

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Modernizace uživatelského rozhraní webové aplikace T_EXonWeb

Bakalářská práce

Vedoucí práce:
Ing. Jan Přichystal, Ph.D.

Josef Vybíhal

Brno, 2015

Stránka se zadáním práce

Děkuji Ing. Janu Přichystalovi, Ph.D. za udělenou možnost pracovat na zajímavém projektu, jeho ochotu se mnou věci konzultovat, velkou trpělivost, cenné připomínky a věcné náměty při řešení.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Modernizace uživatelského rozhraní webové aplikace T_EXonWeb**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 25. června 2015

.....

Abstract

Vybíhal, J. Modernizing the user interface of T_EXonWeb web application. Brno, 2015.

This bachelor thesis deals with analyzing, designing and implementing of a new user interface for web application called T_EXonWeb. It describes why current interface does not meet today users criteria anymore. It also describes and implements changes, that should improve user experience, even on their mobile devices. It is described what components were chosen and the way they were implemented, including reasons of choosing specific programming tools.

Keywords: Bachelor thesis, T_EXonWeb, HTML, CSS, Flexbox, Javascript, Perl, Apache, SVN, UX, UI

Abstrakt

Vybíhal, J. Modernizace uživatelského rozhraní webové aplikace T_EXonWeb. Brno, 2015

Bakalářská práce se zabývá analýzou, návrhem a implementací nového uživatelského rozhraní pro webovou aplikaci T_EXonWeb. Popisuje proč již současné rozhraní není pro některé uživatele vyhovující, navrhuje potřebné změny a nabízí ukázkou uplatnění těchto změn tak, aby uživatelé měli možnost aplikaci používat na svých mobilních zařízeních. Práce popisuje výběr použitých komponent a postup při jejich implementaci, stejně tak odůvodňuje a vysvětluje výběr programových prostředků.

Klíčová slova: Závěrečná práce, bakalářská práce, T_EXonWeb, HTML, CSS, Flexbox, Javascript, Perl, Apache, SVN, UX, UI

Obsah

1	Úvod a cíl práce	7
1.1	Úvod	7
1.2	Cíl práce	8
2	Metodika a vhodný postup práce	9
2.1	Postup	9
2.2	Vývojové prostředky a použité technologie	9
3	Současný stav	11
3.1	Průzkum v tvorbě uživatelských rozhraní	11
3.2	Analýza současného stavu	11
	Z pohledu UI a UX	11
	Z pohledu zdrojového kódu	13
3.3	Alternativy	14
	Share L ^A T _E X	14
	Overleaf	14
3.4	Motivace	15
4	Implementace	18
4.1	Návrh řešení	18
4.2	Návrhy rozhraní	18
	První návrh	18
	Druhý návrh	20
	Třetí, finální návrh	22
4.3	Textový editor, jeho výběr a implementace do rozhraní	23
	Eclipse Orion	24
	CodeMirror	24
	Ace Editor	26
4.4	Výsledná implementace	29
5	Diskuze a zhodnocení	32
5.1	Přínosy řešení	32
5.2	Nedostatky implementace	32
5.3	Kroky v budoucnu	33
5.4	Zhodnocení	34
6	Závěr	35
7	Reference	36
	Přílohy	38

A	Celé výsledné rozhraní aplikace	39
B	Výsledné rozhraní aplikace a otevřené levé menu	40
C	Výsledné rozhraní aplikace a otevřené pravé menu	41

1 Úvod a cíl práce

1.1 Úvod

Tato práce se zabývá analýzou uživatelského rozhraní webové aplikace T_EXonWeb¹ a vytvořením jeho nové modernizované podoby, která by měla eliminovat nedostatky současného stavu a poskytnout tak uživatelům lepší UX².

T_EXonWeb je webové rozhraní, které umožňuje uživatelům pracovat s DTP³ systémem T_EX, popřípadě s nadstavbami jako jsou L^AT_EX či X_YL_AT_EX. S tímto rozhraním pracují ze značné části studenti a zaměstnanci Mendelovy Univerzity v Brně, nicméně je dostupné komukoli z celého světa na veřejné internetové adrese. Nejčastější využití je při výuce systému T_EX v předmětu Zpracování textů na počítači, při psaní vědeckých článků a jako nástroj pro psaní závěrečných prací.

T_EX⁴ a jeho nadstavby fungují na principu značkovacího jazyka (podobně jako HTML⁵). Uživatel do prostého textu vloží značky, kterými definuje, jak se má s konkrétní částí dokumentu naložit. Například v jazyce T_EX značka `\textbf{tučně}` udává, že se text „tučně“ má vypsát **tučně**, podobně jako v jazyce HTML funguje tag `` a ``. Takto označovaný text předá uživatel kompilátoru, který mu ze zdrojového textového souboru vytvoří finální dokument v definovaném formátu. Obvyklé výstupní formáty jsou PostScript⁶ a PDF⁷. Dalšími výstupními formáty mohou být například rastrové formáty jako JPG a PNG⁸.

Aby mohl uživatel T_EX používat, musí mít v dané oblasti alepoň základní znalosti o fungování jazyka, mít nainstalovaný kompilátor a vhodný textový editor. T_EXonWeb je projekt umožňující využívat DTP systém T_EX/L^AT_EX na webovém serveru za použití webového prohlížeče (Přichystal, 2006, online), což přináší uživateli určitý komfort. K práci mu stačí pouze internetový prohlížeč. Odpadá mu tedy nutnost výběru textového editoru, instalace kompilátoru dokumentů a částečně mu odpadá nutnost učit se některé značky jazyka, protože T_EXonWeb nabízí šablony a nástrojovou lištu. Výhodou webové aplikace je, že se uživatel může zaregistrovat a ukládat si své dokumenty přímo na serveru, tím má vždy kopii své práce uloženou na univerzitním serveru, kde je trochu nižší pravděpodobnost ztráty dokumentu⁹.

¹Dostupné z internetu na <http://tex.mendelu.cz>

²Z anglického „User Experience“ – volně lze přeložit jako uživatelský zážitek

³Desktop Publishing – tvorba tištěného dokumentu za pomoci počítače

⁴Vytvořil Donald Knuth ve druhé polovině 70. let 20. století

⁵HyperText Markup Language

⁶zkráceně PS, jazyk vyvinutý společností Adobe, určený ke grafickému popisu tisknutelných dokumentů

⁷„Portable Document Format“ je binární přenosný formát dokumentů vyvinutý společností Adobe

⁸rastrový formát má pevně danou velikost, obvykle udávanou v pixelech a není to formát vhodný pro tisk dokumentů

⁹Toto by ovšem nikdo neměl brát jako zálohu, a každý by měl své dokumenty zálohovat s takovou mírou, jak jsou pro něj dané dokumenty cenné, mimo univerzitní server. T_EXonWeb umožňuje uložit zdrojový `.tex` soubor na disk.

K dokumentu na serveru se po zadání přihlašovacího jména a hesla dostane z libovolného počítače s internetovým připojením a prohlížečem. Dokument může rovnou upravovat, uložit a nechat si vyexportovat PDF soubor.

1.2 Cíl práce

Cílem práce je modernizovat uživatelské rozhraní webové aplikace \TeX onWeb, tak aby odpovídalo moderním standardům, bylo použitelné i na mobilních zařízeních a poskytovalo uživatelům komfort při práci s typografickým systémem \TeX respektive s nadstavbou \LaTeX . Pro mnoho studentů jsou výše uvedené vlastnosti velmi užitečné a díky nim se rozhodnou svou závěrečnou práci psát právě v prostředí \TeX onWeb (I tato práce je v něm napsána). Úmyslem této práce je zpříjemnit studentům, a všem ostatním uživatelům, jejich čas strávený ve webovém prostředí, eliminovat výtky vůči stávajícímu prostředí a vylepšit funkcionalitu tak, aby se kdokoli, kdo s aplikací pracuje, mohl soustředit více na samotnou práci a nemusel bojovat s nedostatky, kterými původní rozhraní oplývá. Podcílem je, neomezovat se pouze na desktopové webové prohlížeče, ale nabídnout možnost pracovat i na mobilních zařízeních, jejichž nárůst mezi uživateli je v posledních letech značný.

Výstupem by měla být aplikace, která vypadá moderně, využívá moderní technologie, pracuje napříč nejpoužívanějšími moderními internetovými prohlížeči a je připravena na rozšiřování funkcionality.

2 Metodika a vhodný postup práce

2.1 Postup

Jelikož je T_EXonWeb aplikace, která je již vytvořená a aktivně se používá, je nutné se nejprve důkladně seznámit s jejím zdrojovým kódem, použitými technologiemi, programovacími jazyky a porozumět zdrojovému kódu, který již byl napsán v minulosti v rámci práce na tomto projektu od mnoha programátorů. Budu muset projít jednotlivé soubory aplikace, vyhodnotit jejich funkce a zjistit nutnost jejich editace pro změny v UI¹⁰.

Dalším krokem bude nutnost seznámit se s moderními trendy v oblasti webových aplikací a vyhodnotit, zda lze na již použitých technologiích a programovacích jazycích stavět nové moderní uživatelské rozhraní, nebo bude nutné navrhnout změnu i v této oblasti.

Před změnami zdrojového kódu, uživatelského prostředí a implementací vlastního designu bude vhodné zjistit, které konkrétní věci uživatelům nevyhovují na současném stavu a vyhodnotit a navrhnout řešení pro tyto nedostatky.

2.2 Vývojové prostředky a použité technologie

Protože práce na živé aplikaci by pravděpodobně měla za následek deformaci celého prostředí pod rukama uživatelů, což není samozřejmě žádoucí, má T_EXonWeb vlastní vývojovou větev nazvanou Devel. V této větvi se provádějí a testují změny. Devel verze je verzována pomocí verzovacího systému SVN¹¹, který umožňuje například sledovat a zachycovat v čase změny v kódu a pokud se zjistí, že nějaká změna není vhodná, je možné vrátit kód zpět do stavu před změnou.

Jelikož na aplikaci momentálně pracuje team 4 vývojářů¹², je vhodné aby si každý každý vývojář mohl samostatně testovat své změny, a až je s úpravou spokojený, tak ji commitnout¹³ do vývojové větve. Z tohoto důvodu má každý vývojář na serveru svůj účet, a ve své domovské složce adresář `public_html`, díky kterému může z webového prohlížeče přistupovat ke své verzi aplikace a testovat ihned změny, které provádí. Za tímto účelem jsem nakonfiguroval modul `mod_userdir` ve webovém serveru Apache.

T_EXonWeb běží na serveru s linuxovým operačním systémem CentOS, jako webový server je využit Apache s modulem CGI¹⁴, který umožňuje spouštět skripty, které dynamicky generují obsah. Backend¹⁵ aplikace je napsaný v jazyce Perl. Pro kompilaci uživatelských dokumentů a generování výsledných PS nebo PDF doku-

¹⁰Uživatelské Prostředí (z anglického User Interface)

¹¹Zkratka pro Subversion – systém pro správu a verzování zdrojových kódů

¹²Jmenovitě Jan Přichystal, Václav Telenský, Pavel Potáček a já, Josef Vybíhal

¹³odeslání provedených změn od posledního commitu do repozitáře

¹⁴Common Gateway Interface

¹⁵serverová programová část

mentů je využita distribuce \TeX Live¹⁶. Na straně frontendu¹⁷ jsou využívány jazyky HTML, CSS, Javascript a javascriptová knihovna jQuery. Všechny jmenované frontendové jazyky jsou používány v jejich starších verzích, což je dáno historií aplikace \TeX onWeb. Při implementaci nového moderního rozhraní bude vhodné se zaměřit na jejich současné verze, hlavně HTML5 a CSS3.

