



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

OSOBNÍ STARTOVACÍ STRÁNKA

PERSONAL STARTING PAGE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ KADLETZ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZŐKE, Ph.D.

BRNO 2014

Abstrakt

Tato práce řeší návrh a realizaci webové aplikace s názvem Osobní startovací stránka. Aplikace umožňuje vytvoření domovské stránky s odkazy a RSS zdroji řazenými do kategorií. Práce klade důraz na responzivní design, přístupnost a ovladatelnost pomocí mobilních telefonů, tabletů a počítačů. Systém je postaven na PHP 5 s použitím Nette frameworku, knihovny jQuery a Bootstrap frameworku. Výsledkem práce je funkční webová aplikace dostupná široké veřejnosti na adrese www.jdinaweb.cz.

Abstract

This thesis is about design and implementation of a web application called Personal Starting Page. The application allows users to create a homepage with links and RSS resources that are systematically classified into categories. The thesis also focuses on responsive design, accessibility and manageability through mobile phones, tablets and computers. The application uses technologies such as PHP 5, Nette framework, jQuery and Bootstrap framework. The result of this thesis is a fully functional web application that is available on the www.jdinaweb.cz website.

Klíčová slova

Osobní startovací stránka, webová aplikace, odkazy, záložky, RSS, PHP, Nette, jQuery

Keywords

Personal Starting Page, web application, links, bookmarks, RSS, PHP, Nette, jQuery

Citace

Lukáš Kadletz: Osobní startovací stránka, bakalářská práce, Brno, FIT VUT v Brně, 2014

Osobní startovací stránka

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szókeho, Ph.D.

.....

Lukáš Kadletz
21. května 2014

Poděkování

Rád bych na tomto místě poděkoval vedoucímu mé bakalářské práce, panu Ing. Igorovi Szókemu, Ph.D., za metodické vedení, osobní přístup a nespočet cenných rad, kterých jsem při tvorbě této práce využil.

© Lukáš Kadletz, 2014.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	Použité technologie	4
2.1	Webové technologie	4
2.2	Serverová část	4
2.2.1	Skriptovací jazyk PHP	4
2.2.2	Nette framework	5
2.2.3	Relační databáze MySQL	6
2.3	Klientská část	6
2.3.1	Značkový jazyk HTML	7
2.3.2	Kaskádové styly CSS	7
2.3.3	Bootstrap framework	8
2.3.4	JavaScript a knihovna jQuery	8
3	Analýza a koncepční návrh aplikace	9
3.1	Rešerše existujících aplikací	9
3.1.1	Start.io	9
3.1.2	3x3links.com	10
3.1.3	Only2clicks.com	11
3.1.4	Startme.com	11
3.1.5	Zhodnocení existujících aplikací	13
3.2	Koncepční návrh aplikace	13
4	Návrh aplikace	16
4.1	Vytvoření diagramu případů užití	16
4.1.1	Návštěvník	16
4.1.2	Uživatel s vlastní URL	18
4.1.3	Přihlášený uživatel	19
4.2	Návrh Entity-Relationship diagramu	20
4.3	Grafický návrh uživatelského prostředí	21
5	Implementace	24
5.1	Vytvoření schématu relační databáze	24
5.2	Struktura aplikace	25
5.3	Implementační detaily vybraných částí aplikace	25
5.3.1	Komponenty	25
5.3.2	Podpůrné třídy	27
5.3.3	Směrování a tvorba URL	28

5.3.4	Funkcionalita Drag and Drop	29
5.3.5	Cacheování obsahu	29
5.4	Grafické uživatelské prostředí	30
6	Testování a zpětná vazba uživatelů	31
6.1	Vyhodnocení testů	31
6.2	Dotazníkové šetření	32
7	Možnosti dalšího vývoje	34
8	Závěr	36
A	Obsah přiloženého CD	38
B	Návrhové diagramy	39
B.1	Diagram případů užití	39
B.2	ER-diagram	40
B.3	Schéma relační databáze	41
C	Materiály k testování uživatelů	42
C.1	Testovací Manuál	42
C.2	Dotazník	43
C.3	Výsledky dotazníku	44
D	Zachycené snímky aplikace	47

Kapitola 1

Úvod

Webové technologie se v dnešní době začínají dostávat do stále většího počtu mobilních zařízení. Příkladem mohou být tablety, mobilní telefony nebo přenosné notebooky. Člověk chce být mobilní a mít svá data dostupná nejlépe ze všech těchto zařízení. Zaměřil jsem se na první stránku, kterou každý z nás navštíví po otevření webového prohlížeče.

Cílem této práce je navrhnout a implementovat webovou aplikaci pro jednoduchou a uživatelsky přívětivou tvorbu osobní startovací stránky. Aplikace by měla uživateli umožnit vytvoření stránky s odkazy a RSS zdroji přehledně řazenými do kategorií. Vytvořené kategorie si uživatel může setřídit dle svého uvážení pouze jednoduchým tahem myši. Jeden uživatel si takovýchto stránek může vytvořit větší množství. Dále aplikace umožní přístup k celé startovací stránce pomocí unikátní webové adresy obsahující uživatelem zvolený řetězec. Soustředím se na přizpůsobení aplikačního rozhraní tak, aby ji bylo možné ovládat se stejným pohodlím i z mobilního telefonu.

V práci bude čtenář obeznámen s použitými technologiemi, koncepčním návrhem i srovnáním s podobnými službami. Čtenáři bude dále vysvětlen návrh aplikace, včetně popisu konkrétních případů užití a vytvořeného entity-relationship diagramu. Přibližně druhá třetina práce je věnována implementačním detailům, které názorně vysvětlují strukturu a funkci aplikace. V závěru práce kladu důraz na testování aplikace a uvedu, které části systému jsem na základě výsledků upravil, přidal nebo naopak úplně odebral. Testování aplikace na uživateli je, dle mého názoru, velmi důležité především proto, že jako tvůrce mám na aplikaci mírně odlišný pohled než uživatelé. Také jsem schopen na základě zpětné vazby odhalit silné a slabé stránky celého řešení.

Kapitola 2

Použité technologie

Práce si klade za cíl především vyjít vstříc uživatelům, kteří chtějí mít svoje často navštěvované stránky po ruce. Dále umožnit spravovat tuto aplikaci odkudkoliv a skrze nejrůznější platformy. Nyní následují pasáže rozděleny na dvě hlavní části z pohledu webové aplikace a to na část *serverovou* a *klientskou*. Pozastavím se i nad webovými technologiemi a pokusím se zhodnotit jejich přínosy. Dále uvedu důvody výběru konkrétních technologií.

2.1 Webové technologie

Značná výhoda při použití webových technologií je přenositelnost skrze různé operační systémy, především ty mobilní. Všechny moderních webové prohlížeče, které jsem měl možnost použít, podporují základní standardy značkovacího jazyka HTML5 a kaskádových stylů CSS3. Patří mezi ně *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*, *Dolphin Browser* a *Maxthon*. S těmito prohlížeči lze dosáhnout, až na výjimky, shodného zobrazení funkčních prvků aplikace a udržet konzistenci v jejím ovládání.

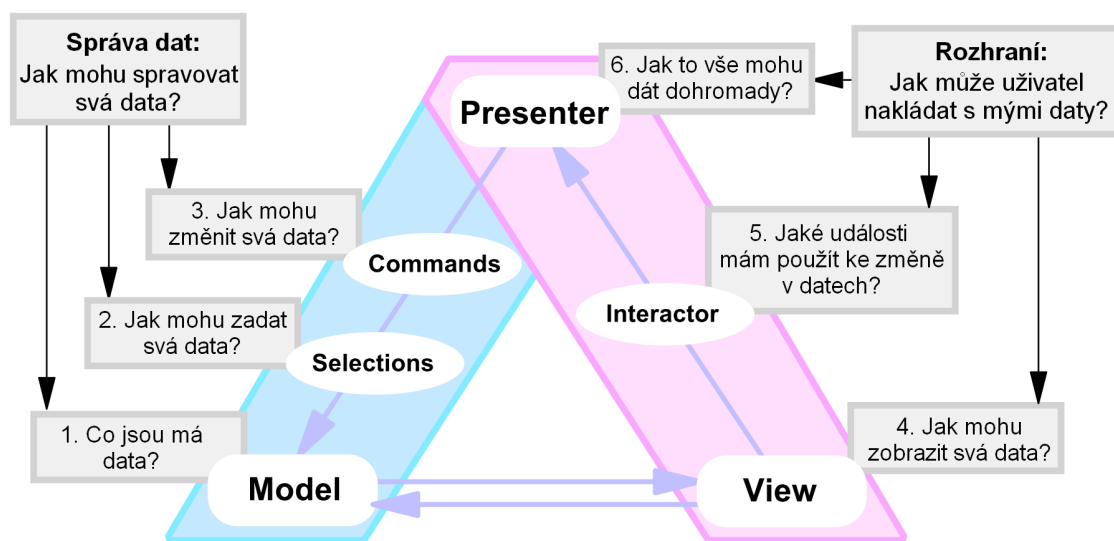
Oproti aplikacím pro stolní nebo mobilní systémy jsou webové aplikace vždy nutně závislé na připojení k internetu. Lze sice dosáhnout dočasného uložení webové stránky do krátkodobé paměti webového prohlížeče, toto ovšem není funkčnost samotné webové aplikace. Tato skutečnost se mohla ještě před několika lety zdát, jako větší překážka, ovšem dnes s rozmachem mobilního internetu to už tak jednoznačné není. Mnoho aplikací pro mobilní systémy Android a iOS jsou i tak z velké části závislé na připojení k internetu a mnohdy se bez aktivního připojení k internetu aplikace ani nespustí.

2.2 Serverová část

Serverová část obsahuje funkční logiku celé aplikace. Mimo jiné slouží pro uchování uživatelských dat, vyhodnocení a zpracování požadavků. Tato část má také velký podíl na výkonu celé aplikace. Jednotlivé podkapitoly se již věnují konkrétním technologiím na kterých je aplikace postavena.

2.2.1 Skriptovací jazyk PHP

PHP (PHP: Hypertext Preprocessor) je výkonný skriptovací programovací jazyk určený pro tvorbu dynamických webových stránek. Původně byl jazyk PHP určen pro vytváření HTML obsahu k zobrazení na klientské stanici a stále je jedním z nejlepších [12]. Jeho nesporná výhoda spočívá v rozšířenosti a podpoře poskytovatelů hostingových služeb.



Obrázek 2.1: Architektura MVP, otázky správy a reprezentace dat. (vlastní překlad [9])

Moje aplikace je od začátku vyvíjena na serveru s běžícím PHP verze 5.4. Tato verze byla uvedena 1. března 2012 a stále je oficiálně podporována (poslední aktualizace na verzi 5.4.27 byla provedena dne 3. dubna 2014¹). S ohledem na využití frameworků popsaných v následujících kapitolách nebude nijak obtížné v budoucnu přejít na vyšší verze PHP. V měsíci dubnu roku 2014 byla uvedena první betaverze PHP 5.6 (PHP 5.6.0beta1) jako Development Preview², bude však poměrně dlouho trvat, než bude PHP 5.6 oficiálně vydané a následně také podporované poskytovateli serverových hostingů a aplikací.

2.2.2 Nette framework

Jedná se o český PHP framework napsaný v PHP 5 s využitím objektově orientovaného programování (OOP). Framework je určen pro vytváření moderních dynamických webových stránek a aplikací. Jeho autorem je *David Grudl*, který Nette vyvíjí již od roku 2004. V roce 2008 jej vydal jako open source [4]. Vývoj Nette spolu s početnou komunitou zastřešuje společnost *Nette Foundation* založená taktéž v roce 2008 [4].

Nette, jako většina PHP frameworků, nabízí softwarovou architekturu MVP³, která dělí aplikaci na tři nezávislé avšak spolupracující bloky. Architekturu MVP názorně vysvětluje diagram na obrázku 2.1, kde každá část MVP modelu odpovídá na otázku k ní přiloženou. Následuje stručný popis hlavních částí MVP z pohledu Nette frameworku [9].

1. **Modelová vrstva** – zajišťuje komunikaci s databázovým systémem. Zde je také umístěno logické jádro aplikace. Příslušné metody pro změnu, získání či smazání dat ostatní vrstvy jen používají.

¹Blogová zpráva autorů PHP: <http://www.php.net/archive/2014.php#id2014-04-03-1>

²Zpráva autorů PHP: <http://php.net/archive/2014.php#id2014-04-11-1>

³MVP: Model-View-Presenter

2. **Pohled** (View) – má na starost zobrazení dat uživateli například pomocí HTML a CSS. Defnuje nejen, způsob zobrazení dat, ale jsou zde také umístěny veškeré dynamické jevy probíhající na straně klientského webového prohlížeče – JavaScript, knihovna jQuery a další technologie.
3. **Prezentační vrstva** (Presenter) – jedná se o vrstvu zajišťující komunikaci s modelem a na základě získaných výsledků nabízí uživateli správný *pohled*. Veškeré požadavky od uživatele (např.: *Zobraz mi článek číslo 7*) zpracovává právě prezentační vrstva.

Hlavními důvody pro zvolení Nette frameworku jsou především početná a aktivní česká komunita vývojářů, výborné zabezpečení, nativní ochrana proti XSS (Cross-site scripting) útokům, přímá podpora AJAX⁴ a mimo jiné také *licence BSD*. Framework lze tedy volně používat i v komerčních projektech. Aplikace je postavena na stabilní verzi *Nette 2.0.13*. K prvnímu lednu roku 2014 bylo oznámeno vydání Nette 2.1. Tuto verzi se budu snažit co nejdříve nasadit, jsou v ní mnohé novinky a podstatné změny, které přispějí k lepšímu výkonu celé aplikace.

2.2.3 Relační databáze MySQL

Pro uchování uživatelských dat a nastavení jsem využil databázový systém MySQL. Jedná se o systém typu RDBMS (Relational Database Management System) [2, str. 4]. MySQL má poměrně velkou uživatelskou základnu především z důvodu velmi liberálních licenčních podmínek. Dalším důvodem k výběru tohoto databázového systému je i fakt, že poskytovatel mého webového hostingu podporuje mimo jiné systémy právě MySQL. Pro chod aplikace využiji MySQL ve verzi 5.6.12. U vytvořených databázových tabulek je třeba aplikovat určitá integritní omezení, proto využiji tabulky typu *InnoDB* [2, str. 262]. Zvolil jsem kódování ve znakové sadě UTF-8.

Aplikace však není přímo závislá na systému MySQL. V aplikaci je použita databázová vrstva *Nette Database*, která je součástí Nette frameworku popsáno v předešlé kapitole 2.2.2. Tato databázová vrstva představuje rozhraní skrze které jsem schopen komunikovat s libovolnou databází pomocí zřetěženého volání metod. V průběhu volání se dotaz zkompile do formy, dle použitého typu databázového systému a v závěru se provede. Lze tedy v případě potřeby migrovat aplikaci na konkurenční databázové systémy jakou jsou například PostgreSQL nebo Oracle. *Nette Database* také přináší optimalizaci výsledného SQL dotazu, díky které jsou skutečně provedené dotazy minimalistické. Dotaz tedy získá z databáze jen takové údaje, které se skutečně v aplikaci použijí nebo vypíše klientovy.

2.3 Klientská část

Klientskou částí je myšlena vizuální stránka aplikace spolu s technologiemi běžícími na straně webového prohlížeče. Při výběru technologií jsem dbal na to, aby byly podporovány co největším množstvím webových prohlížečů.

⁴AJAJ – Asynchronous JavaScript and JSON (technologie asynchronního zpracování webových stránek využívající textový JSON formát pro přenos dat), velmi podobná technologii AJAX [4].

2.3.1 Značkovací jazyk HTML

HTML (Hypertext Markup Language) je značkovací jazyk sloužící k zobrazení dat předem daným způsobem. Základní HTML kód je podrobný návod pro webový prohlížeč, jak zobrazit přijatá data na obrazovce [1]. Jedná se o typ dokumentu z rodiny univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language), který umožňuje definovat typ dokumentu pomocí metadat [10].

Pro účely aplikace využijí nejnovějšího standardu HTML5, který podporují všechny mnou testované webové prohlížeče. HTML5 zavádí mnoho nových funkcí včetně pokročilé práce s multimédií, formuláři, podporuje nejrozličnější rozhraní pro programování aplikací a přináší také novou sémantiku HTML prvků [7]. V neposlední řadě přináší podporu pro funkci **Drag and Drop** [13], kterou budou v aplikaci využívat. HTML5 také přináší zlepšení z hlediska přístupnosti stránek pro zrakově nebo jinak zdravotně postižené uživatele.

V úvodní části HTML5 dokumentu deklarují tzv. doctype⁵ ve tvaru: `<!DOCTYPE html>` [6]. Nyní bude webový prohlížeč připraven na zpracování HTML5 obsahu.

2.3.2 Kaskádové styly CSS

Prvky HTML mají definovanou základní sémantiku, kterou různé webové prohlížeče mohou interpretovat různým způsobem pro účely zobrazení. Proto v roce 1996 organizace W3C (World Wide Web Consortium) vytvořila návrh šablon stylů CSS (Cascading Style Sheets). Dva roky poté vznikl návrh normy CSS2, který původní verzi rozšiřuje a stává se poměrně rozsáhlý [3].

Postupem času programátoři volali po změnách a modernizaci CSS2 a proto organizace W3C vydala třetí verzi CSS3. Nová verze přináší spoustu novinek a vylepšení. Některé z hlavních uvedu v následujícím výčtu [3]:

- **Vícesloupcový layout** – rozvržení obsahu do sloupců bez použití vlastnosti *float*
- **Dotazy na média** (Media Queries) – podpora responzivního designu reakcí na dynamicky měnící se stav použitého zobrazovacího média.
- **Transformace a přechody** – otáčení překlápění a jiné deformace objektů
- **Gradienty a vícenásobná pozadí** – lineární a radiální gradient, více obrázků na pozadí stránky a objektů.
- **Grafické efekty** – rozostření a rotace objektů, zaoblené rohy, stíny atd. ...

Na těchto a dalších stěžejních novinkách CSS3 jsou postaveny tzv. CSS frameworky. Jedním ze zástupců CSS frameworků, který jsem použil při vývoji této aplikace je Bootstrap framework blíže popsany v kapitole 2.3.3. Ze začátku bylo velmi obtížné standard CSS3 použít, protože nebyl podporován většinou webových prohlížečů. Tento stav se však již změnil a nyní mohu použít většinu vlastností CSS3, aniž bych byl limitován určitým webovým prohlížečem klientské stanice.

⁵Doctype – deklarace typu dokumentu, kterou zavedl jazyk SGML [10]

2.3.3 Bootstrap framework

Bootstrap je open source CSS framework vytvořen pány Mark Otto a Jacob Thornton. Vývoj byl zahájen v srpnu roku 2011 pro potřeby sociální sítě Twitter. Jedná se čistě o CSS framework s podporou JavaScriptu. V podstatě se soustřeďuje na responzivní design, tedy na způsob, jakým lze vytvořit dokument měnící dynamicky svůj vzhled v závislosti na velikosti stránky. Bootstrap framework užívá mnoho vlastností standardu CSS3, díky kterým je pro programátora nebo designera snazší vytvořit moderní webovou prezentaci. Jsou tu však i další vylepšení které Bootstrap přináší, jako je například standardizace CSS vlastností [11]. Standardizaci bylo třeba zavést pro zvýšení kompatibility mezi dnešními webovými prohlížeči. Framework nabízí propracovaný systém mřížek tzv. grid. V jednom elementu může být až 12 sloupců, které se přizpůsobí celkem ve čtyřech možných šířkách obrazovky. To dává programátorovi značné možnosti při návrhu a implementaci aplikace.

