



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Generátor prezentací se stromovou strukturou
Student:	Marek erný
Vedoucí:	Ing. Ji í Pavelka
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Tématem práce je nástroj na tvorbu nelineárních prezentací s hierarchickou (stromovou) strukturou obsahu. Vedle základního pr chodu p es obrazovky nejvyšší úrovn vznikne prezentujícímu v každém bod možnost rekurzivního zano ení.

- Analyzujete sou asnou situaci a požadavky.
- Navrhn te a implementujte nástroj, který umožní:
 - vytvo ení sady HTML soubor prezentace derivací obsahu webových stránek, resp. dokumentu (Google/Word),
 - globální orientaci a navigaci ve struktu e prezentace,
 - ovládání všemi sm ry s podporou externích ovlada (prezenter),
 - zobrazení v p izp sobivém nebo zvoleném pom ru stran,
 - export prezentace jako balík HTML soubor a soubory ve formátu LaTeX a PDF.
- Prove te vhodné testování.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 3. ledna 2017

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



Bakalářská práce

Generátor prezentací se stromovou strukturou

Marek Černý

Vedoucí práce: Ing. Jiří Pavelka

11. května 2017

Poděkování

Děkuji především Ing. Jiřímu Pavelkovi za velmi aktivní přístup k vedení práce a cenné rady. Dále chci poděkovat všem, kteří se podíleli na uživatelském průzkumu a v neposlední řadě rodině a přátelům za morální podporu.

Prohlášení

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

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

V Praze dne 11. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Marek Černý. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Černý, Marek. *Generátor prezentací se stromovou strukturou*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Bakalářská práce se zabývá tvorbou nástroje, jenž uživatelům umožňuje vytváření webových prezentací s hierarchickou strukturou. Prezentace je generována z obsahu, který lze vytvářet v aplikacích Google Dokumenty, Word nebo přímo v jazyce HTML.

První část práce se věnuje analýze, v které jsou prozkoumány existující řešení a potřeby uživatelů při prezentování. Na základě analýzy je navržen a následně implementován prezentační nástroj, ke kterému je současně vytvořen uživatelský manuál.

Výsledné řešení kromě vytvoření hierarchické prezentace nabízí i možnost přizpůsobení a export do dokumentů ve standardním formátu. V příloze lze nalézt metodiku, podle které byla práce vedena, původní návrhy a alternativní způsoby řešení vybraných problémů.

Klíčová slova prezentace, hierarchická, stromová, nelineární, HTML, JavaScript

Abstract

The goal of this bachelor thesis is to make a tool that enables its users to create web presentations with a hierarchical structure. A user can choose between Google Docs, Word or HTML to write the content of a presentation.

The first part of this thesis is an analysis of existing solutions and also of the requirements users have when creating presentations. The analysis is then used to design and subsequently implement the presentation tool, which then leads to creation of a user manual.

The resulting solution, in addition to the creation of hierarchical presentations, also offers the ability to customize presentation and its export to documents in a standard format. The attachment to the thesis contains the methodology according to which the thesis was conducted, original designs and alternative ways of solving selected problems.

Keywords presentation, hierarchical, tree, nonlinear, HTML, JavaScript

Obsah

Úvod	1
1 Cíl práce	3
2 Analytická fáze	5
2.1 Průzkum existujících řešení	5
2.2 Uživatelský průzkum	9
2.3 Požadavky na funkce	16
2.4 Požadavky na kvality	18
2.5 Posouzení existujících řešení vůči požadavkům na funkce	20
3 Návrhová fáze	23
3.1 Aplikační architektura	23
3.2 Procesy	29
3.3 Případy užití	33
3.4 Prototypování	36
4 Implementační fáze	43
4.1 Koncepty řešení dílčích požadavků	43
4.2 Technická specifikace	50
4.3 Testování	50
Závěr	53
Literatura	55
A Pravidla a zásady projektů FIT	57
B Modul derivace obsahu	67
B.1 Logika 0 {L0}	68
B.2 Logika 1 {L1}	68

C	Linearizace průchodu	71
D	Uživatelský manuál	73
E	Vzorové dokumenty	79
F	Seznam použitých zkratk	83
G	Obsah přiloženého CD	85

Seznam obrázků

3.1	Schéma aplikační architektury	24
3.2	Schéma derivace obsahu	26
3.3	Schéma tvorby stromové struktury z HTML5	28
3.4	Proces 1	30
3.5	Proces 2	30
3.6	Proces 3	31
3.7	Proces 4	32
3.8	Proces 5	32
3.9	Proces 6	33
3.10	Prototyp – navigační obrazovka	37
3.11	Prototyp – obrazovka nastavení	38
3.12	Prototyp – obrazovka klávesové zkratky	38
3.13	Prototyp – obrazovka stažení	39
3.14	Prototyp – rozložení navigačních prvků	40
3.15	Primární navigace	40
B.1	Struktura vytvořená modulem derivace obsahu	68
C.1	Schéma linearizace průchodu	72

Seznam tabulek

2.1	Posouzení existujících řešení vůči požadavkům na funkce	21
4.1	Klávesové zkratky	47
4.2	Dotyková gesta umožňující ovládání prezentace	48
B.1	Výchozí nastavení derivace	69

Úvod

V dnešní době existuje nespočet webových aplikací a knihoven, které se zabývají tvorbou prezentací a téměř všechny mají společnou jednu vlastnost a to, že jejich výstupem je prezentace s lineární strukturou. Nicméně tento typ prezentace nemusí být dostačující pro všechny uživatele.

Výsledná práce umožní vytvářet prezentace s hierarchickou neboli stromovou strukturou, která bude vytvořena na základě derivace obsahu dokumentu Google, Word nebo HTML.

Téma jsem si zvolil, neboť v současné době není problém spojený s tvorbou hierarchických prezentací uspokojivě vyřešen a jeho řešení by mohlo být přínosné zejména pro náročnější a aktivně přednášející uživatele.

Celá práce je rozdělaná do tří fází podle metodiky Pravidla a zásady projektů FIT (příloha A). První část se zabývá analýzou existujících řešení a potřeb uživatelů, na základě kterých jsou vytvořeny požadavky na funkce a kvality výsledného nástroje.

Druhá část se věnuje návrhu řešení požadavků z analytické fáze a vytvoření sady případu užití a procesů. Poslední část se zabývá implementací dle návrhu, popisem použitých technologií a testováním.

Cíl práce

Cílem rešeršní práce je analýza existujících řešení, kvalitativní a následně kvantitativní uživatelským průzkum, na jehož základě budou sestaveny funkční a nefunkční požadavky pro výsledný nástroj. Na závěr budou porovnány navrhované funkční požadavky s vlastnostmi již existujících řešení.

Praktická část má za cíl návrh a implementaci nástroje, který bude umožňovat tvorbu prezentací se stromovou strukturou s možností jejich generování ze standardních formátů. Nástroj dále bude umožňovat vizuální přizpůsobení prezentace a export do dokumentů PDF a \LaTeX .

Analytická fáze

V rámci analytické fáze byla provedena analýza existujících řešení, kde byla prozkoumána celá řada webových prezentačních nástrojů. Nicméně žádný z posuzovaných nástrojů nenabízel veškeré požadované funkce.

Přestože v současné době není dostupný odpovídající nástroj, tak uživatelský průzkum ukázal, že uživatelé o něj mají zájem. A tedy na základě kvalitativního a kvantitativního průzkumu byl vytvořen seznam vlastností, které jsou pro uživatele klíčové.

Nakonec bylo provedeno srovnání funkčních vlastností navrhovaného nástroje s již existujícími nástroji formou tabulky (manažerské shrnutí), která umožňuje informované rozhodnutí o dalších krocích návrhu.

2.1 Průzkum existujících řešení

Dokument se zabývá výběrem a popisem vlastností existujících nástrojů pro tvorbu webových prezentací. V současné době je na internetu k dispozici velké množství takových nástrojů, ať už ve formě aplikací nebo knihoven.

2.1.1 Existující řešení

Většina vybraných nástrojů využívá vlastní formát HTML, jako zdroj dat a kaskádové styly pro vytváření barevných témat. Některé nástroje umožňují nelineární průchod. Žádný z nalezených nástrojů nenabízí možnost vytvoření hierarchické prezentace nebo generování prezentace z webové prezentace.

2.1.1.1 Reveal.js¹

Framework pro vytváření prezentací pomocí značkovacího jazyka HTML, nabízející speciální zobrazení pro prezentátora, podporu mobilních zařízení a

¹<http://lab.hakim.se/reveal-js>

možnost zanořování slajdů. Výslednou prezentaci lze vygenerovat ve formátu PDF.

- Nelineární průchod - každý slajd (horizontální průchod) je možné doplnit libovolným počtem slajdů (vertikální průchod).
- Čtyřbodová navigace pro průchod prezentací (nahoru/dolů, doleva/doprava).
- Zobrazení struktury celé prezentace formou náhledu na samostatné obrazovce včetně vyznačení aktuální pozice.
- Podpora gest na mobilních zařízeních.
- Zobrazovací mód pro prezentátora na druhý monitor, který obsahuje aktuální a následující slajd, čas a poznámky.
- Export prezentace do PDF.

2.1.1.2 Impress.js²

Umožňuje vytvářet online prezentace, které využívají plný potenciál přechodů a transformací kaskádových stylů CSS3 v moderních prohlížečích. To dává prezentátorům možnost kromě klasických slajdových prezentací vytvářet i nejruznější vizualizace.

- Canvas neomezených rozměrů, v kterém lze obsah prezentace libovolně umísťovat, škálovat a otáčet.
- CSS3 transformace vytvářející efekt 3D prezentace.
- Přesun na předcházející nebo následující slajd pomocí šipek na klávesnici.
- Myš umožňuje přeskakovat libovolně mezi slajdy prezentace v případě, že je její rozvržení tomu přizpůsobené.

2.1.1.3 Deck.js³

HTML5 framework, který kromě základních funkcionalit, jako je navigace doleva/doprava a volba přechodů dává prezentátorům možnost i přidávání rozšíření, ať už vlastních nebo originálních od vývojářů samotného frameworku.

- Přidávání a odebírání rozšíření.
- Možnost vytvoření vlastních rozšíření.

²<https://github.com/impress/impress.js>

³<http://imakewebthings.com/deck.js/>

- Aktuální pozice slajdu v rámci celé prezentace zobrazena číselným indikátorem (3/15).
- Navigační prvky umožňující přechod mezi předcházejícím a následujícím slajdem pomocí klávesnice i myši.

2.1.1.4 Shower⁴

Prezentační engine postavený na HTML, CSS a Vanilla JS. Vytvořené prezentace fungují ve všech moderních internetových prohlížečích a nabízejí možnost exportu do souboru ve formátu PDF.

- Export prezentace do PDF.
- Horizontální ukazatel průběhu (progress bar) orientačně znázorňující aktuální pozici v prezentaci.
- Podpora klávesových zkratk pro spouštění/vypnutí prezentace a přechod mezi předcházejícím a následujícím slajdem.
- Lze zvolit vizuální téma prezentace.

2.1.1.5 Prezi⁵

Prezentační software, jenž je alternativou k tradičním na slajdech založeným prezentacím. Základem prezentací je neomezeně velká mapa, na kterou lze kreslit slajdy libovolných tvarů a velikostí. Tím lze vytvořit i efekt vizuální hierarchizace, avšak stále pouze s lineárním procházením.

Vytváření, editaci a prezentaci prezentací lze provádět přímo ve webovém prohlížeči nebo v mobilní a desktopové aplikaci, která je dostupná pouze pro platící uživatele spolu s dalšími funkcemi, jako export prezentace do PDF, editace obrázků a online podpora.

- Bezplatná verze nabízející základní funkce.
- Webová, mobilní a desktopová aplikace pro prezentování offline.
- Tvorba grafů, diagramů a editace obrázků přímo v prezentaci.
- Interaktivní tutoriály vysvětlující základní funkčnost a online podpora.
- Velké množství hotových šablon.
- Aplikace lze ukládat a sdílet prostřednictvím cloudu.

⁴<https://shwr.me/>

⁵<https://prezi.com/>

2.1.1.6 Prezentace Google⁶

Webový nástroj pro tvorbu prezentací, mezi jehož hlavní přednosti patří jednoduché sdílení, možnost úpravy jedné prezentace několika vybranými uživateli zároveň a export do několika formátů, kterými jsou například PDF, JPEG, SVG a PowerPoint. Také nabízí velké množství barevných motivů, přechodů a rozložení slajdů.