¹⁶<https://www.tug.org/texlive/>

¹⁷část aplikace, která se zpracovává na straně klienta (typicky webový prohlížeč), někdy označováno CSD (Client Side Development)

3 Současný stav

3.1 Průzkum v tvorbě uživatelských rozhraní

Uživatelské rozhraní, anglicky user interface (UI), je prostředek, pomocí kterého je uživatel schopný pracovat a komunikovat s nějakým systémem. Aby byl uživatel schopný s rozhraním pracovat, tak musí splňovat určité znaky použitelnosti (anglicky usability). Takovými znaky použitelnosti jsou hlavně, jak rychle je uživatel se schopen dané rozhraní naučit a používat, jak rychle je schopen si dané rozhraní zapamatovat, jak efektivní v něm je a jestli v takovém rozhraní dělá chyby. V roce 1995 Jakob Nielsen sestavil 10 heuristických pravidel použitelnosti pro vytváření uživatelských rozhraní (Nielsen, 1995).

Každý může k použitelnosti přistupovat odlišným způsobem a rady k větší použitelnosti se mohou lišit či navzájem doplňovat, všechny ale mají společné jádro: uživatelé potřebují použitelné produkty. To je nejuniverzálnější uživatelská potřeba ze všech (Garrett, 2011).

Design uživatelského rozhraní je založený na správném výběru prvků pro úlohy, které se snaží uživatel dosáhnout a jejich rozmístění takovým způsobem, aby byly snadno pochopitelné a použitelné. Základními prvky rozhraní bývají *vstupní komponenty* (tlačítka, zaškrtnutá tlačítka, textové boxy, radio buttony, rolovací menu, atd.), *navigační komponenty* (drobečkové menu, stránkování, ikony, menu) a *informační komponenty* (ikonky, vyskakovací bubliny, ukazatel pokroku, notifikace, modální okna). Úspěšné rozhraní by mělo být takové, aby uživatel byl schopný v rychlosti rozlišit důležité a funkční prvky (Garrett, 2011).

V rozlišení různých funkčních prvků při návrhu rozhraní nám mohou výrazným způsobem pomoci barvy. Jak popisuje Jeff Johnson (2014, s. 85), naše vědomí je více optimalizováno na vnímání kontrastu, než-li na vnímání jasu, proto je vhodné při přípravě rozhraní použít protikladné barvy (například bílou a tmavě šedou).

V dnešní době používají uživatelé mnoho různých zařízení, která se liší jak svým použitím (osobní zařízení ovládaná dotyky, počítače ovládané periferiemi), tak i velikostí displaye a jeho rozlišením. Aby nemuseli vývojáři vytvářet několik verzí stejné aplikace, uchytil se v praxi trend zvaný Responsive (web)design. Jako první tento pojem definoval Ethan Marcotte¹⁸. Jde o způsob, jak psát aplikace a rozhraní takovým způsobem, aby se byla schopna přizpůsobit velikosti obrazovky na uživatelském zařízení (Marcotte, 2011).

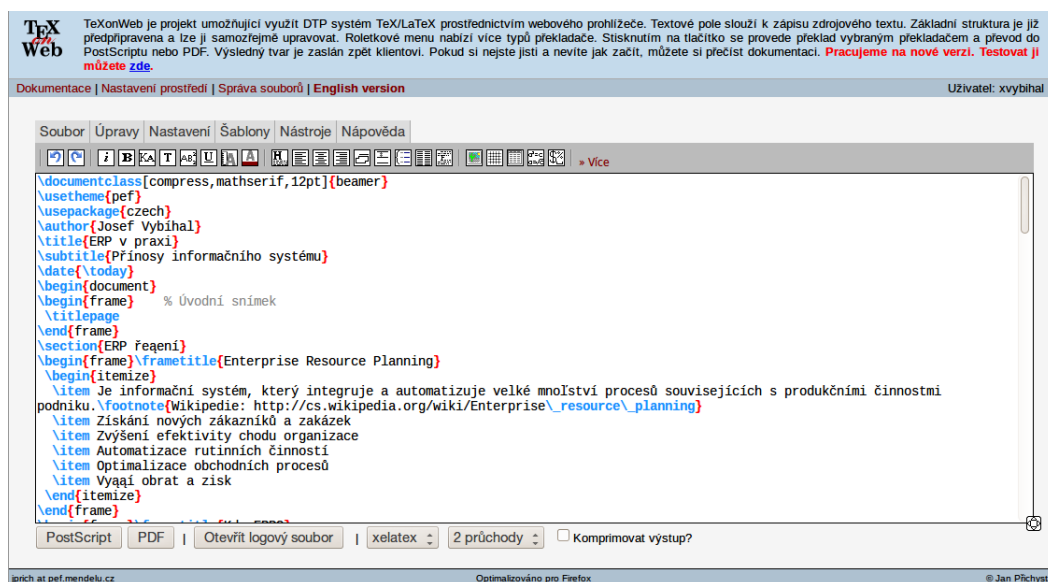
3.2 Analýza současného stavu

Z pohledu UI a UX

Na obrázku č. 1 je zachyceno rozhraní přihlášeného uživatele před změnama, kterými se zabývá tato práce. Uživatelské rozhraní aplikace se sestává z horního ban-

¹⁸<http://alistapart.com/article/responsive-web-design>

neru s logem a doprovodného textu. Banner je doplněn o odkazy na dokumentaci, nastavení, správu souborů a přepnutí do anglické verze. Pod popisovaným bannerem je souborové menu, podobně jako je tomu v desktopových aplikacích. Pod tímto menu se nachází nástrojová lišta obsahující možnosti vložení značek systému \TeX do dokumentu. Dále se v rozhraní nachází textový editor zvýrazňující syntaxi, který zabírá výraznou plochu celé aplikace a pod ním jsou tlačítka pro vygenerování PS a PDF souborů, možnost otevřít si logový soubor, zvolit překladač a počet průchodů překladače. Dole v zápatí stránky je lišta bez výrazné informační hodnoty, pouze upozorňuje uživatele, že aplikace je optimalizována pro Firefox¹⁹. Pokud se \TeX onWeb otevře například v internetovém prohlížeči Google Chrome, textový editor má zvýrazňování syntaxe zcela vypnuté. K tomuto bylo přistoupeno z důvodu chybné, či dokonce žádné, funkčnosti editoru v jiném prohlížeči, než je Firefox. Na toto bude určitě nutné se zaměřit při návrhu nového rozhraní.



Obrázek 1: Původní uživatelské rozhraní \TeX onWeb

Kromě již zmíněného problému s editorem jsou aplikaci vytýkány i další věci. Sběr připomínek k aplikaci nebyl prováděn přímým průzkumem mezi uživateli aplikace, ale byl mi tlumočen Ing. Janem Přichystalem od uživatelů, kteří s \TeX onWeb pracují v rámci předmětu Zpracování textů na počítači. Jednou z výtek je zastaralý vzhled aplikace a s tím spojené velmi těžkopádné a nepohodlné ovládání na mobilních zařízeních (náhled z tabletu Nexus 7 je vidět na obrázku 2). Menu se někdy špatně otevírá, protože jsou nápisy příliš malé, to samé platí o ikonách, které jsou velmi titěrné. Obecně je celé toto uživatelské rozhraní navrhnuté bez ohledu na mobilní zařízení a menší displaye těchto zařízení, což je dáno dobou, kdy bylo toto rozhraní tvořeno. Tehdy ještě zařízením jako tablety, iPady a smartphony s vy-

¹⁹Internetový prohlížeč od společnosti Mozilla

sokým DPI²⁰ neexistovala nebo byla ve svém raném počátku a nebylo zcela jasné, zda budou mít vůbec úspěch a šanci na rozšíření na trhu. Proto se responzivita²¹ mobilních aplikací moc neřešila.

Z pohledu zdrojového kódu

Při otevření zdrojového HTML kódu vygenerované stránky aplikace jsem podle hlavičky zjistil, že je použita specifikace jazyka HTML4, který je dnes již považován za zastaralý. Trendem je používat značkovací jazyk HTML5, který se od HTML4 liší, přidává nové elementy a atributy. Příkladem nového elementu mohou být `header`, `section`, `article`, `aside`, `footer` a příkladem nového atributu může být např. `reversed` u elementu `ol` (Goldstein, Lazaris, Weyl, 2011).

V dynamicky vygenerovaném HTML kódu ze strany serveru je možné si všimnout, že podstatná část aplikace (menu, ikony v nástrojovém panelu, editor) zcela chybí a tyto komponenty jsou generovány až javascriptovou částí aplikace. Ta se nahází v souboru `menuBeta.js` a volá se na straně klienta. Až u uživatele v prohlížeči vygeneruje „chybějící“ HTML kód a ten následně přes `document.write` zapíše do HTML stránky. Toto může být pomalé, způsobovat prodlevu v načítání stránky a z uživatelského pohledu způsobovat zadrhnutí. Dle mého názoru není důvod, proč komponenty aplikace, které se nemění, generovat touto cestou a bylo by lepší je generovat již na straně serveru, v závislosti na tom, zda je uživatel přihlášen, či nikoli. Další z věcí, kterých jsem si v souboru `menuBeta.js` všiml je, že obsahuje i definice vzhledu některých elementů na stránce, což podle mého názoru není vůbec správný přístup ke stylování elementů, ale raději by jim měly být přidělovány třídy a grafické vlastnosti těchto tříd by měly být definovány separátně v CSS souborech.

Aplikace sice využívá CSS souborů ke stylování některých elementů a několik jich je v hlavičce dokumentu referencováno, ale jejich přehlednost a hierarchie je značně sporná. Zkoušel jsem cvičně měnit vlastnosti některých selektorů pomocí vývojářské konzole prohlížeče, nicméně jsem zjistil, že se mnoho definic přebíjí (např. několikrát definovaná barva elementu na více místech dokumentu), názvy tříd jsou někdy velmi nepřesné a často z nich vlastně není ani poznat jaký mají účel a použití. Jako jeden příklad za všechny si dovoluji uvést následující část kódu:

```
.menu { /*trida obrazku img*/
  background: \#FFFFFF;
  margin: 0px;
  margin-top: 6px;
  margin-right: 2px;
  margin-left: 2px;
  width: 16px;
  height: 16px;
```

²⁰dots per inch

²¹stylování aplikace tak, aby se dokázala přizpůsobit různým velikostem obrazovky

```
padding: 0px;  
border: 1px solid #000000;  
}
```

Na obrázku 1 bylo vidět, že menu je šedé, ale zde je definice bílé barvy. Z takového kódu musel být nejeden vývojář aplikace dost zmaten a jelikož na aplikaci pracovala již řada programátorů, tak si někteří chtěli ulehčit práci a namísto studování takového CSS pekla přilili svou kapkou do ohně, vytvořili nové `id` či `class` a v něm nadefinovali vzhled dle své potřeby, namísto dodržení jednotného stylu. Našel jsem i části, kdy se někdo rozhodl vzhled vynutit deklarací `!important`, čímž se problém ještě eskaluje a kaskádový styl se stává ještě méně přehledný a dávající smysl (The `!important` exception, 2015, online).