V aplikaci *Osobní startovací stránka* je použit Bootstrap framework ve verzi 3.0.3. Na webových stránkách frameworku je dostupná také kompletní dokumentace spolu s pokyny pro dodržení typografických konceptů, řízení formulářů a správu JavaScriptových pluginů⁶.

2.3.4 JavaScript a knihovna jQuery

JavaScript je objektově orientovaný skriptovací jazyk určený právě pro webové prezentace. Jedná se o interpretovaný jazyk, který se provádí na straně klienta v internetovém prohlížeči. Je však poměrně obtížné (podobně jako je tomu u CSS) implementovat potřebnou funkcionalitu tak, aby byla v co největší míře kompatibilní s nejrůznějšími webovými prohlížeči. Proto bylo nad JavaScriptem vyvinuto mnoho knihoven, které řeší v první řadě kompatibilitu a shodné chování v různých prohlížečích. Dále se JavaScriptové knihovny snaží abstraktně popsat složitou funkcionalitu tak, aby ji mohl programátor snáze využít. Jedním příkladem za všechny je knihovna jQuery⁷.

V aplikaci je použita knihovna jQuery v aktuální verzi 2.1.0. Hlavním motem knihovny je věta: „Pište méně, dělejte více.“ [8], která perfektně vystihuje jeho filozofii. Knihovna usnadňuje procházení a manipulaci s dokumenty HTML, zpracování událostí webového prohlížeče, animace nad modelem DOM, interakce prostřednictvím technologie Ajax a programování skriptů v JavaScriptu [8, str. 21]. Knihovnu jQuery jsem zvolil z těchto důvodů:

- **Oblíbená knihovna** s početnou komunitou uživatelů.
- **Minimalizuje rozdíly** mezi webovými prohlížeči a normalizuje výsledné chování.
- **Rozsáhlá dokumentace** s praktickými příklady včetně zdrojových kódů.
- **Licence** – MIT licence, která dovoluje knihovnu využít téměř libovolně i v komerčních projektech.
- Jedná se o **open source** knihovnu, kde každý může opravovat, vylepšovat nebo se jiným způsobem podílet na vývoji.

Vedle samotné knihovny vznikly také menší doplňky, které tuto knihovnu využívají, jako je například *jQueryUI* (jQuery User Interface), *jQuery Mobile*, *Sizzle*. Je také vhodné zmínit existenci *QUnit* frameworku, který se hojně používá právě k testování obecného JavaScriptového kódu.

⁶Webové stránky Bootstrap frameworku: <http://getbootstrap.com/>, <http://getbootstrap.com/css/>

⁷Oficiální webové stránky knihovny jQuery: <http://jquery.com/>

Kapitola 3

Analýza a koncepční návrh aplikace

Pro úspěšnou webovou aplikaci je velmi důležitá hlavní myšlenka následovaná propracovaným grafickým prostředím. Tato kapitola si klade za cíl sdělit koncepci aplikace a upozornit na místa, u kterých by mohl v průběhu návrhu či implementace vzniknout problém. Lze předpokládat, že některé koncepční prvky budou mít ve finálním návrhu aplikace mírně odlišnou podobu. Nejprve uvedu soupis vyhledaných informací o již existujících aplikacích. Zde zmíním případy, které mne u aplikací zaujaly a které jsem naopak shledal jako nefunkční či nadbytečné. Dále se soustředím na koncepci aplikace. Zde se budu snažit obeznámit čtenáře s vizí mého projektu tak, jak ji vidím já a zmínit specifické aspekty aplikace.

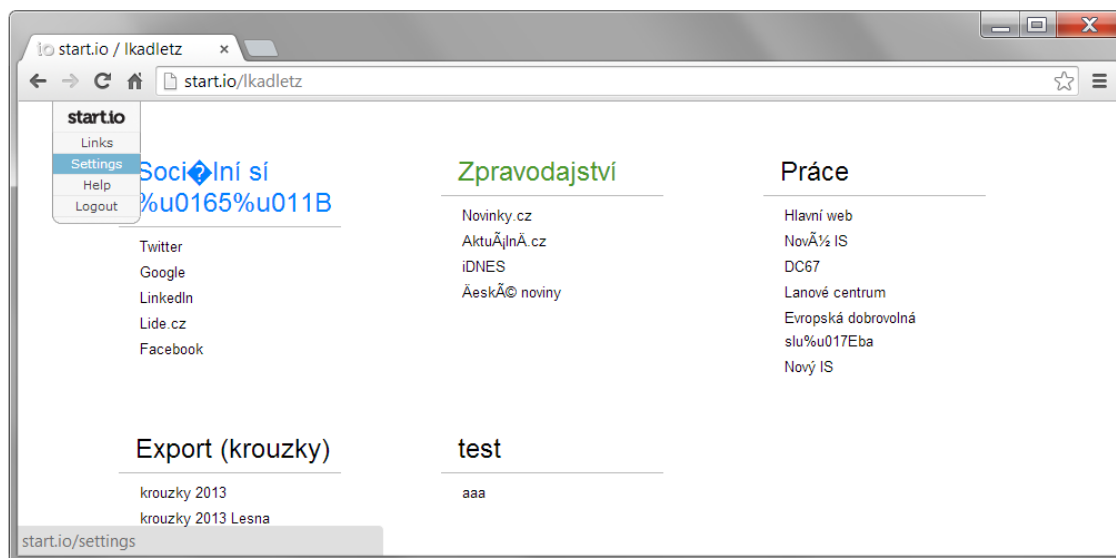
3.1 Rešerše existujících aplikací

Jak jsem již v úvodu kapitoly naznačil, existuje mnoho aplikací, které implementují některé společné prvky s aplikací *Osobní startovací stránka*. Zde je velký potenciál k inspiraci, ale i k poučení se z chyb nebo nepříliš zdařilého řešení. V následujících kapitolách uvedu některé konkrétní, dle mého názoru, důstojné zástupce aplikační problematiky. Pokusím se je objektivně zhodnotit a přidat moje názory a postřehy.

3.1.1 Start.io

Služba *Start.io* dostupná na webové adrese <http://start.io/>, je základním nástrojem pro uchování a zobrazení odkazů. Nabízí několik témat pro zobrazení odkazů včetně možnosti sdružovat odkazy podobného charakteru do kategorií. Jednotlivým kategoriím lze přiřadit barvu, která je v různých tématech zobrazena v okolí záhlaví.

Hlavní nevýhoda této aplikace spočívá v podpoře kódování. Použité kódování (nebo způsob implementace) není přizpůsobeno pro českou diakritiku a pro ostatní UTF-8 znaky. Názorně je tento problém viditelný na obrázku 3.1, kde je zobrazena stránka s odkazy seřazenými do několika kategorií. Dále není základní design stránky dostatečně optimalizován pro mobilní zařízení. Při zobrazení v desktopovém prohlížeči se sice jednotlivé kategorie přizpůsobují rozměrům displeje, ale na jejich řazení má uživatel jen minimální vliv. Negativní dojem na mne zanechala skutečnost, že si služba *Start.io* vytváří záznamy o mé aktivitě ve smyslu jednotlivých kliků na odkazy. Odkazy totiž nejsou dostupné přímo, ale pouze skrze přesměrování, což mi z hlediska uživatele vadí. Chápu, že některé statistiky mohou slou-



Obrázek 3.1: Snímek aplikace *Start.io* ukazuje chybné zobrazení diakritiky.

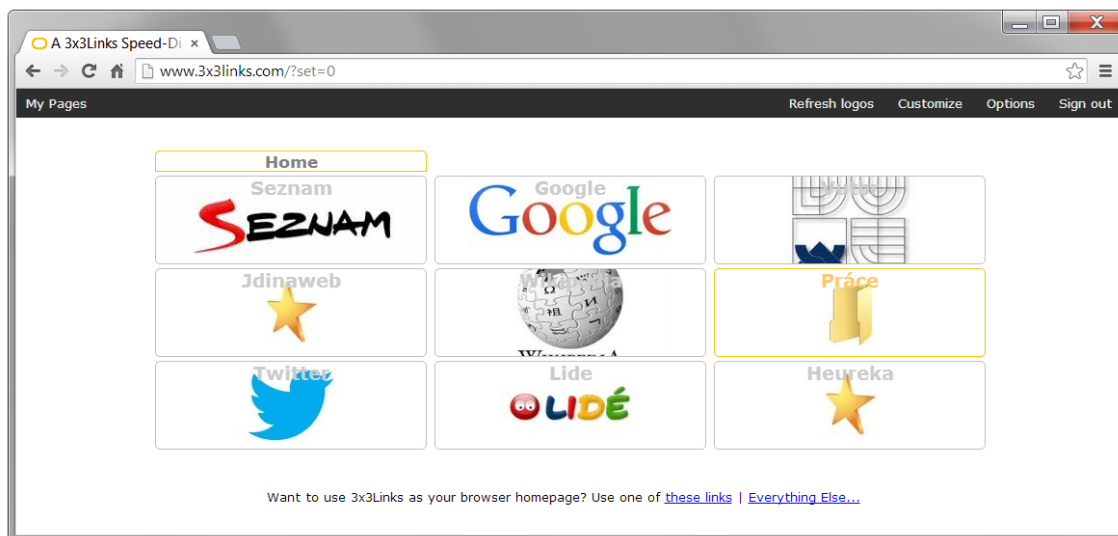
žit jako podklad pro vylepšení aplikace, v tomto případě je však tato statistika dle mého názoru zcela irelevantní a spíše narušuje uživatelské soukromí.

Naopak výhodu vidím v práci aplikace se šablonami. Pokročilý uživatel znalý značkovacího jazyka HTML a kaskádových stylů CSS je schopen přizpůsobit si zobrazení stránky podle sebe. Zde vidím potenciál, ale nakonec tuto funkcionalitu, dle mého názoru, využije jen zlomek uživatelů. Pokud by totiž uživatel vložil svůj čas do tvorby vlastní šablony, musel by se vyrovnat s již zmíněnými zápory celé aplikace. U aplikace *Start.io* musím ještě vyzdvihnout intuitivnost grafického uživatelského rozhraní, které netvoří novým uživatelům zábrany ve formě zmatení nebo dlouhého zkoumání. Aplikace pracuje s technologií *Drag and Drop* fungující dle očekávání a až na výjimky plynule. Při testování se mi často stávalo, že jsem nedostal od aplikace zpětnou vazbu o provádění nějaké operace na pozadí a nabyl jsem přesvědčení, že se aplikace zasekla. Například při mazání kategorie jsem nedostal vizuální odezvu, že byla kategorie smazána a stále se na stránce zobrazovala. Teprve až po manuálním obnovení okna jsem se přesvědčil, že prováděná operace dopadla úspěšně. Těchto aspektů bych se měl při vývoji aplikace vyvarovat a snažit se, aby uživatel neustále vnímal, že se něco děje a dostal vizuální zpětnou vazbu o stavu operace.

3.1.2 3x3links.com

U aplikace *3x3links.com* objevuji čistý design a originální nápad. Prostředí aplikace se skládá z devíti polí s odkazy. Kliknutím na složku přesune je uživatel přesunut do další úrovně s celkem osmi odkazy a jedním tlačítkem *zpět*. Naopak kliknutím na odkaz je uživatel rovnou přesměrován na požadovanou stránku. Snímek aplikace zobrazuje obrázek 3.2. Tento koncept by se mohl zdát zdlouhavý, ovšem je tu podpora i pro klávesové zkratky, s těmi se jistě použitelnost zvýší. V této fázi lze uvažovat o použití klávesových zkratk i v aplikaci *Osobní startovací stránka*.

Zklamáním je jednoznačně registrace, která nenabízí jinou autentizaci než pomocí *Google účtu*. To by mohl být pro značné množství uživatelů nepřekonatelný problém, na druhou stranu méně početná skupina tuto funkci ocení. Možnosti nastavení jsou rozsáhlé včetně



Obrázek 3.2: Aplikace *3x3links.com* s několika odkazy a složkou.

vložení vlastního souboru s kaskádovými styly. Dále aplikace umožňuje vkládat uživatelem vytvořené HTML elementy. Podobně jako u aplikace *Start.io* (3.1.1) je však vyžadována znalost značkovacího jazyka HTML. Zmíním ještě možnost stáhnout si kompletní stránku jako HTML a zobrazovat ji přímo uloženou na disku počítače. Toto řešení má nespornou výhodu v rychlosti zobrazení, ovšem změna parametrů nebo odkazů není z povahy věci možná. Grafické zpracování aplikace je částečně přizpůsobeno pro mobilní zařízení.

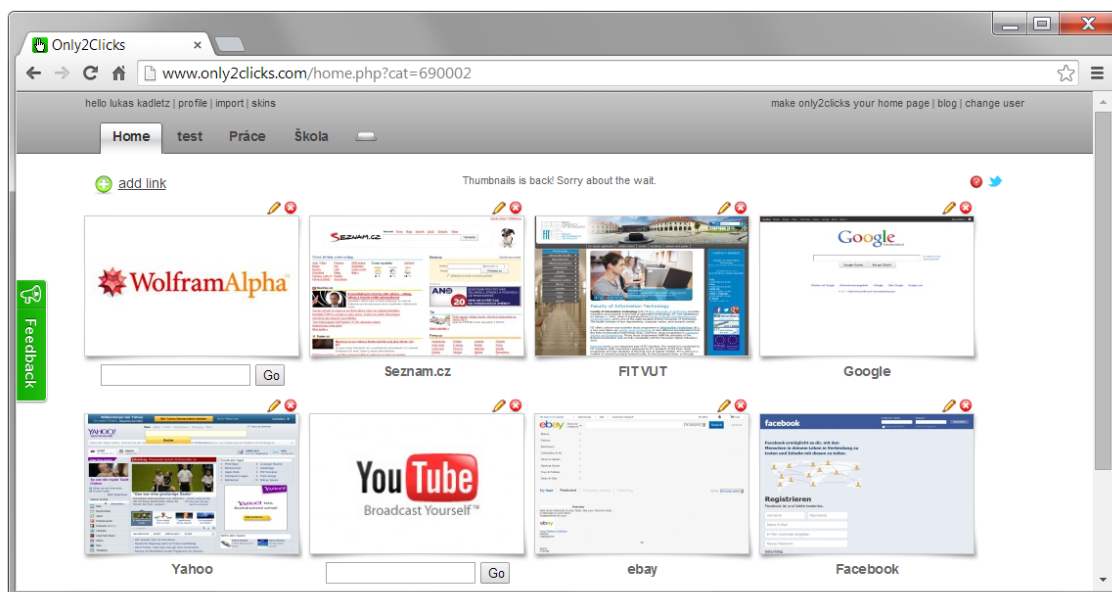
3.1.3 Only2clicks.com

Jedná se o graficky zajímavou aplikaci s využitím náhledů stránek namísto odkazů. Což se zdá být pro uživatele na první pohled srozumitelnější a dodává aplikaci přehlednost. Stálá přítomnost editačních ikon ovšem může uživatele rušit při používání aplikace, jak lze názorně vidět na obrázku 3.3. Používané odkazy se nemění příliš často, a proto by měl být editační režim oddělen od režimu prohlížení. Stránka je dále rozdělena na složky, což je dle mého názoru povedené řešení. Uživatelé s mnoha odkazy to jistě ocení, zvláště, když jen osm odkazů (políček) zabere prakticky celou stránku.

Výhrady mám především k rušícím elementům jako jsou stále přítomné ikony nastavení a spodní lišta, kterou bych spíše ukotvil do spodní části stránky. Co se týče nastavení aplikace, tak to je řešené základním způsobem. Je dostupná možnost změny e-mailové adresy, což je jistě dobrá věc. Dále se v nastavení vyskytuje možnost upravit si přístupovou URL adresu. Stránka s nastavením, včetně hlavní stránky, disponuje přizpůsobením pro různou velikost obrazovky, ale často se stane, že se některý prvek stránky zobrazí chybně. Na závěr zmíním možnost velmi snadno změnit vzhled stránky. Otázkou zůstává, jak často si bude uživatel tento vzhled měnit.

3.1.4 Startme.com

Aplikace je dle mého názoru velice povedená. Uživatelské rozhraní je jednoduché, čisté a do jisté míry i intuitivní. Jedná se, podobně jako u služby *Start.io* popsané v kapitole 3.1.1, o aplikaci založenou na blocích s odkazy. Tyto bloky se zobrazují ve třech pevných sloupcích



Obrázek 3.3: Snímek prostředí aplikace *Only2Clicks.com*

pod sebou. Každý z odkazů v bloku je doprovázen faviconou¹ webové stránky. Toto řešení velmi pomáhá orientaci na stránce, díky miniaturním ikonám je uživatel schopen v krátkém čase potřebný odkaz nalézt. Problémem je však skutečnost, že je aplikace schopna zobrazit přibližně jen 50% těchto icon. Ačkoliv všechny testované webové stránky faviconu měly, zobrazovala se místo ní jen ikona planetky. Je možné, že aplikace nepočítá s některými způsoby uložení této ikony.

Další funkcionalitou je RSS čtečka. Čtečka zobrazuje požadovaný kanál včetně fotografie nebo loga kanálu a několik úvodních vět prvního odkazu. Aplikace také disponuje vícestránkovým provedením, kde na každé stránce je možné zobrazit libovolný počet bloků, případně RSS zdrojů. Co se týče možnosti ovládání aplikace z mobilního telefonu, tak ty jsou vesměs obstojné. Bloky se při nižším rozlišení seskládají nejprve do dvou a při dalším snižování šířky obrazovky do jednoho sloupce. Na široké obrazovce však aplikace stále dodržuje rozvržení do tří sloupců. Toto řešení není na dnes už běžném rozlišení FullHD² ideální. Především z důvodu zbytečného prázdného místa na pravé straně každého z bloků. Jen u RSS odkazů tento hendikep není přítomný, protože nadpisy článků jsou většinou delší než čtyři slova. Při větším množství odkazů a záložek bude muset uživatel častěji posunovat obrazovku směrem dolů. Tento problém se pokusím v aplikaci Osobní startovací stránka eliminovat. Jinak si z této aplikace odnáším vesměs pozitivní dojmy a je zde jistě několik bodů, kde je možné získat inspiraci.

Služba prošla na začátku roku 2014 modernizací a v pondělí 21.dubna 2014 se přesunula na stránky *start.me*. Aplikace dostala zcela nový responzivní vzhled. Nově se lze do aplikace přihlásit skrze sociální sítě v čele se sítí *Facebook* a *Google+*.

¹Favicon je miniaturní ikona webové stránky umístěná většinou v hlavním adresáři stránky mající název *favicon.ico*

²FullHD rozlišením myslím 1920 bodů horizontálně a 1080 bodů vertikálně při poměru stran 16:9.