- Možnost vkládání animací, videí a obrázků.
- Mobilní aplikace pro Android a iOS.
- Sdílení prostřednictvím odkazů nebo uživatelských účtů.
- Bezdrátové prezentování pomocí Chromecast⁷.

2.1.2 Klíčové vlastnosti

Mezi klíčové vlastnosti existujících nástrojů patří samotná prezentace a její ovládání. Pro účel naší aplikace jsou klíčová zdrojová data a struktura výsledné prezentace. Důležitým faktorem výběru či vývoje aplikace je bezesporu přizpůsobivost a customizace.

- Prezentace s nelineární strukturou,
 - s možností zanoření se do hloubky, hierarchická,
 - lineární průchod nelineární prezentace.
- Generování prezentace ze standardního formátu,
 - z HTML nebo dokumentu GD⁸/Word.
- Export do standardního formátu,
 - L^AT_EX/ PDF.
- Spuštění prezentace,
 - od začátku,
 - z konkrétního místa.
- Navigace a orientace v prezentaci,
 - číselně nebo pomocí schématu.

⁶www.google.com/intl/cs_cz/slides/about

⁷Produkt od společnosti Google pro streamování digitálního obsahu.

⁸Google Dokumenty

- Ovládání prezentace,
 - externí ovladače,
 - gesta na dotykových zařízeních,
 - klávesové zkratky.
- Vizualní přizpůsobení prezentace,
 - pomocí barevných témat,
 - vlastní.
- Offline prezentování.

2.2 Uživatelský průzkum

Uživatelský průzkum zjišťoval, co je pro uživatele nejdůležitější při práci s prezentačními nástroji, jaké funkce jim u současných řešení chybí a jejich pohled na nelineární prezentace. Z průzkumu plyne, že by uživatelé nástroj využili, a to zejména kvůli možnosti tvorby nelineárních prezentací.

Průzkum se skládal z dvou částí - kvalitativní a kvantitativní. Kvalitativní průzkum byl prováděn formou rozhovorů se zástupci uživatelských skupin za účelem vytvoření hypotéz vycházejících z používání prezentačních nástrojů. Kvantitativní průzkum byl tvořen dotazníkem, který sloužil k ověření hypotéz a prioritizaci funkcí navrhovaného nástroje.

2.2.1 Uživatelské skupiny

Potencionální uživatelé prezenteru byli rozděleni do dvou uživatelských skupin – Akademici a Zaměstnanci a studenti.

- Akademici:
 - pedagogové a školitelé,
 - vysokoškolsky vzdělaní,
 - pravidelně přednášející s prezentací,
 - alespoň s minimálními znalostmi webových technologií,
 - se zájmem o prezentace s nelineární strukturou.
- Zaměstnanci a studenti:
 - z oboru informatiky,
 - se zálibou ve webových technologiích,
 - prezentující svou práci,
 - vlastníci webovou prezentaci,
 - rádi zkoušející nové věci týkající se daného oboru.

2.2.2 Témata k rozhovorům

Témata k rozhovorům vycházejí z klíčových vlastností. Protože výsledná aplikace bude využívána více skupiny uživatelů, přičemž každý z nich má různé potřeby a představy o jejím fungování, tak je nezbytné zjistit, co je pro uživatele nejdůležitější, a kterým věcem naopak nepřikládají takovou důležitost.

1. Prezentace s hierarchickou strukturou.
2. Generování prezentace ze standardního formátu.
3. Export prezentace do standardního formátu.
4. Spuštění prezentace.
5. Navigace a orientace v prezentaci.
6. Ovládání prezentace.
7. Vizuelní přizpůsobení prezentace.
8. Vlastní či existující rozšíření a konfigurace.
9. Prezentování bez přístupu k internetu.

2.2.3 Kvalitativní průzkum

Cílem kvalitativního průzkumu bylo zjištění, jaký pohled mají uživatelé na nelineární prezentace, jaké vlastnosti prezentačního softwaru jsou pro ně nejdůležitější a naopak, co nepovažují za důležité nebo vůbec nevyužívají během prezentování.

Průzkum byl prováděn formou rozhovorů s vybranými zástupci uživatelských skupin. Rozhovory byly uskutečněny s každým zástupcem zvlášť, pokládané otázky vycházely z kapitoly 2.2.2 a délka jednotlivých rozhovorů byla 20–30 minut.

2.2.3.1 Ing. Ladislav Vagner, Ph.D., odborný asistent a vyučující na FIT, 28. listopadu 2016

- U prezentací považuje za nejdůležitější verzovatelnost, kompatibilitu napříč zařízeními (portabilitu) a strojově modifikovatelné zdroje (L^AT_EX).
 - Prezentační mód na Eduxu mu nevyhovuje zejména pro omezenost syntaxe (rovnice).
- Generování prezentace jako derivaci z webových stránek považuje za zajímavé.

- Sám má spoustu textů pro studenty, které do prezentace nepatří. Takové texty by byly k dispozici na webu. Z tohoto (společného) zdroje by se generovala prezentace bez popisných textů.
- Další zdroje pro prezentace vidí ve stávajícím e-learningu pro předměty PA1, PA2, PS1 a další, které využívají DokuWiki.
- Nelineární prezentaci (prezentaci s odbočkami) považuje za zajímavý nápad.
 - Nedovede si však představit popsané chování v praxi.
 - Problém spatřuje v přizpůsobování jakékoli automatizace (tagy, štítky, třídy).
 - Také si nedovede představit nelineární prezentaci v PDF.

2.2.3.2 Ing. Filip Suk, vývojář a podnikatel v IT, 30. listopadu 2016

- Možnost vytváření prezentací s nelineární strukturou se mu líbí a největší potenciál vidí ve využívání již existujících dat, díky čemuž by se vyvaroval zbytečnému duplikování obsahu.
 - Dle jeho názoru by prezentace měly být zároveň co nejjednodušší a zanoření o více, než jednu úroveň nepovažuje za přínosné.
- Export prezentace by uvítal hlavně pro prezentování v cizím prostředí, kde se nemůže spolehnout na internetové připojení.
 - Za ideální by považoval export prezentace do souboru HTML.
- Ovládání pomocí myši, klávesnice a dálkových ovladačů považuje v dnešní době za samozřejmost.
- Nevyžaduje široké možnosti vizuálního přizpůsobení prezentace. Volbu barvy písma a pozadí považuje za dostačující.
 - Velmi by se mu líbila podpora zvýraznění syntaxe programovacích jazyků.
- Za velmi důležité považuje kvalitní dokumentaci, aby věděl, jakým způsobem se výsledná prezentace generuje.
 - Také by ocenil konfiguraci prezentace například prostřednictvím textového souboru.

2.2.3.3 Tomáš Martínek, student na FIT, 1. prosince 2016

- Nelineární prezentace ho zaujala. Hlavně možnost zabývat se nějakými tématy v prezentaci pouze povrchově a v případě potřeby se zanořit a probrat je detailněji.
- Generování prezentace z HTML nebo GD se mu líbí.
 - Vidí v tom příležitost, jak zaujmout.
 - Uvítal by i generování stručné (rychlé) nebo podrobné prezentace.
- Líbila by se mu možnost exportu prezentace do PDF pro případ, že by jí chtěl shlédnout například v metru.
- Spuštění prezentace z libovolné podkategorie webové stránky považuje za užitečné.
 - V případě takové funkcionality by chtěl i možnost volby zda se prezentace spustí z dané podkategorie nebo od začátku.
- Za ideální navigaci ve smyslu orientace v prezentaci považuje minimapu.
- Vizuaální přizpůsobení prezentace je pro něj velmi důležité.
 - Ocenil by výběr z barevných profilů nebo možnost vytvoření vlastních.

2.2.3.4 Ing. Lukáš Bařinka, odborný asistent na KSI a ICT, 5. prosince 2016

- Nelineárních prezentace mu nejsou cizí a jejich generování ze standardních zdrojů by využil.
 - Jako zdroj pro generování by preferoval GD, Word nebo již existující webové stránky.
- Myšlenka exportu prezentace do souboru se mu líbí a dovede si jí představit.
 - Jako výstupní formát by preferoval PDF, který mu mimo jiné umožňuje snadné vytisknutí.
- Pro ovládání prezentace je pro něj klíčový prezentér (dálkové ovládání).
 - Ocenil by možnost namapování tlačítek prezentéru pro plnohodnotné ovládání nelineární prezentace.
- Ukazatel aktuální pozice v rámci celé prezentace by preferoval formou indikátoru průběhu (progress bar) nebo minimapy.

- Vizualní přizpůsobení prezentace by si představoval výběrem z přednastavených témat s možností jejich upravení.
 - Za velmi důležité považuje korektní zobrazování na monitorech různých velikostí a poměrů stran.
- Jako zajímavé rozšíření zmínil mobilní aplikaci, kde by prezentující měl k dispozici náhled následujících slajdů, ovládání prezentace a časomíru.

2.2.3.5 Jan Havlůj, Softwarový architekt, 12. prosince 2016

- Prezentace s nelineární strukturou se mu líbí a umí si jejich využití představit v praxi.
- Generování prezentací z webových stránek ho příliš nezaujalo.
 - Dával by přednost tvorbě obsahu například ve značkovacím jazyce XML, protože je přesvědčen, že tak je schopen vytvořit kvalitnější prezentaci, díky stoprocentní kontrole nad obsahem.
- V případě využití webových stránek, jako zdroje dat pro prezentaci by nevyžadoval možnost export do standardního formátu.
 - Z toho důvodu, že všude, kde se pohybuje je k dispozici internetové připojení a tedy dostupnost na webových stránkách považuje za dostačující.
- Prvky znázorňující aktuální pozici v prezentaci považuje za velmi důležité hlavně v případě nelineárních prezentací.
 - Také by uvítal zobrazení pro prezentujícího na druhém monitoru, kde by mohl vidět náhled následujících slajdů.
- Prezentace ovládá zásadně dálkovým ovládáním nebo klávesnicí.
 - Ovládání myši nepoužívá z důvodu nutnosti zaměření se na aktuální pozici na obrazovce, což je pro něj zbytečné rozptýlení.
- Přizpůsobení vzhledu prezentace by si představoval pomocí předpřipravených šablon.

2.2.3.6 Vyhodnocení rozhovorů

Z rozhovorů vyplývá, že dotazovaným se líbí možnost generování prezentace z dokumentů ve standardním formátu. Nelineární strukturu prezentace považují také za zajímavou, i když je to pro některé uživatele hůře představitelné.

Dále se ukázalo, že export prezentace a podpora dálkových ovládaní je pro dotazované klíčová. Na rozdíl od vizuálního přizpůsobení, kde nevyžadují žádné speciální možnosti a vystačí si s připravenými šablonami.

2.2.4 Kvantitativní průzkum

Při kvantitativním průzkumu bylo postupováno podle metodiky 6 kroků, jak vytvořit dotazník[1].

2.2.4.1 Krok 1) Co je potřeba zjistit a jak budou zjištěné informace využity

Cílem dotazníku je zjištění, jak často uživatelé prezentují, jaké prezentační nástroje k tomu využívají a jaké funkce jsou pro ně nejdůležitější při používání daných nástrojů. Zjištěné informace umožní prioritizaci jednotlivých funkcí.

2.2.4.2 Krok 2) Příprava otázek

Při vytváření otázek byla snaha dávat přednost uzavřeným otázkám před otevřenými. Dále byl kladen důraz na jejich stručnost a jednoduchost.

- Do kterých skupin uživatelů byste se zařadil/a?
- Jak často prezentujete?
- Využíváte prezentace k jiným, než prezentačním účelům, např. jako studijní materiál?
- Jaký využíváte nástroj pro tvorbu prezentací?
- Ocenil/a byste následující funkce u nástroje pro tvorbu webových prezentací?
 - Stažení souborů pro účely offline prezentace.
 - Export prezentace do PDF/L^AT_EX.
 - Zanoření se v prezentaci (podkapitoly k danému snímku).
 - Režim pro prezentujícího (náhled následujících slajdů a poznámek na druhé obrazovce).
 - Podpora dálkových ovládaní.
 - Navigace znázorňující aktuální pozici v prezentaci (číselná 1/23, posuvník apod.).
 - Výběr z několika barevných motivů prezentace.
 - Volba stylu přechodů mezi slajdy.
 - Zvýraznění syntaxe programovacích jazyků.
 - Podpora zobrazení matematických vzorců (ve formátu TeX).