3.3 Alternativy

Než začnu popisovat, jak řešit výše zmíněné nedostatky a implementovat nové řešení, rád bych na tomto místě krátce zmínil existující alternativy k aplikaci `TeXonWeb`. Nejedná se zdaleka o kompletní výčet, ale jde o aplikace, které jsou si s `TeXonWeb` myšlenkově podobné, či mají podobnou funkcionalitu. Srovnání těchto alternativ se blíže věnuje článek Možnosti tvorby dokumentů v `TeXu` pomocí webového prohlížeče (Přichystal, 2014).

Share \LaTeX

Aplikace disponuje čistým moderním prostředím, pokročilým editorem zdrojového kódu, správcem souborů, umožňuje soubory verzovat a spolupráci více uživatelů na jednom dokumentu. Vedle editoru, je možné zobrazit přímo náhled dokumentu, který uživatel zrovna edituje. Ve verzi zdarma není k dispozici nástrojová lišta, jako má `TeXonWeb`, takže je u uživatele vyžadována znalost značkovacího jazyka. Na tabletu je ovládání poměrně nepohodlné.

Dostupný na <https://cs.sharelatex.com>

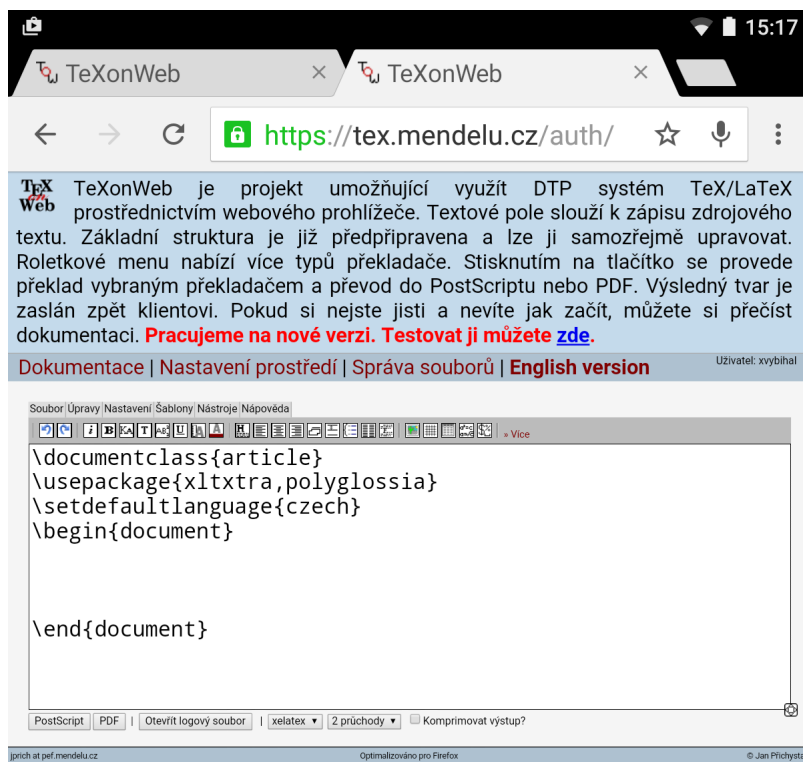
Overleaf

Dříve se tento projekt jmenoval `Write \LaTeX` . Disponuje rovněž pokročilým editorem zdrojového kódu a umí vytvářet náhledy dokumentu vedle editoru. Obsahuje jednoduchou nástrojovou lištu, která umožňuje základní úkony, jako např tučný text, kurzívu nebo (ne)číslované seznamy. Oproti `TeXonWeb` například postrádá, ve zdarma dostupné verzi, pokročilejší editor tabulek.

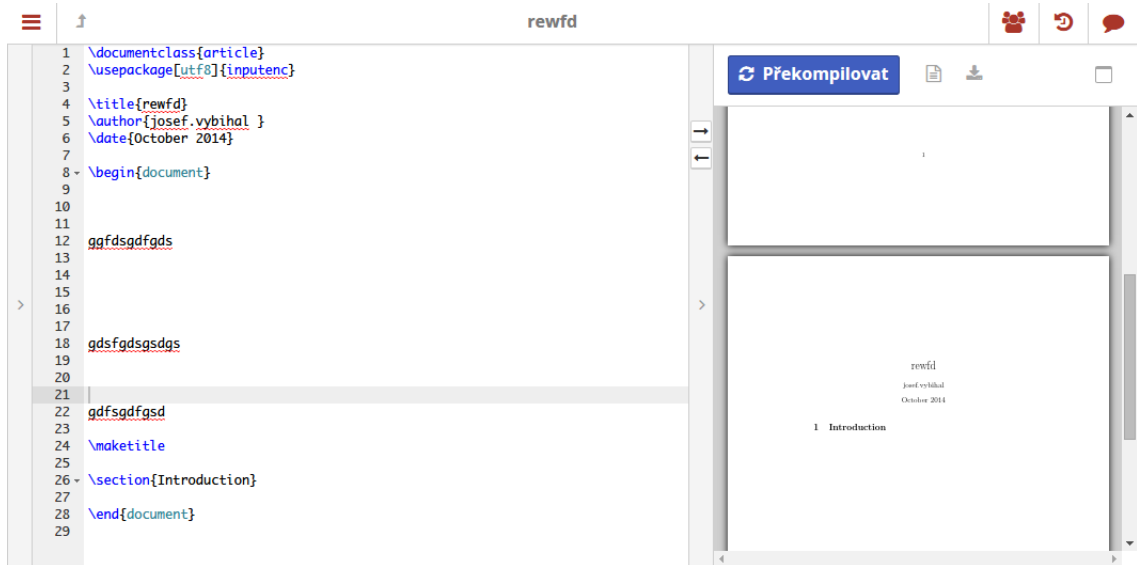
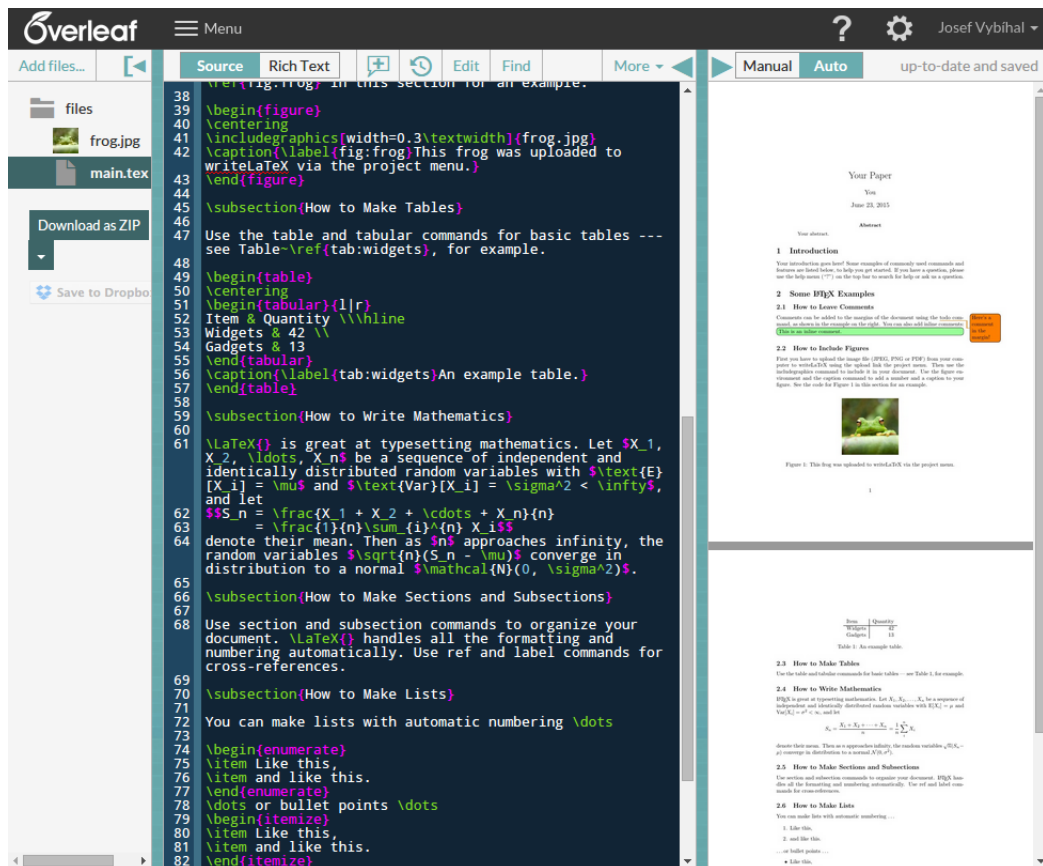
Dostupný na <https://www.overleaf.com>

3.4 Motivace

Ačkoli k \TeX onWeb existují alternativy, které se mohou zdát vyspělejší a některé z nich mají i funkce, kterými \TeX onWeb zatím nedisponuje, tak má stále smysl tuto aplikaci tvořit a vylepšovat. Na \TeX onWeb provedlo za mnoho let mnoho programátorů skvělou práci a vytvořili funkcionalitu, která usnadňuje práci jak studentům, tak i akademickým pracovníkům nejen Mendelovy Univerzity (Přichystal, 2014). Při správném přístupu lze ze zdánlivě zastaralé aplikace vytvořit moderní aplikaci, kterou budou moci její uživatelé dále využívat nejen na desktopech, ale také na svých mobilních zařízeních.



Obrázek 2: Původní uživatelské rozhraní TeXonWeb na tabletu Nexus 7

Obrázek 3: Psaní dokumentu v Share L^AT_EX

Obrázek 4: Psaní dokumentu v Overleaf

4 Implementace

4.1 Návrh řešení

Při návrhu řešení nového uživatelského rozhraní bylo nutné se rozhodnout, zda budu vycházet z již vytvořené zastaralé struktury, nebo přistoupím k vytvoření rozhraní úplně od začátku. Jak již bylo naznačeno v analýze, tak zdrojový kód původního rozhraní je značně zastaralý a dost nepřehledný na to, aby se dalo úpravami docílit potřebného výsledku.

Přistoupil jsem proto k započetí práce na novém rozhraní úplně od začátku. Jako základ pro projekt využívám HTML5 Boilerplate²². Boilerplate umožňuje projekt začít rychleji, protože seskupuje do jednoho místa zažité postupy pro přístup ke tvorbě HTML5 stránek a webových aplikací. Obsahuje také soubor `normalize.css`²³, který slouží k normalizaci prvků na stránce. To znamená, že se v jednotlivých prohlížečích všechny prvky vykreslují co možná nejkonzistentněji. Tento otevřený projekt obsahuje ověřené způsoby, jak konzistence napříč prohlížeci nejlépe dosáhnout. Tato normalizace je testována napříč následujícími internetovými prohlížeči: Google Chrome, Mozilla Firefox, Mozilla Firefox ESR, Opera, Apple Safari 6+, Internet Explorer 8+.