3.1.5 Zhodnocení existujících aplikací

Na základě provedené rešerše mohu říci, že na trhu existuje mnoho aplikací, které by úlohu startovací stránky mohly zastávat. Některé, jako například *Start.io*, jsou zaměřeny pouze na odkazy. Jiné služby jako jsou například *3x3links.com* nebo *Only2clicks.com* umožňují vložit na stránku i vyhledávací pole nebo interaktivní složky. Aplikace *Startme.com* přináší RSS čtečku a pevné sloupcové zobrazení. U každé aplikace jsem vybral několik kladných a několik záporných bodů, které zobrazuje tabulka 3.1.

Název aplikace	Klady	Zápory
Start.io	<ul style="list-style-type: none">– čistý a přehledný design– snadné ovládání– vlastní URL adresa	<ul style="list-style-type: none">– chybné zobrazení diakritiky– zobrazení na mobilním telefonu– řazení kategorií
3x3links.com	<ul style="list-style-type: none">– částečně responzivní design– ovládání klávesami– vlastní adresa	<ul style="list-style-type: none">– registrace skrze Google účet– málo místa pro odkazy– nepřehledné nastavení
Only2clicks.com	<ul style="list-style-type: none">– náhledy webových stránek– práce se složkami– vyhledávací pole	<ul style="list-style-type: none">– přítomnost editačních ikon– málo místa pro odkazy– chudé nastavení
Startme.com	<ul style="list-style-type: none">– čistý design s odkazy– RSS čtečka– icony webových stránek	<ul style="list-style-type: none">– tři pevné sloupce– absence vlastní URL– nelze zobrazit bez přihlášení

Tabulka 3.1: Shrnutí kladů a záporů zjištěných u testovaných aplikací.

3.2 Koncepční návrh aplikace

Jak už bylo naznačeno v úvodu, cílem mé práce je vytvořit aplikaci *Osobní startovací stránka*. Pod těmito třemi slovy si každý může představit něco mírně odlišného, proto se v následujících odstavcích pokusím sdělit můj pohled. Soustředím se na konceptní rámec a pokusím se uvést a podrobně vysvětlit cíle, kterých chci touto aplikací dosáhnout.

Obecně si pod názvem této aplikace představuji jednu čistou stránku, na které budu já jako uživatel mít zobrazeny moje záložky. Záložky bych si chtěl seskupovat do logických celků, abych se v nich vyznal a dokázal rychle najít to, co potřebuji. Jenže to mi nestačí, mám mnoho RSS zdrojů, které odebírá a používám tedy více RSS čteček. Chci, aby aplikace měla RSS čtečku integrovanou. Dále chci aplikaci používat na mobilním telefonu a tabletu, zkrátka všude tam, kde jsem připojen k internetu. Častěji si budu stránku jen zobrazovat, než měnit, proto by měla být rychlá a svižná. Když budu chtít něco jednou za čas upravit, nechci pořád dokola zkoumat, jak se to dělá. Chci mít jistotu v ovládání a chci čistou aplikaci bez obtěžující grafiky, která spíše aplikaci znepřehledňuje. Také chci přistupovat ke své stránce z jedné adresy URL, kterou si nakopíruji do všech mobilních zařízení a nastavím ji jako domovskou. Takto si aplikaci *Osobní startovací stránka* představuji a nyní zkusím jít trochu do hloubky a zaměřit se na detaily.

Aplikace bude zaměřena na jednoduchost ovládání a přizpůsobivost platformě na které je užívána. Ze své osobní zkušenosti vím, že uživatelé o sobě nechtějí příliš informací prozrazovat a vkládat je do webových formulářů, proto bude aplikace při registraci vyžadovat jen nutné údaje. Aplikace však bude mít možnost některé údaje doplnit později v nastavení, ale

půjde o volitelnou možnost. Mým cílem je do aplikace přidat také možnost registrace skrze nejrozšířenější sociální sítě jako jsou *Twitter*, *Google+* nebo *Facebook*. Na základě malého průzkumu jsem od lidí v mém okolí získal kladné ohlasy. Většina z nich totiž registrace na webové stránky skrze některou ze sociálních sítí využila. Často také upozorňovali na fakt, že se jedná o rychlou a nenáročnou činnost, která oproti registračním formulářům netvoří překážku před použitím nové webové služby.

Odkaz je bezesporu hlavní entita aplikace a bude se vkládat do *kategorií*, které budou asociovány se *stránkami*. Kategorie i stránky budou mít svoje specifické nastavení stejně jako jednotlivé odkazy. Chci umožnit uživateli toto nastavení měnit i globálně, respektive nastavit globální výchozí hodnoty. Nastavení bude uloženo v databázi takovou formou, aby umožňovalo bez větších problémů rozšíření. Další vlastností bude časová závislost zobrazení prvků. Tedy v některém čase chci například zobrazit kategorii „Práce“ a v jiném kategorii „Blog“. Chci do aplikace zahrnout stránkování. Sice si nejsem jistý, jak uživatelé stránkování přijmou a jestli se jedná o správnou cestu, ale zastávám názor, že to smysl má. Proto chci dát uživateli prostor k vytvoření více stránek, ale pokud je chtít nebude, může používat jednu základní, v tom případě bude ovládání aplikace stále stejné. Čas ukáže, jestli tuto funkcionalitu uživatelé přijmou nebo ne.

Jak vyplývá z povahy věci, data uložená v aplikaci se nebudou příliš často měnit. Proto se zaměřím na cacheování³ obsahu. Chci zajistit, aby se hotová stránka zobrazila uživateli pokud možno jako statická. Pokud se ji rozhodne uživatel editovat, přesune se do editačního režimu a stránka se z dočasné paměti odebere. Jakmile uživatel ukončí nastavení a vrátí se na hlavní stránku, budou stránky uloženy do mezipaměti pro příští rychlejší přístup.

Ovládat aplikaci bude možné pomocí myši a doteků prstu (především na tabletech a mobilních telefonech). HTML5 přináší mnoho vylepšení mezi které patří i technologie **Drag and Drop** (2.3.1), kterou v aplikaci využiji. Mělo by se jednat o posun kategorií na stránce v rámci mřížky. Tato technologie je pro uživatele srozumitelná a myslím, že její implementaci pomohu k efektivnější práci v prostředí. Kategorie budu zároveň seskládat tak, abych co nejlépe využil šířky zobrazovací jednotky.

Vzhled startovací stránky bude střídavý, pokud bude uživatel stránku používat (nebude provádět změny na stránce nebo v konfiguraci), tak se mu nebudou zobrazovat nadbytečné ikony. Cílem je přehledná stránka na niž uživatel během krátkého času najde co hledá. Co se týče přístupu na stránku, zde si představuji tři možnosti:

1. Klasické přihlášení do uživatelského prostředí.
2. Skrze jedinečnou URL adresu, která bude obsahovat argument ve tvaru generované náhodné posloupnosti znaků a čísel.
3. Skrze zapamatovatelnou URL, kde argumentem bude uživatelem zvolený řetězec.

Druhá možnost bude pravděpodobně využívána nejčastěji, protože uživatel bude mít možnost přístupu na stránku pomocí URL, kterou zná jen on a která je obtížně generovaná webovými roboty. Tato možnost tedy nabízí kompromis mezi bezpečností a rychlostí přístupu. U třetího způsobu umožním uživateli zvolit si i více řetězců, případně některé řetězce se mohou odkazovat na jiné stránky z množiny. U posledních dvou zmíněných možností platí, že upravovat stránku uživatele bude možná až po standardním přihlášení.

Předpokládám, že někteří aktivní uživatelé budou mít o aplikaci zájem a budou se snažit sdělit administrátorovi své postřehy a náměty. Aplikace by tedy měla obsahovat kontaktní

³Cacheování – Obecná metoda zvyšování rychlosti přenosu dat [5, str. 64]

stránku a stránku podpory. Na kontaktní stránce by bylo dobré uvést e-mailový kontakt v podobě dostatečně zabezpečené proti webovým robotům. Dále chci na kontaktní stránku umístit formulář, skrze který může uživatel snadno a rychle napsat svůj názor a odeslat jej. Předpokládám, že uživatelé častěji využijí formuláře, než přímý e-mailový kontakt. Stránky podpory by měly obsahovat odpovědi na základní otázky týkající se užití aplikace. Měla by zde být vysvětlena podstata aplikace a její přednosti. Obsah doplním o krátké motivační video a několik snímků aplikace.

Kapitola 4

Návrh aplikace

Kapitola o návrhu aplikace obsahem zasahuje do databázového systému, případů užití aplikace, rozvržení modelových vrstev a v neposlední řadě do grafického uživatelského rozhraní. Pokusím se nadefinovat většinu případů užití, které vyplývají z konceptuálního návrhu uvedeného v kapitole 3.2. Jelikož se jedná o aplikaci, která se z velké části opírá o názory, doporučení či nápady uživatelů, mohou být některé detaily návrhu během implementace pozměněny. Vždy se však budu snažit na tuto skutečnost upozornit a řádně ji zdůvodnit.

4.1 Vytvoření diagramu případů užití

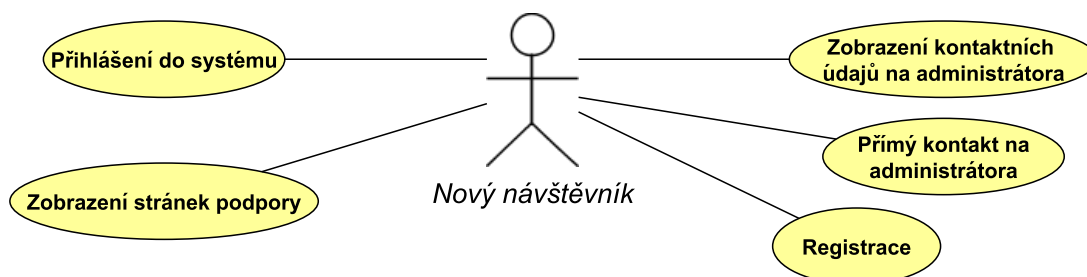
Důvodem k vytvoření diagramu případů užití je zachycení co největšího množství požadavků na funkčnost aplikace. Právě díky případům užití lze rozdělit celek systému na podproblémy a zabývat se jimi zvlášť. Jedná se o vhodný prostředek, protože při pohledu na celek systému se často stane, že drobnosti nebo podstatné detaily se snadno přehlédnou. V následujících podkapitolách se zaměřím na vybrané konkrétní případy užití a upozorním právě na ty podstatné detaily. Pro přehlednost bude v textu zobrazen vždy jen výřez diagramu užití, kompletní diagram naleznete v příloze B.1.

4.1.1 Návštěvník

V následujících odstavcích se budu věnovat vybraným případům užití z pohledu uživatele přicházející na webové stránky aplikace poprvé nebo se ze systému právě odhlásil (dále jen návštěvník). Návštěvník bude disponovat základními možnostmi, jak se systémem zacházet. Jednotlivé případy užití, zmíněné v následujících odstavcích, zobrazuje diagram na obrázku 4.1.

Registrace

Registrace je zajisté nejdůležitější krok z hlediska návštěvníka, který jej dělí od použití aplikace. Proto nechci návštěvníkům provedení registrace nijak komplikovat. Systém návštěvníka vyzve k zadání jeho e-mailové adresy, která je nezbytná pro komunikaci s budoucím uživatelem a ke zvolení nového hesla. Nové heslo je třeba zopakovat pro kontrolu, tím zabráním jednorázovým překlepům. Jakmile uživatel zdárně dokončí registraci, bude mu odeslán potvrzující e-mail s jedinečným odkazem pro potvrzení vlastnictví e-mailu. Účet uživatele se stává aktivním až po potvrzení odkazu v e-mailu. Pokud by však z nějakého důvodu (např.: uživatel si e-mail sám omylem smazal) došlo ke ztrátě aktivačního odkazu,



Obrázek 4.1: Výřez digramu užití z pohledu nového návštěvníka (Kompletní diagram naleznete v příloze [B.1](#))

může si uživatel vyžádat obnovení hesla. Tímto procesem jsem schopen e-mailovou adresu znovu ověřit, aktivovat účet a zároveň vyzvat uživatele k zadání nového hesla.

Přihlášení do systému

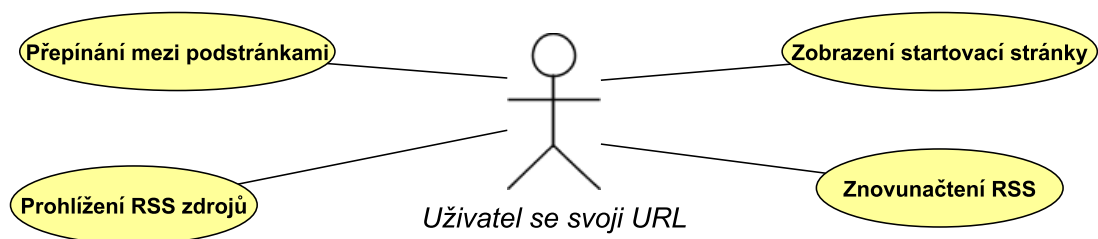
Přihlášení do systému bude spolu s formulářem pro registraci dostupné jak na hlavní stránce aplikace, tak na samostatné adrese. Proces přihlášení se skládá ze dvou kroků. Nejprve systém vyzve uživatele k zadání správných přihlašovacích údajů (e-mailové adresy a hesla) a následně nabídne možnost dlouhodobého nebo krátkodobého přihlášení. Dlouhodobé přihlášení bude platné přibližně po dobu jednoho měsíce, naopak krátkodobé přihlášení bude trvat jen do zavření prohlížeče. Návštěvník se po přihlášení do systému stane v kontextu diagramu užití přihlášeným uživatelem, jehož další případy užití aplikace popisuje kapitola [4.1.3](#).

Přímý kontakt na administrátora

Při návrhu aplikace jsem se rozhodl zařadit tento případ užití především proto, že chci uživatelům umožnit odeslat svůj dotaz či podnět přímo. Chci tím získat více zpětné vazby od uživatelů. Zároveň mají uživatelé možnost odeslat klasický e-mail z jejich poštovního klienta. Rozhodl jsem se, že kontaktní formulář bude obsahovat mimo e-mailové adresy návštěvníka také předmět, který uživatel sám specifikuje. Mnoho webových služeb do svých kontaktních formulářů formu předmětu nezavádí, ale já věřím, že každý z uživatelů je schopen svůj problém popsat několika slovy právě do předmětu zprávy. Po odeslání dotazu bude uživatel o úspěšnosti operace upozorněn.

Stránky podpory

Jakmile přijde nový návštěvník na webovou stránku, chci, aby měl možnost dozvědět se o aplikaci více. Občas nestačí přečíst si několik přesně mířených vět a shlédnout pár náhledů aplikace. Stránky podpory budou proto obsahovat strukturovaný popis aplikace. Dále blok často kladených otázek a samozřejmě kontaktní údaje případně odkaz na kontaktní formulář. Součástí stránek podpory budou také odkazy na videa obsahující návody, jak s aplikací zacházet. Rozhodl jsem se takové video vytvořit a formou zachycení aktuální obrazovky aplikace v doprovodu s ústním komentářem. Jsem přesvědčen, že video jakožto multimediální komunikační kanál pomůže uživatelům v práci s aplikací více, než dlouhé odstavce popisu prostředí.



Obrázek 4.2: Výřez digramu užití z pohledu uživatele s vlastní URL (Kompletní diagram naleznete v příloze B.1)

4.1.2 Uživatel s vlastní URL

Uživatелеm s vlastní URL mám na mysli toho, který je již registrovaný navštíví webovou aplikaci pod jeho adresou, kterou dostal přidělenou po registraci. Případy užití jsou zobrazeny na obrázku 4.2. Takovýto uživatel bude na stránky přistupovat nejčastěji. Předpokládám, že si přidělené URL vloží do svých chytrých telefonů, tabletů nebo stolního počítače a budou ji používat jako domovskou stránku. Tedy při každém otevření prohlížeče se tato stránka znovu načte.

Zobrazení startovací stránky

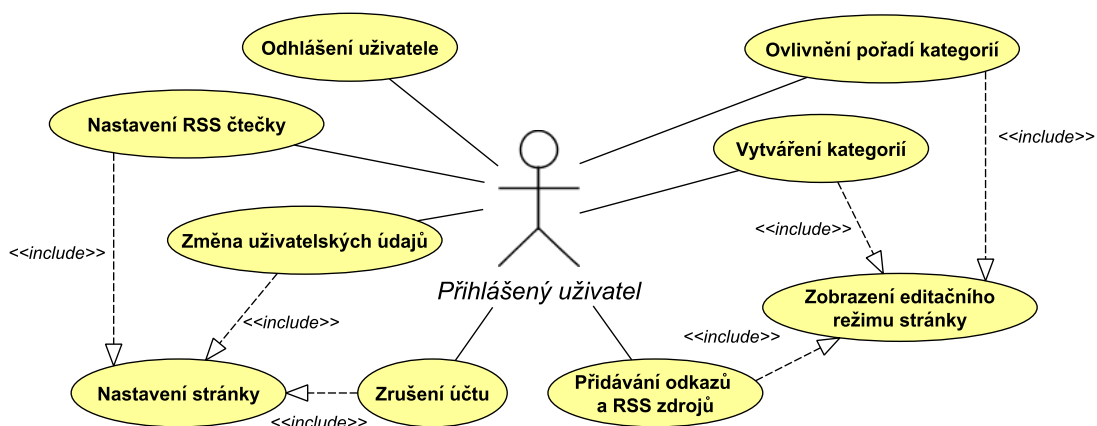
Zobrazení stránky musí být svižné a přesné. Stránka bude optimalizována na různé šířky obrazovky s využitím CSS frameworku a ostatních klientských technologiích popsaných v kapitole 2.3. Na stránce budou zobrazeny také RSS zdroje a další widgety. Uživatel však nemá možnost stránku jakkoliv měnit, aniž by se před tím řádně přihlásil. Bude se jednat čistě o read-only přístup. Zde se také soustředím na rychlou odezvu aplikace, pokusím se jednou načíst z databáze potřebné data, vytvořit stránku a poté ji uložit jako statickou do mezipaměti. Stránka uložená v mezipaměti se bude uživateli zobrazovat zřetelně rychleji.

Znovunačtení RSS

Uživatel si na svoji stránku může vložit libovolné množství RSS zdrojů. Pokud však budu vytvořenou stránku ukládat do mezipaměti, tak by mohli uživatelé o ty nejnovější zprávy z RSS kanálů přijít. Proto jsem se rozhodl přidat do stránky tlačítko pro obnovení všech RSS zdrojů na aktuální stránce. Ovšem i bez použití tohoto tlačítka by měly být informace alespoň částečně aktuální, proto bude načtená stránka v mezipaměti jen omezenou dobu. Přesný údaj zvolím na základě testování, v průběhu návrhu nemá smysl s tímto údajem pracovat.

Přepínání mezi podstránkami

Navrhl jsem použití více stránek mezi kterými může uživatel přepínat. Přepínací pole bude ve formě horizontálního panelu s tlačítky. Uživatel v jedné chvíli uvidí všechny svoje stránky na jednom řádku a může si jednu z nich k zobrazení vybrat. Zde také myslím na použití v mobilních zařízeních, proto musí být tlačítka přijatelně velká, aby je mohl uživatel pohodlně aktivovat.