– Podpora různých velikostí obrazovek (s možností volby poměru stran).

- Jaké další funkce prezentačních nástrojů využíváte?
- Jaké funkce Vám u prezentačních nástrojů chybí?

2.2.4.3 Krok 3) Otestování otázek

Otázky byly testovány na skupině, která byla tvořena 5 lidmi ve věkovém rozmezí 21 až 50 let, kde 3 z dotazovaných měli technické vzdělání. Skupina byla po přečtení otázek dotázána, zda všechny otázky považují za dobře formulované a rozumí jim. Nikdo z dotazovaných neměl s pochopením nebo formulací otázek problém, takže nebyla nutná jejich úprava.

2.2.4.4 Krok 4) Sestavení dotazníku

Pro tvorbu dotazníku byla zvolena aplikace Formuláře Google⁹, která nabízí přívětivé uživatelské rozhraní, díky kterému je vytvoření i následná správa dotazníku rychlá a snadná. Navíc poskytuje přehledné statistiky zodpovězených otázek ve formě výsečových grafů.

Dotazník se skládal z jedné otázky rozčleněné na 11 podotázek, kde uživatel u každé podotázky mohl vybírat z 5 odpovědí (určitě ano | spíše ano | nevím | spíše ne | určitě ne), přičemž označit šlo pouze jednu z nich. Dále byly v dotazníku 4 otázky s volbou odpovědi a 2 otevřené otázky. Žádná otázka nebyla definovaná, jako povinná. Na konci dotazníku měli dotazující možnost zadat svojí e-mailovou adresu v případě zájmu se podílet na dalším testování.

2.2.4.5 Krok 5) Otestování dotazníku

Před finálním rozesláním dotazníku byla jeho funkčnost a dostupnost otestována třemi lidmi, kteří dotazník zvládli vyplnit do 3 minut. Dva z dotazovaných si nebyli jisti, co mají vyplnit u otevřených otázek v případě, že je nic nenapadá. Na základě jejich pochybností byly otázky upraveny.

2.2.4.6 Krok 6) Rozeslání dotazníku

Rozeslání dotazníku probíhalo ve dvou fázích. V první fázi byl dotazník rozeslán menší skupině respondentů a následně proběhla kontrola jejich odpovědí. Žádná z odpovědí nebyla nevhodná nebo nepoužitelná pro účely dotazníku, takže již nebylo třeba dalších úprav. V druhé fázi byl dotazník zveřejněn na sociální síti za účelem oslovení většího počtu respondentů.

⁹https://www.google.com/intl/cs_CZ/forms/about/

2.2.4.7 Vyhodnocení dotazníku

Průzkumu se zúčastnilo celkem 50 lidí. Největší skupinu respondentů tvořili studenti spolu se zaměstnanci, kteří jsou zároveň jednou z cílových uživatelských skupin.

Ukázalo se, že většina respondentů prezentuje méně, než jednou měsíčně, avšak velká část z nich využívá často prezentace k jiným, než prezentačním účelům. Mezi tři nejpoužívanější nástroje pro tvorbu prezentací patří PowerPoint, Google Prezentace a \LaTeX . Dále z výsledků dotazníku vyplývá následující:

- Většina uživatelů by ocenila možnost stažení souborů prezentace a její export do PDF / \LaTeX .
- Uživatelé by využili možnost zanoření se v prezentaci.
- Během prezentování je důležitá podpora dálkových ovládaní a navigace znázorňující aktuální pozici v prezentaci.
- Velká část uživatelů si vystačí s možností výběru barevných motivů. Naopak o volbu přechodů mezi slajdy nestojí.
- Podporu zobrazení matematických vzorců a zvýraznění syntaxe programovacích jazyků by uživatelé také využili.

Mezi funkce, které uživatelé využívají, ale nebyly v dotazníku zmíněny patří možnost komentování slajdů, přímá provázanost s aplikací Excel a noční režim při využívání prezentace, jako studijního materiálu v noci.

2.3 Požadavky na funkce

Požadavky na funkce vznikly v závislosti na uživatelském průzkumu a představují funkce, které by měl výsledný nástroj obsahovat. U následujících požadavků (není-li uvedeno jinak) je [MUST] výchozí úroveň nezbytnosti dle klíčových slov[2].

2.3.1 Nelineární struktura prezentace

Nelineární struktura prezentace je jednou ze základních funkcí nástroje, která umožní uživateli vytvořit prezentaci s hierarchickou (stromovou) strukturou obsahu.

- Vytvoření nelineární struktury prezentace.
- Možnost rekurzivního zanoření v každém bodě prezentace.

2.3.2 Generování prezentace ze standardního formátu

Vytvoření prezentace ve formě HTML souboru, jako derivace obsahu webových stránek nebo dokumentů GD a Word.

- Generování z HTML souboru.
- Generování z GD.
- Generování z Word dokumentu.

2.3.3 Navigace a orientace v prezentaci

Navigace znázorňující:

- aktuální pozici v rámci celé prezentace,
- jakým směrem se lze pohybovat v závislosti na pozici a obsahu prezentace.

2.3.4 Ovládání prezentace

Ovládání prezentace pomocí:

- dálkových ovládaní (prezenterů),
- klávesnice a myši,
- dotykových gest na zařízení s dotykovou obrazovkou [MAY].

2.3.5 Volba poměru stran

Zobrazení prezentace na obrazovkách s různými poměry stran lze přizpůsobit pomocí:

- přizpůsobivého režimu, který automaticky zvolí vhodný poměr stran podle parametrů zobrazovacího zařízení,
- nastavení výchozího poměru stran.

2.3.6 Export prezentace

Výslednou prezentaci lze exportovat, jako soubor ve formátu:

- HTML,
- PDF,
- \LaTeX .

2.3.7 Vizualní přizpůsobení prezentace

Vizuální vzhled prezentace lze přizpůsobit:

- výběrem barevného motivu prezentace [SHOULD].

2.4 Požadavky na kvality

Požadavky na kvality představují nefunkční požadavky, které jsou nezbytné pro správné fungování a případný další rozvoj aplikace. Tyto požadavky vycházejí z dokumentu Pravidla a zásady projektů FIT (příloha A). Níže jsou uvedeny požadavky, od kterých bylo v této práci ustoupeno nebo jsou řešeny jinak.

2.4.1 Dokumenty

- Uživatelská dokumentace není dostupná z webu FIT, protože aplikace není součástí informačního systému FIT.

2.4.2 Architektura a integrace do infrastruktury

- Aplikace nevyužívá existující technologie a služby FIT a není součástí katalogu služeb FIT¹⁰.
- Aplikace je implementována v jazycích:
 - PHP,
 - JavaScript (ES6),
 - HTML5,
 - CSS3.

2.4.3 Kód aplikace

- Kód aplikace je vyvíjen na revizním systému Git¹¹.
- Vývoj kódu není řízen testy a nejsou používány stanovené kvalitativní metriky. V rámci QA¹² budou popsány způsoby a formy kontroly pro klíčové funkce. Například vznikne sada dokumentů se specifickým obsahem a očekávanou výstupní strukturou prezentace. Vizualní stránka bude kontrolována pomocí externího nástroje.

¹⁰<https://ict.fit.cvut.cz/web/current/web/ict/Katalog/>

¹¹Distribuovaný systém správy verzí.

¹²Quality assurance

2.4.4 Bezpečnost a ochrana osobních údajů

- Nařízení rektora o ochraně osobních údajů¹³ není pro aplikaci relevantní.

2.4.5 Další koncepty webových aplikací (W2.0)

Aplikace využívá konceptů:

- RSS,
 - upozornění na nové změny v prezentaci [MAY],
- personalizace,
 - zobrazení v přizpůsobivém režimu, který automaticky nastaví poměr stran prezentace v závislosti na zařízení uživatele [MUST],
- customizace,
 - výběr z barevných motivů prezentace [SHOULD],
- folksonomie,
 - libovolnou pozici v prezentaci lze označit (vytvořit záložku) [MAY],
- social networking,
 - přímé sdílení prezentace na sociální síť [MAY],
- real-time web¹⁴,
 - distribuované promítání, které umožňuje posluchačům sledovat prezentaci společně s přednášejícím [MAY],
- crowdsourcing¹⁵,
 - editace, návrhy či komentáře ke slajdům prezentace, které jsou viditelné i pro další uživatele [MAY],
- kolaborace,
 - na obsahu prezentace se může podílet více uživatelů zároveň [MAY],
- průvodce (wizardy),
 - tutoriál popisující ovládání a možnosti aplikace [MAY],
- konfiguratory,

¹³<https://drive.google.com/file/d/0ByRrmJsDN0TwRzhSWn10UHowQ28/view>

¹⁴https://en.wikipedia.org/wiki/Real-time_web

¹⁵<https://en.wikipedia.org/wiki/Crowdsourcing>

- generování výsledné prezentace ze standardního formátu lze nastavit pomocí konfiguračního souboru [MAY],
- gamifikace¹⁶,
 - prezentaci lze ohodnotit na číselné škále 1 až 5, kde vyšší číslo představuje lepší hodnocení [MAY],
- mikrodata¹⁷,
 - obsah prezentace je označen pomocí mikrodat pro její lepší strojové zpracování vyhledávači [MAY].

2.5 Posouzení existujících řešení vůči požadavkům na funkce

Požadavky na funkce navrhovaného nástroje jsou porovnány s funkčními vlastnostmi všech nástrojů z kapitoly Existující řešení 2.1.1. Díky tomu získáme přehled o tom, co všechno nabízí ve srovnání s již existujícími prezentačními nástroji. Porovnání je provedeno pomocí srovnávací tabulky 2.1.

¹⁶<https://en.wikipedia.org/wiki/Gamification>

¹⁷[https://en.wikipedia.org/wiki/Microdata_\(HTML\)](https://en.wikipedia.org/wiki/Microdata_(HTML))

2.5. Posouzení existujících řešení vůči požadavkům na funkce

Tabulka 2.1: Posouzení existujících řešení vůči požadavkům na funkce

Funkce/Nástroj	Reveal.js	Impress.js	Deck.js	Shower	Prezi	Prezentace Google
Nelineární struktura prezentace	Ano	Ne	Ne	Ne	Ne	Ne
Možnost rekurzivního zanoření	Ne	Ne	Ne	Ne	Ne	Ne
Generování prezentace z GD	Ne	Ne	Ne	Ne	Ne	Ne
Generování prezentace z Wordu	Ne	Ne	Ne	Ne	Ne	Ne
Generování prezentace z HTML	Ano	Ano	Ano	Ano	Ano	Ne
Znázornění aktuální pozice v prezentaci	Ano	Ano	Ano	Ano	Ano	Ano
Ovládání prezentace dálkovým ovládáním	Ano	Ano	Ano	Ano	Ano	Ano
Zobrazení v přizpůsobivém režimu	Ano	Ano	Ano	Ne	Ano	Ne
Zvolení poměru stran	Ano	Ne	Ne	Ne	Ano	Ano
Export prezentace do PDF	Ano	Ne	Ne	Ano	Ano	Ano
Export prezentace do L ^A T _E X	Ne	Ne	Ne	Ne	Ne	Ne
Export prezentace do HTML	Ano	Ano	Ano	Ano	Ano	Ne
Výběr barevného motivu prezentace	Ano	Ne	Ano	Ano	Ano	Ano

Návrhová fáze

V této kapitole byla navržena aplikační architektura, která rozděluje dílčí problémy tvorby prezentace, jako derivace obsahu, export do PDF nebo vytvoření stromové struktury do několika částí. Do návrhu se navíc povedlo začlenit již existující službu usnadňující práci se vstupními dokumenty GD a Word.

Poté na základě analytické fáze a funkčních požadavků byly vytvořeny vzorové případy užití aplikace, které jsou pokryty sadou procesů.

Na závěr byly navrženy prototypy jednotlivých obrazovek v prezentaci, včetně navigačních a orientačních prvků.