4.2 Návrhy rozhraní

První návrh

Při návrhu nového rozhraní jsem původně vycházel z původního vzhledu aplikace (obrázek 5). Vlevo nahoře jsem umístil logo aplikace, pod ním bylo horizontální souborové menu a pod tímto menu měla být nástrojová lišta s tlačítky. Následoval editor kódu a vedle něj tlačítka pro kompilaci a generování výstupních dokumentů. Myšlenka byla taková, že při psaní dokumentu se bude uživateli editor kódu rozšiřovat stále dolů, to znamená, že by neměl pevně stanovenou velikost, jako dříve, ale měl by velikost definovanou na základě počtu řádků v dokumentu. Uživatel by si potom pohyboval celou stránkou aplikace. Toto jsem implementoval na základě použití editoru kódu CodeMirror. Použití a výběru editoru kódu se věnuje sekce 3 této 4. kapitoly. Ukázka implementace:

```
/*
 * Vytvoreni editoru a~pocatecni nastaveni
 */
var textovePole = document.getElementById('textovePole'), // IE
    myCodeMirror = CodeMirror.fromTextArea(textovePole,
    {
        theme: "tex-on-web",
```

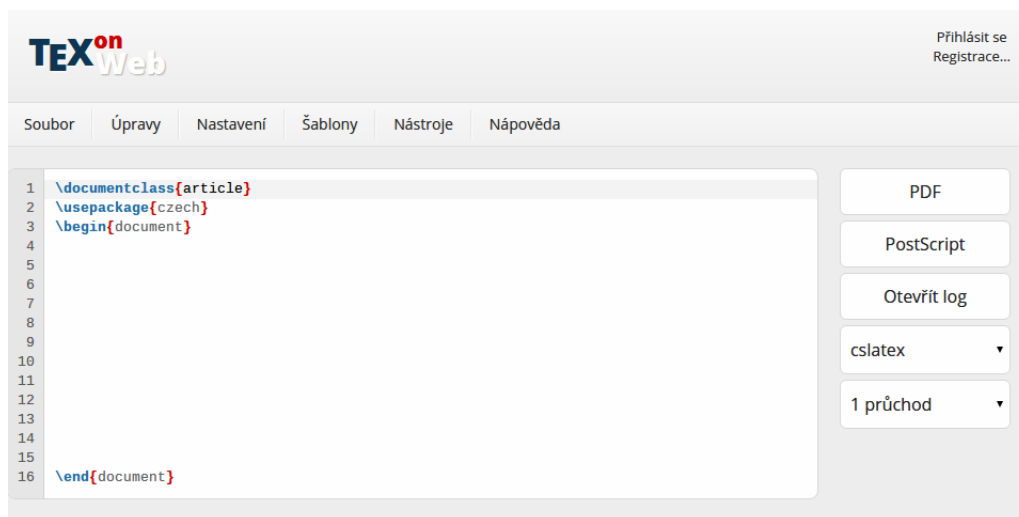
²²<https://html5boilerplate.com/>

²³github.com/necolas/normalize.css

```

        lineNumbers: true,
        lineWrapping: true,
        undoDepth: 40,
        cursorBlinkRate: 800,
        matchBrackets: true,
        findMatchingBracket: true,
        styleActiveLine: true,
        autoCloseBrackets: true
    }),
    cm = $(".CodeMirror"),
    cmStartWidth = cm.width();
/*
  Editace velikosti editoru pomoci jQuery UI
  */
cm.resizable({
  stop: function() { myCodeMirror.refresh(); },
  handles: 'e, w',
  maxWidth: cmStartWidth, //$ (this).width(),
  resize: function() {
    myCodeMirror.setSize($(this).width());
    myCodeMirror.refresh();
  }
});

```



Obrázek 5: První návrh uživatelského rozhraní

Jako návrh změny chování mi to přišlo dobré a neatřelé (nedohledal jsem podobnou aplikaci, která by využívala dynamicky se měnící velikost editoru), ale při testování, jak na desktopu, tak na mobilních zařízeních, jsme v teamu T_EXonWeb

zjistili, že tudy cesta nevede (Vybíhal; Přichystal, 2014). Pohyb v dokumentu a v celé aplikaci byl díky tomu těžkopádný, uživatel neměl potřebné nástroje k dispozici, takže musel mnohem více scrollovat, než kdy dříve. Nakonec jsme od této varianty zcela upustili a rozhraní jsem se rozhodl nasměrovat úplně jinou cestou.

Druhý návrh

Druhý návrh rozhraní nevyužívá žádné části z prvního návrhu a namísto toho, aby vytvářel stránku, která se stále rozšiřuje, tak svou velikost přizpůsobuje velikosti okna prohlížeče. K tomu využívám Flexbox, což je nový efektivní způsob rozložení objektů na stránce, který zarovnává objekty tak, že vyplňují dynamicky místo na stránce, i když není definována přímo jejich velikost (MDN, 2015, online).

Toto chování je dle mého názoru pro naši aplikaci velmi výhodné, protože tak mohou vytvořit layout, který se bude automaticky přizpůsobovat velikosti mého okna prohlížeče. Na následujícím příkladu se pokusím demonstrovat, jakým způsobem jsem nové dynamické rozhraní pro aplikaci `TeXonWeb` vytvořil. V příkladu jsou použity generické blokové elementy `div`, zatímco v implementaci rozhraní jsou použity HTML5 elementy jako např. `nav` nebo `header`, ale myšlenka je úplně stejná.

Relevantní část HTML

```
<div id="main">
  <div id="cerny">
    <div id="cerveny"></div>
  </div>
  <div id="zeleny"></div>
  <div id="wrap">
    <div id="modry"></div>
    <div id="zluty"></div>
  </div>
  <div id="fialovy"></div>
</div>
```

Kompletní CSS

```
html, body { height: 100%; margin: 0; }
```

```
#main {
  height: 100%;
  display: flex;
  flex-direction: column;
}
```

```
#cerny {
  background-color: rgb(40,40,40);
  height: 50px;
```

```
}  
#cerveny {  
  background-color: red;  
  width: 10%;  
  margin-left: auto;  
  height: 100%;  
}  
  
#zeleny {  
  background-color: rgb(95, 215, 95);  
  height: 50px;  
}  
  
#wrap {  
  height: 100%;  
  display: flex;  
  flex-direction: row;  
}  
  
#modry {  
  background-color: rgb(132, 206, 239);  
  width: 80%;  
}  
  
#zluty {  
  background-color: rgb(226, 249, 85);  
  width: 20%;  
}  
  
#fialovy {  
  background-color: purple;  
  height: 50px;  
}
```

Ukázkové rozhraní je složeno z několika vnořených bloků. Hlavní blok `#main` má nastavenou vlastnost `display` na `flex`. Tím je zdefinováno, že blok `#main` je flex kontejner, a všechny položky v něm jsou tzv. „flex items“. Definicí `flex-direction: column`; je nastaveno, že se všechny položky v kontejneru budou ve sloupečku (lze chápat „pod sebou“). Následuje černý blok, jenž obsahuje červený blok. Jelikož potřebuji červený blok zarovnat doprava, nastavím mu `margin-left: auto`, to znamená, že vše vlevo se flexibilně vyplní místem, které červený box nevyužívá (pro účely demonstrace bylo nastaveno, že červený box využívá 10 % šířky stránky). Zelený blok nebylo nutné nijak speciálně upravovat. Nyní následuje blok `#wrap`. To-



Obrázek 6: Ukázka použití Flexbox

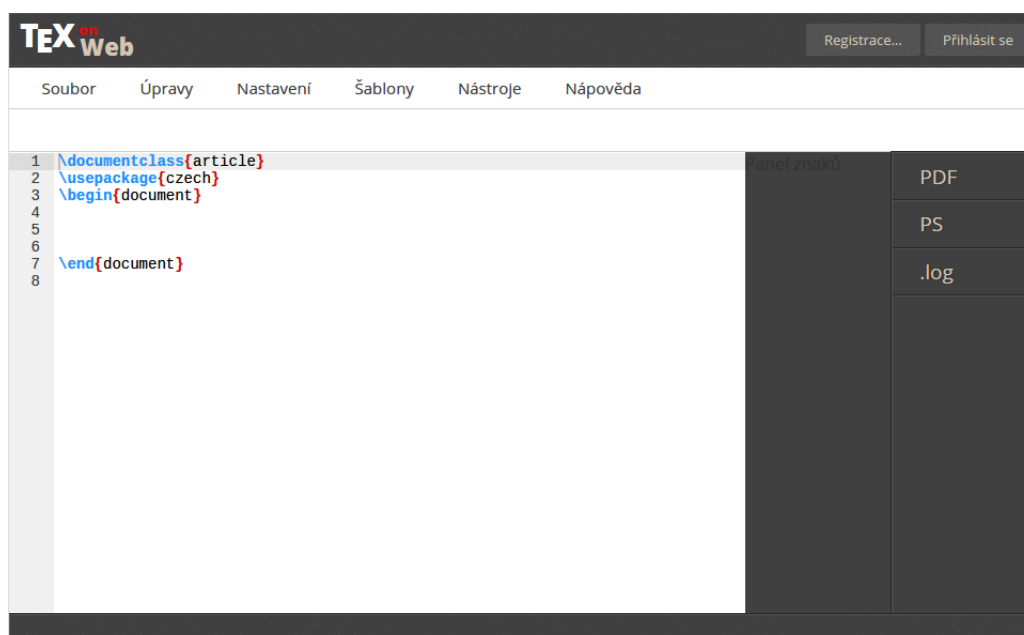
mužto bloku budu potřebovat nastavit, že se jedná o flex kontejner pomocí `display: flex`; a změním flow položek na `flex-direction: row`;. Je to z toho důvodu, abych mohl zobrazit dva bloky vedle sebe, ale nebyly vnořeny, jako černý a červený. Pro začátečníka může být toto trochu zmatečné, já jsem si na to vytvořil svou vlastní pomůcku „Položky v řádku, plynou vedle sebe. Položky ve sloupečku plynou nad sebou.“. Poslední blok je fialový, ten již není v kontejneru `#wrap`, takže je to klasická sloupečková položka, stejně jako zelený.

Druhý návrh (obrázek 7) již tedy využívá této flexibilní technologie. V horním banneru je vlevo logo a na druhé straně se nachází tlačítka pro přihlášení a registraci. Pod bannerem se nachází souborové menu a pod tímto menu je oblast připravená pro nástrojovou lištu. Pod nástrojovou lištou jsou v tomto návrhu připraveny tři sloupečky. V prvním se nachází editor zdrojového kódu, v druhém je oblast, do které bylo plánováno přidat panel často používaných specifických znaků (jako jsou např. `\`, `&`, `~` a další.) a v třetím sloupci jsou tlačítka pro generování výstupu a prohlížení logu. Dole se pak nachází úzká oblast v které by se měly zobrazovat informace týkající se dokumentu (například název otevřeného dokumentu, počet znaků, slov, atd.)

Tento druhý návrh se již přibližuje naší představě o vzhledu aplikace, ale stále má své nedostatky. Hlavním nedostatkem je zde ono souborové menu, které se na tabletu a iPadu velmi špatně rozlikává a skrývá a proto byl vytvořen ještě třetí návrh.

Třetí, finální návrh

Tento návrh z podstatné části staví na druhém návrhu. Využívá techniky Flexbox úplně stejným způsobem a hlavní změnou je odstranění viditelnosti souborového menu. Na obrázku 8 je vidět, že toto menu bylo nahrazeno dvěma ikonami. Ikona



Obrázek 7: Druhý návrh uživatelského rozhraní

trojitě čárky, předzivaná „hamburger“, otevírá jedno postranní menu, které má položky Otevřít, Šablony, Nový, Nová karta, Uložit, Soubory, log a PDF. Druhá ikona, se znakem ozubeného kolečka vysouvá vpravo druhé menu, které slouží pro nastavená překladače, počtu překladů a nastavení písma i motivu textového editoru.