Obrázek 4.3: Výřez digramu užití soustředěného přihlášeného uživatele (Kompletní diagram naleznete v příloze [B.1](#))

4.1.3 Přihlášený uživatel

Přihlášený uživatel je takový, který se právě do systému přihlásil nebo ten, který má aktivní sezení (dlouhé nebo krátké). Po zobrazení domovské stránky je přihlášený uživatel automaticky přesměrován na svoji startovací stránku. Na stránce jsou uživateli nabídnuty rozšířené případy užití dle diagramu na obrázku [4.3](#). V následujících odstavcích provedu podrobnější zamyšlení nad jednotlivými případy užití.

Nastavení stránky

Nastavení systému bude probíhat na jedné stránce přístupné z menu. Nastavení obsahuje úpravu profilových informací včetně změny e-mailu, URL stránky a podobně. Dále je zde uvedeno globální nastavení vztahující se k Osobní stránce. Tady je důležité zmínit, že globální nastavení bude aplikováno na elementy u kterých není toto nastavení explicitně změněno. Výchozí hodnoty lze měnit, případně resetovat u jednotlivých komponent. Nastavení stránky je prostředník k dalším případům užití jakou jsou například tyto:

- **Nastavení RSS čtečky** – Uživatel bude mít možnost nastavit si stránkování zobrazených odkazů, protože RSS zdroj nabízí zpravidla několik desítek odkazů a zobrazovat je všechny naráz by nedávalo smysl. Dále bude umožněno uživateli rozhodnout se, jestli chce zobrazovat ikony RSS kanálů, nebo jejich textový popis, který si sám vytvoří.
- **Změna uživatelských údajů** – Samozřejmostí u dnešních aplikací jsou úkony typu: změna hesla, změna e-mailového účtu, úprava uživatelského jména a podobně. Uživateli bude umožněno tyto hlavní úkony provést spolu se změnou své soukromé adresy URL.
- **Zrušení účtu** – Dle mého názoru je zrušení účtu velmi důležité. Pokud již uživatel nebude chtít aplikaci používat, měl by mít možnost účet zrušit a odstranit tedy všechny své osobní údaje ze serveru. Zrušení účtu bude tedy součástí nastavení aplikace.

Zobrazení editačního režimu

Editační režim sice nebude navštěvovaný tak často jako přímé zobrazení startovací stránky, ale právě proto je třeba jeho uživatelské rozhraní vytvořit intuitivně. Uživatel by měl najít co nejrychleji to potřebné a vědět jak to použít. Vzhled editačního režimu se bude na základě zpětné vazby uživatelů přizpůsobovat.

Při editování kategorií chci uživateli alespoň částečně zachovat pohled na aplikaci tak, jak vypadá startovací stránka read-only režimu. Pouze přibudou ikony a panely umožňující přidání *RSS zdrojů*, *kategorií* a *widgetů*. Funkce změny pořadí kategorií bude doprovázeno o odkazy ve tvaru šipek, které budou pořadí měnit. Ne vždy je pro uživatele pohodlné využít funkci **Drag and Drop**, proto chci zachovat oba způsoby.

Odhlášení uživatele

Odhlášení uživatele bude dostupné z menu tak, aby bylo jasné patrné, že se o tuto funkci jedná. K samotnému odkazu pravděpodobně přidám i doprovodný symbol odhlášení ze systému. Odhlášení uživatele bude probíhat i automaticky, respektive v závislosti na čase. Pokud se uživatel přihlásí na kratší dobu, bude automaticky odhlášen po zavření webového prohlížeče nebo po uběhnutí doby relace. Ve druhém případě se uživatel automaticky odhlásí, pokud na svoji stránku nevstoupí delší dobu, například jeden měsíc. Konkrétní dobu zatím nejsem schopen stanovit, ale dokáží si ji představit v řádu několika týdnů.

4.2 Návrh Entity-Relationship diagramu

Ke správnému návrhu relační databáze je vhodné vytvořit konceptuální model zvaný Entity-Relationship Diagram (dále jen ER diagram). V diagramu se nacházejí entitní množiny, tedy množiny entit, kde každá entita představuje jednu samostatnou jednotku mající svůj jednoznačný identifikátor v rámci množiny. V diagramu tyto identifikátory vyznačuji takto: **<<PK>> NazevIdentifikatoru**. Zkratka PK pohází z anglického **Primary Key** a značí, že se jedná právě o jedinečný identifikátor. Každá entita obsahuje také množinu atributů pro lepší představu, jaká data se budou později ukládat do databáze. V následujících odstavcích vysvětlím podstatné entitní množiny a vztahy mezi nimi. Kompletní ER diagram aplikace naleznete v příloze **B.2**.

Entitní množina User a množina Page

Entitní množina **User** je základem pro funkčnost celé aplikace. Jednotlivé entity představují vždy jednoho uživatele majícího několik atributů. Atribut **UserID** uchovává jedinečný identifikátor v rámci entitní množiny. Atribut **passwordHash** bude obsahovat otisk uživatelského hesla, protože v případě útoku na databázi a získání dat třetí stranou, nebude možné uživatelské heslo získat v čitelné podobě. Tímto je zajištěno alespoň základní zabezpečení. Dále ještě zmíním atribut **PrivateHash**, který bude obsahovat náhodnou posloupnost znaků a čísel jedinečnou pro každého uživatele. Tuto posloupnost bude mít uživatel možnost změnit nebo si ji nechat vygenerovat znovu.

Každá z entit v množině **Page** obsahuje několik atributů. Hlavním atributem je stejně jako u entity z množiny **User** primární klíč **PageID**. Atribut **linearOrder** udržuje informaci o pořadí stránky ve skupině stránek přiřazených jednomu uživateli. Toto pořadí bude důležité pro zobrazení seznamu stránek v aplikaci. Následuje další důležitý atribut a tím je **caption**, uchovávající název stránky k zobrazení.

Mezi entitními množinami **User** a **Page** je vztah s kardinalitou 1:M a popisem *vlastní*. Vztah vyjadřuje závislost mezi jednotlivými entitami. V tomto případě má každá entita množiny **Page** právě jednoho *vlastníka*, entitu z množiny **User** s jednoznačným identifikátorem **UserID**. Tedy každá entita z množiny **User** může *vlastnit* libovolné množství entit z množiny **Page**, nebo také žádnou.

Entitní množina **Folder**

Entitní množina **Folder** zastupuje všechny kategorie na stránkách. Při návrhu jsem se rozhodl vztah mezi entitními množinami **Folder** a **Page** modelovat s kardinalitou vztahu M:N. Vedla mě k tomu myšlenka mít na více stránkách stejné kategorie a při změně jedné z nich by se změnily i ostatní, protože se bude jednat stále o tu stejnou kategorii. Dokáží si představit využití této funkcionality například u kategorií s RSS zdroji. Zde si uživatel může vytvořit jednu kategorii a tu si pak nechat od aplikace zobrazit na všech, případně jen na některých, stránkách. Součástí takového vztahu je i atribut, který se na rozdíl od všech ostatních neváže k jediné entitní množině, ale k oběma zároveň. Jedná se o atribut **linearOrder**, který podobně jako u entitní množiny **Page** určuje pořadí kategorií náležících jedné stránce.

4.3 Grafický návrh uživatelského prostředí

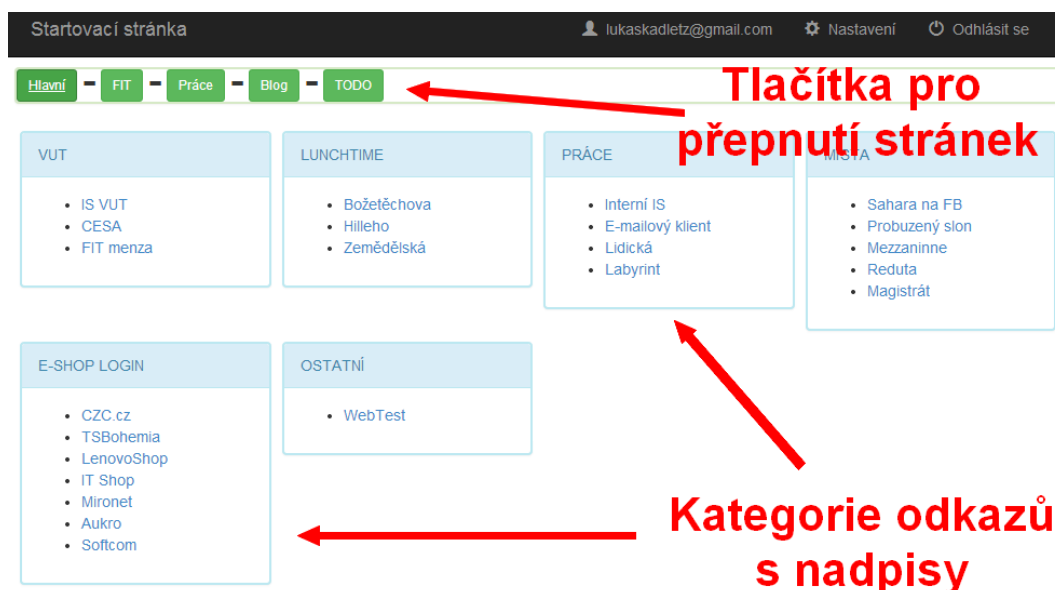
Hlavní požadavky na grafické uživatelské prostředí již nastínila koncepce v kapitole 3.2. Pro jejich realizaci postavím základ grafického prostředí na open-source CSS frameworku *Bootstrap*, blíže je tento framework popsán v kapitole 2.3.3. Při návrhu se pokusím co nejvíce využít vlastností, které mi *Bootstrap* nabízí. Navržený design aplikace se skládá ze tří hlavních částí a to *horní lišty*, *obsahu* a *patičky*. V následujících odstavcích tyto části blíže představím.

Horní lišta

Horní lišta obsahuje několik základních položek jakými jsou: *Přihlášení*, *Registrace* a *Kontakt*. Po úspěšném přihlášení do aplikace se v menu zobrazují položky: *Upravit stránku*, *Nastavení* a *Odhlášení*. Horní lišta by má mít především navigační charakter a zobrazit vždy relevantní informace. Proto jsem se rozhodl menu skrývat jen při minimální šířce obrazovky. Menu se na mobilním telefonu skryje pod jedno pole opatřené grafickým znázorněním menu. Po kliknutí na toto pole se celé menu vysune. Podobně je tomu i u aplikací v mobilních telefonech. Samotné vysunutí zajistí JavaScriptový framework *jQuery*. Zmíněné vlastnosti jsou typické pro aplikace postavené na *Bootstrap* frameworku.

Obsah aplikace

Hlavní stránka aplikace by měla jasně sdělit uživateli, co mu aplikace nabízí. Obsah zobrazí uživateli základní informaci o službě, možnost přihlášení a registrace. Dále bude následovat část vypichující hlavní přednosti aplikace společně s odkazy na video a návody. Soustředím se na zmínění hlavních důvodů, proč a jak by mohla aplikace uživateli pomoci. Na hlavní stránce by měl být také prvek, který nově příchozího uživatele zaujme. Dle mého názoru bude nejlepší umístit na hlavní stránku také screenshoty aplikace. Dobře vytvořený screenshot řekne více, než několik správně mířených vět.



Obrázek 4.4: První grafický návrh uživatelského prostředí startovací stránky

Další důležitá stránka aplikace se uživateli zobrazí ihned po přihlášení. Bude se jednat o startovací stránku v režimu prohlížení. První grafický návrh prostředí startovací stránky znázorňuje obrázek 4.4. Na obrázku lze vidět horní lištu, na které jsou zobrazeny základní prvky menu. Dále pak pod horní lištou nalezneme tlačítka, kdy každé z tlačítek představuje jednu stránku, na kterou se bude moci přepnout. Tlačítka budou reagovat na klik (nebo dotek prstu) grafickým znázorněním zmáčknutého tlačítka. Především z toho důvodu, že stránky budu přepínat technologií AJAX, a uživatel by měl ihned po zmáčknutí tlačítka dostat zpětnou vazbu, i když se stránka fyzicky přepne až za krátký okamžik. Pole s kategoriemi bude obsahovat uživatelem definovaný počet bloků. Tyto bloky se budou řadit dle aktuální velikosti obrazovky. Pokud si tedy uživatel stránku zobrazí na mobilním telefonu, bude mít vedle sebe maximálně dvě kategorie. Vytvoří se tedy maximálně dva sloupce. V opačném případě kdy bude obrazovka dostatečně široká se počet sloupců zvýší přibližně na pět. Konkrétní hodnoty zvolím v průběhu implementace. Grafický vzhled kategorií zajistí CSS třída `panel` a `panel-info`.

Editační režim chci co možná nejvíce připodobnit režimu prohlížení. Uživatel tak bude mít možnost po každé úpravě vidět výsledek operace tak, jak bude vypadat při obyčejném prohlížení. Navrhl jsem dvě možnosti zobrazení editačních prvků, které nyní popíši:

1. **Kategorie uvnitř ostatních** – první variantou je nechat uživatele zvolit, kterou kategorii bude chtít upravit. Po výběru se kategorie mírně zvětší a jednotlivé odkazy budou doplněny o editační ikony. Kliknutím na editační ikonu odkazu se uživateli zobrazí pole, které může přímo editovat. Po každé změně sice uživatel uvidí opět toto editační okno, ale nebude mít přesnou představu, jak bude kategorie finálně vypadat, protože bude stále zvětšená.
2. **Oddělený blok** – druhou možností je vytvořit nový blok na straně od ostatních kategorií a v něm editaci provádět. Po jakékoliv změně bude uživatel ihned vidět, jak kategorie vypadá a jak se chovají odkazy v ní. Může si tak lépe rozvrhnout strukturu popisků u odkazů a ihned si jejich funkčnost vyzkoušet.

Obě výše zmíněné varianty počítají s obsluhou operací na pozadí pomocí technologie AJAX. Pokud dojde k chybě v komunikaci s databází v průběhu provádění libovolného dotazu, bude o tom uživatel řádně informován chybovou hláškou. Tato hláška bude opatřena kontaktem na administrátora a omluvou uživateli za vzniklý problém.

Přihlášený uživatel bude mít možnost z libovolné stránky přejít do sekce nastavení. Design této sekce bude, stejně jako v předchozích případech, využívat možností *Bootstrap* frameworku tak, aby byl udržen jednotný styl zobrazení. Vzhled tlačítek zajistí CSS třída `btn` a `btn-info`. Při návrhu se držím konvencí uvedených na oficiálních webových stránkách frameworku¹.

Patička

Aplikaci zakončuje patička fixně připojená ke spodnímu okraji stránky. Patička obsahuje informace o aplikaci, kontakty na autora a dynamická tlačítka sociálních sítí. Především sociální sítě jsou pro aplikaci velmi důležité. Nejlepší reklamu každého produktu dělají spokojení uživatelé, kteří se o své dojmy z aplikace mohou ihned podělit s přáteli. Patička se bude zobrazovat pouze na úvodní obrazovce, případně na stránkách přímo nesouvisejících s uživatelským účtem. Dále se patička zobrazí také v nastavení uživatelského účtu. Nebudu ji zobrazovat na osobní stránce, protože by rušila uživatele při používání aplikace. V případě potřeby bude mít uživatel možnost přejít na hlavní stránku případně na stránku s kontaktním formulářem. Kontaktní formulář nebude přímo součástí patičky, ale bude na něj v patičce uveden odkaz. Zmenšením velikosti okna se patička rozdělí na více řádků zobrazených pod sebou, opět z důvodu lepší čitelnosti na mobilních zařízeních.

¹Webové adresy: <http://getbootstrap.com/css/>, <http://getbootstrap.com/components/>

Kapitola 5

Implementace

Po představení návrhu aplikace je na čase podrobně rozebrat implementační část. Tato kapitola pojednává o vývoji finální aplikace a upozorňuje na důležité aspekty vývoje. Každá z podkapitol se zabývá několika konkrétními, logicky souvisejícími celky. Nejprve se soustředím na serverovou část aplikace, tedy na databázový systém. Dále následují podkapitoly zaměřující se na podstatné části aplikace případně na neobvyklé či jiným způsobem zajímavé implementační řešení.

5.1 Vytvoření schématu relační databáze

Aplikační a uživatelská data budou ukládána do databázového systému MySQL. Důvody výběru tohoto systému jsou popsány v kapitole 2.2.3 pojednávající právě o systému MySQL. Součástí návrhu aplikace je ED diagram, který jsem pomocí transformací převedl na model relační databáze. Kompletní schéma relační databáze naleznete v příloze B.3. Dále jsem vytvořil script, který schéma relační databáze v systému MySQL inicializuje včetně dodržení všech integritních omezení a datových typů. Tento skript naleznete spolu na přiloženém CD, jehož obsah je součástí přílohy A. Následující odstavce zachycují důležité fáze transformačního procesu.

Transformace ED diagramu na schéma relační databáze

Nejprve je třeba analyzovat vytvořený ER diagram z hlediska vztahů mezi entitními množinami. V ED diagramu se nacházejí pouze tyto typy:

1. Vztahy s kardinalitou 1:M jsou to například vztahy mezi množinami **User** a **Settings**, **Page** a **Page_meta** nebo **FolderType** a **Folder**.
2. Vztah s kardinalitou M:N, ten je zastoupen vztahem mezi množinami **Page** a **Folder**.

Uvedu příklad pro transformaci vztahu entitních množin s kardinalitou 1:M. Vztahy transformuji do schématu relační databáze vytvořením dvou databázových tabulek, tabulky **Settings** a **User**. Tabulka zastupující entitní množinu s kardinalitou vztahu M, tedy tabulka **Settings**, bude obsahovat sloupec s cizím klíčem (FK). Cizí klíč bude obsahovat primární klíč (PK) druhé tabulky s názvem **User** zastupující entitní množinu s kardinalitou vztahu 1. Tímto řešením dosáhnou jednoznačné závislosti mezi oběma tabulkami. Nyní tedy pro každý jeden záznam v tabulce **User** existuje M záznamů v tabulce **Settings**. V tomto

konkrétním případě, ale i ve všech ostatních, může být hodnota M rovna libovolnému nezápornému celému číslu.

Vztah s kardinalitou $M:N$ u entitních množin **Page** a **Folder** jsem transformoval následujícím způsobem. Prvním krokem bylo vytvoření vazební tabulky s názvem **PageToFolder** obsahující sloupce dle atributů entitní množiny **PageToFolder** a navíc dva cizí klíče tabulek **Page** a **Folder**. Opět je zde zajištěno jednoznačné logické spojení mezi záznamy skrze právě jednu vazební tabulku, přičemž samotné spojení obsahuje atributy množiny **PageToFolder**. Hodnota M i N mohou nabývat hodnot libovolných nezáporných celých čísel.

5.2 Struktura aplikace

Struktura aplikace vychází z balíku *SandBox*, který je součástí *Nette* frameworku. Obsahuje tyto hlavní adresáře:

- **/app** – Hlavní adresář aplikace. Dle modelu MVP (viz. 2.2.2) obsahuje modelovou, prezentační a pohledovou vrstvu.
- **/lib** – Adresář s použitými knihovnami a frameworky.
- **/log** – Tento adresář slouží k uchování záznamů o chybách aplikace
- **/temp** – Dočasný adresář obsahující cache paměť *Nette* frameworku a zkompileované šablony
- **/www** – Prezentační adresář do kterého jsou pomocí souboru **.htaccess** přesměrovány veškeré požadavky na aplikaci. Dále tento adresář obsahuje soubory JavaScriptu ***.js**, CSS ***.css** a **index.php**.