3.1 Aplikační architektura

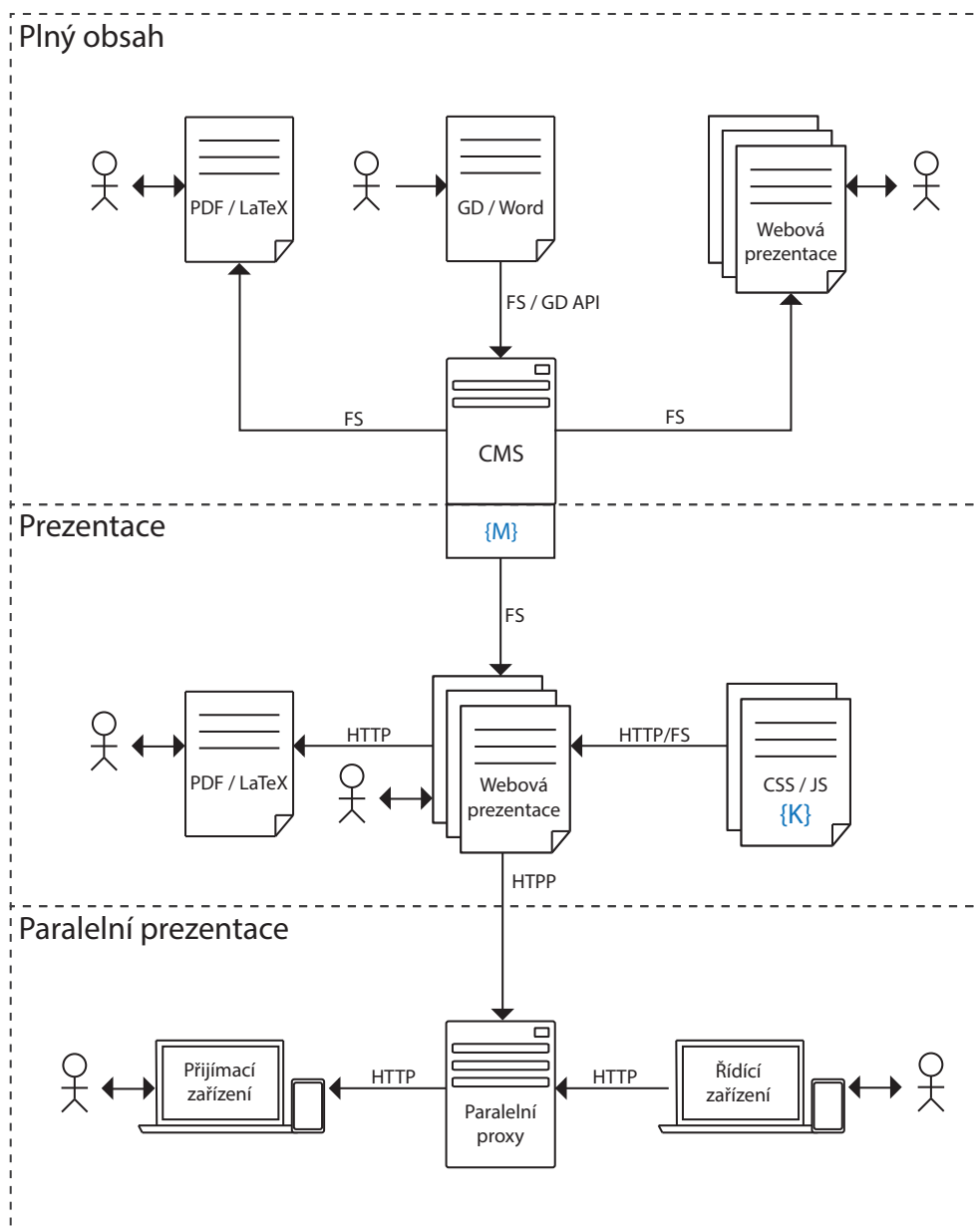
Navrhovaná aplikační architektura znázorněná na obrázku 3.1 zahrnuje veškeré služby a moduly, které jsou nezbytné pro vytvoření prezentace ve formátu PDF/ \LaTeX nebo HTML z uživatelem vytvořeného dokumentu. Požadovaná funkcionální aplikace je rozdělena do dvou částí skládající se z modulu a knihovny s identifikátory $\{M\}$ a $\{K\}$. Schéma je rozděleno do tří sekcí:

1. Plný obsah – nezměněný vstupní obsah, který je základem pro vytvoření prezentace a může z něj být vytvořena webová stránka nebo dokument ve formátu PDF/ \LaTeX .
2. Prezentace – derivovaný obsah, který je ve formátu vhodném pro prezentační účely.
3. Paralelní prezentace – využívá prezentaci s derivovaným obsahem pro účely sdíleného promítání.

3.1.1 CMS

CMS je již existující komerční systém pro správu obsahu, který umí mimo jiné zpracovávat data získaná z GD a Word dokumentů a vytvářet z nich webové

3. NÁVRHOVÁ FÁZE



Obrázek 3.1: Schéma aplikační architektury

stránky. Pro účely BP je systém potřeba rozšířit o návrh modulu {M}, jehož implementace bude provedena přímo autory systému.

3.1.2 Modul exportu do HTML {M}

Modul bude implementovat standard vyžadovaný knihovnou {K}. Ten definuje požadovanou strukturu HTML dokumentu, která bude vytvořena na základě dat získaných z dokumentu GD nebo Word. Požadavky na danou strukturu jsou popsány v kapitole Syntaxe zdrojových dat 3.1.3.2.

V rámci návrhu jsme se po konzultaci rozhodli, že výsledná prezentace nebude tvořena sadou HTML dokumentů, jak je uvedeno v zadání, ale pouze jedním HTML dokumentem. Tento přístup značně usnadňuje použití samotné knihovny Tree Presenter {K}, protože uživatel již nemusí vytvářet několik HTML dokumentů včetně navigace a dalších metadat, ale postačí mu jediný dokument s obsahem prezentace.

Vše ostatní bude automaticky vygenerováno knihovnou. Další výhodou je minimalizace požadavků na server při prohlížení prezentace a možnost efektivního vytvoření schématu stromové struktury prezentace, jako SVG obrázku.

Přestože tento přístup vyžaduje generování navigace a derivace na straně klienta pomocí jazyku JavaScript, tak to neznamená, že uživatelé využívající internetový prohlížeč bez jeho podpory nebudou mít přístup k obsahu prezentace. Pouze bude omezena funkcionální prezentace, která souvisí s jazykem JavaScript.

3.1.3 Knihovna Tree Presenter {K}

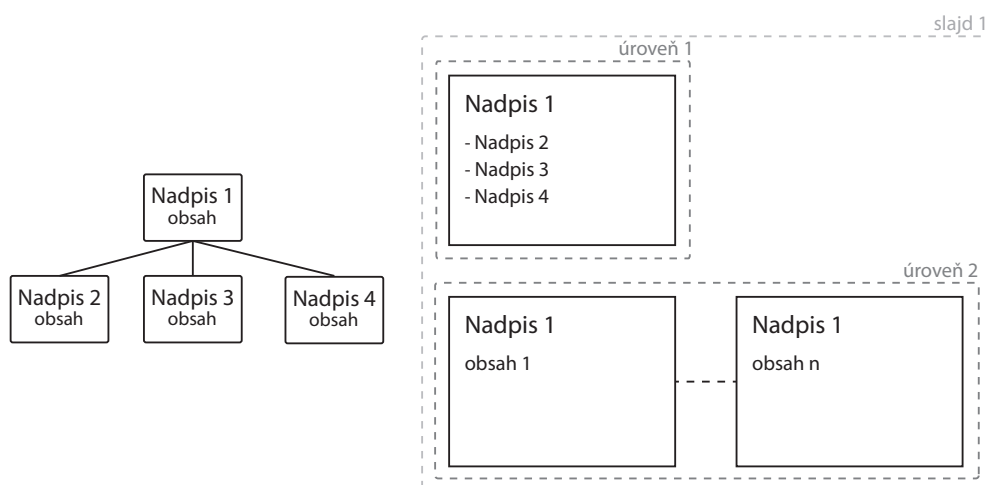
Knihovna se skládá z CSS a JS souborů, které jsou importovány dokumentem získaného z modulu {M} a zajišťuje funkcionální popsanou v kapitole Požadavky na funkce 2.3, vyjma generování prezentace z GD/Word dokumentů.

3.1.3.1 Derivace obsahu

Derivace představuje jiný pohled na obsah vytvářený v dokumentech GD a Word. Ten je založen na metodě obrácené pyramidy[3] tedy, že informace s nejvyšší prioritou se nachází na nejvyšší úrovni dokumentu a se zvětšujícím se zanořením jejich důležitost klesá.

Obsah výsledné prezentace je zobrazován ve dvou úrovních. První úroveň zobrazuje derivovaný obsah, jehož slajdy jsou vždy tvořeny nadpisem kapitoly a seznamem. Nadpis kapitoly chápeme jako vnitřní uzel ve stromové struktuře prezentace. Položky seznamu jsou tvořeny podnadpisy nadpisu kapitoly neboli potomky uzlu.

Pokud kapitola, z které je slajd vytvořen má nějaký obsah, tak se lze přepnout do tzv. druhé úrovně. Prezentace na této úrovni má pouze lineární strukturu a její obsah už není derivovaný a je shodný s obsahem kapitoly.



Obrázek 3.2: Schéma derivace obsahu

To znamená, že první úroveň umožňuje pohyb celou stromovou prezentací a v případě potřeby se lze přepnout do druhé úrovně v rámci konkrétního slajdu.

U kapitol, které se neskládají z žádných podkapitol nelze vytvořit derivaci, protože jsou listy stromu a tedy nemají žádné potomky, kteří tvoří obsah slajdů na první úrovni. V takovém případě slajd nemá první úroveň a při jeho navštívení uživatel vstoupí rovnou do úrovně druhé. Pro lepší pochopení je derivace znázorněna na obrázku 3.2, kde jsou zobrazeny výsledné úrovně pro první slajd prezentace.

Původní návrh derivace, který byl navrhován jako další modul do CMS je součástí přílohy B.

3.1.3.2 Syntaxe zdrojových dat

Aby bylo možné zdrojová data korektně zpracovat a následně z nich vytvořit stromovou prezentaci, tak uživatel musí v dokumentech využívající {K} dodržovat syntaxi definovanou v uživatelském manuálu (příloha D). Odůvodnění jednotlivých požadavků:

- Jeden komplexní HTML soubor.
 - Sada souborů, kde by každý slajd prezentace představoval jeden soubor by uživateli značně zesložila tvorbu prezentace. Z tohoto důvodu je vyžadován veškerý obsah v jednom souboru.
- Jazyk HTML5.
 - Knihovna je v současnosti omezena na standard HTML5, který byl zvolen z důvodu jeho rozšířenosti. V budoucnu lze podporu rozšířit o standardy HTML4 a XHTML.

- Validní syntaxe.
 - Algoritmus vytvářející stromovou strukturu předpokládá na vstupu validní kód. V opačném případě není zaručeno správné vygenerování prezentace a její funkčnost.
- Existence identifikátoru nadpisů.
 - Identifikátory umožňují odkazování na slajdy prezentace i bez použití JavaScriptu a mají tedy přímý vliv na robustnost prezentace.