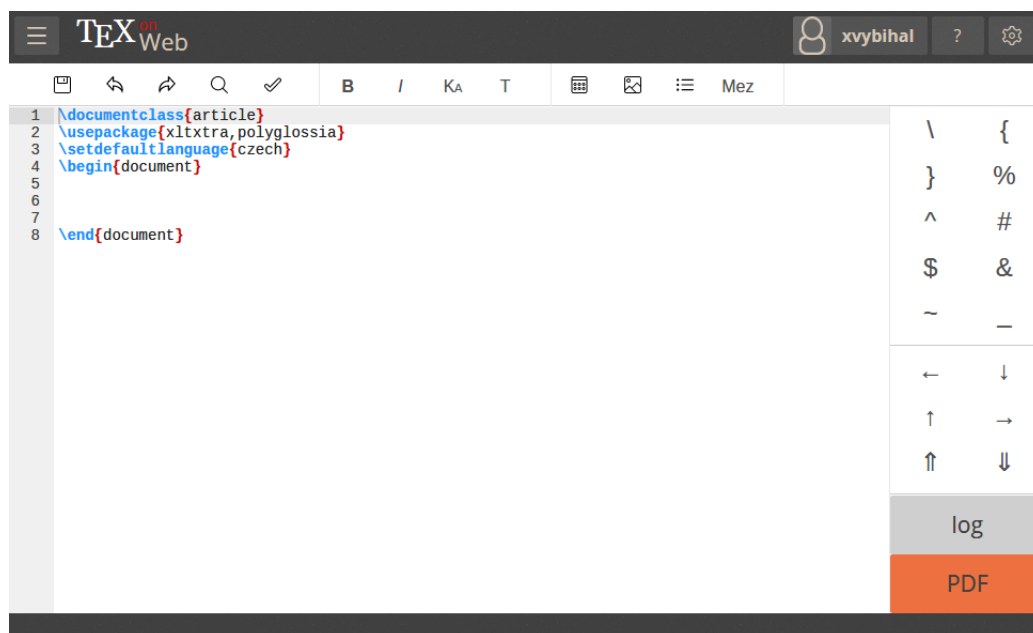
Nástrojová lišta přidává již nové viditelné ikonky. Ikonky jsou ve vektorovém formátu SVG, takže by při libovolné velikosti měly být krásně ostře vykresleny. Původní design obsahoval malé rastrové ikony, které při změně velikosti nevypadaly hezky.

Pod nástrojovou lištou se nachází dva sloupce. První širší sloupec obsahuje textový editor a ve druhém sloupci jsou přidány často používané speciální znaky, po jejichž kliknutí se vloží do dokumentu na pozici kurzoru. Pod znaky se nachází seznam šipek, ty slouží pro navigaci v dokumentu. Tato část je vhodná pro uživatele mobilních zařízení, kteří se pomocí těchto šipek mohou pohybovat po svém dokumentu. Dole se pak nachází tlačítko pro překlad do PDF a tlačítko pro prohlížení logu z překladu dokumentu.

4.3 Textový editor, jeho výběr a implementace do rozhraní

Jak již bylo dříve zmíněno, aplikace obsahuje textový editor. Editor textu je pro podobnou aplikaci velmi důležitá a poměrně komplikovaná komponenta se kterou uživatelé pracují většinu času. Při psaní dokumentu se tomu tomu nevyhnout. Proto jsem stál před volbou, co s textovým editorem udělat.

Jednou z možností bylo využít editoru, který již aplikace obsahuje. Toto byla ale velmi problémová varianta, protože tento editor se zvýrazňováním syntaxe funguje



Obrázek 8: Třetí, finální návrh uživatelského rozhraní

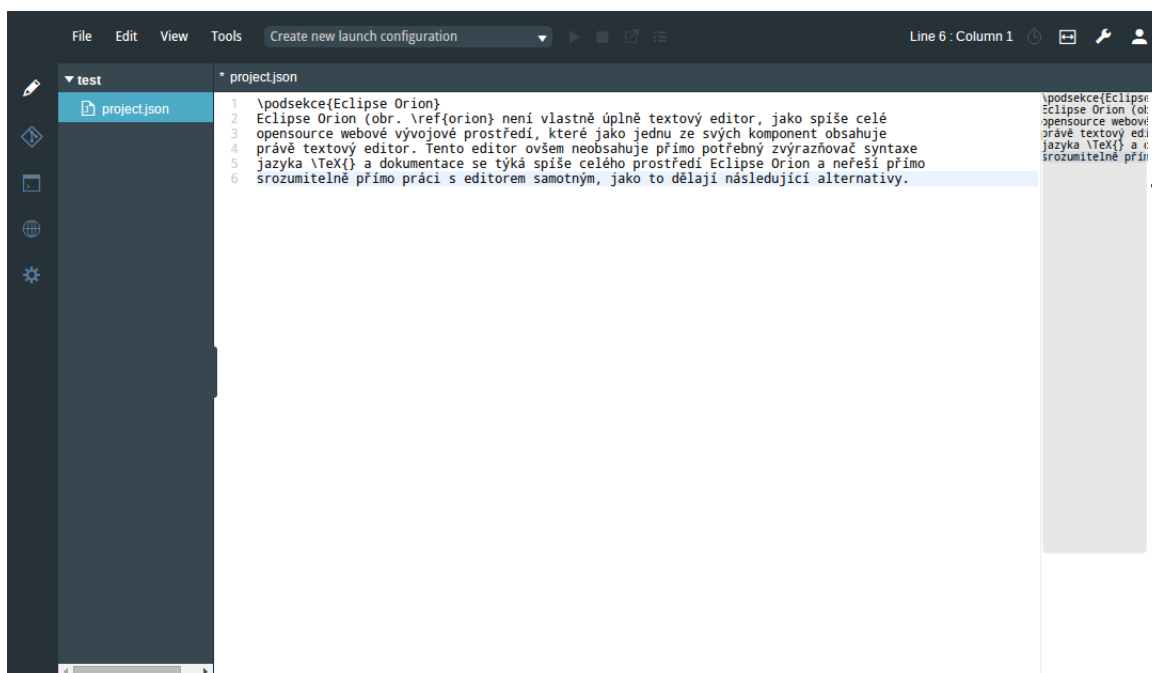
prakticky jen na Firefoxu a to ne ještě úplně bezchybně. Další věc, která hrála v neprospěch k zachování editoru byla, že nemá velmi dobře zdokumentované rozhraní pro práci s ním a kromě zvýrazňování syntaxe nemá žádné další funkce. Jelikož se práce zabývá modernizací uživatelského rozhraní, chtěl jsem uživatelům nabídnout něco víc, proto jsem udělal průzkum mezi alternativami, které by se daly namísto původního editoru použít.

Eclipse Orion

Eclipse Orion (obrázek 9) není vlastně úplně textový editor, jako spíše celé open-source webové vývojové prostředí, které jako jednu ze svých komponent obsahuje právě textový editor. Tento editor ovšem neobsahuje přímo potřebný zvýrazňovač syntaxe jazyka \TeX a dokumentace se týká spíše celého prostředí Eclipse Orion a neřeší přímo srozumitelně přímo práci s editorem samotným, jako to dělají následující alternativy.

CodeMirror

CodeMirror (obrázek 10) je samostatný univerzální editor kódu napsaný v jazyce Javascript, CSS a HTML. Podporuje zvýrazňování syntaxe mnoha jazyků, mezi nimiž je i \TeX a \LaTeX . Kromě zvýrazňování syntaxe podporuje editor i různá témata vzhledu a je poměrně snadné vytvořit i téma svoje, buď zcela od nuly a nebo modifikací některého z již existujících téma. Projekt je open-source sdílený pod MIT licenci, takže jeho použití a imodifikace v souladu s licenci jsou možné. Další ze zají-



Obrázek 9: Eclipse Orion: Orionhub

mavých funkcí a rozšíření tohoto editoru mohou být např. automatické dokončování, konfigurovatelné klávesové zkratky, schopnost přizpůsobit se obsahu, automatické odsazování obsahu nebo třeba code folding²⁴. Funkce editoru jsou programátorovi přístup přes velmi dobré a dobře zdokumentované API²⁵.

This is CodeMirror

```

1 <!-- Create a simple CodeMirror instance -->
2 <link rel="stylesheet" href="lib/codemirror.css">
3 <script src="lib/codemirror.js"></script>
4 <script>
5     var editor = CodeMirror.fromTextArea(myTextarea, {
6         lineNumbers: true
7     });
8 </script>

```

Obrázek 10: Editor CodeMirror

Vložení samotného editoru do aplikace je poměrně jednoduchou záležitostí. Stačí mít v dokumentu element `TextArea`²⁶ a poté jej tímto příkazem jsem schopný nahradit CodeMirror editorem:

²⁴schopnost skrýt část dokumentu pod vysouvací roletku

²⁵rozhraní pro práci s aplikací

²⁶http://www.w3schools.com/tags/tag_textarea.asp

```
var myCodeMirror = CodeMirror.fromTextArea(myTextArea);
```

Jak je uvedeno v podsekcí První návrh, je možné jako druhý parametr předat objekt, který může obsahovat různá nastavení editoru (např. jak rychle má blikat kurzor, zda zobrazit čísla řádků, a podobně.).

CodeMirror je bezesporu editor na velmi vysoké úrovni, svědčí o tom i fakt, že jej používají projekty jako je Firefox, Chrome, Adobe Brackets nebo třeba Bitbucket. Nicméně pro aplikaci `TeXonWeb` jsme se jej v teamu rozhodli nepoužít na základě našeho testování v mobilních zařízeních. Zjistili jsme, že trpí jednou docela zásadní vadou, která pro nás byla pro použití nepřekonatelná. Na androidových zařízeních nejsou totiž uživatelé editoru za určitých okolností schopni vůbec mazat znaky. O tomto problému existuje na Githubu projektu poměrně zajímavá diskuze (Github, `backspace doesn't work properly`, 2013, online), ze které stručně řečeno vyplývá, že jde o způsob chování klávesnice v Androidu. Při kliknutí do některého místa v editoru, se vytvoří nový (ale dočasný) element `TextArea` (takto funguje editor CodeMirror) a když uživatel začne mačkat na androidové klávesnici `backspace`, tak se ve skutečnosti vůbec nepošle signál, že uživatel `backspace` stiskl protože je `TextArea` v tomto okamžiku prázdná a Google tvrdí, že je to správný design chování, že není potřeba posílat signál, když v elementu není co mazat. Z tohoto důvodu se autoři označit problém jako `WONTFIX`.

Toto testování problému s mazáním znaků probíhalo ze strany teamu `TeXonWeb` naposledy na přelomu let 2014 a 2015. V čase kdy píší tuto práci (červen 2015) jsem vyzkoušel poslední verzi CodeMirror 5.3 a situace se zdá být odlišná. Mazání znaků mi při krátkém testu fungovalo vždy, takže je možné, že se vývoj tímto směrem posunul (testuji ale na rozdílné verzi prohlížeče a systému Android). Naopak ale zaznamenávám problémy připisované, kdy mi softwarová klávesnice bez pochopitelného důvodu zmizí a musím ji otevírat znovu.

Ace Editor

Ace (obrázek 11) je stejně jako CodeMirror samostatný a univerzální editor kódu, napsaný v jazyce Javascript, HTML a CSS. Jeho autoři o něm tvrdí, že je to, doslova přeloženo, „vysoce výkonný editor kódu pro web“. Dále tvrdí, že svou funkcionalitou se dokáže vyrovnat nativním editorům, jako jsou Sublime, Vim, a TextMate.