Aplikace byla z počátku vyvíjena na lokálním stroji, ale s potřebou testovat uživatelské rozhraní jsem ji v lednu roku 2014 zpřístupnil na webové adrese <http://start.kadletz.cz/>. Tento krok výrazně přispěl k získání zpětné vazby od prvních uživatelů, tedy lépe řečeno, od prvních testerů. S pokročilou fází vývoje vznikla potřeba přesunout aplikaci na samostatnou doménu. V průběhu měsíce dubna roku 2014 jsem zaregistroval doménu druhého řádu:

<http://www.jdinaweb.cz/>

Na této doméně se nyní aplikace nachází a je přístupná široké veřejnosti.

5.3 Implementační detaily vybraných částí aplikace

Vybral jsem několik částí systému, jejichž implementační stránka je v jistých směrech zajímavá. V následujících podkapitolách podám podrobné informace o implementaci těchto částí, soustředím se na detaily a přínos k tématu.

5.3.1 Komponenty

Veškeré implementované komponenty se nacházejí v adresáři **/app/components**. Komponenty se v *Nette* frameworku s výhodou využívají pro oddělení jednoho logického celku a zpřehlednění zdrojového textu. Každá z komponent je pak aktivována tzv. *továrničkou*, tedy metodou v libovolném *presenteru* s názvem **createComponent[Nazev]**. Metoda má za

úkol vrátit vytvořený objekt komponenty, která rozšiřuje abstraktní třídu:

`\Nette\Application\UI\Control`. K inicializaci komponenty dojde až po jejím *zavolání* v šabloně pomocí makra: `{Control [Nazev]}`. Velkou výhodou je použití komponenty opakovaně v různých kontextech aplikace pouze se změnou výchozích hodnot. V následujících odstavcích představím vybrané komponenty.

FoldersControl

Komponenta `FoldersControl` slouží k zobrazení kategorií v obou režimech, tedy jak v režimu čtení, tak v režimu editačním. Při vytváření objektu předávám komponentě *životně důležité* objekty modelové vrstvy. Dále specifikuji pomocí metody `setEditMode($bool)` režim určený k vykreslení kategorií s editačními prvky. Komponenta zajišťuje zpracování *signálů* týkajících se mazání kategorie nebo změny pořadí v rámci dané stránky.

Metoda `render()` je volána až v případě, že jádro frameworku požádá o vykreslení šablony. Před tím, než se šablona začne vykreslovat je třeba naplnit proměnné šablony pomocí konstrukce `$this->template->[...]`. Metoda naplní proměnnou `$foldersArray` údaji o kategoriích, odkazech a stylu zobrazení. Má-li načíst údaje z RSS zdroje, načte je právě zde. O výpis údajů se postará hlavní šablona `FoldersControl.latte` spolu se šablonami `FoldersControlLinks.latte`, `FoldersControlNew.latte` a `FoldersControlRSS.latte`. Tyto šablony jsou pevně přiřazeny komponentě v průběhu vykonávání metody `render`. Šablony se následně postarají o správné vykreslení prvků na stránce.

AjaxFormControl

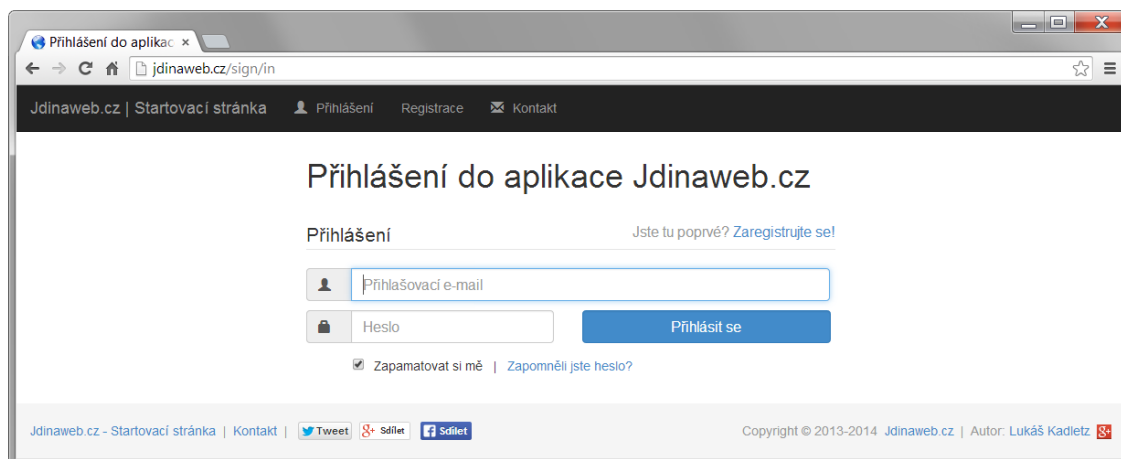
Komponenta `AjaxFormControl` slouží k odeslání dotazu technologií AJAX tam, kde je třeba tento dotaz odeslat bez interakce ze serverem pomocí JavaScriptu. Rozhodl jsem se jej implementovat jako skrytý formulář, který pomocí frameworku *jQuery* naplním a na pozadí stránky odešlu. Formulář je skryt pomocí kaskádových stylů CSS. Tuto komponentu lze použít i univerzálně, já ji používám především u změny pořadí elementů na stránce pomocí *Drag and Drop*.

SignInControl

Komponentu `SignInControl` jsem zde zařadil především pro její typické použití na kterém chci vysvětlit, proč jsem aplikaci vystavěl především s použitím komponent. Komponenta `SignInControl` vykresluje formulář pro přihlášení uživatele spolu s texty okolo něj. Komponentu používám na dvou místech:

- **Samostatný přihlašovací formulář** dostupný na adrese `/sign/in`. Zde je komponenta nastavena tak, aby zobrazovala kromě standardního formuláře i informaci o umístění registračního formuláře. Důležité je to především pro návštěvníky, kteří by na přihlašovací stránku přišli nedopatřením a nevěděli by jak dál. Také rozvržení prvků je mírně odlišné.
- **Hlavní stránka** je typickým místem k umístění přihlašovacího formuláře. Zde je formulář jednoduchý, bez zbytečných textů a mírně odlišný, než v předchozím případě.

Komponenta mimo výtisk formuláře a textů také obsluhuje celý proces přihlášení uživatele. Při vytváření objektu je komponentě skrze konstruktor předána modelová vrstva.



Obrázek 5.1: Komponenta `SignInControl` na samostatné přihlašovací stránce.

Modelovou vrstvu komponenta použije pro ověření uživatele žádajícího o přihlášení. Autentizaci uživatele zprostředkuje objekt `\Nette\Security\User` invokací metody `login()`. Pokud uživatel vložil chybné údaje, metoda vyvolá výjimku a uživatel je o této skutečnosti informován.

Největší výhoda této a dalších komponent je v použitelnosti v různých kontextech aplikace. Jsem schopen tuto komponentu použít na více stránkách. Obrázek 5.1 zobrazuje samostatnou komponentu na přihlašovací stránce. Tato konkrétní komponenta může být s drobnými úpravami použita v libovolné aplikaci postavené na *Nette* frameworku.

5.3.2 Podpůrné třídy

Podpůrné objekty jsem umístil do adresáře `/app/support`. Jedná se o objekty podporující chod aplikace. Zvolil jsem jejich oddělené umístění, protože většina z implementovaných objektů lze použít i v jiných aplikacích. Tím jsem logicky oddělil prvky závislé od nezávislých. Nyní se soustředím na některé významné třídy a podrobně je představím.

Třída `EmailSender`

Jedná se o třídu určenou pro odesílání e-mailových zpráv uživatelům. Rozšiřuje třídu `\Nette\Mail\Message`, která je součástí frameworku. Třída zajišťuje kromě samotného odeslání e-mailové zprávy také formát zprávy. Jednotlivé metody začínající řetězcem `setBody` nastaví tělo zprávy a předmět podle vložených parametrů. Mám tedy na jednom místě přístupné všechny texty, včetně patiček a podpisů a při přidání nového e-mailu je snazší udržet jednotný formát. Dále tato třída zajišťuje odeslání kontrolních e-mailů na účet administrátora. Je to především z důvodu kontroly možných chyb či nedorozumění. Administrátor si pak v případě potřeby může najít libovolný e-mail v historii. Třída používá objekt `Smtplib`, který je součástí frameworku a zajišťuje zabezpečené SSL spojení s e-mailovým serverem. Využívám server `smtp-53463.m63.wedos.net`, který je součástí hostingu.

Třída `RSSReader`

`RSSReader` je třída zajišťující korektní načtení RSS kanálu z daného zdroje. Třída podporuje tyto nejčastěji používané verze RSS: RSS 0.91, RSS 1.0 a RSS 2.0. Objekt se vytváří

přímo s volitelným parametrem `$URL`, který obsahuje adresu RSS kanálu. Ihned po inicializaci se ověří platnost adresy. Jakmile dojde k invokaci metody `parse()`, stáhnu pomocí PHP knihovny `cURL` zdrojový kód, který se pod adresou skrývá. Následují metody, které otestují, jestli se skutečně jedná o příchozí XML dokument a jestli tento dokument je RSS zdrojem. V případě úspěšné detekce RSS obsahu vrátím uživateli strukturované pole s kompletním obsahem RSS kanálu. Uživatel si také může vyžádat jen některé informace o kanálu, k tomuto účelu slouží metoda `getChannelArray`, která opět vrací strukturované informace o kanálu. V případě neúspěšného pokusu detekovat v příchozích datech RSS zdroj, je metodou `parse()` vrácena hodnota `FALSE`.

Třída Navigator

Původní myšlenka byla sjednotit navigaci na stránkách a řídit všechny odkazy v aplikaci z jednoho místa. Myšlenku naplňuje třída `Navigator`, skrze kterou tvořím všechny odkazy napříč celou aplikací včetně komponent, presenterů i šablon. Obsahuje metody zajišťující přesměrování uživatele z míst, ke kterým nemá oprávnění. Pokud tedy nepřihlášený uživatel navštíví stránku s nastavením, je již na úrovni prezentační vrstvy přesměrován na přihlášení. Tímto ošetřím zpracování formuláře vytvořeného za sezení, které již není platné. Jednotlivé řetězce zastupující adresy stránek ukládám formou konstant třídy.

5.3.3 Směrování a tvorba URL

Jedním z požadavků na aplikaci je umožnit přístup skrze soukromou URL, kterou bude každý jednotlivý uživatel vlastnit. Nechtěl jsem vytvářet adresy příliš dlouhé a nepřehledné, jako by tomu bylo při použití standardních nástrojů frameworku. Proto jsem vytvořil třídu `PageRoute`, která bude směrovat adresy pro přístup ke stránce do čitelné podoby. Třída je vytvořena rozšířením stávající třídy `Route` a implementací metody `match()`, která odpovídá za nalezení shody s adresou uživatelského požadavku. Bylo nutné, aby třída měla přístup k modelové vrstvě modelu, proto jsem implementoval metodu `setRepository` zajišťující předání správných objektů modelu.

Zde je znázorněna výsledná podoba adresy URL:

Vzor:

`http://www.jdinaweb.cz/+privateHash/pageID`

Příklady reálných adres:

`http://www.jdinaweb.cz/+cc996b780e865400/`

`http://www.jdinaweb.cz/+lukaskadletz`

`http://www.jdinaweb.cz/+ondrej/15`

Řetězec `privateHash` zastupuje jedinečnou posloupnost znaků a číslic patřící každému registrovanému uživateli. Uživatel si může původní, náhodně vygenerovaný řetězec změnit v nastavení. Před samotným řetězcem je zobrazen znak `+`, abych oddělil adresy aplikačního rozhraní od adres uživatelů. Díky tomu nedochází ke kolizím. Volitelně lze přidat na konci adresy `pageID`, tedy identifikační číslo stránky, kterou chce uživatel otevřít. Samozřejmě nelze, aby si uživatel na konci své adresy dosadil libovolné číslo, ale pouze to, které mu v sdělí aplikace. Ostatní adresy, které si případně uživatel upraví podle sebe fungovat nebudou. Takový uživatel bude přesměrován na domovskou stránku s hlášením o chybě v adrese.

5.3.4 Funkcionalita Drag and Drop

V návrhu jsem se rozhodl umožnit uživateli měnit pořadí kategorií pomocí funkce **Drag and Drop**. Použil jsem tedy plugin ke knihovně *jQuery* s názvem *Shapeshift*¹. Pomocí aplikačního API² jsem nadefinoval, které objekty se mají na stránce chovat jako kategorie a které jako kontejnery. Bylo třeba některé triggeru propojit s JavaScriptovým jádrem *Nette* frameworku tak, aby řazení a funkcionalita drag and drop fungovala i po změně stránky skrze technologii AJAX. Proto jsem v souboru `main.js` uloženého na adrese `/www/js` vytvořil funkci `editContentInit()`. Funkce je volána v prvním případě po kompletním načtení stránky a v druhém případě po úspěšném provedení AJAXového dotazu. Tímto jsem schopen zajistit správné chování pluginu v kombinaci s *Nette* frameworkem.

Součástí inicializační funkce `editContentInit()` je nastavení výchozích hodnot pluginu *Shapeshift* spolu s registrací triggeru na událost `ss-rearranged`. Tato událost se provede poté, co uživatel přemístil kategorii z jednoho místa na druhé. Pro tento případ jsem vytvořil komponentu `AjaxFormControl` popsanou v kapitole 5.3.1. Pomocí JavaScriptu naplním formulář této komponenty hodnotami nového pořadí prvků a na pozadí formulář odešlu technologií AJAX. Po zpracování požadavku se stránka nemusí celá obnovovat, změnám jen nejdůležitější části HTML struktury. Uživateli se tato operace bude jevit jako plynulá.

5.3.5 Cacheování obsahu

Abych co možná nejvíce urychlil zobrazování osobní stránky³, použil jsem cacheování obsahu. Uživatel bude na svoji osobní stránku přistupovat mnohokrát častěji než na stránku editační. Proto jsem cacheování použil právě na osobní stránce. *Nette* framework poskytuje cacheování na více úrovních, já jsem zvolil cacheování na úrovni celé šablony. Ukládám tedy kompletní obsah stránky ve formě HTML dokumentu. Toto řešení přináší výrazné zrychlení při přepínání stránek a vůbec při používání aplikace.

V aplikaci jsem také využil cacheování favicon⁴ jednotlivých webových stránek. K získání favicony jsem se rozhodl použít služby společnosti *Google, Inc.*, jelikož se jedná o výpočetně nejúspornější variantu. Jakmile faviconu získám, uložím ji v binárním formátu do cache paměti. V případě potřeby faviconu z paměti načtu, převedu do formátu `base64`⁵ a přímo vložím do atributu `src` HTML tagu `` tímto způsobem:

```

```

Celá mezipaměť aplikace je uložena v adresáři `/temp/cache`. Jednotlivé favicony se ukládají do podadresáře `_Nette.Favicon.Cache`. Zvolil jsem oddělený adresář, aby se nemíchala cache paměť šablony, frameworku a uživatelských dat.

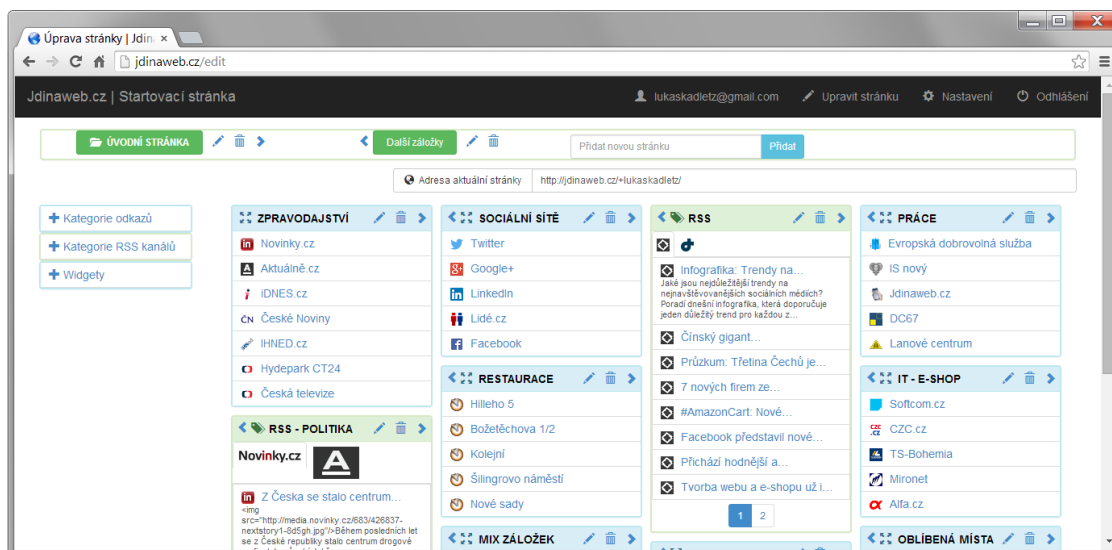
¹Adresa repozitáře na serveru GitHub.com: <https://github.com/McPants/jquery.shapeshift>

²API – Application Programming Interface (aplikační rozhraní)

³Osobní stránkou mám na mysli stránku v režimu čtení bez možnosti úprav.

⁴Favicona je miniaturní ikona určité webové stránky.

⁵Base64 je kód, kterým lze převést datový vstup na posloupnost ASCII znaků a naopak.



Obrázek 5.2: Snímek grafického prostředí aplikace v editačním režimu.

5.4 Grafické uživatelské prostředí

Grafické prostředí jsem implementoval podle návrhu. Editací režim aplikace můžete vidět na obrázku 5.2. Horní část obsahuje panel se stránkami. U levého okraje je umístěn panel pro přidání kategorií a po stisknutí ikony *pera* se nad ním ještě zobrazí editační blok zvolené kategorie. V příloze D naleznete několik dalších snímků aplikace v různém rozlišení obrazovky a také fotografii aplikace na mobilním telefonu. Také jsem vytvořil motivační video, na kterém divák uvidí grafické prostředí na různých přístrojích včetně tabletu a mobilního telefonu. Celé video naleznete spolu s ostatními soubory na přiloženém CD, jehož obsah je popsán v příloze A.

Kapitola 6

Testování a zpětná vazba uživatelů

Rozhodl jsem se provést testování vytvořeného uživatelského rozhraní, abych mohl zhodnotit, jestli je mnou navržené rozhraní dostatečně intuitivní a jestli dává smysl. Navrhl jsem tedy testy zaměřené právě na práci s rozhraním aplikace.

Testování aplikace se zúčastnilo 14 osob ve věku od 19 do 47 let. Tři čtvrtiny z nich aplikaci nikdy před tím neviděli, ostatní se s aplikací setkali již v průběhu vývoje. Před testem byla uživateli představena myšlenka aplikace a průběh testování. Uživatel měl během testování na stole před sebou manuál, podle kterého dané úkony vykonával. Přitom měl možnost v průběhu klást otázky, pokud by nevěděl, jak má test dokončit.