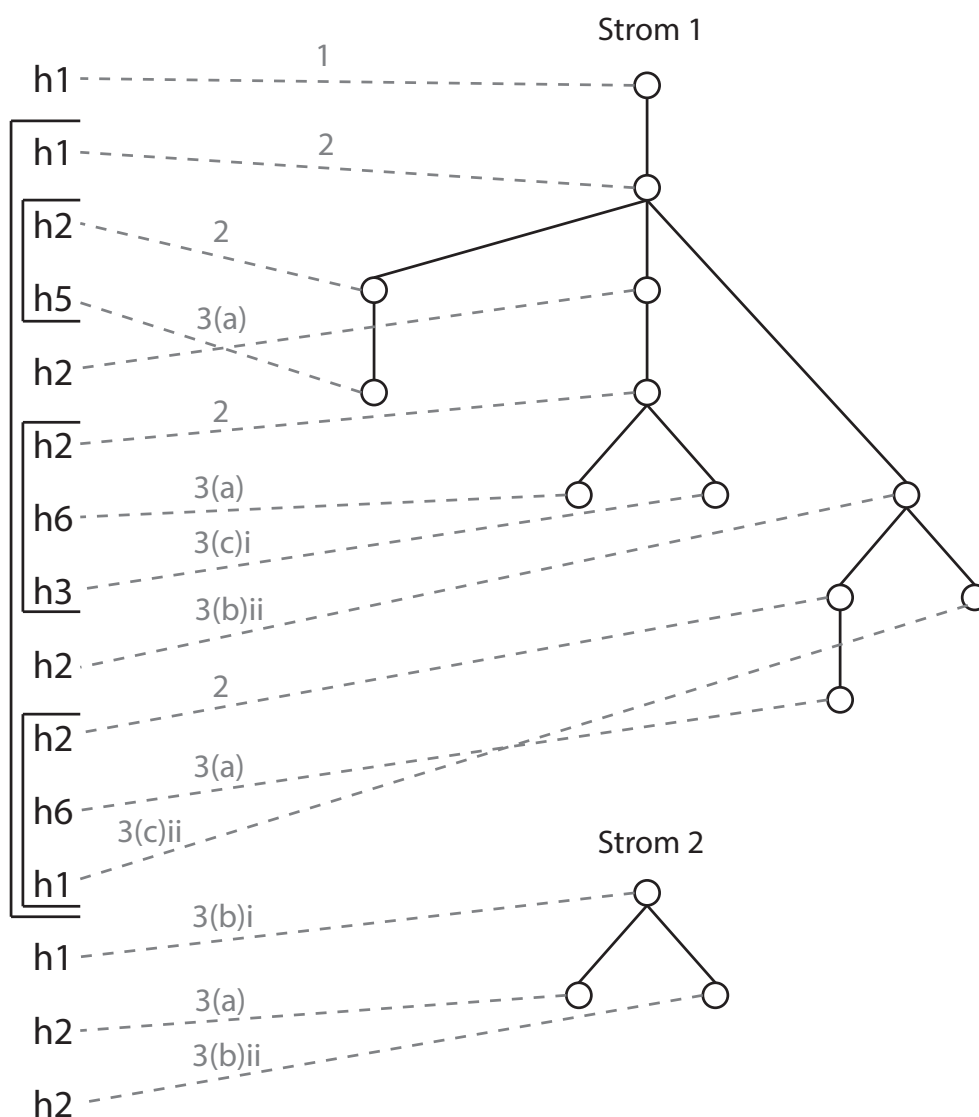
V případě použití jiných technologií nebo nevalidních souborů není zaručena kompatibilita a knihovna by měla o této skutečnosti uživatele informovat.

3.1.3.3 Algoritmus tvorby stromové struktury z HTML5

Zdrojové data ve formátu HTML5 jsou zpracovány podle následujícího algoritmu, který vychází ze specifikace W3S[4], kde jednotlivé nadpisy představují slajdy prezentace.

Algoritmus se skládá z dvou hlavních kroků: načtení všech nadpisů a postupné vyhodnocení jejich úrovně. Úroveň nadpisů záleží na jeho hodnotě uvedené v kódu (*h1* má nejvyšší hodnotu a *h6* nejnižší), zanoření v elementu *section* a na předcházejících nadpisech ve stejné sekci. Vyhodnocení výsledné úrovně nadpisů probíhá následujícím způsobem pro každý nadpis:

1. První nadpis je kořenem stromu a tedy má úroveň 1 bez ohledu na ostatní faktory (zanoření a úroveň nadpisu v kódu).
2. První nadpis uvnitř elementu *section* má úroveň rovnou jeho rodiči zvýšené o 1.
3. V případě, že nadpis není první v rámci elementu *section* nebo *body*, tak dojde k vyhodnocení podle relativní úrovně vůči předcházejícím nadpisům.
 - a) Pokud hodnota nadpisu je menší, než nadpisu předcházejícího ve stejné sekci, tak jeho výsledná úroveň je rovna úrovni předcházejícího nadpisu zvýšené o 1.
 - b) Pokud hodnota nadpisu je rovna hodnotě předcházejícího nadpisu ve stejné sekci a:
 - i. *body* je rodičovským elementem, tak je vytvořen nový strom, jehož kořen je daný nadpis,
 - ii. *section* je rodičovským elementem, tak úroveň nadpisu je rovna jeho předchůdci.
 - c) Pokud hodnota nadpisu je větší, než hodnota nadpisu předcházejícího ve stejné sekci a zároveň je:



Obrázek 3.3: Schéma tvorby stromové struktury z HTML5

- i. menší, než hodnost hlavního nadpisu sekce, tak úroveň je rovna předcházejícímu nadpisu,
- ii. větší, než hodnost hlavního nadpisu sekce, tak úroveň nadpisu je rovna úrovni hlavního nadpisu dané sekce.

Pro lepší pochopení algoritmu je postup zpracování zobrazen na obrázku 3.3, kde je u každého nadpisu je vyznačeno pravidlo z popisu algoritmu, podle kterého byl uzel ve stromu vytvářen.

Některé struktury dokumentu mohou vést k neintuitivním výsledkům, které jsou ovšem v souladu se specifikací HTML5. Například v případě 3(b)i, kde se vytváří další strom a výsledkem tedy není jediná stromová struktura, ale celý les. V takovém případě bude vytvořeno více prezentací s možností jejich přepínání.

3.1.3.4 Exportu do PDF/L^AT_EX

Prezentaci vytvořenou z HTML lze exportovat do dokumentu PDF nebo L^AT_EX. Prezentace v těchto dokumentech bude mít lineární strukturu na rozdíl od webové prezentace, která má strukturu stromovou a je tedy potřeba obsah linearizovat. K tomu bude využit algoritmus pro průchod stromem v pořadí preorder[5]. Alternativní algoritmus je součástí přílohy C.

Obsah exportovaných prezentací je shodný s derivací obsahu první úrovně, která je popsána v kapitole 3.1.3.1. Tedy slajdy prezentace jsou tvořeny vnitřními uzly stromové struktury a jejich obsah je roven seznamu, jehož položky jsou nadpisy potomků vnitřního uzlu.

3.1.4 Paralelní proxy

Jedná se o službu, která využívá webovou prezentaci s knihovnou {K} pro tzv. paralelní promítání. Paralelní promítání umožňuje distribuovat jednu instanci prezentace, jenž je řízená prezentujícím mezi libovolný počet uživatelů. Během promítání mohou ke službě být připojeny dva typy zařízení:

1. Řídící zařízení – neboli prezentující sdílející URL adresu své prezentace, kterou může sdílet s libovolným počtem posluchačů.
2. Přijímací zařízení – neboli posluchač, který využívá URL adresu poskytnutou prezentujícím. To mu umožňuje sledovat prezentaci spolu s přednášením na vlastním zařízení.

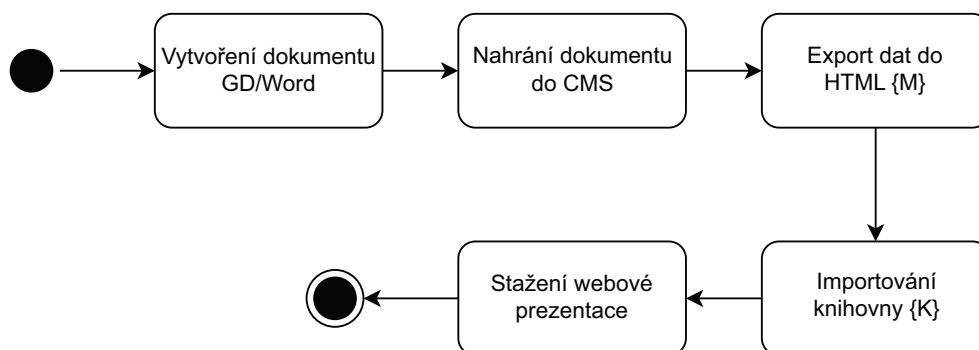
3.2 Procesy

Požadavky na funkce jsou zpracovány do čtyř základních procesů s identifikátory {P1–4}, přičemž každý pro svojí realizaci využívá modul {M} nebo knihovnu {K}. Procesy {P5–6} představují již existující funkce služby CMS, které mohou být využity při tvorbě prezentace.

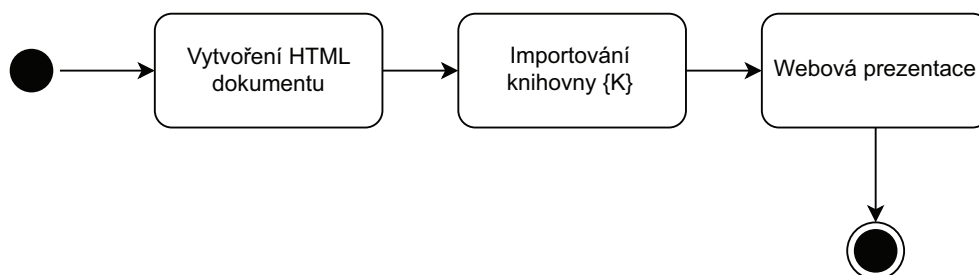
3.2.1 Vytvoření prezentace z GD/Word {P1}

Uživatel vytvoří GD nebo Word dokument, z kterého je následně prostřednictvím CMS vytvořena online prezentace (diagram 3.4).

1. Vytvoření GD/Word dokumentu.



Obrázek 3.4: Proces 1



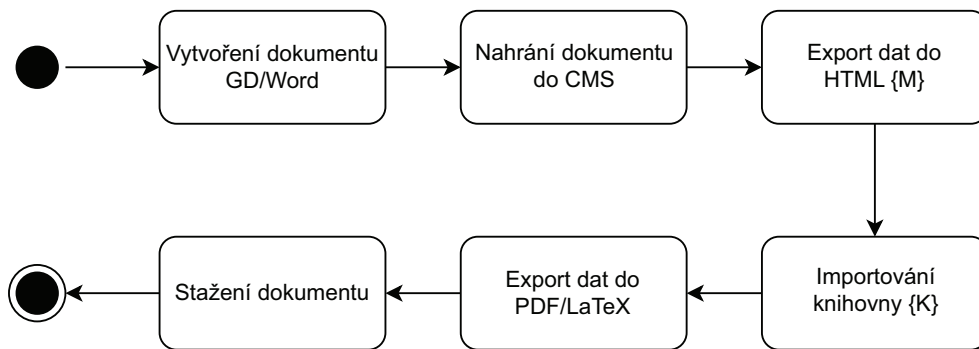
Obrázek 3.5: Proces 2

2. Nahrání dokumentu do CMS.
3. Export dat do HTML {M}.
4. Importování knihovny {K}.
5. Stažení webové prezentace.

3.2.2 Vytvoření prezentace z HTML {P2}

Jedná se o nejjednodušší způsob vytvoření stromové prezentace, protože nevyužívá generování dokumentů pomocí služby CMS. Uživatel vytvoří pouze HTML dokument dodržující syntaxi zdrojových dat a importující knihovnu {K} (diagram 3.5).

1. Vytvoření HTML dokumentu.
2. Importování knihovny {K}.
3. Webová prezentace.



Obrázek 3.6: Proces 3

3.2.3 Vytvoření offline prezentace z GD/Word {P3}

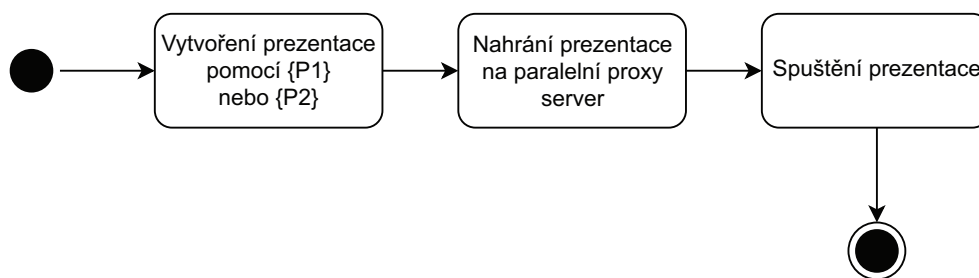
Uživatel vytvoří GD nebo Word dokument, z kterého je pomocí CMS a knihovny {K} vygenerován soubor typu PDF/LaTeX. Tyto soubory mají shodný obsah s první úrovní online prezentace, ale jejich struktura je lineární (diagram 3.6).