Původně byl Ace samostatně vyvíjený projekt paralelně s projekty Skywriter/Bespin, které vznikaly v Mozilla Labs. V roce 2010 na konferenci JSConf.eu v Berlíně se team projektu Skywriter²⁷ rozhodl, že sloučí pluginovací systém ze Skywriteru do Ace. Všechny jejich změny, které provedli, byly sloučeny do editoru Ace a ten plně nahrazuje již neaktivní Skywriter. Ace je primárním editorem webového IDE²⁸ Cloud9 IDE. Na vývoji a udržování textového editoru Ace se podílí jak vývojáři z Cloud9, tak i vývojáři z Mozilla.

²⁷<https://mozillalabs.com/en-US/skywriter/>

²⁸vývojové prostředí

```
1- /**
2  * In fact, you're looking at ACE right now. Go ahead and play with it!
3  *
4  * We are currently showing off the JavaScript mode. ACE has support for 45
5  * language modes and 24 color themes!
6  */
7
8- function add(x, y) {
9     var resultString = "Hello, ACE! The result of your math is: ";
10    var result = x + y;
11    return resultString + result;
12 }
13
14 var addResult = add(3, 2);
15 console.log(addResult);
16
```

Obrázek 11: Editor Ace

Editor vyniká opravdu velkou škálou funkcí a jsem přesvědčen, že mnoho z nich uživatelé aplikace `TeXonWeb` jistě ocení. Umí zvýrazňovat syntaxi přes 110 jazyků, mezi nimiž nechybí ani `TeX`, stejně jako `CodeMirror` podporuje i různá témata. Dále pak automatické odsazování, klávesové zkratky, má integrované vyhledávání a nahrazování, dokáže vkládat tabelátory a nebo několik prázdných znaků při stisku tabelátoru, zobrazovat zkrývání znaků, přesouvat text označením a tažením myši a samozřejmě je „vyjmi, kopíruj, vlož“ funkcionalita. Další užitečnou vlastí editoru může být fakt, že podporuje několik kurzorů najednou. Uživatel si navolí několik kurzorů a může editovat několik míst v dokumentu zároveň. Toto se musím přiznat, že je velmi návyková vlastnost na kterou jsem zvyklý z desktopové aplikace `Sublime Text`. Poslední vlastností ze seznamu na webu editoru `Ace`, kterou bych tu chtěl zmínit, je schopnost zvládat velké dokumenty, což je samozřejmě v aplikaci typu `TeXonWeb` velmi důležité, protože se dá předpokládat, že budou uživatelé editovat soubory, které mohou mít stovky až tisíce řádků. Na zkoušku jsem si zkusil v editoru `CodeMirror` a `Ace` otevřít stejný dokument, který obsahoval přibližně 5000 řádků a nastavil zvýrazňování `TeX`ové syntaxe. Zcela z pocitového hlediska na mě editor `Ace` působil opravdu svižněji a vypadá to, že autoři na optimalizaci opravdu pracovali, ale dovoluji si zopakovat, že jde pouze o pocitové hodnocení jednoho uživatele a není podloženo žádnými konkrétními měřeními.

Vložení editoru `ace` do aplikace je velmi podobné, jako vložení editoru `CodeMirror`. Následující ukázka vložení je vzata přímo z finální verze Třetího, finálního návrhu rozhraní aplikace `TeXonWeb`:

```
ace.require("ace/ext/language_tools");
var editor = ace.edit("codeEdit");
editor.setTheme("ace/theme/texonweb");
editor.getSession().setMode("ace/mode/texonweb");
editor.getSession().setUseWrapMode(true);
editor.setOptions({
enableBasicAutocompletion: true,
```

```

fontFamily: 'monospace',
fontSize: '15px'
});
editor.setShowPrintMargin(false);

```

Editor Ace ve složí namísto elementu na stránce s id `codeEdit`, poté se mu nastaví vzhled tématu `texonweb` a režim zvýrazňování syntaxe rovněž pod názvem `texonweb`. Téma a zvýrazňovač syntaxe jsou speciálně uzpůsobeny pro aplikaci `TeXonWeb` a nejsou součástí standardní distribuce a kódu Ace.

Téma `texonweb` je vzhled, který je záměrně barevně velmi podobný tématu původního editoru `aplka`, jelikož existoval požadavek, aby byl původní vzhled editoru zachován. Přidání vlastního vzhledu se dá provést následujícím způsobem:

Z githubu si stáhnou poslední verzi editoru

```
git clone https://github.com/ajaxorg/ace.git
```

Ve složce `./ace/lib/ace/theme/` vytvořím css soubor s názvem nového stylu, např. `texonweb.css`. Ve stejné složce ještě vytvořím soubor `texonweb.js` jehož obsah je následující:

```

define(function(require, exports, module) {

exports.isDark = false;
exports.cssClass = "ace-tow";
exports.cssText = require("../requirejs/text!./texonweb.css");

var dom = require("../lib/dom");
dom.importCssString(exports.cssText, exports.cssClass);
});

```

CSS soubor se stylem můžeme vytvořit zcela od nuly a nebo jako výchozí bod využít již nějaký, který Ace obsahuje. Pro vytvoření stylu `texonweb.css` byl jako základ využit styl `xcode.css` a do něj byly přidávány potřebné definice a modifikovány atributy. Následuje příklad definice vzhledu, v tomto případě vzhled zvýraznění klíčových slov:

```

.ace-tow .ace_storage,
.ace-tow .ace_keyword {
  color: #1E90FF;
  font-weight: bold;
}

```

Při testování jsme si všimli, že režim zvýrazňování syntaxe nezvýrazňuje některé symboly, které původní editor zvýrazňoval (např. `@`) a zvýrazňuje kolaté a hranaté závorky, u kterých naopak nechceme, aby byly zvýrazňovány. Proto jsem se rozhodl původní definice zvýrazňovače syntaxe, aby lépe odpovídal na-

ším potřebám. Abych neupravoval přímo původní zvyrazňovač, vytvořil jsem nový s názvem `texonweb`. Stěžejními soubory jsou `./ace/lib/ace/mode/texonweb.js` a `./ace/lib/ace/mode/texonweb_highlight_rules.js`. V druhém jmenovaném souboru stačilo upravit několik regulárních výrazů tak, aby (ne)obsahovaly konkrétní znaky.

Jelikož výše popsané úpravy byly prováděny přímo ve zdrojovém kódu editoru, zbývá ještě jeden krok, aby mohl být Ace nasazen a obsahoval naše změny. Tímto krokem je tzv. build. V prostředí, kde má takový build vzniknout je potřeba mít nainstalovaný NodeJS²⁹ a DryIce³⁰. Poté stačí spustit build pomocí příkazu `make build` a počkat přibližně minutu (v závislosti na výkonnosti počítače, na kterém build probíhá). Při buildu se kontroluje syntaxe, závislosti, přidání všech pluginů ze složky `lib` a rovněž se provádí minifikace³¹. Výsledek buildu, který nás zajímá je uložen v `./ace/build/src-min-noconflict/`. Kód je minifikovaný a má přívlástek `noconflict`, to znamená, že se v kódu používá `ace.require` namísto `require`, aby se zamezilo možné kolizi s jinými Javascriptovými komponentami na stránce.

API editoru a jeho dokumentace jsou na dobré úrovni a práce s editorem je velmi intuitivní. Například vložení znaku, případně celého řetězce, na pozici kurzoru lze provést pomocí `editor.insert('&');`

Editor Ace jsem se po dohodě s ostatními členy teamu rozhodl začlenit do výsledného nového rozhraní. Nabízí výbornou funkcionalitu, funguje napříč moderními internetovými prohlížeči, jeho vývoj je velmi aktivní³² a vývojářům lze na Githubu hlásit problémy, které pomáhají řešit. Fakt, že jde o velmi pokročilý editor potvrzuje i to, že již v základu obsahuje mnoho užitečných klávesových zkratk³³. Poslední věcí, kterou bych rád u Ace zmínil je, že netrpí problémem s mazáním znaků na mobilních zařízeních, jako jeho alternativa CodeMirror. Během našeho testování jsme neměli s vkládáním a mazáním znaků žádný problém a to hrálo nemalou roli, v našem rozhodování o výběru.

4.4 Výsledná implementace

Výslednou implementací nového rozhraní je aplikace s vzhledem popisovaným v podsekcí týkající se třetího návrhu rozhraní, založená na moderních technologiích jako jsou HTML5, CSS3, Flexbox, Ace Editor a rovněž využívá výhod poskytovaných javascriptovou knihovnou jQuery. Mezi výhody použití jQuery může patřit např. snadná manipulace s dokumentem, práce s událostmi, animace nebo Ajaxová³⁴ volání, která fungují napříč prohlížeči.

²⁹<https://nodejs.org>

³⁰<https://github.com/mozilla/dryice>

³¹sloučení zdrojového kódu a zmenšení jeho velikosti bez vlivu na výslednou funkčnost

³²<https://github.com/ajaxorg/ace/commits/master>

³³<https://github.com/ajaxorg/ace/wiki/Default-Key-Board-Shortcuts>

³⁴AJAX – asynchronní Javascript a XML

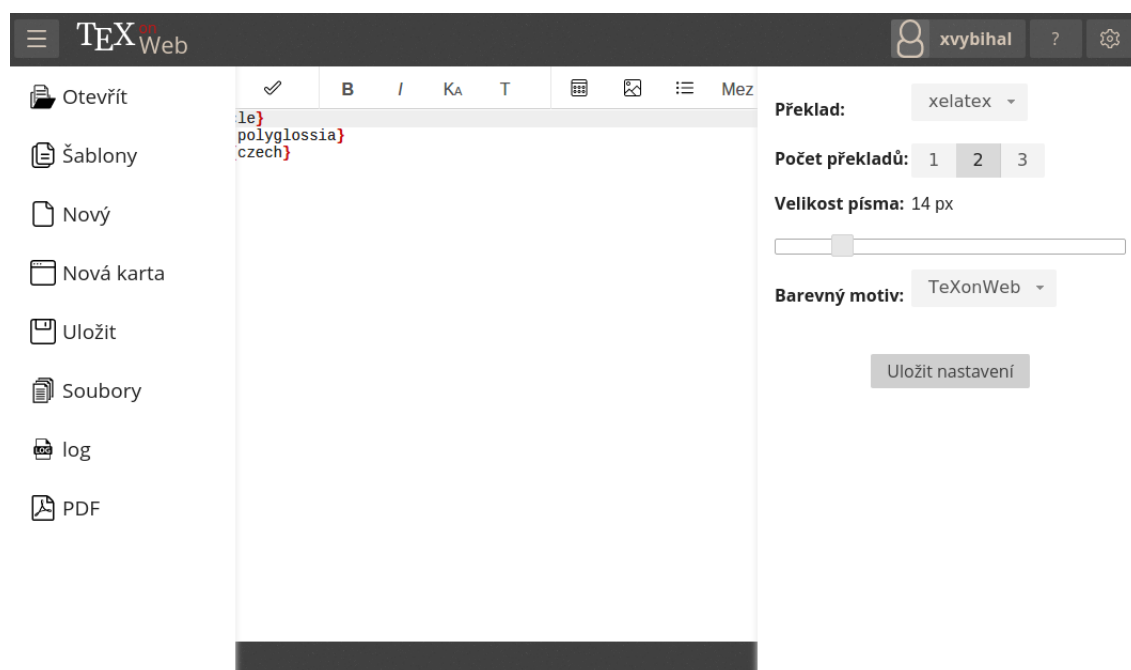
```
1 \documentclass{article}
2 \usepackage{xltextra,polyglossia}
3 \setdefaultlanguage{czech}
4 \begin{document}
5
6 Ukázka kulatých závorek ()
7 Ukázka hranatých závorek []
8 Ukázka složených závorek {}
9
10 Speciální symboly: ^ # $ & ~ _
11 Komentáře: % vše za procentem je komentář
12
13 \end{document}
14
```

Obrázek 12: Ukázka vzhledu xcode a původní zvýraznění syntaxe

```
1 \documentclass{article}
2 \usepackage{xltextra,polyglossia}
3 \setdefaultlanguage{czech}
4 \begin{document}
5
6 Ukázka kulatých závorek ()
7 Ukázka hranatých závorek []
8 Ukázka složených závorek {}
9
10 Speciální symboly: ^ # $ & ~ _
11 Komentáře: % vše za procentem je komentář
12
13 \end{document}
```

Obrázek 13: Ukázka upraveného vzhledu a modifikované zvýraznění syntaxe

Vzhled rozhraní, který byl vytvořen v rámci této práce, je vyobrazen na obrázku 14. Na obrázku jsou vidět otevřena obě menu zároveň. Pozorný čtenář si může povšimnout, že rozhraní obsahuje i ikony. Jedná se o sadu vektorových ikon (v rozhraní jsou využity jako SVG obrázky), která je volně k dispozici a k použití na adrese <http://www.flaticon.com/packs/bigmug-line>.