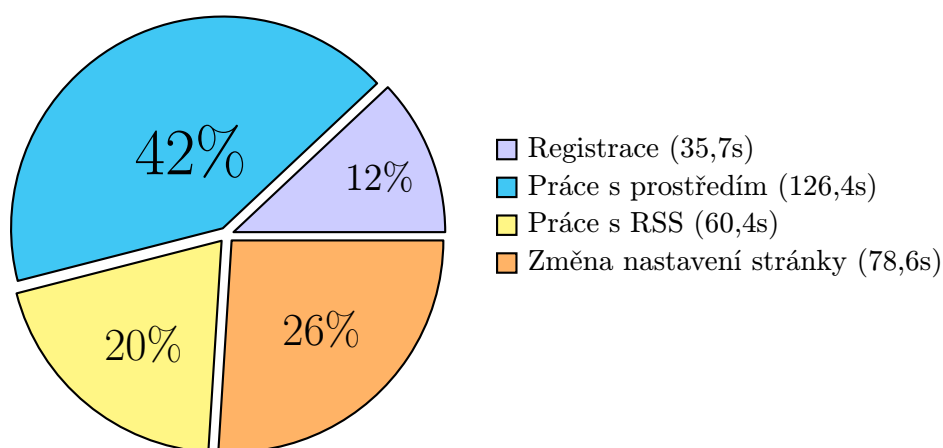
Test zasahoval do následujících oblastí:

- **Registrace** – provedení nové registrace, reakce na příchozí e-mail, přihlášení
- **Práce s prostředím aplikace** – vytváření a odebírání kategorií, práce s odkazy, změna pořadí
- **Práce s RSS** – připojit RSS kanál, nastavit parametry kanálu
- **Změna nastavení stránky** – změna atributů nastavení, změna hesla

Vypracování testů bylo doprovázeno ústním komentářem. Kompletní testovací manuál, který byl k testování využit, naleznete v příloze [C.1](#).

6.1 Vyhodnocení testů

Během testování uživatelského rozhraní aplikace jsem od uživatelů získal mnoho postřehů a námětů na vylepšení. Většina z nich zvládla všechny testy bez potřeby konzultace. Přibližně čtvrtina uživatelů se již s aplikací setkala při vývoji a dokázala tedy posoudit, jaký měla aplikace posun. Několik uživatelů upozorňovalo na horší odezvu aplikace v editačním rozhraní, kdy jim aplikace připadala zaseknutá. Dámy také zmiňovaly, že by rády provedly barevné změny v designu aplikace a přidaly fotku na pozadí. Jen přibližně polovina uživatelů zkusila kategorii přesunout tahem myši pomocí **Drag and Drop**, proto jsem do záhlaví kategorie přidal ikonu značící možnost kategorii přesunout. Upravil jsem také vzhled kurzoru tak, aby naznačoval možnost přesunutí kategorie tahem myši. Byl jsem dále upozorněn na několik nelogičností a chyb v popisích nastavení a doprovodných textů, které jsem následně opravil. Na grafu [6.1](#) je znázorněna průměrná časová náročnost každého bodu testovacího manuálu.



Obrázek 6.1: Rozložení *průměrné* časové náročnosti jednotlivých bodů testovacího manuálu.

Na základě podnětů a připomínek uživatelů jsem provedl následující úpravy:

- **Drag and drop** – přidání ikony do záhlaví kategorie, změna kurzoru při najetí myši (pouze v některých webových prohlížečích)
- **Barvy** – kategorie RSS má barvu zelenou, klasická kategorie modrou, prozatím jsem větší barevné úpravy nerealizoval.
- **Ikony místo textů** – nahradil jsem texty *Odebrat*, *Upravit* a *Přidat* odpovídajícími ikonami. Použil jsem ikony, které jsou součástí *Nette* frameworku. Přitom jsem textovou část zachoval v HTML atributu `title`, který se ve formě *bubliny* objeví nad prvkem po najetí myši.
- **Úprava textů** – Na základě upozornění od uživatelů jsem upravil popisky prvků v nastavení aplikace.
- **Optimalizace** – Snížil jsem velikost přenášených dat při vykonávání asynchronních požadavků.

6.2 Dotazníkové šetření

Vytvořil jsem dotazník obsahující několik otázek týkajících se aplikačního rozhraní a aplikace samotné. Po provedení testů jsem každého uživatele požádal o jeho vyplnění. Dotazník se skládá z následujících otázek:

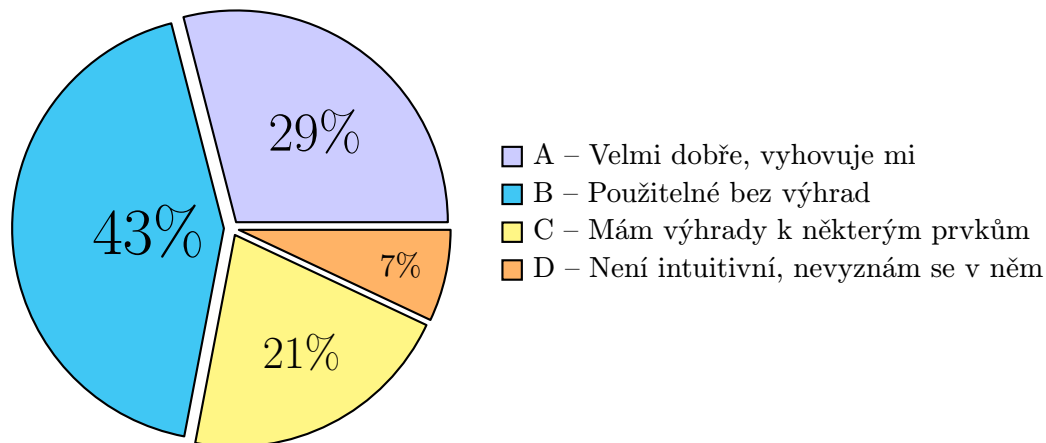
1. Jak na Vás působí rozvržení stránky?
2. Jak byste zhodnotil(a) grafické provedení aplikace?
3. Jak byste zhodnotil(a) intuitivnost ovládání?
4. Jak hodnotíte textové a grafické popisky?
5. Jak hodnotíte rychlost aplikace?

6. Využil(a) byste tuto aplikaci jako domovskou stránku na některém z Vašich zařízení?
7. Zde je místo pro Vaše sdělení autorovi aplikace.

Vzor dotazníku včetně odpovědí naleznete v příloze C.2. Dotazník byl vyplňován elektronicky skrze službu *Google Docs*.

Vyhodnocení dotazníku

Více než dvě třetiny testovaných uživatelů kladně hodnotili rozvržení stránky, jak ukazuje graf 6.2. Většina uživatelů ocenila jednoduchý grafický design aplikace, i když někteří z nich měli k některým prvkům poznámky. Ovládání aplikace označilo 79% uživatelů za intuitivní, přičemž téměř polovina z nich uvedla, že vědí, jak aplikaci použít. K textovým popiskům měli někteří uživatelé výhrady již v průběhu testování a každý takový podnět jsem ihned v aplikaci opravil. Celkově 65% respondentů označilo textové a grafické popisky za velmi zdařilé. Uživatelé v průběhu testování upozorňovali na mírně pomalejší odezvy aplikace, což se promítlo i do výsledků dotazníkového šetření. Téměř polovina uživatelů uvedla, že by aplikace mohla fungovat rychleji. Zaměřil jsem se tedy na přenášená data při provádění asynchronních požadavků a podařilo se mi nalézt a opravit chybu a zrychlit tak některé části aplikace. Grafické znázornění všech odpovědí naleznete v příloze C.3.



Obrázek 6.2: Rozložení odpovědí na otázku: *Jak na Vás působí rozvržení stránky?* (Kompletní výsledky dotazníku naleznete v příloze C.3)

Kapitola 7

Možnosti dalšího vývoje

Již během implementace této aplikace mne napadaly různé možnosti dalšího vývoje. Jak aplikaci vylepšit, jakou funkcionalitu přidat a jak případně rozšířit zhotovený návrh. Jsem si také vědom toho, že ne všechny části koncepčního návrhu uvedeného v kapitole 3.2 se podařilo implementovat. Například časová závislost zobrazovaných prvků nebo přihlašování uživatele skrze sociální sítě. Následující odstavce popisují moji vizi dalšího směřování aplikace.

Miniaplikace

V aktuální implementaci existují dvě skupiny kategorií: kategorie s *odkazy* a kategorie s *RSS zdroji*. Mám v plánu vedle této skupiny vytvořit i skupinu miniaplikací. Miniaplikace jsou samostatné bloky zajišťující konkrétní funkcionalitu, jako například *Google kalendář*, *počasí* nebo *poznámkový blok*. Startovací stránka tak dostane další rozměr v případech užití. Miniaplikace se budou vkládat na stránku podobně jako nyní kategorie. Systém zobrazování zůstane tedy stejný, změní se však obsah bloku dle zvolené miniaplikace.

Jednou z miniaplikací bude *vlastní HTML*. Zde si uživatel jednoduše nadefinuje velikost bloku a do obsahu vloží svůj HTML kód. Tato funkce jistě zajme ty uživatele, kteří již nějakou podobou stránku mají, jen si ji spravují v čistém HTML. Půjde tak jednoduše zobrazit například skupina obrázků, tabulky nebo libovolný formátovaný text. Zde je však z hlediska implementace nutné udržet blok oddělen od ostatních kategorií a povolit jen prvky nenarušující běh aplikace. Zde je na zvážení, zda povolit provedení *JavaScriptového* kódu či nikoliv. Zdrojový kód psaný v jazyce PHP interpretován jistě nebude, ačkoliv i to by bylo možné zařídit pomocí PHP funkce `eval()`.

Modulární architektura

Modulární architekturou mám na mysli systém s propracovaným API¹, který by mohli ostatní programátoři použít pro vytvoření vlastního pluginu. Plugin by pak programátor jednoduše nahrál jako balík ke své stránce a mohl si jej přidat do stránky. Aby tato architektura dávala smysl, každý takto vytvořený plugin by měl možnost ukládat některá základní data a pak s nimi operovat. Při použití na stránce by bylo možné tyto data uživatelsky měnit a přizpůsobovat vytvořený plugin vzhledově nebo funkčně. Na takto postaveném API bych vytvořil několik demonstračních miniaplikací.

¹API – Application Programming Interface (aplikační rozhraní)

Dalším krokem by bylo vytvořit sesterský web, kde by uživatelé mohli svoje vytvořené pluginy veřejně sdílet s ostatními. Sdílením by zároveň zvětšovali množství miniaplikací, které by mohli ostatní uživatelé využít. Tímto krokem bych jistě otevřel dveře kreativním nápadům a novým možnostem využití této aplikace.

Sociální síť

Plánuji aplikaci více propojit se sociálními sítěmi. Jednak na úrovni přihlašování a registrace nových uživatelů, tak na úrovni obsahu. Uživatel by měl možnost propojit si svůj účet na sociální síti s aplikací. Propojením by aplikace získala určitá oprávnění dostačující pro zobrazení kontextu sociální sítě na startovací stránce. Dokáži si představit, že s aplikací propojím svůj vlastní *Twitter*² účet a budu si místo odkazů v kategorii zobrazovat nově příchozí zprávy právě této sociální sítě.

Import a Export

Abych především novým uživatelům usnadnil přechod z klasických záložek v jejich webovém prohlížeči na startovací stránku, plánuji přidat možnost *Importu* a *Exportu*. Většina prohlížečů nabízí export uživatelských dat do formátu *XML*, jen některé mobilní prohlížeče, jako například *Dolphin Browser*, tuto možnost nemají. Uživatel vloží exportovaný soubor a aplikace jej zanalyzuje a přidá nalezené odkazy spolu s popisem na startovací stránku. Funkce *exportu* by naopak data uložila do formy čitelné pro zvolený webový prohlížeč. Výstupní soubor by pak mohl uživatel do prohlížeče vložit. Export by dále sloužil i jako záloha nebo materiál pro přesun na jiný účet.

Takové *XML* soubory bych mohl také sám vytvořit a nabídnout uživateli ke vložení přímo na stránku. Uživatel by si pak pomocí jednoho kliknutí vložil do stránky i desítky odkazů dle výběru. Zde je nutné zamyslet se nad způsobem vytváření souborů, aby bylo možné co největší množství dat uložit a následně jinde použít.

Optimalizace a změna serveru

V následujících měsících se chci zaměřit na zvýšení výkonu aplikace. Především v editačním režimu jsou odezvy aplikace vyšší, než by bylo třeba. Což vyplývá i z výsledků dotazníkového šetření z kapitoly 6.2. Odezvy aplikace v ostatních částech jsou dobré, ale i zde je místo k optimalizaci. Je možné, že je rychlost aplikace ovlivněna i ze strany poskytovatele hostingových služeb. Rozhodl jsem se změnit aktuálního poskytovatele a otestovat aplikaci na jiném serveru.

²Twitter – sociální síť určená pro sdílení krátkých zpráv. <http://twitter.com/>

Kapitola 8

Závěr

Cílem této práce bylo vytvořit webovou aplikaci pro snadnou tvorbu osobní startovací stránky. Při návrhu aplikace jsem kladl důraz na vytvoření čistého a přehledného prostředí, abych uživateli tvorbu stránky co nejvíce zjednodušil. Vyzkoušel jsem některé, myšlenkou podobné, existující aplikace a zhodnotil jejich silné a slabé stránky, jak lze vidět v tabulce 3.1. Zjištěné poznatky jsem promítl do návrhu aplikace tak, abych neopakoval chyby konkurenčních řešení. Prostudoval jsem dostupné technologie pro implementaci webových aplikací a zvolil PHP framework *Nette*, knihovnu *jQuery* a CSS framework *Bootstrap* jako základní stavební prvky celého systému.

Podařilo se mi vytvořit webovou aplikaci splňující vytyčené cíle. Uživatel si snadno vytvoří startovací stránku s oblíbenými odkazy řazenými do kategorií, připojí odebírané RSS kanály a seřadí je na stránce pomocí tahu myši. Při změně šířky obrazovky se dynamicky mění rozmístění ovládacích prvků pro co nejsnadnější ovládání aplikace. Lze ji tedy pohodlně ovládat ze všech zařízení mající webový prohlížeč a přístup k internetu. Uživatel má možnost vytvořit si vlastní URL adresu, kterou si pak může nastavit jako domovskou stránku ve svém webovém prohlížeči. Navíc si takovýchto stránek může vytvořit více a ke každé z nich přistupovat zvlášť. Po načtení uvidí startovací stránku přesně tak, jak si ji vytvořil. Stránka se uživateli zobrazí v krátkém čase, jelikož využívám cacheování a uživateli nabídnu již dříve vytvořenou statickou stránku. Vytvořil jsem také motivační video prezentující myšlenku startovací stránky a důvody, proč by ji mohli lidé s výhodou využít. Kompletní a funkční aplikace je nyní veřejně dostupná na webové adrese www.jdinaweb.cz. Po uplynutí třech týdnů od zveřejnění aplikace jsem zaznamenal celkem 26 registrovaných uživatelů.

Díky této práci se mi podařilo proniknout do světa moderních webových technologií, čehož si velmi cením. Mám v plánu aplikaci dále vyvíjet a budu se přitom snažit reagovat na podněty uživatelů. Z pohledu dalšího vývoje bych se chtěl soustředit na optimalizaci editačního režimu, možnost importu záložek z webového prohlížeče a práci se sociálními sítěmi, jak popisuji v kapitole 7. Jsem přesvědčen o smyslu a použitelnosti této aplikace a věřím, že se mi podaří rozšířit uživatelskou základnu a získat tak ještě větší osobní motivaci pro zlepšování aplikace.

Literatura

- [1] BROŽA, P.: *Programování WWW stránek pro úplné začátečníky*. Computer Press, a.s., první vydání, 2000, ISBN 80-7226-278-5.
- [2] CONVERSE, T.; PARK, J.; MORGAN, C.: *PHP5 and MySQL Bible*. Wiley Publishing, Inc., první vydání, 2004, ISBN 0-7645-5746-7.
- [3] GOLDSTEINOVÁ, A.; LAZARIS, L.; WEYLOVÁ, E.: *HTML5 a CSS3 - pro webové designéry*. ZONER software, a.s., 2011, ISBN 978-80-7413-166-0.
- [4] GRUDL, D.: Začínáme s Nette Framework [online]. *Zdrojak.cz*, 2009-03-10 [cit. 2013-12-25], ISSN 1803-5620, <http://www.zdrojak.cz/serialy/zaciname-s-nette-framework/>.
- [5] HLAVENKA, J.; kolektiv: *Výkladový slovník výpočetní techniky a komunikací*. Computer Press, a.s., třetí vydání, 1997, ISBN 80-7226-023-5, 456 s.
- [6] HOGAN, B. P.: *HTML5 a CSS3 - výukový kurz webového vývojáře*. Computer Press, a.s., 2011, ISBN 978-80-251-3576-1.
- [7] HUNT, L.: A Preview of HTML 5 [online]. <http://alistapart.com/article/previewofhtml5>, 2007-12-04 [cit. 2013-12-27].
- [8] LINDLEY, C.; kolektiv; [překlad O. BAŠE]: *jQuery Kuchařka programátora*. Computer Press a.s., 2010, ISBN 978-80-251-3152-7.
- [9] POTEL, M.: MVP: Model-View-Presenter The Taligent Programming Model for C++ and Java [online]. <http://www.wildcrest.com/Potel/Portfolio/mvp.pdf>, Taligent, Inc. 1996 [cit. 2014-04-12].
- [10] SPERBERG-McQueen, C.; BURNARD, L.: A Gentle Introduction to SGML [online]. <http://www.tug.org/applications/pdftex/gentlesgml.pdf>, 1996 [cit. 2014-02-16].
- [11] SPURLOCK, J.: *Bootstrap - Responsive Web Development*. O'Reilly, 2013, ISBN 978-1-44934-391-0, 128 s., dostupné online <http://it-ebooks.info/book/2331/>.
- [12] TATROE, K.; MACINTYRE, P.; LERDORF, R.: *Programing PHP*. O'Reilly, třetí vydání, 2013, ISBN 978-1-449-39277-2.
- [13] VIEIRA, S.: How to use HTML5's drag and drop [online]. <http://www.webdesignerdepot.com/2013/08/how-to-use-html5s-drag-and-drop/>, 2013-08-12 [cit. 2013-12-27].