1. Vytvoření GD/Word dokumentu.
2. Nahrání dokumentu do CMS.
3. Export dat do HTML {M}.
4. Importování knihovny {K}.
5. Export dat do PDF/LaTeX.
6. Stažení dokumentu.

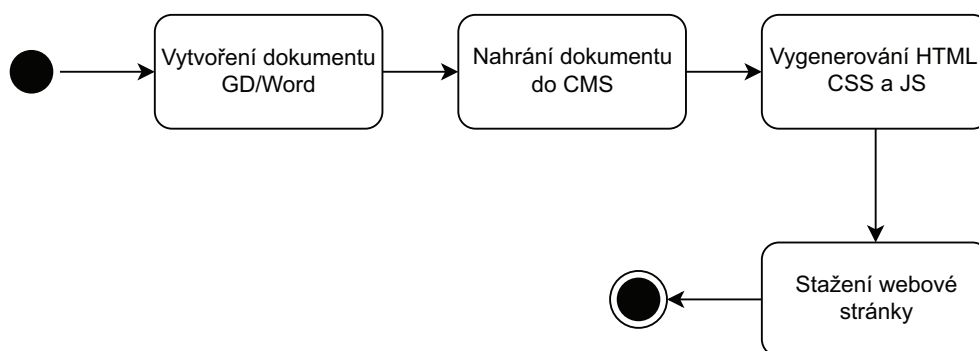
3.2.4 Paralelní promítání {P4}

Pro využití sdíleného neboli paralelního promítání je zapotřebí online prezentace, kterou lze vytvořit pomocí procesů {P1} nebo {P2}. Výsledná prezentace musí být nahrána na paralelní proxy server, který zajišťuje připojení a obsluhu řídicích a přijímacích zařízení (diagram 3.7).

1. Vytvoření prezentace pomocí {P1} nebo {P2}.
2. Nahrání prezentace na paralelní proxy server.



Obrázek 3.7: Proces 4



Obrázek 3.8: Proces 5

3.2.5 Vytvoření webové stránky z GD/Word {P5}

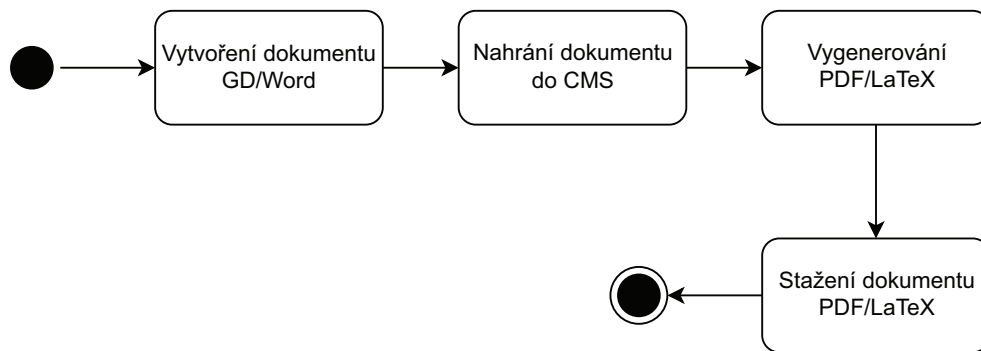
Uživatel vytvoří GD nebo Word dokument, který CMS zpracuje a na základě jeho obsahu vygeneruje sadu souborů HTML, CSS a JS tvořící webovou stránku (diagram 3.8).

1. Vytvoření GD/Word dokumentu.
2. Nahrání dokumentu do CMS.
3. Vygenerování HTML, CSS a JS.
4. Stažení webové stránky.

3.2.6 Export GD/Word do PDF/L^AT_EX dokumentu {P6}

Dokumenty ve formátu GD nebo Word jsou prostřednictvím CMS exportovány do souborů PDF a L^AT_EX (diagram 3.9).

1. Vytvoření GD/Word dokumentu.
2. Nahrání dokumentu do CMS.



Obrázek 3.9: Proces 6

3. Vygenerování PDF/L^AT_EX.
4. Stažení dokumentů PDF/L^AT_EX.

3.3 Případy užití

Případy užití přímo vycházejí z předcházející kapitoly Procesy 3.2, přičemž každý se skládá z krátkého příběhu, pokrývaných procesů a možného toku událostí.

3.3.1 UC1: Vytvoření prezentace z dokumentu

Kamarád vám ukázal stromovou prezentaci, která vás velmi zaujala. Ovšem jste nikdy neslyšel/a o HTML5 a vytváření prezentace pomocí daného značkovacího jazyka vás odrazuje od použití nástroje pro tvorbu stromových prezentací.

- Pokryté procesy

– {P1}

- Tok událostí

1. Uživatel vytvoří GD/Word dokument.
2. Nahraje ho do CMS.
3. CMS vytvoří webovou prezentaci se stromovou strukturou.

3.3.2 UC2: Prezentace pomocí

Už vám nepostačují prezentace vytvořené v PowerPointu a chcete vyzkoušet nějaký nový nástroj s nelineární strukturou výsledného obsahu, který vám navíc umožní správu prezentací pomocí verzovacího systému.

- Pokryté procesy
 - {P2}
- Tok událostí
 1. Uživatel vytvoří HTML5 soubor.
 2. Do vytvořeného souboru importuje knihovnu Tree Presenter {K}.
 3. Vytvoří obsah prezentace na základě uživatelského manuálu.

3.3.3 UC3: Offline prezentace z dokumentu

Využíváte webové prezentace s použitím Tree Presenteru a chcete si je vždy znova prostudovat, než je budete prezentovat. Preferujete papírovou formu, protože vám umožňuje vytvoření poznámek u jednotlivých slajdů, které používáte během prezentování. Takže si prezentaci stáhnete ve formátu PDF a vytisknete.

- Pokryté procesy
 - {P3}
- Tok událostí
 1. Uživatel nahraje GD/Word dokument do CMS.
 2. Vybere požadovaný výstupní formát prezentace (PDF/L^AT_EX).
 3. Stáhne prezentaci ve zvoleném formátu.

3.3.4 UC4: Sdílené promítání

Přednášíte na vysoké škole a v dalším semestru budete pravidelně přednášet v místnosti, kde je nekvalitní projektor a studenti si často stěžují, že obsah prezentace je nečitelný. Rozhodnete se jim dát možnost sledovat slajdy s vámi prostřednictvím paralelního promítání.

- Pokryté procesy
 - {P1}
 - {P2}

– {P4}

- Tok událostí
 1. Uživatel si vytvoří stromovou prezentaci.
 2. Nastaví jí na paralelním proxy serveru.
 3. Během přednášky studentům nasdílí odkaz na prezentaci, který mu poskytl server.

3.3.5 UC5: Vytvoření webové stránky

Jste vlastníkem hotelu a pro zvýšení jeho návštěvnosti si chcete zařídit webové stránky. Avšak nemáte zkušenosti ani dostatek prostředků. Vytvoříte si tedy obsah webu v dokumentu Word a využijete služby CMS.

- Pokryté procesy
 - {P5}
- Tok událostí
 1. Uživatel vytvoří GD/Word dokument.
 2. Nahraje ho do CMS.
 3. Stáhne výslednou webovou stránku ve formě sady HTML, CSS, JS souborů.

3.3.6 UC6: Export ve formátu \LaTeX

Máte velké množství obsahu ve formátu GD/Word a rozhodl jste se vyzkoušet sázeací nástroj \LaTeX . Namísto přepisování využijete službu CMS.

- Pokryté procesy
 - {P6}
- Tok událostí
 1. Uživatel nahraje dokument ve formátu GD/Word do CMS.
 2. Zvolí požadovaný výstupní formát.
 3. Stáhne dokument v požadovaném formátu.

3.4 Prototypování

Z počátku byly veškeré prototypy vytvářeny v papírové podobě a až poté byly překresleny pomocí nástroje Adobe Illustrator do formátu SVG. Při jejich návrhu byl kladen důraz na funkční vzhled uživatelského prostředí, jehož výsledná podoba je založena na existujících aplikacích.

Principiální rozložení obsahu prezentace je inspirováno webovou aplikací Prezentace Google a navigační prvky, zejména primární navigace knihovnou Reveal.js.

Při návrhu nebyly opominuty ani prohlížeče bez podpory jazyku JavaScript, které budou mít přístup k veškerému obsahu prezentace s omezením funkcí spojených s tímto jazykem.

3.4.1 Typy stránek

Výsledná webová prezentace se skládá z několika typů stránek, které buď zobrazují přímo obsah prezentace nebo slouží k navigaci v prezentaci a jejímu nastavení. Každá stránka se zobrazuje přes plnou velikost průhledu obrazovky a je přístupná pomocí klávesových zkratk.

3.4.1.1 Šablony vzhledu

Prezentace využívá šablony s různým principiálním rozložením obsahu na stránce. Tyto šablony jsou převzaty z webové aplikace Prezentace Google¹⁸ a byly zvoleny z důvodu jejich jednoduchého rozložení prvků, které je již otestováno širokou škálou uživatelů a tedy vymýšlení nových by bylo nepřínosné a neefektivní.

3.4.1.2 Navigační obrazovka

Pro lepší přehled o struktuře a aktuální pozici v prezentaci se generuje schéma celé stromové struktury prezentace. Jednotlivé uzly stromu (slajdy) jsou reprezentovány nadpisem daného slajdu.

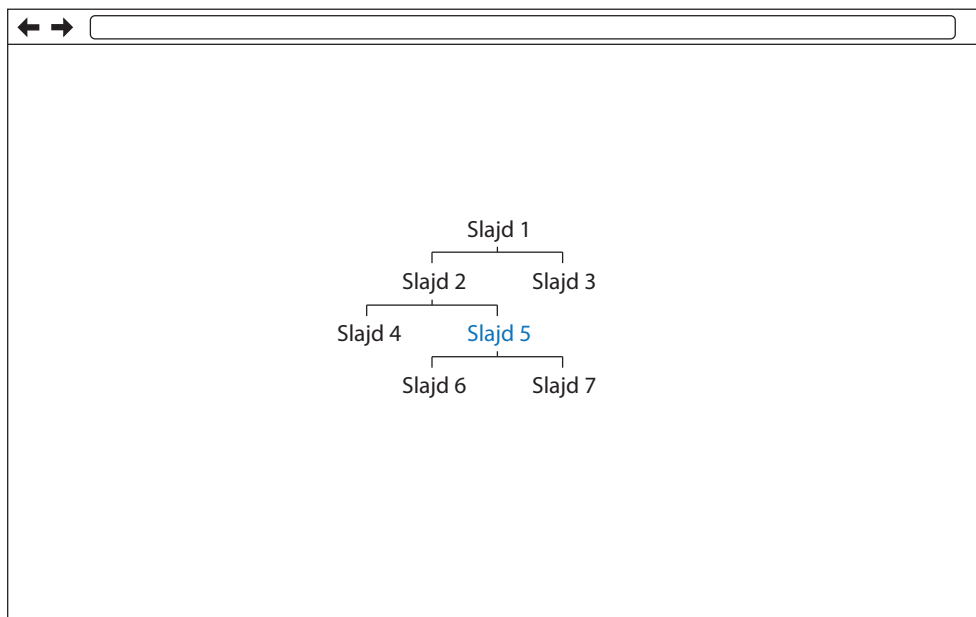
Aktuální pozice v prezentaci je barevně zvýrazněna a pozici lze změnit kliknutím myši na libovolný uzel. Návrh obrazovky 3.10.

3.4.1.3 Nastavení

Customizace prezentace je možná z této stránky, která se zobrazí přes celou obrazovku, jejíž prototyp je na obrázku 3.11. Nastavení se skládá z posuvných tlačítek symbolizující stav zapnuto/vypnuto a výběrových tlačítek. V nastavení lze:

- zapnout/vypnout schovávací navigaci,

¹⁸https://www.google.com/intl/cs_cz/slides/about/



Obrázek 3.10: Prototyp – navigační obrazovka

- zapnout lineární průchod prezentací,
- zvolit poměr stran,
- zvolit barevné téma.

