značkovací jazyk sloužící pro tvorbu webových stránek
- AJAX Asynchronous JavaScript and XML, technologie měnící obsah webových stránek za pomoci asynchronního zpracování
- MVC Model-View-Controller, softwarová architektura oddělující datový model aplikace, uživatelské rozhraní a řídicí logiku
- CSS Cascading Style Sheets, jazyk pro popis způsobu zobrazení elementů napsaných v jazyce HTML
- Licence MIT Licence s původem na Massachusetts Institute of Technology, umožňující volné šíření software s podmínkou, že text licence je dodáván spolu s tímto software
- IDE Integrated Development Environment, vývojové prostředí
- Licence BSD Berkeley Software Distribution, licence umožňující volné šíření licencovaného obsahu, vyžaduje uvedení autora a informace o licenci spolu s upozorněním na zřeknutí se odpovědnosti za dílo
- URL Uniform Resource Locator, řetězec znaků s definovanou strukturou, který slouží k specifikaci umístění zdrojů na internetu
- ČNK Český národní korpus

Příloha C

Pre-test dotazník

C.1 Otázky dotazníku

1. Účastnil/a jste se někdy testování uživatelského rozhraní?
2. Máte zkušenosti s používáním čtečky literárních děl Vokabuláře webového?
3. Setkal/a jste se někdy s podobným nástrojem, sloužícím k prohlížení knih/literárních děl v elektronické podobě? Jakým?
4. Používal/a jste někdy nějaký nástroj sloužící k přehrávání audio knih? Jaký?

C.2 Odpovědi na dotazník

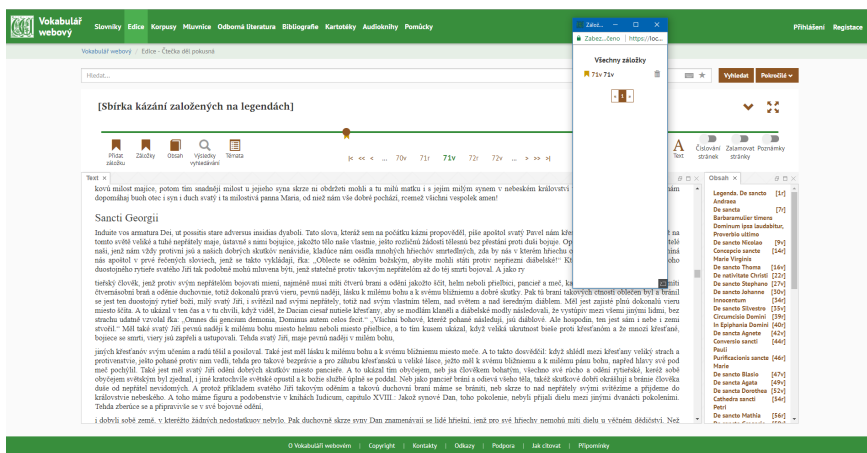
Participant	Účastnil/a jste se někdy testování uživatelského rozhraní?	Máte zkušenosti s používáním čtečky literárních děl Vokabuláře webového?	Setkal/a jste se někdy s podobným nástrojem, sloužícím k prohlížení knih/literárních děl v elektronické podobě?	Pokud ano, jakým?	Používal/a jste někdy nějaký nástroj sloužící k přehrávání audioknih?	Pokud ano, jaký?
1	Ano	Ne	Ano	Google Books	Ne	
2	Ano	Ne	Ano	ibooks, ebook,	Ano	audible
3	Ne	Ne	Ano	ČNK, Frantext	Ne	
4	Ano	Ano	Ano	hardwarová čtečka e-knih	Ano	Vokabulář webový
5	Ano	Ano	Ano	hardwarová čtečka, prohlížení pdf v mobilu	Ano	klasický audiopřehrávač
6	Ano	Ano	Ano	Wooky, DarkHorse	Ano	Librivox (aplikace na Android)

Příloha D

Testované úkoly pro participandy testování

Desktopová varianta čtečky

1. Vytvořte záložku na začátku kapitoly *Sancti Georgii* v díle *Sbírka kázání založených na legendách*
2. Zobrazte detailní informace o tomto díle
3. Na stránce *130r* naleznete výskyt jména *Vavřinec*
4. Pomocí nástrojového panelu se vraťte na místo, kde byla vytvořena dříve záložka
5. Upravte rozložení jednotlivých panelů podobně jako na obrázku níže



6. V edici *Text pro workshop* otevřete obrazový zdroj tohoto díla
7. Textový zdroj pro toto dílo zavřete

8. Otevřete audio knihu *Frantovy práva*
9. Stáhněte kapitolu s názvem *O ženách*
10. Přehrajte kapitolu s číslem 14

Mobilní varianta čtečky

1. Otevřete edici s názvem *Chirurgie Lékařství*
2. Na straně 44r si na základě poznámky k textu zjistěte původní tvar slova *lístem*
3. Přejděte na kapitolu *Naučenie*, odkud pokračujte ve čtení dalších stran knihy
4. Zobrazte patičku *Vokabuláře webového*
5. V režimu celého okna přejděte na stou stranu(v pořadí) v knize

Příloha E

Post-test dotazník

E.1 Otázky dotazníku

1. Zdál se průchod jednotlivými úkoly intuitivní?
2. Nastala při průchodu úkoly situace, kdy jste nevěděl jak dál? Pokud ano, o jakou situaci se jednalo?
3. Bylo ovládání čtečky knih pohodlné a bezproblémové? Pokud ne, o jaký prvek čtečky se jednalo?
4. Chyběla Vám v čtečce nějaká funkce?
5. Byl/a jste spokojen/a s možností přizpůsobení rozložení panelů čtečky?
6. Pokládáte provedené změny za přínosné oproti původní čtečce?

E.2 Odpovědi na dotazník

Participant	Zdál se průchod jednotlivými úkoly intuitivní?	Nastala při průchodu úkoly situace, kdy jste nevěděli/a jak dále postupovat?	Pokud ano, o jakou situaci se jednalo?	Bylo ovládání čtečky knih pohodlné a bezproblémové?	Pokud ne, o jaký prvek čtečky se jednalo?	Chyběla Vám v čtečce nějaká funkce?	Pokud ano, jaká?	Byl/a jste spokojen/a s možností přizpůsobení rozložení panelů čtečky?	Pokládáte provedené změny za přínosné oproti původní čtečce? (nevypíňujte, pokud s předchozí podobou čtečky nemáte zkušenost)
1	Spiše ano	Ano	Během úkolu č. 2 jsem nevěděl, kde zobrazit detailní informace.	Rozhodně ano		Ne		Rozhodně ano	
2	Spiše ano	Ne		Spiše ano		Ne		Nevím	
3	Spiše ano	Ano	přesouvání panelů, nevěděla jsem, co znamená slovo patička, nalezení detailních informací taky nebylo pro mě hned zřejmé	Spiše ano		Ne		Spiše ano	
4	Spiše ano	Ano	nastal problém s rozlišením štítků a záložek; při úkolu číslo 2 nebylo zřejmé, kde naléznout informace o díle	Spiše ano	problém s rozlišením co je obsah a co témata	Ano	Tisk díla	Rozhodně ano	
5	Spiše ano	Ano	celkově se na stránce nacházelo příliš mnoho nových informací při prvním seznamování	Rozhodně ano		Ne		Spiše ano	Nevím
6	Spiše ano	Ano	při přesunu obsahového panelu na druhou stranu čtečky	Spiše ano	více uživatelsky přívětivá byla mobilní podoba čtečky než desktopová	Ne		Rozhodně ano	Rozhodně ano