Obrázek 14: Hotové a implementované rozhraní aplikace TeXonWeb

5 Diskuze a zhodnocení

5.1 Přínosy řešení

Myslím si, že implementované řešení by mělo mít (a z prvotních ohlasů, např. z konference PEFnet 2014, tomu tak je) příznivý vliv na uživatelský dojem z aplikace. Oproti původnímu vzhledu aplikace \TeX onWeb působí moderněji, zobrazuje se korektně ve všech moderních kvalitních internetových prohlížečích, včetně schopnosti se automaticky přizpůsobit velikosti okna a nový textový editor Ace nejen, že správně zvýrazňuje syntaxi napříč prohlížeči, ale přináší i mnoho nových funkcí jako jsou např. změny vzhledu, klávesové zkratky, automatické odsazování obsahu, zvýrazňování párových závorek apod. Všechny tyto funkce, jakkoli se mnou zdát banální, uživateli mohou zpříjemnit práci a pozvednout komfort, který mu rozhraní nabízí.

Další věcí se kterou jsem spokojen, je fakt, že i když rozhraní aplikace prošlo velkou změnou, tak si udrželo rozmístění prvků takovým způsobem, že i neznalý uživatel nemá problém si nyní k \TeX onWeb sednout a ihned s ním začít pracovat. Z druhé strany, uživatelé, kteří s \TeX onWeb již dříve pracovali, tak nebudou cítit, že jim nutím něco těžkopádného a neintuitivního, alespoň v to doufám. Víím, že zavděčit se všem uživatelům není vůbec lehký úkol. Podněty na zlepšení ze strany uživatelů jsou pro celý team \TeX onWeb velmi vřele vítány.

Za velký přínos považuji i to, že je rozhraní mnohem lépe optimalizováno pro mobilní zařízení, hlavně androidové tablety a iPady. Editor zvýrazňuje syntaxi, menu se dají mnohem příjemněji rozkliknout a tlačítka jsou dostatečně velká bez toho, aby musel uživatel nějak zvětšovat stránku a snažit se najít vhodnou pozici do které má kliknout.

5.2 Nedostatky implementace

I když jsem s výsledkem modernizovaného rozhraní spokojený, musím zmínit některé nedostatky o kterých víím, jinak bych nemohl tuto práci považovat za objektivní.

Ačkoli je layout postavený na Flexboxu zobrazován bezproblémově na moderních desktopových prohlížečích, včetně Internet Exploreru 11, tak na mobilních zařízeních s operačním systémem Windows Mobile jsem narazil na problém s načítáním aplikace. Buď se nevykreslovala zcela správně a určité části obsahu přetékal, nebo se vůbec nezobrazovaly. Několikrát se i stalo, že se správně nenačetl textový editor Ace. Tato problémová chování se projevovala na nejstahovanějších prohlížečích této mobilní platformy (Internet Explorer, Maxthon³⁵ a UC Browser³⁶) a ke dni dokončování této práce se nepodařilo problém úspěšně vyřešit.

Zobrazení aplikace na platformách Android a iOS je oproti Windows Mobile, znatelně lepší a výše popsané problémy jsem byl schopný eliminovat, nebo vůbec nenastaly. Jediným omezením, které na mobilních zařízeních pozoruji, je velikost

³⁵<http://goo.gl/y91vnT>

³⁶<http://goo.gl/0Pfv1w>

zařízení. Pokud má uživatel zařízení s displayem menším jako 7 palců, tak nemůže čekat příliš pohodlnou práci s editorem. Prostě a jednoduše, když uživatel vysune softwarovou klávesnici, tak ta na malém displayi zabere klidně i 85 % veškerého místa na obrazovce a pak už moc místa na zobrazení editoru nezůstává. Toto není ani tak chyba aplikace, jako spíše omezení daného zařízení.

Další nedostatek, opět na mobilních zařízeních, vidím v tom, že se editorem Ace nedá v delších dokumentech úplně pohodlně scrollovat. Scrollování funguje, ale uživatel musí přesně vědět, v které části editoru má prstem táhnout, nemůže například umístit prst doprostřed textu a pohnout s ním nahoru nebo dolů a očekávat, že se tak bude pohybovat dokumentem. Musí toto činit na hraně dokumentu, což není úplně intuitivní a dokážu si představit nejednoho člověka, který by tím mohl být rozladěn. Proto byly v pravé části vytvořeny tlačítka dvojitých šipek, díky kterým se dá v dokumentu rychleji pohybovat, jejich funkcionalita je podobná klávesám `Page Up` a `Page Down` na klasické klávesnici. Alternativa `CodeMirror` očekávané scrollování umí mnohem lépe, ale při výběru byla důležitá funkcionalita, a jak již bylo poznamenáno v kapitole 4, nemožnost mazat znaky byla dost závažný důvod editor nezvolit.

Jako drobný nedostatek vnímám i samotnou grafickou podobu rozhraní. Tím mám na mysli hlavně vzhled prvků a tlačítek, které nejsou zcela konzistentní. Jde o pozůstatek různých testů s rozložením prvků, testy vkládání ikon jako rastry, fonty a vektorovou grafiku apod. Výsledkem je, že některé prvky mají mezi sebou trochu jinou mezeru, než jiné, nejsou vždy zarovnány stejným způsobem a nezabírají stejný prostor i když jde o tlačítko stejného typu. Na odstranění těchto drobností již pracuji, ale pro úplost je přesto uvádím.

Posledním nedostatkem, který bych tu chtěl uvést, je funkcionalita. V původním uživatelském rozhraní měl uživatel v nástrojové liště mnohem více nástrojů a tlačítek, které mohl použít. Jelikož ale bylo rozhraní psáno zcela od začátku, tak se nedají všechny funkce naráz přenést pouhým překopírováním, ale musí se jich část přepsat. Team `TeXonWeb` na přenosu nástrojů intenzivně pracuje. Některá tlačítka, která jen vkládají specifický řetězec do stránky, by bylo samozřejmě možné přenést jednoduše, ale různé matematické znaky a symboly jsou tak specifické, že pro ně momentálně nemáme vhodnou vektorovou ikonu, a bude potřeba ikonky dokreslit, což zabere určitý čas.

5.3 Kroky v budoucnu

Další kroky, které budou s rozhraním činěny, bude postupné eliminování nedostatků uvedených v předchozí sekci. Některá odstranění nedostatků budou lehčí a je jasné co se má přímo dělat (vyladění grafické podoby, dokreslování ikon) a jiná budou potřebovat hlubší analýzu, porozumění problému, a navrhnutí robustního řešení (kompatibilita s mobilním Internet Explorerem, vylepšení scrollování u editoru Ace).

Dále potom plánuji lepší využití volného místa v rozhraní, například lišta na spodním okraji může zobrazovat některé užitečné informace o dokumentu (jméno

souboru, stav uložení, verze souboru, počet slov, počet znaků, atd.) a rovněž bych se chtěl zabývat více backendovou částí, která celé rozhraní generuje. Backend je psán v jazyce Perl a vidím v něm prostor pro zlepšení, sjednocení volání některých funkcí, generování statických částí aplikace a upravení přihlašovací a registrační stránky aplikace.

Rozhodně by také bylo vhodné se zabývat možnostmi, aby si uživatel mohl rozhraní nastavit podle svého, například kromě vzhledu editoru mít i možnost nastavit si barvy celé aplikace, nahrát vlastní ikonu profilu a v budoucnu třeba i prvky rozhraní zobrazovat, skrývat a přesouvat podle vlastních preferencí, které by si mohl uložit.

5.4 Zhodnocení

Když bych měl novou implementaci rozhraní \TeX onWeb porovnat s konkurencí jako jsou Share \LaTeX nebo OverLeaf, tak cítím, že \TeX onWeb nepůsobí stejným dojmem. U obou zmíněných, když si otevřete úvodní stránku, tak působí jako profesionální korporátní aplikace, která má potřebu na Vás od začátku chrlit, co všechno umí a když si ji chcete vyzkoušet, tak Vás donutí se hned registrovat (nutnou podotknout, že je možné se registrovat bezplatně). \TeX onWeb jde od svého začátku svou cestou. Na adrese <https://tex.mendelu.cz> má uživatel ihned k dispozici textový editor, napíše text a může si z něj ihned vygenerovat PDF. Pokud si chce své soubory ukládat na server, tak po krátké registraci může. V novém rozhraní jsem tuto funkcionalitu neměnil. Myslím si, že je to výhoda a uživatelé ji ocení, což svým kladným hodnocením potvrzují i studenti předmětu Zpracování textů na počítači, kteří nové rozhraní již dva semestry za sebou testují.

Co se týče samotného vzhledu rozhraní, tak si myslím, že \TeX onWeb nyní nijak nezaostává, naopak bych si subjektivně troufl tvrdit, že v některých aspektech vyčnívá. Například práce v mobilních zařízeních je v \TeX onWeb díky vysouvajícím se menu a nástrojovým tlačítkům příjemnější, než v alternativách. Rozhraní aplikací Share \LaTeX a OverLeaf se mi občas při práci z nějakého důvodu zadržovala a musel jsem čekat, než budu moci něco napsat. Na druhou stranu, oba zmíněné projekty zatím funkcemi \TeX onWeb přebíjí. Bližší porovnání popisují v kapitole 3.2 Alternativy.

6 Závěr

Cílem této práce bylo modernizovat uživatelské rozhraní tak, aby odpovídalo moderním standardům, bylo použitelné i na mobilních zařízeních a poskytovalo uživatelům komfort při práci s typografickým systémem $\text{T}_{\text{E}}\text{X}$ a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. Pro implementaci funkcionality v rámci splnění tohoto cíle byly použity technologie jazyků Perl, Javascript, jQuery, HTML5, CSS3, Flexbox, operační systém CentOS a webový server Apache.