3.4.1.4 Klávesové zkratky

Vzhledem k tomu, že prezentace nabízí několik typů stánek, které jsou primárně přístupné přes klávesové zkratky, tak byla navržena stránka zobrazující jejich kompletní výčet a fungující, jako nápověda. Přístup pomocí klávesových zkratek byl zvolen, protože větší počet prvků umístěných přímo v prezentaci umožňující přístup i pomocí myši by mohl působit rušivě. Návrh obrazovky 3.12.

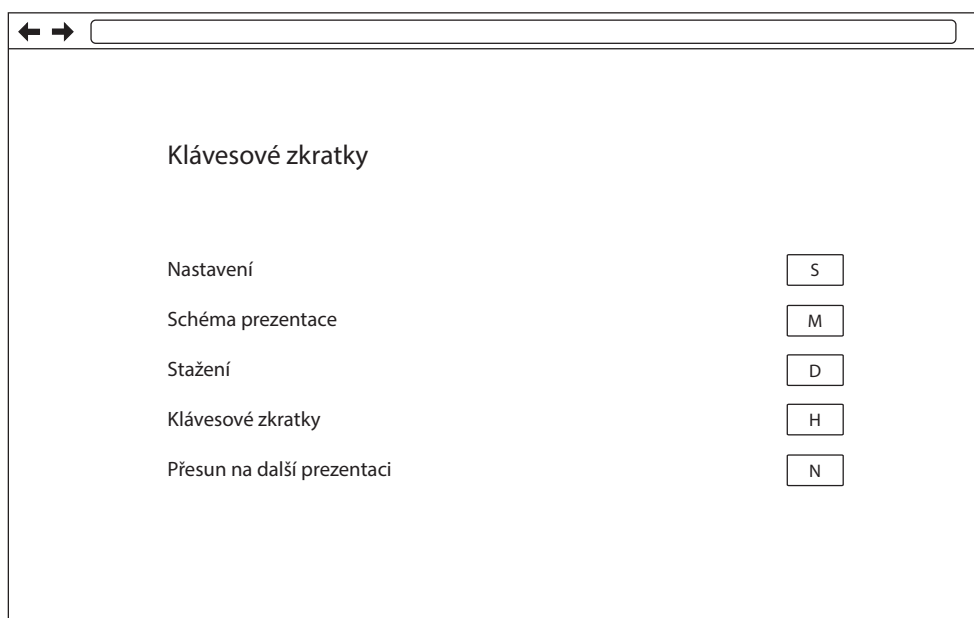
3.4.1.5 Stažení

Stránka stažení dává uživateli možnost výběru formátu, v kterém bude prezentace stažena. Na výběr je z možností HTML, PDF nebo $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a ke každé existuje přidružené tlačítko „Stáhnout“. Návrh obrazovky 3.13.

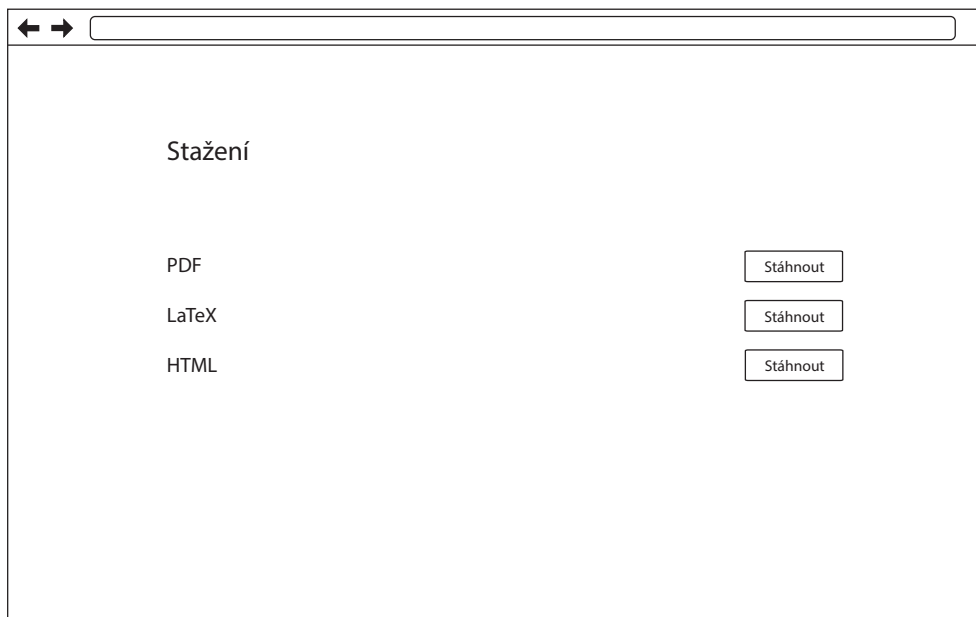
3. NÁVRHOVÁ FÁZE



Obrázek 3.11: Prototyp – obrazovka nastavení



Obrázek 3.12: Prototyp – obrazovka klávesové zkratky



Obrázek 3.13: Prototyp – obrazovka stažení

3.4.2 Navigační prvky

Na stránce se nachází dva typy navigačních prvků 3.14, které je možné v případě potřeby skrýt v nastavení prezentace. Primární navigace se nachází v pravém dolním rohu a navigační ikony jsou ukotveny v levém dolním rohu.

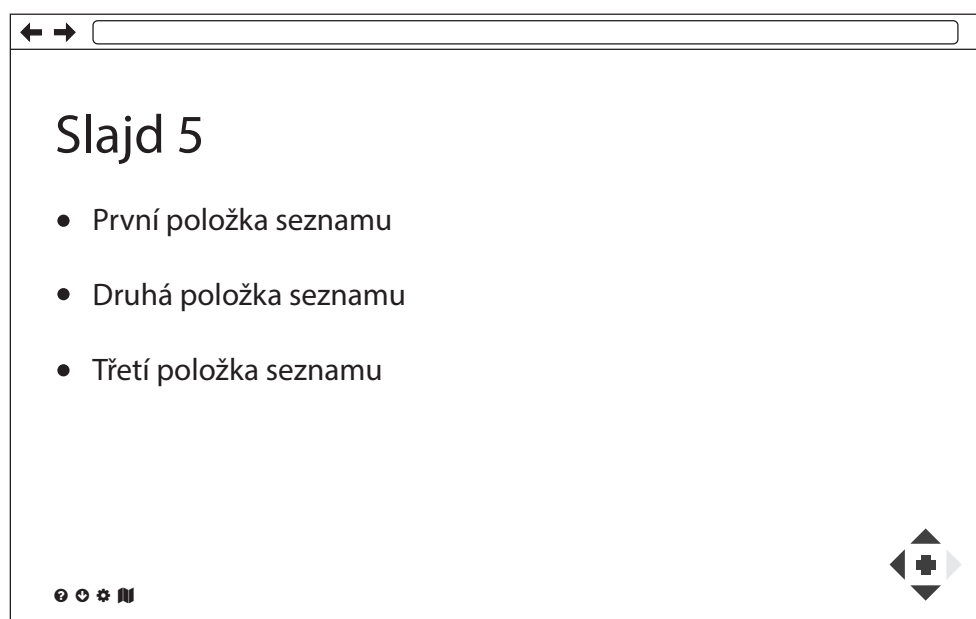
3.4.2.1 Primární navigace

Primární navigace se skládá z pěti prvků, které jsou zobrazeny na 3.15 a umožňují přesun na:

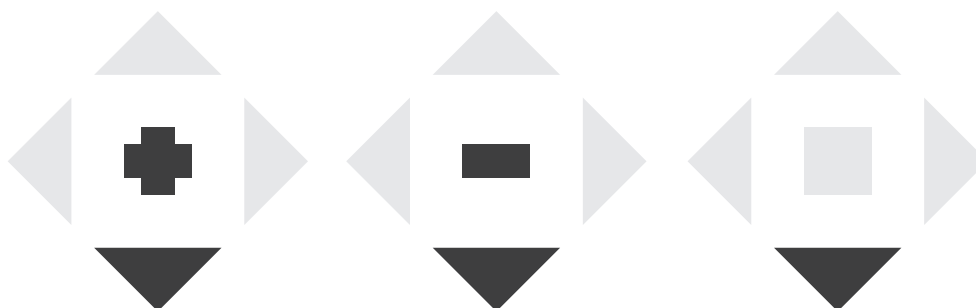
- rodiče,
- nejlevějšího potomka,
- další slajd,
- předcházející slajd,
- druhou nebo první úroveň prezentace.

Pokud obsah prezentace přesun daným směrem umožňuje, tak je trojúhelník barevně zvýrazněn. Prostřední navigační prvek umožňující přesun na jinou úroveň prezentace může být tvořen třemi symboly:

1. plus – přesun z první úrovně na druhou,



Obrázek 3.14: Prototyp – rozložení navigačních prvků



Obrázek 3.15: Primární navigace

2. mínus – přesun z druhé úrovně na první,
3. čtverec – slajd se skládá pouze z jedné úrovně.

3.4.2.2 Navigační ikony

V levém dolním rohu jsou ukotveny celkem čtyři navigační ikony, které zobrazí příslušnou obrazovku v případě jejich vybrání. Použité ikony jsou vybrané ze sady Font Awesome¹⁹. Ikony přiřazené k jednotlivým stránkám:

- otazník – klávesové zkratky,

¹⁹<http://fontawesome.io>

- šipka dolů – stažení,
- ozubené kolo – nastavení,
- mapa – navigační obrazovka.

3.4.3 Nepodporovaný JavaScript

Pokud prohlížeč nenabízí podporu nebo nemá aktivní jazyk JavaScript je funkčnost prezentace omezena. Nejsou vygenerovány navigační prvky a prezentaci nelze customizovat. Ovšem veškerý obsah prezentace je stále přístupný a to v základní podobě po aplikování robustních stylů. Ty jsou určeny pouze pro prohlížeče bez JavaScriptu a sdílejí základní styly, jako font písma a odřádky seznamů spolu s dynamickými styly.

3.4.4 Prezentace ve formátu PDF a \LaTeX

Struktura prezentací generovaných ve formátu PDF a \LaTeX je zjednodušená a slajdy se skládají pouze z nadpisu slajdu a jeho obsahu, který je tvořen seznamem. Zjednodušení bylo provedeno, protože prezentace mají pouze lineární strukturu a tedy vícesměrová navigace nebo schéma prezentace nedává v tomto případě smysl.

Implementační fáze

Na základě návrhu byla vytvořena knihovna Tree Presenter v jazyce JavaScript. Ta až na generování prezentace z dokumentů GD a Word, které je řešeno na straně CMS splňuje veškeré požadavky na funkčnost a kvality s prioritou [MUST] a [SHOULD].

První část implementační fáze se věnuje konceptům řešení jednotlivých požadavků. Poté jsou v rámci technické specifikace shrnuty technologie a nástroje, které jsou knihovnou využívány a nakonec je popsáno testování korektního zobrazení a generování prezentace.

4.1 Koncepty řešení dílčích požadavků

Kapitola popisuje výslednou implementaci vybraných požadavků na funkce a kvality.

4.1.1 Stromová struktura prezentace

Stromová struktura je vytvořena na základě obsahu HTML dokumentu pomocí HTML5 outliner²⁰, který je dostupný pod licencí MIT²¹, jenž umožňuje využití v dané práci. Implementace algoritmu popsaného v kapitole 3.1.3.3 nebyla nutná z důvodu nalezení již existujícího řešení v jazyce JavaScript zmíněného výše.

Algoritmus zpracuje obsah HTML5 a jeho výstupem je struktura typu les nebo strom, který ve svých uzlech obsahuje referenci na nadpis a jeho potomky v DOM²², kde jednotlivé uzly stromu představují slajdy prezentace. Pokud je výstupem les, tak každý jeho strom vytvoří jednu prezentaci. Pro naše účely je potřeba výsledný strom modifikovat tak, aby v jednotlivých uzlech byl i obsah prezentace a jeho derivace.

²⁰<https://github.com/h5o/h5o-js>

²¹https://cs.wikipedia.org/wiki/Licence_MIT

²²<https://www.w3.org/DOM/>

Modifikace je provedena průchodem stromu, kde v každém navštíveném uzlu je nalezen obsah prezentace a uložen, jako reference. Obsah je tvořen elementy, které následují za nadpisem až po element tvořící sekci (article, aside, nav, section) nebo další nadpis. Pokud se jedná o uzel vnitřní, tak je vytvořena i jeho derivace, která je popsána v kapitole 3.1.3.1.