Příloha A

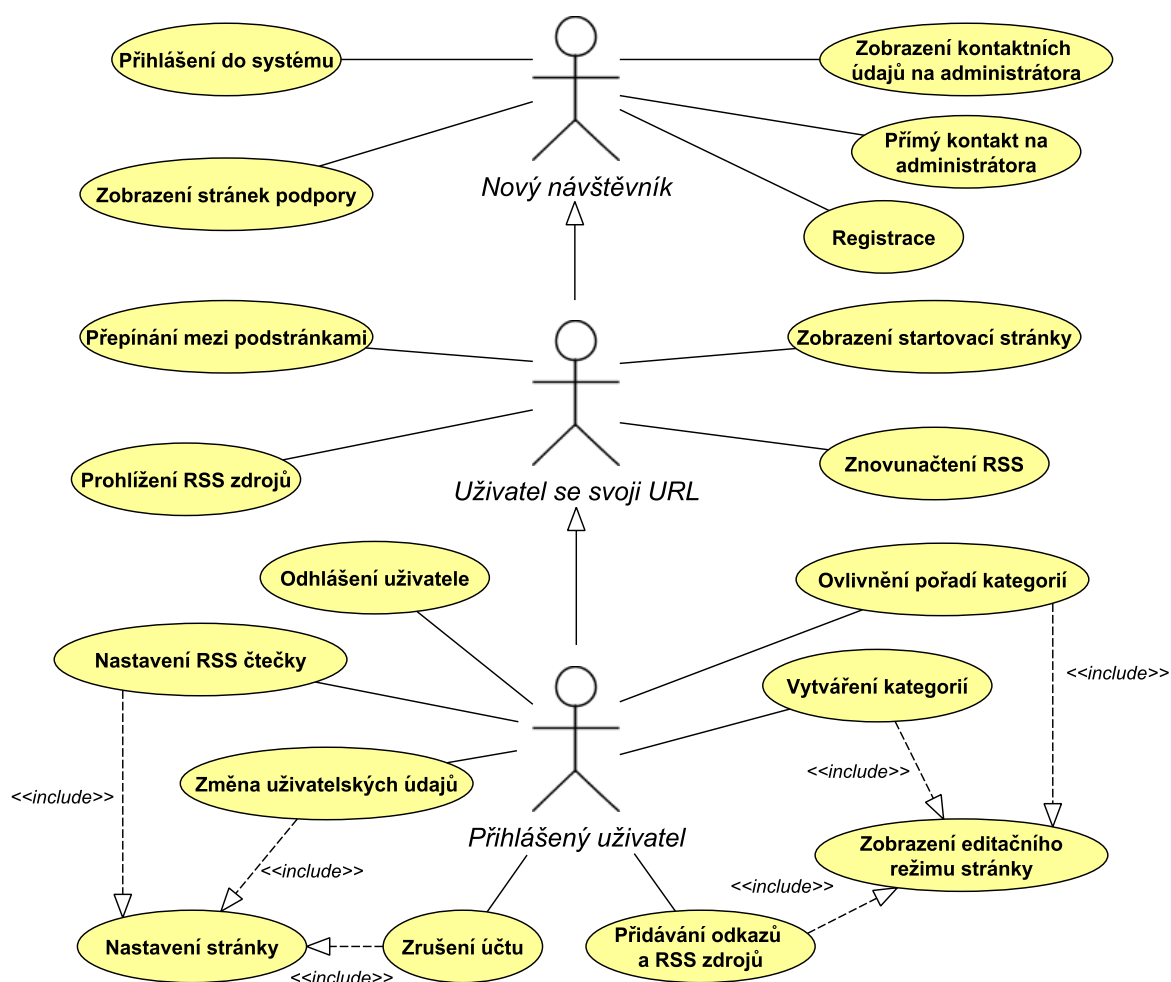
Obsah přiloženého CD

- /xkadle29_aplikace/ – zdrojové kódy aplikace, instalační SQL skript, knihovny
- /xkadle29_aplikace_prilohy/ – návrhové diagramy, fotografie, snímky aplikace
- /xkadle29_bp_zdroj/ – zdrojové kódy práce v programu L^AT_EX, obrázky a diagramy
- /xkadle29_video/
 - zdrojove_soubory/ – zdrojové soubory videa, fotografie, animace
 - 2014-xkadle29-osobni-startovaci-stranka.mp4 – motivační video, 1080p
 - 2014-xkadle29-osobni-startovaci-stranka-nahled.jpg – náhled videa
 - 2014-xkadle29-osobni-startovaci-stranka-popis.xml – XML popis videa
- /xkadle29_aplikace.zip – komprimovaný adresář ve formátu ZIP
- /xkadle29_aplikace_prilohy.zip – komprimovaný adresář ve formátu ZIP
- /xkadle29_bp_zdroj.zip – komprimovaný adresář ve formátu ZIP
- /xkadle29_video.zip – komprimovaný adresář ve formátu ZIP
- /xkadle29_osobni-startovaci-stranka.pdf – bakalářská práce ve formátu PDF
- /xkadle29_osobni-startovaci-stranka_print.pdf – tisknutelná forma práce
- /xkadle29_readme.txt – podrobný popis adresářů a souborů

Příloha B

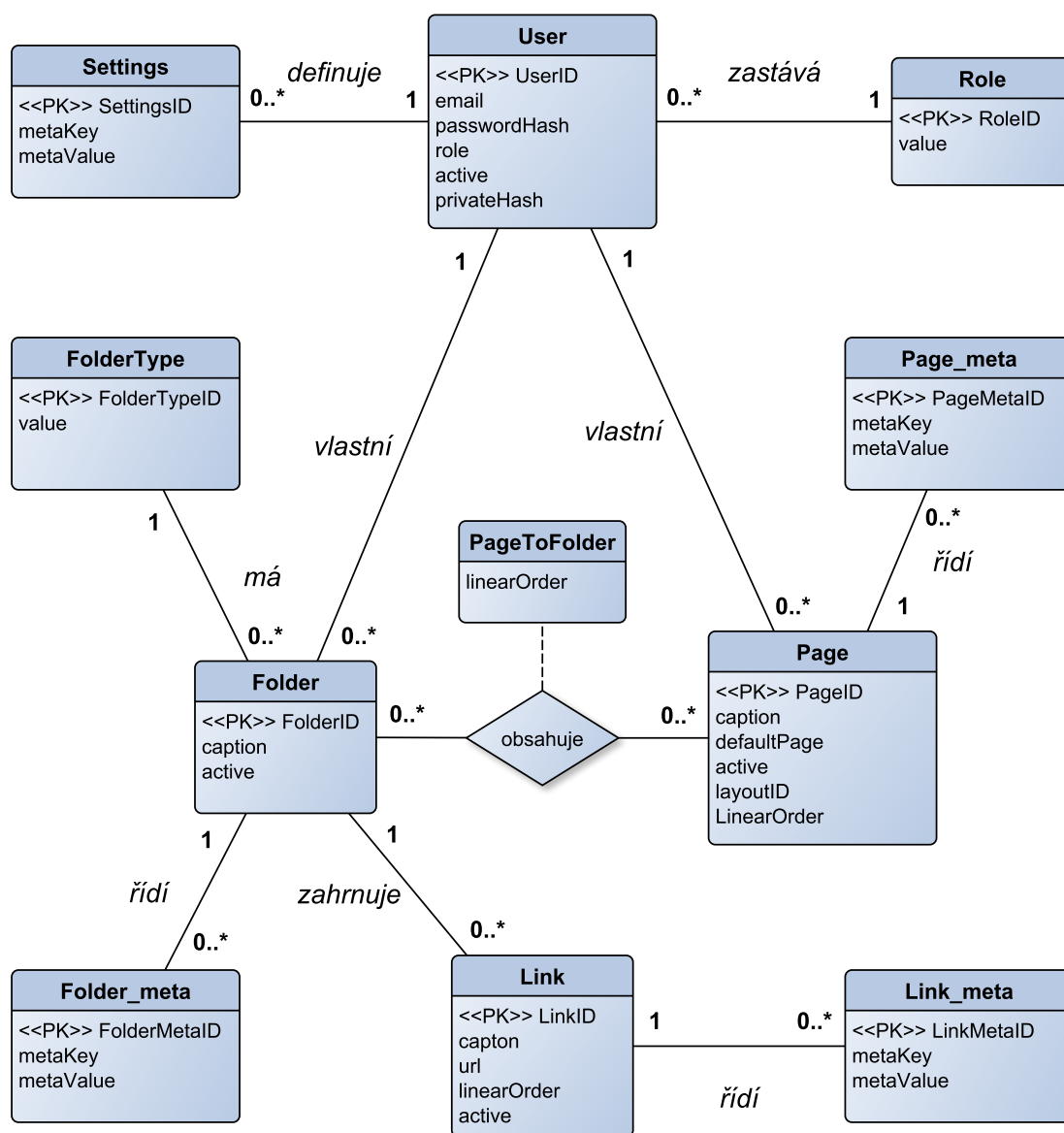
Návrhové diagramy

B.1 Diagram případů užití



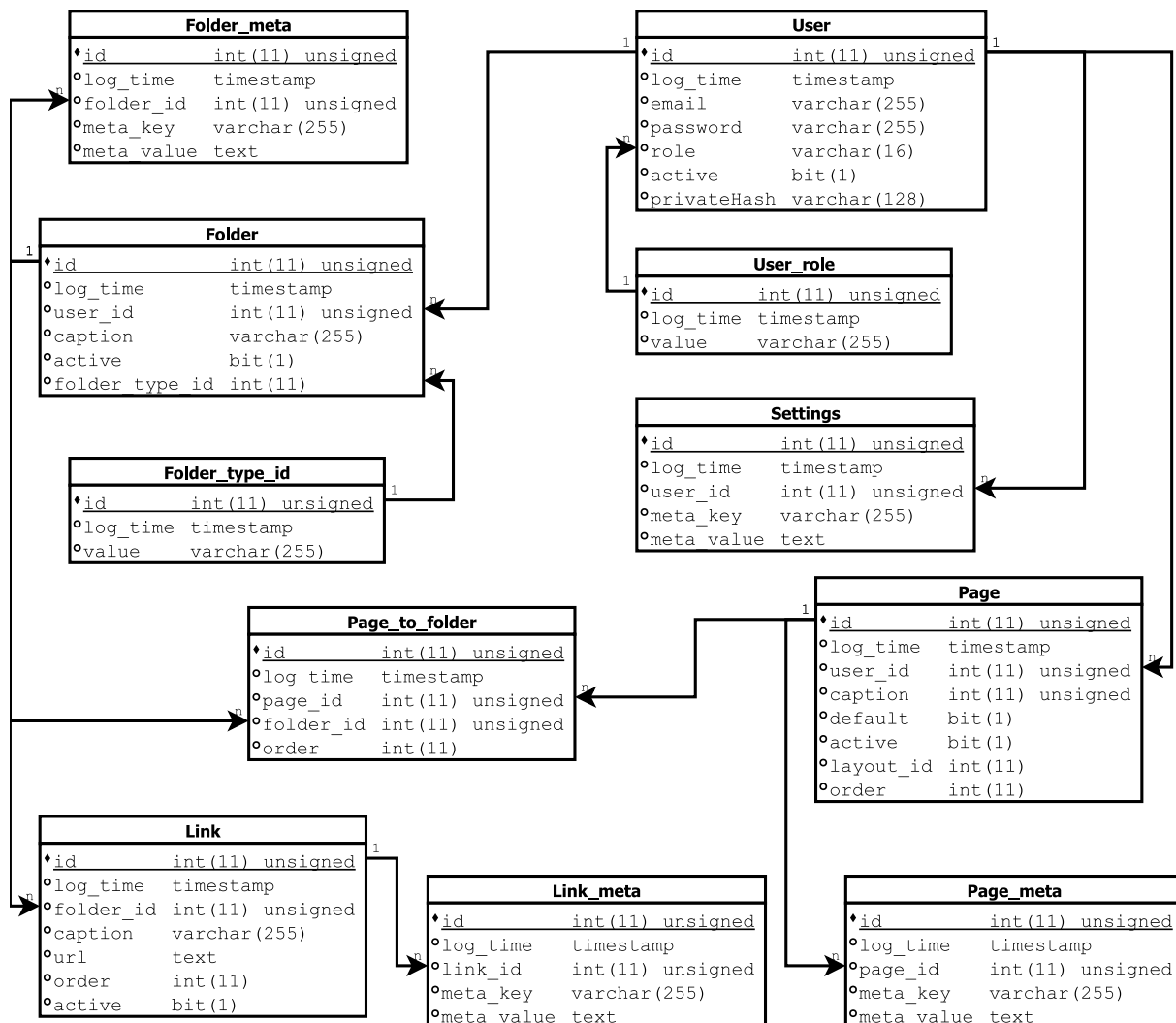
Obrázek B.1: Kompletní diagram případů užití aplikace Osobní startovací stránka

B.2 ER-diagram



Obrázek B.2: ER-diagram zachycující vztahy mezi entitními množinami aplikace.

B.3 Schéma relační databáze



Obrázek B.3: Kompletní schéma relační databáze aplikace Osobní startovací stránka

Příloha C

Materiály k testování uživatelů

C.1 Testovací Manuál

Testovací manuál dostal každý účastník testu na papír před sebe. Měl za úkol postupně vypracovat všechny body manuálu. Přitom se na mne mohl kdykoliv obrátit s otázkou. Nyní následuje kompletní testovací manuál.

Testování uživatelského rozhraní aplikace www.jdinaweb.cz

Před malou chvílí Vám byla představena webová aplikace www.jdinaweb.cz. Vaším úkolem bude provést několik po sobě jdoucích operací, kterými otestujete uživatelské rozhraní aplikace. Pokud budete mít v průběhu testování nějaký problém, nebo nebudete vědět jak dál, neváhejte mne oslovit. Rád vám s úkoly poradím. Samotný průběh je rozdělen na dvě části. V první části budete s aplikací pracovat, měnit a přidávat prvky. Ve druhé části Vás požádám o vyplnění krátkého dotazníku.

1. Registrace

- **Zadejte** do webového prohlížeče adresu: www.jdinaweb.cz a oevřete ji.
- **Zaregistrujte se.**

2. Práce s prostředím

- **Odeberte** všechny kategorie a vytvořte jednu novou s názvem **Atlantída**.
- **Přidejte** do kategorie **Atlantída** všechny tyto **odkazy** (název odkazu a adresu).
 - **Seznam.cz** <http://www.seznam.cz/>
 - **Google.cz** <http://www.google.cz/>
 - **Startovací stránka** <http://www.jdinaweb.cz/>
 - **Brno online** <http://www.brno.cz/>

3. Práce s RSS

- **Přidejte** novou kategorii RSS zdrojů s názvem **Novinky**.

- Připojte k této kategorii **RSS zdroj** ze serveru Root.cz.
 - **Root.cz** <http://www.root.cz/rss/clanky/>

4. Nastavení

- Změňte **nastavení** Vaší stránky
 - Zobrazovat **loga** RSS zdrojů místo textového popisku
 - Změňte si heslo (*klidně znovu na to stejné*)
 - Proveďte odhlášení a přihlášení

Tímto je praktické testování aplikace ukončeno. Bude následovat krátký dotazník, kde aplikaci zhodnotíte. Děkuji za Váš čas a za vyplnění dotazníku.

C.2 Dotazník

Tento dotazník dostali všichni uživatelé po provedení testů. Měli za úkol vybrat jednu odpověď, která se jim zdála nejbližší. Při podání dotazníku bylo řečeno, že pro libovolnou zpětnou vazbu slouží poslední otázka dotazníku. Respondent vyplňoval tento dotazník on-line přímo ve webovém prohlížeči.

1. Jak na Vás působí rozvržení stránky? (Odpovědi zobrazuje graf C.1)

- (a) Velmi dobře, vyhovuje mi
- (b) Použitelné, bez výhrad
- (c) Mám výhrady k některým prvkům
- (d) Není intuitivní, nevyznám se v něm
- (e) Nedovedu posoudit

2. Jak byste zhodnotil(a) grafické provedení aplikace? (Odpovědi zobrazuje graf C.2)

- (a) Graficky velmi povedené a praktické
- (b) Základní barvy, čisté, jednoduché
- (c) Najdu i nelogičnosti, ale vesměs pozitivní dojem
- (d) Rozhraní mi nevyhovuje
- (e) Nedokáži se rozhodnout

3. Jak byste zhodnotil(a) intuitivnost ovládání? (Odpovědi zobrazuje graf C.3)

- (a) Vždy jsem našel/šla co potřebuji rychle a snadno
- (b) Víím, jak aplikaci používat
- (c) Občas jsem se v aplikaci ztratil(a)
- (d) Často jsem nevěděl jak dál, nevím k čemu jsou některé prvky
- (e) Nevím

4. **Jak hodnotíte textové a grafické popisky?** (Odpovědi zobrazuje graf C.4)

- (a) Vše jak má být, velmi zdařilé
- (b) Dostačující a logické
- (c) Občas jsem nevěděl(a), co to přesně udělá
- (d) V aplikaci jsem se nevyznal, ani intuice často nepomohla
- (e) Nevím jak formulovat moji odpověď

5. **Jak hodnotíte rychlost aplikace?** (Odpovědi zobrazuje graf C.5)

- (a) Aplikace je svižná, nemám co vytknout
- (b) Někdy čekám déle, ale nic hrozného
- (c) Často nemám odezvu na provedenou operaci včas
- (d) Zdá se mi, že je aplikace zaseknutá, ale pak se najednou něco stane
- (e) Aplikace je velmi pomalá

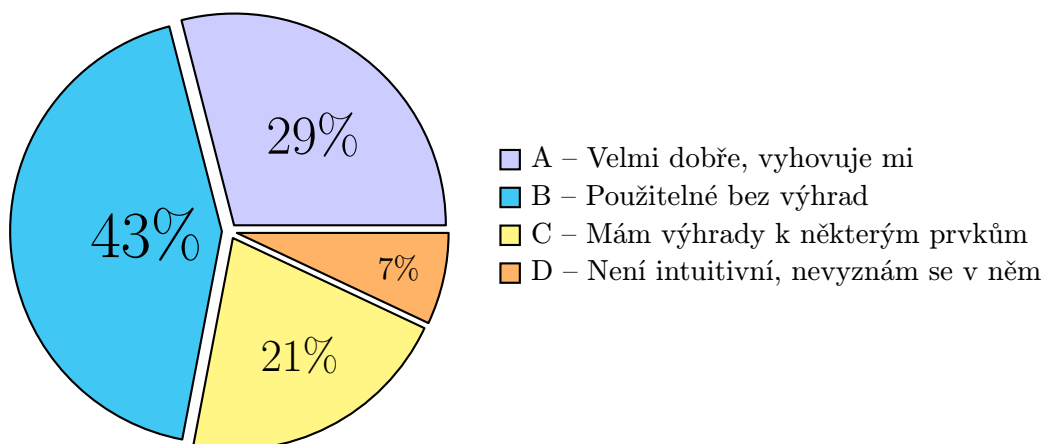
6. **Využil(a) byste tuto aplikaci jako domovskou stránku na některém z Vašich zařízení?** (Odpovědi zobrazuje graf C.6)

- (a) Určitě se zaregistruji a budu ji používat
- (b) Určitě ji zkusím, třeba se mi časem zalíbí
- (c) Nevím, jestli bych ji využil(a)
- (d) Tuto aplikaci nechci a nepotřebuji
- (e) Vůbec nevím, nemám na to názor