V práci byl analyzován a určen vhodný postup při řešení stanoveného cíle, zvoleny vývojové prostředky pro jeho dosažení a byla provedena analýza vývoje uživatelského rozhraní. V analýze byly stručně popsány hlavní komponenty tvořící běžná uživatelská rozhraní a byl definován trend responzivního designu. Dále byl vyhodnocen současný stav aplikace a to jak z pohledu uživatelského rozhraní a uživatelské přívětivosti, tak i z pohledu kvality zdrojového kódu frontendu.

V práci bylo dále definováno, proč uživatelské rozhraní modernizovat, popsány motivace k takovému kroku a vyjmenovány alternativy a konkurenti, kteří se podobným tématem zabývají.

V části věnující se implementaci řešení jsem popsal jednotlivé návrhy rozhraní a přiblížil, jakým způsobem došlo k jejich realizaci. Podstatná část se věnuje výběru textového editoru, který tvoří jednu z nepostradatelných částí celé aplikace. Bylo odůvodněno, proč byl konkrétně vybrán Ace editor a byly popsány úpravy v jeho vzhledu a zvyraňování syntaxe.

Při práci na rozhraní byly prováděny konzultace s Ing. Janem Přichystalem, Ph.D., a později i s dalšími členy teamu $\text{T}_{\text{E}}\text{XonWeb}$, čímž jsem na své nápady a myšlenky získal často velmi cennou vazbu a informace, které mi pomohly k dokončení řešení celé práce.

Implementované řešení daný cíl práce splňuje a je k dispozici ve vývojové části aplikace na kterou je veden odkaz z hlavní stránky aplikace. Rozhraní testují studenti předmětu Zpracování textů na počítači Mendelovy univerzity v Brně a je již nyní k dispozici prakticky komukoli, kdo má zájem si je vyzkoušet. Po odstranění některých drobných nedostatků, které jsou uvedeny v kapitole 5.2, předpokládám, že dojde k přesunutí na hlavní stránku projektu, <https://tex.mendelu.cz> a původní verze bude odsunuta na nějakou archivní adresu. Všechny vytvořené a modifikované soubory, které vznikly při návrhu rozhraní, jsou uloženy jako příloha v Univerzitním informačním systému Mendelovy univerzity v Brně.

7 Reference

- CODEMIRROR ISSUES *Android native app using WebView, backspace doesn't work properly #1196*. In: CodeMirror Issues [online]. [cit. 2015-06-25]. Dostupné z: <https://github.com/codemirror/CodeMirror/issues/1196>.
- GARRETT, J. J. *The elements of user experience: user-centered design for the Web and beyond*. 2nd ed. Berkeley, CA: New Riders, 2011, xviii, 172 p. Voices that matter. ISBN 0321683684.
- GOLDSTEIN, ALEXIS, LOUIS LAZARIS A ESTELLE WEYL., 2011: HTML5 a CSS3 pro webové designéry. Vyd. 1. Brno: Zoner Press, 2011, 286 s. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
- HOLZNER, S. *JavaScript : profesionálně*. 1. vyd. Brno: Mobil Media, 2003. 1071 s. ISBN 80-86593-40-1.
- JOHNSON, J. *Designing with the mind in mind simple: simple guide to understanding user interface design guidelines*. Second edition. 234 p. ISBN 9780124079144.
- MARCOTTE, E.; FOREWORD BY KEITH J.] *Responsive web design*. New York: A Book Apart, 2011. ISBN 9780984442577.
- MOZILLA DEVELOPER NETWORK AND INDIVIDUAL CONTRIBUTORS *Using CSS flexible boxes [online]*. 2015 [cit. 2015-06-20]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes.
- MOZILLA DEVELOPER NETWORK *The !important exception*. In: Mozilla Developer Network [online]. [cit. 2015-06-25]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity#The_!important_exception.
- NIELSEN, J. *10 Usability Heuristics for User Interface Design*. Nielsen Norman Group [online]. 1995 [cit. 2015-06-17]. Dostupné z: <http://www.nngroup.com/articles/ten-usability-heuristics>.
- ODELL, D. *JavaScript: průvodce programováním ajaxových aplikací*. Vyd. 1. Brno: Computer Press, 2010, 368 s. ISBN 978-80-251-2733-9.
- PŘICHYSTAL, J.; RYBIČKA, J. *Webové rozhraní pro sazbu dokumentů*. In OLŠÁK, P. SLT 2004 – sborník semináře o Linuxu a TeXu. 1. vyd. Brno: Konvoj, CSTUG, CZLUG, 2004, s. 73–78. ISBN 80-7302-068-8.
- PŘICHYSTAL, J. *Dokumentace systému TeXonWeb*. [online]. 2006, Dostupné z: <http://tex.mendelu.cz/navod.pl>.
- PŘICHYSTAL, J. *Možnosti tvorby dokumentů v TeXu pomocí webového prohlížeče*. Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach. Zborník príspevkov

- medzinárodnej konferencie OSSConf 2014. 1. vyd. Bratislava: Spoločnosť pre otvorené informačné technológie, 2014, s. 23–30. ISBN 978-80-970457-4-6.
- PŘICHYSTAL, J. *Nové vlastnosti systému TeXonWeb*. In MendelNet 2005 (sborník abstraktů z evropské vědecké konference doktorandů). 1. vyd. Brno: Konvoj, 2005, s. 121–122. ISBN 80-7302-107-2.
- SATRAPA, P. *Perl pro zelenáče*. 2. vyd. Praha: Neokortex, 2000. 224 s. ISBN 80-86330-02-8.
- STANÍČEK, P. *CSS kaskádové styly : kompletní průvodce*. 1. vyd. Praha: Computer Press, 2003. 178 s. ISBN 80-7226-872-4.
- VYBÍHAL, J.; PŘICHYSTAL, J. *Tvorba nového uživatelského rozhraní aplikace TeXonWeb*. PEFnet 2014. 1. vyd. Brno: Mendel Univerzity in Brno, 2014, s. 1–10. ISBN 978-80-7509-152-9.

Přílohy

A Celé výsledné rozhraní aplikace

The screenshot shows the TeX Web editor interface. The main window displays the source code of a LaTeX document, and the sidebar on the right contains navigation and utility buttons.

TeX Web Editor Interface:

- Header:** TeX^{on}Web logo, user profile 'xvybihal', and settings icons.
- Toolbar:** Includes icons for file operations (save, undo, redo, search, check), text formatting (bold, italic, underline, strikethrough), and other tools (calendar, image, list, menu).
- Main Editor:** Displays LaTeX source code for a document titled 'Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}'. The code includes preamble, title, author, abstract, and table of contents sections.
- Right Sidebar:** Contains a table of symbols and their corresponding LaTeX commands, and two buttons: 'log' and 'PDF'.

LaTeX Source Code (Line 1 to 32):

```

1 % preamble dokumentu
2 \documentclass[12pt]{article}
3 \usepackage[czech]{babel}
4 \usepackage{dipp,xltextra}
5 \begin{document}
6 \pagestyle{headings}
7
8 \def\typrace{Bakalářská}
9 % uvodní část závěrečné práce
10 \titul{Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}}{Josef Vybíhal}{Ing
11 . Jan Přichystal, Ph.D.}{Brno, 2015}
12 \podekovani{Děkuji Ing. Janu Přichystalovi, Ph.D. za udělenou možnost pracovat na
13 zajímavém projektu, jeho ochotu se mnou věci konzultovat, velkou trpělivost a-cenné
14 připomínky a-věcné náměty při řešení.}
15 \prohlasenimuz{V-Brně dne \today}
16 \%prohlasenizena{místo a-datium prohlášení}
17
18 \abstract{Vybíhal, J. Modernizing the user interface of \TeX{onWeb} web application. Brno
19 ,~2015.}{This bachelor thesis deals with analyzing, designing and implementing of
20 a-new user interface for web application called \TeX{onWeb}. It describes why current
21 interface does not meet today users criteria anymore. It also describes and
22 implements changes, that should improve user experience, even on their mobile devices
23 . It is described what components were chosen and the way they were implemented,
24 including reasons of choosing specific programming tools.}
25 \textbf{Keywords:} Bachelor thesis, \TeX{onWeb}, HTML, CSS, Flexbox, Javascript, Perl,
26 Apache, SVN, UX, UI}
27
28 \abstrakt{Vybíhal, J. Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}.
29 Brno,~2015.}{Bakalářská práce se zabývá analýzou, návrhem a-implementací nového
30 uživatelského rozhraní pro webovou aplikaci \TeX{onWeb}. Popisuje proč již současné
31 rozhraní není pro některé uživatele vyhovující, navrhuje potřebné změny a-nabízí
32 ukázkou uplatnění těchto změn tak, aby uživatelé měli možnost aplikaci používat na
33 svých mobilních zařízeních. Práce popisuje výběr použitých komponent a-postup při
34 jejich implementaci, stejně tak odůvodňuje a-vysvětluje výběr programových prostředků
35 .}
36 \textbf{Klíčová slova:} Závěrečná práce, bakalářská práce, \TeX{onWeb}, HTML, CSS,
37 Flexbox, Javascript, Perl, Apache, SVN, UX, UI}
38
39 \obsah
40 \cislovat{2}
41
42 % jednotlivé kapitoly dokumentu
43 \input{bp-01_uvod_a_cil.tex}
44 \input{bp-02_metodika_a_postup.tex}
45 \input{bp-03_soucasny_stav.tex}
46 \input{bp-04_implementace.tex}
47 \input{bp-05_diskuze_a_zhodnoceni.tex}
48 \input{bp-06_zaver.tex}
49 \input{bp-99_literatura.tex}
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Right Sidebar:

\	{
}	%
^	#
\$	&
~	_
←	↓
↑	→
⇧	⇩

log
PDF

• bp-00_main.tex

B Výsledné rozhraní aplikace a otevřené levé menu

The screenshot displays the TeXonWeb web application interface. On the left, a sidebar menu contains icons and labels for 'Otevřít', 'Šablony', 'Nový', 'Nová karta', 'Uložit', 'Uložit jako', 'Soubory', 'log', and 'PDF'. The central editor area shows LaTeX source code for a document, including package declarations like `\documentclass{article}`, `\usepackage{babel}`, and `\usepackage{xltextra}`. The code also contains a title in Czech: 'Závěrečná práce, bakalářská práce' and a subtitle: 'Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}'. The right sidebar features a symbol table with LaTeX symbols and their corresponding characters, and two buttons labeled 'log' and 'PDF'.

C Výsledné rozhraní aplikace a otevřené pravé menu

The screenshot displays the TeXonWeb web editor interface. The top bar includes the TeXonWeb logo, a user profile for 'xvybihal', and navigation icons. Below the bar is a toolbar with icons for file operations and text formatting (bold, italic, underline, strikethrough, text color, background color). The main area is split into two panes:

- Left Pane (Source Code):** Shows a LaTeX source file named 'bp-00_main.tex'. The code includes a preamble with document class 'article', packages 'babel', 'dipp', and 'xltextra', and a title 'Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}'. It also contains an abstract in English and Czech, and a list of keywords.
- Right Pane (Rendered Output):** Shows the rendered PDF document. The settings on the right indicate:
 - Překlad: xelatex
 - Počet překladů: 1, 2, 3 (1 is selected)
 - Velikost písma: 14 px
 - Barevný motiv: TeXonWeb
 - Uložit nastavení button

The rendered output shows the title 'Modernizace uživatelského rozhraní webové aplikace \TeX{onWeb}' and the abstract text in both English and Czech. The interface is clean and professional, with a dark theme.