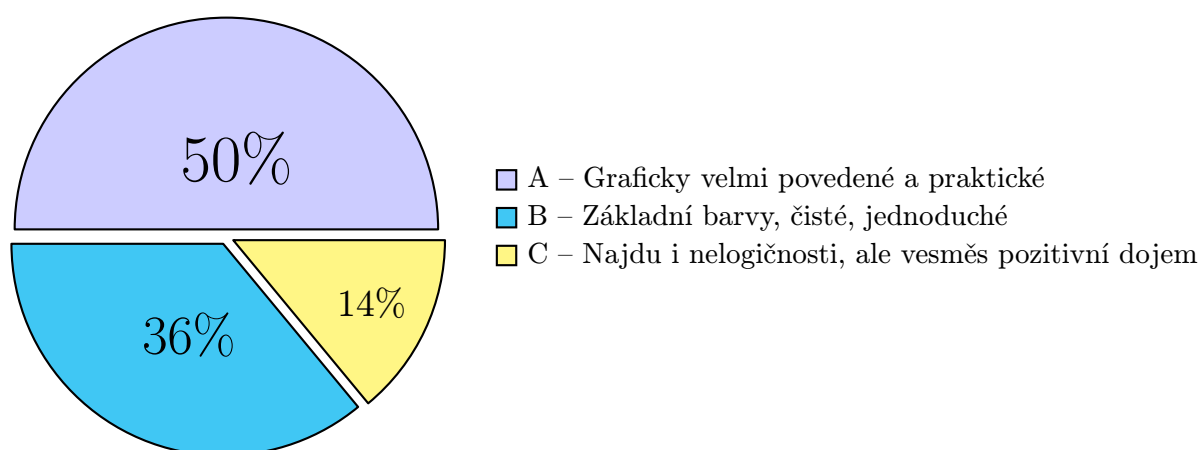
7. **Zde je místo pro Vaše sdělení autorovi aplikace.**

- *Textové pole pro zpětnou vazbu...*

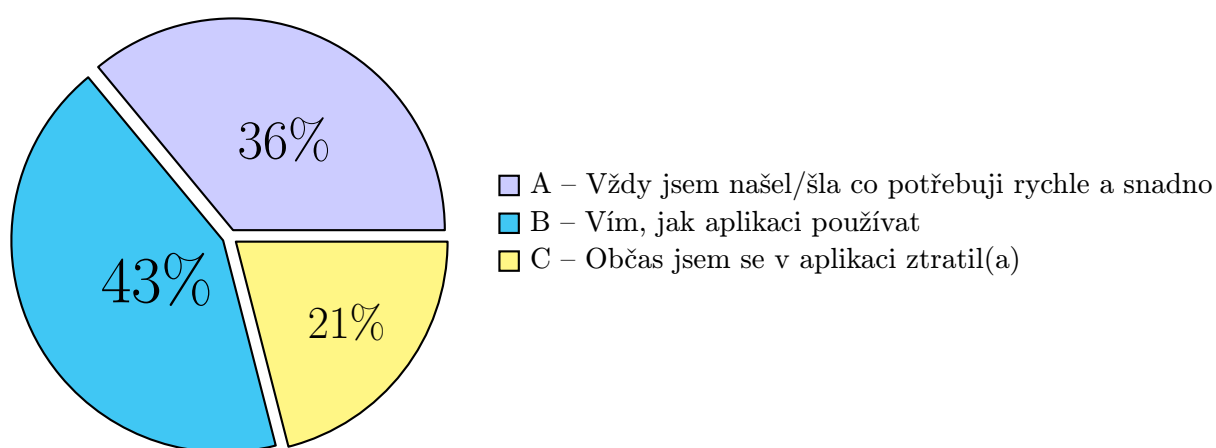
C.3 Výsledky dotazníku



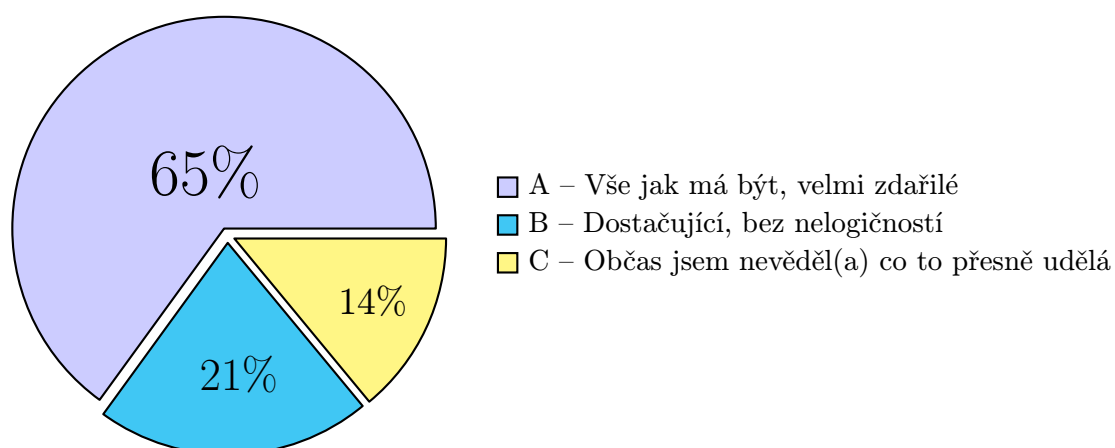
Obrázek C.1: Jak na Vás působí rozvržení stránky?



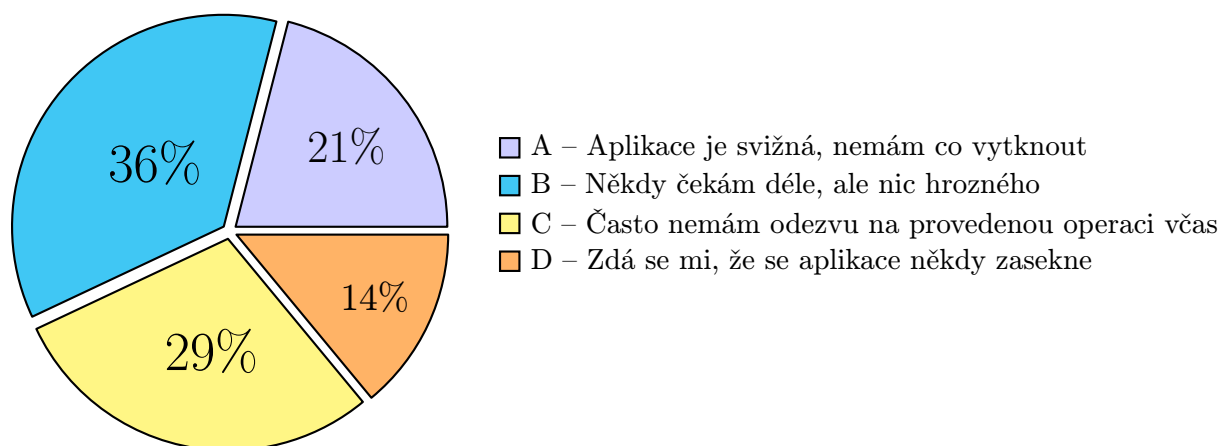
Obrázek C.2: Jak byste zhodnotil(a) grafické provedení aplikace?



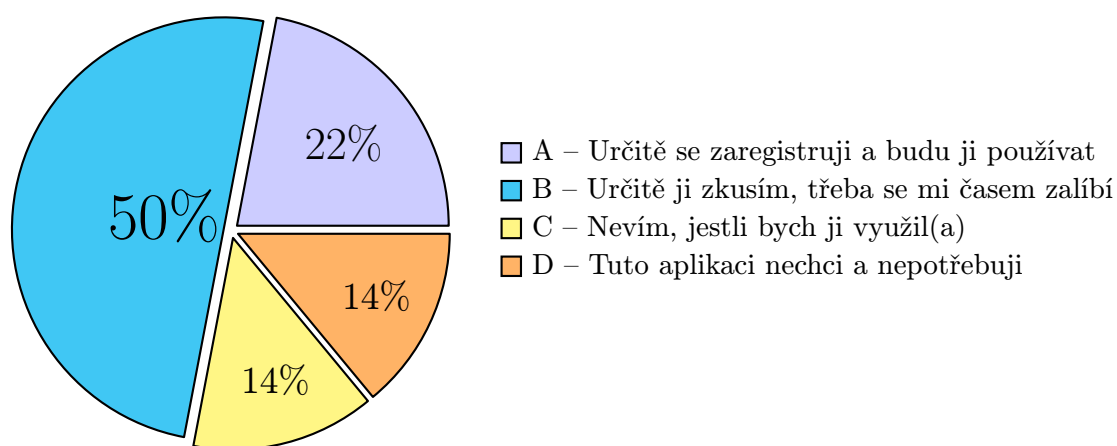
Obrázek C.3: Jak byste zhodnotil(a) intuitivnost ovládání?



Obrázek C.4: Jak hodnotíte textové a grafické popisky?



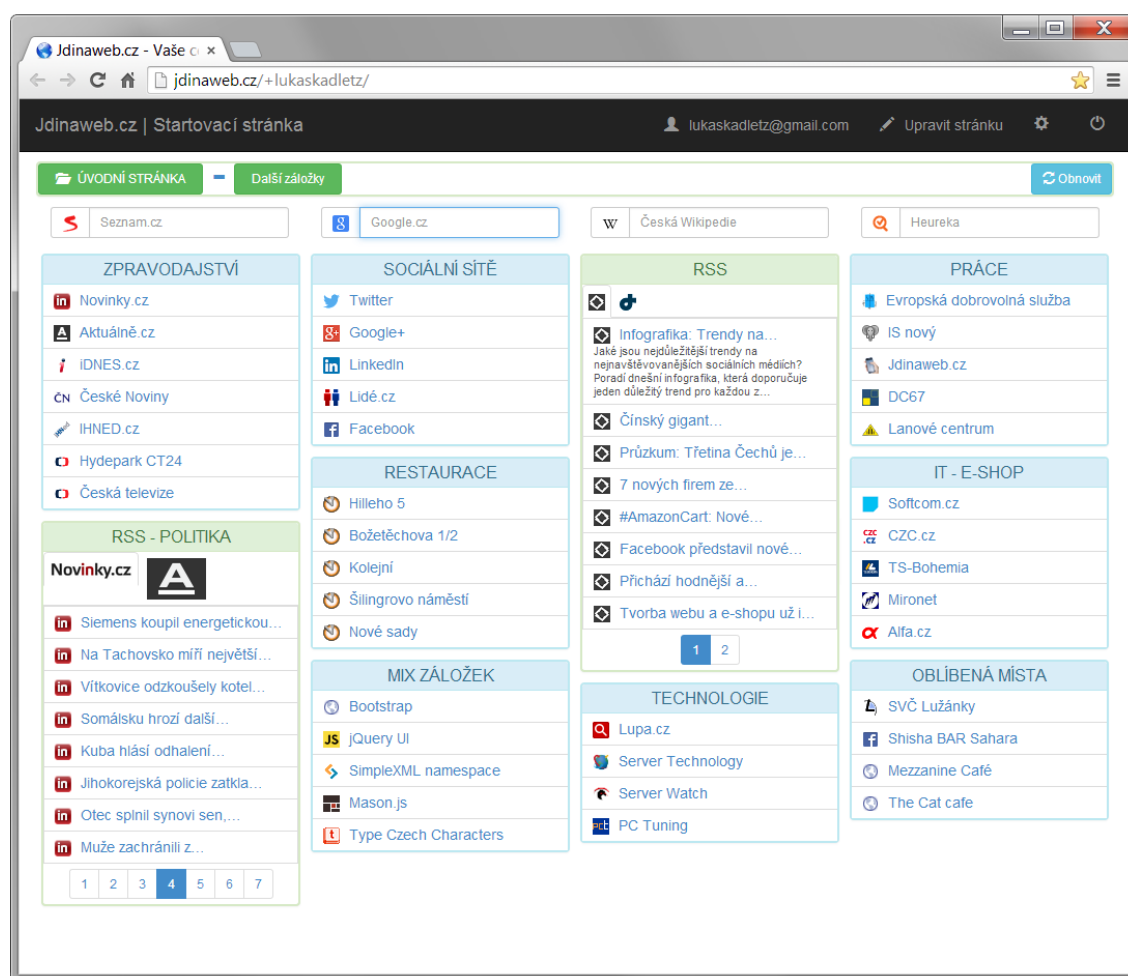
Obrázek C.5: Jak hodnotíte rychlost aplikace?



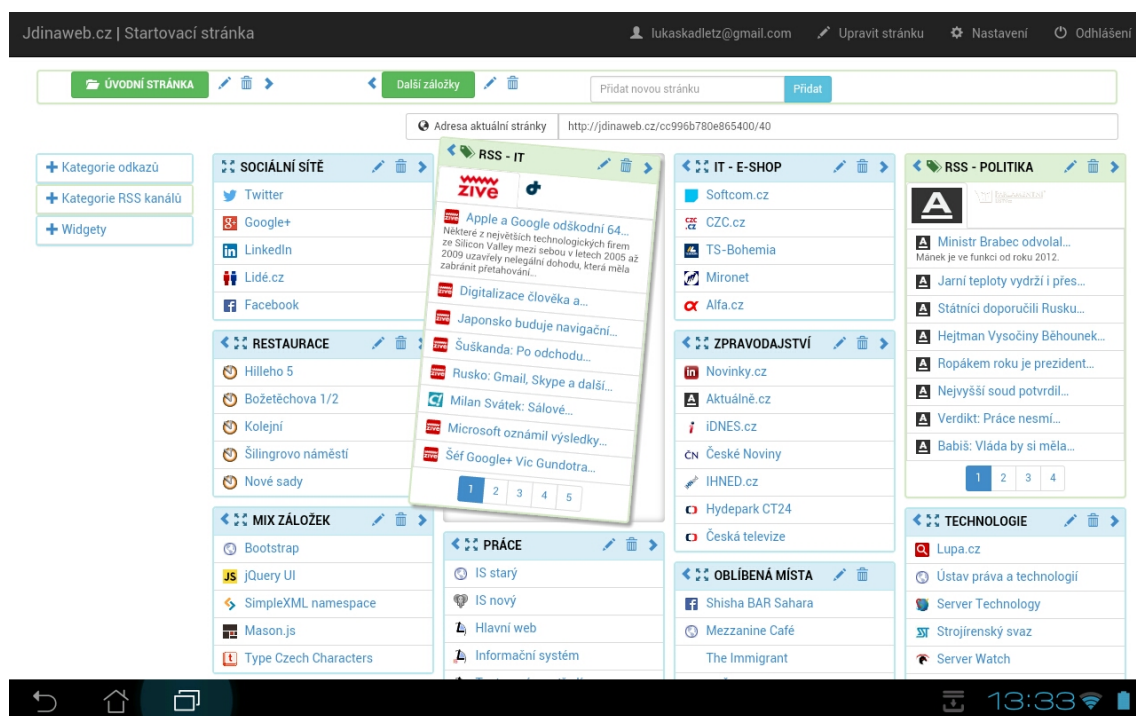
Obrázek C.6: Využil(a) byste tuto aplikaci jako domovskou stránku na některém z Vašich zařízení?

Příloha D

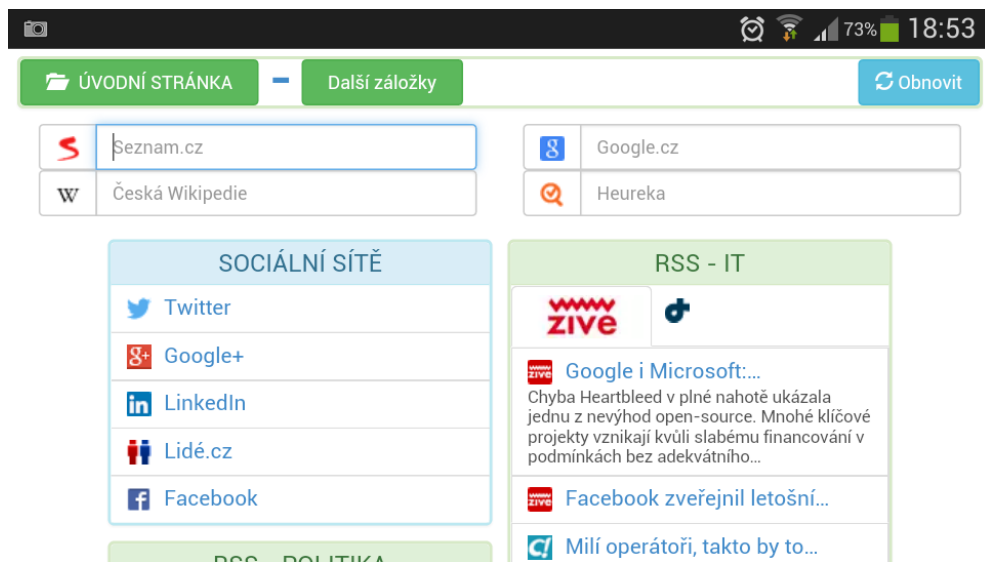
Zachycené snímky aplikace



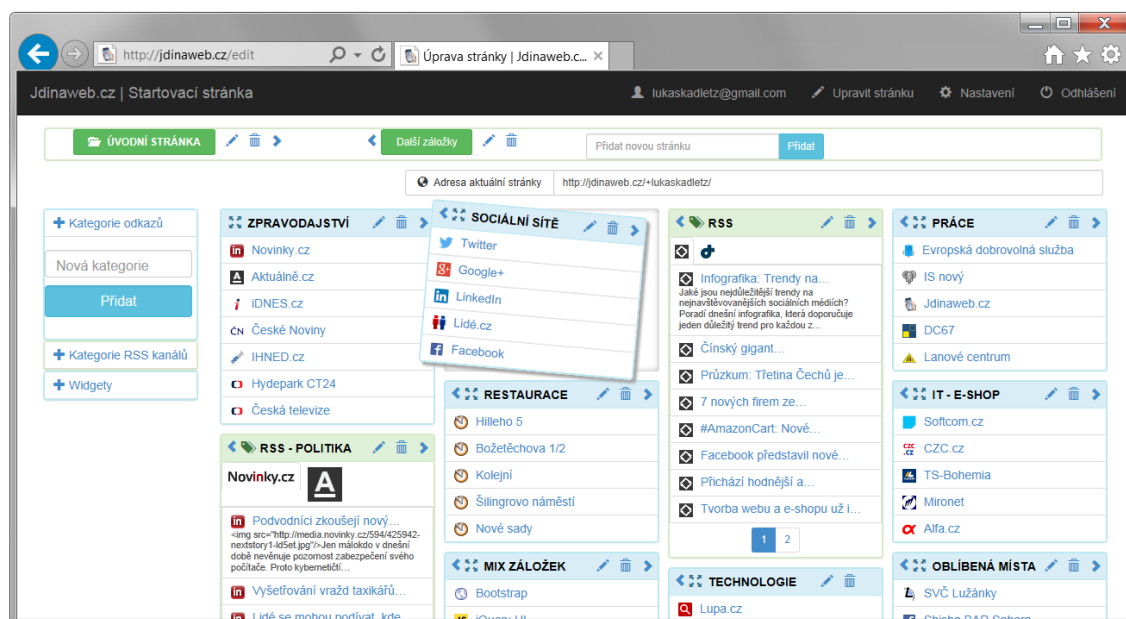
Obrázek D.1: Snímek aplikace v prohlížeči Google Chrome 34.0, rozměry obrazovky při zachycení snímku: 1070x850 bodů



Obrázek D.2: Snímek aplikace v editačním režimu při přesunu RSS kategorie. Snímek proveden na tabletu s operačním systémem *Android* s použitím webového prohlížeče Google Chrome.



Obrázek D.3: Snímek aplikace na mobilním telefonu s použitím integrovaného webového prohlížeče.



Obrázek D.4: Snímek editačního rozhraní v prohlížeči Internet Explorer 11. Snímek zachycuje přesun kategorie *Sociální sítě* na jinou pozici pomocí Drag and Drop.



Obrázek D.5: Fotografie mobilního telefonu při ovládání aplikace v mobilním prohlížeči Dolphin Browser 10.3.0.