4.1.2 Vizualizace prezentace

Výsledná prezentace a obrazovky umožňující zobrazení schématu, nápovědy nebo nastavení prezentace jsou zobrazeny přes plnou šířku a výšku průhledu obrazovky. Pro tyto účely je do DOM vygenerována následující struktura:

```
<div id="tree-presenter">
  <div id="tp-tree-structure"></div>
  <div id="tp-download"></div>
  <div id="tp-help"></div>
  <div id="tp-settings"></div>
  <div id="tp-slides"></div>
</div>
```

Vnější element s identifikátorem „tree-presenter“ zabírá plnou šířku a výšku obrazovky. Uvnitř je pět elementů typu *div*, přičemž viditelný neboli v popředí je vždy pouze jeden z nich, což závisí na akcích uživatele. Jednotlivé elementy obsahují:

- schéma stromové struktury,
- odkazy ke stažení prezentace v PDF a \LaTeX ,
- nápovědu s výčtem klávesových zkratk,
- nastavení prezentace,
- slajdy prezentace.

Obsah elementu s identifikátorem „tp-slides“ se mění v závislosti na pozici v prezentaci. Tedy v případě přesunu na jiný slajd se změní aktivní uzel ve stromu a jeho obsah je vložen dovnitř daného elementu.

Tento přístup, kdy pro vizualizaci prezentace využíváme vlastní strukturu namísto uživatelem vytvořené, z které je generována stromová struktura přináší řadu výhod.

1. Jediný zobrazovaný obsah je uvnitř obalujícího elementu s identifikátorem „tree-presenter“, takže prezentaci nemohou ovlivnit případné další elementy vytvořené uživatelem uvnitř dokumentu.
2. Jednoduché přizpůsobení prezentace v podobě nastavení poměru stran nebo barev, kdy stačí pracovat s obalujícím elementem.

3. Případné rozšíření prezentace o zápatí, záhlaví nebo logo lze realizovat pouhým vložením obsahu do elementu s identifikátorem „tp-slides“.

Korektní zobrazení prezentace na různých projektorech a obrazovkách je zajištěno pomocí CSS3 media pravidel[6], které definují velikosti písma a ovládacích prvků pro různé rozlišení a typy zobrazovacích zařízení.

4.1.3 Export do PDF a \LaTeX

Export je implementován na základě návrhu 3.1.3.4. Tedy obsah je tvořen první úrovní prezentace, která byla linearizována průchodem stromu v pořadí preorder. Pro tvorbu PDF je využita knihovna pdfmake²³ umožňující generování PDF přímo na straně uživatele pomocí JavaScriptu.

V případě \LaTeX je vytvořen soubor, jehož obsah dodržuje syntaxi definovanou rozšiřující třídou sazecího systému Beamer²⁴, která se běžně využívá pro tvorbu \LaTeX prezentací.

4.1.4 Navigace a orientace v prezentaci

V prezentaci jsou dva typy navigace, které zároveň slouží, jako orientační prvky.

4.1.4.1 Primární navigace

Primární navigace je tvořena pěti prvky, které jsou vygenerovány do HTML, jako elementy typu *div* a nastýlovány pomocí CSS. Jejich zvýraznění a tvar se mění dynamicky v závislosti na aktuální úrovni a pozici ve stromové struktuře prezentace. V případě první úrovně, pokud uzel má:

- rodiče – zvýrazní se šipka nahoru,
- potomka – zvýrazní se šipka dolů,
- levého sourozence – zvýrazní se šipka doleva,
- pravého sourozence – zvýrazní se šipka doprava,
- nederivovaný obsah – zobrazí se symbol plus.

V druhé úrovni:

- šipka nahoru je vždy zvýrazněna,
- šipka dolů není nikdy zvýrazněna,

²³www.pdfmake.org

²⁴<https://cs.sharelatex.com/learn/Beamer#/Introduction>

4. IMPLEMENTAČNÍ FÁZE

- šipka doprava zvýrazněna, pokud je dostupný další obsah nebo existuje pravý sourozenec,
- šipka doleva zvýrazněna, pokud se nejedná o první část obsahu v rámci slajdu nebo existuje levý sourozenec,
- symbol mínus se zobrazí, pokud se jedná o vnitřní uzel.

4.1.4.2 Schéma prezentace

Schéma prezentace je generováno do stránky ve formě SVG. Pro tento účel je využita knihovna Treant-js²⁵. Ta pro vytvoření schématu vyžaduje přesně strukturovaný JSON²⁶ dle specifikace²⁷. Požadovaný JSON je vytvořen průchodem stromu prezentace, kde jako název uzlu je použit obsah nadpisu slajdu prezentace.

Aby schéma nesloužilo pouze k orientačním účelům, ale dalo se využít i k navigaci v prezentaci, tak při průchodu stromu je k jednotlivým uzlům přiřazena CSS třída s hodnotou rovnou identifikátoru uzlu ve stromu. Tato třída je následně využita k identifikaci uzlu v případě, kdy uživatel na některý z nich klikne ve výsledném schématu.

4.1.5 Ovládání prezentace

Pro ovládání prezentace lze využít následující nástroje a gesta.

4.1.5.1 Klávesnice a myš

Součástí knihovny TP jsou posluchače[7], které naslouchají na stránce s aktivní webovou prezentací. Ve chvíli, kdy uživatel stiskne tlačítko na klávesnici je tato akce zaznamenána. Pokud stisknutá klávesa odpovídá některé z definovaných zkratek, tak je vyvolána příslušná akce. Definované klávesové zkratky jsou popsány v tabulce 4.1.

Ovládání myší je možné pomocí hlavní navigace, navigačních ikon nebo schématu prezentace a funguje na stejném principu, jako ovládání klávesnicí. Tedy na HTML elementy tvořící příslušnou navigaci jsou navěšené posluchače, které zachytí případné kliknutí a vykonají předdefinovanou akci.

4.1.5.2 Dálkové ovládání (prezentéry)

Pro umožnění ovládání prezentace pomocí dálkových ovládání není v případě běžných prezentací zapotřebí další implementace. Protože stisk tlačítka pro následující nebo předcházející slajd na dálkovém ovládání je shodný se stiskem

²⁵<http://fperucic.github.io/treant-js/>

²⁶https://cs.wikipedia.org/wiki/JavaScript_Object_Notation

²⁷<http://fperucic.github.io/treant-js/#orgchart-initialization>

Tabulka 4.1: Klávesové zkratky

Klávesová zkratka	Akce
Šipky	Přesun mezi slajdy
Enter	Přesun mezi druhou a první úrovní prezentace
n/N	Přesun na další/předchozí prezentaci (pokud existuje)
S	Nastavení
M	Schéma prezentace
D	Stahování
H	Klávesové zkratky/nápověda

šipky doprava nebo doleva na klávesnici a je tedy zachycen stejným posluchačem, který umožňuje ovládání pomocí klávesnice.

U stromové prezentace může nastat problém, pokud dálkové ovládání má pouze tlačítka umožňující přesun na předcházející a následující slajd a tedy neumožňuje přesun na potomka nebo rodiče.

Z tohoto důvodu byla do nastavení prezentace přidána možnost, která spustí režim lineární navigace. Ten umožňuje pohyb po první úrovni stromové prezentace pouze pomocí tlačítek vpřed a vzad. Pohyb vpřed je umožněn pomocí následujícího algoritmu, který vychází z průchodu stromem v pořadí preorder, ale oproti známým implementacím musí rozeznávat uzly s derivací obsahu a bez ní. Další možností by bylo použití algoritmu popsáném v příloze C. Pohyb zpět je řešen obdobně.

```

findNext(node) {
  if (node.parent === null) {
    return null;
  }
  let nextSibling =
    this.findNextSibling(node.parent.rightSibling);
  if (nextSibling) {
    return nextSibling;
  }
  return this.findNext(node.parent);
}

findNextSibling(node) {
  if (node === null) {
    return null;
  }
  if (node.derivation) {
    return node;
  } else {
    return this.findNextSibling(node.rightSibling);
  }
}

```

Tabulka 4.2: Dotyková gesta umožňující ovládání prezentace

Posunutí prstu	Přesun na
Shora dolů	Rodiče
Zdola nahoru	Potomka
Zleva doprava	Levého sourozence
Zprava doleva	Pravého sourozence

```
findFollower(node) {  
  // case 1: null  
  if (node === null) {  
    return null;  
  }  
  // case 2: return first child with derivation  
  for (let i = 0; i < node.children.length; i++) {  
    if (node.children[i].derivation) {  
      return node.children[i];  
    }  
  }  
  // case3: return next sibling with derivation  
  let nextSibling = this.findNextSibling(node.rightSibling);  
  if (nextSibling) {  
    return nextSibling;  
  }  
  // case4: find next node in order  
  return this.findNext(node);  
}
```

4.1.5.3 Dotykové gesta

Pokud obrazovka, na které běží webová prezentace disponuje dotykovým ovládáním, tak je možné průchod prezentací ovládat pomocí gest popsanych v tabulce 4.2.

Rozpoznání jednotlivých gest²⁸ umožňují posluchače, které odposlouchávají akce typu:

- Touchstart,
 - uloží počáteční souřadnice dotyku na obrazovce.
- Touchmove,
 - zpracuje nově získané souřadnice dotyku a odečte je od počátečních souřadnic. Na základě výsledných hodnot je určen směr pohybu a vykonána k němu definovaná akce.

²⁸Implementace rozpoznávání gest vychází z dokumentace[8].

4.1.6 Nastavení

Nastavení umožňuje customizovat výslednou prezentaci přímo ve webovém prohlížeči prostřednictvím grafického uživatelského rozhraní.

4.1.6.1 Volba poměru stran

Prezentaci lze zobrazit ve čtyřech režimech s různým poměrem stran:

1. 4:3,
2. 16:9,
3. 16:10,
4. auto.

Čtvrté nastavení představuje plně přizpůsobivý režim, při kterém je prezentace zobrazena přes celou obrazovku. U režimů 1–3 je potřeba dopočítat požadovanou výšku a šířku prezentace. Výpočet je prováděn v následujících krocích:

1. Je spočítán poměr stran obrazovky jako:

$$pomerStran = sirkaObrazovky / vyskaObrazovky$$

2. Pokud je poměr stran obrazovky větší, než zvolený poměr stran, tak je prezentace zobrazena přes plnou výšku obrazovky a je dopočítána šířka. V opačném případě je prezentace zobrazena přes plnou šířku obrazovky a je dopočítána výška.
3. Šířka je vypočtena jako:

$$sirka = zvolenyPomer \cdot vyskaObrazovky$$

4. Výška je vypočtena jako:

$$vyska = sirkaObrazovky / zvolenyPomer$$

4.1.6.2 Vizualní přizpůsobení prezentace

Vizualní přizpůsobení lze provést výběrem barevného tématu. Pokud uživatele zvolí jedno z barevných témat, tak je elementu obalující obsah prezentace přidána CSS třída, jejíž hodnota odpovídá barvě zvoleného tématu.

Ve stylech prezentace je pro každou třídu definovaná příslušná barva, která ovlivní barvu nadpisů, hlavní navigace a schématu prezentace.

4.1.6.3 Uložení nastavení

Nastavení webové prezentace je kompletně řešeno na straně uživatele v jazyce JavaScript, takže při zavření prohlížeče, v kterém běží prezentace dojde k ztracení nastavení, jenž uživatel aplikoval na prezentaci.

Aby nastavení bylo zachováno i po zavření nebo opětovném načtení prohlížeče, tak je využito HTML5 Local Storage[9], které umožňuje ukládání dat přímo v prohlížeči uživatele. Tedy veškeré nastavení zvolené uživatelem je ukládáno do Local Storage ve formátu *klíč:hodnota*. Poté je při vytváření prezentace kontrolován obsah úložiště zda neobsahuje data relevantní pro nastavení. V kladném případě jsou uložené hodnoty aplikovány na prezentaci.

4.2 Technická specifikace

V rámci implementace vznikl uživatelský manuál²⁹ v anglickém jazyce, který je součástí repozitáře knihovny Tree Presenter³⁰ na službě GitLab a přílohy D. Mimo jiné manuál obsahuje souhrn použitých technologií a nezbytné informace pro uživatele, kteří by chtěli knihovnu technicky použít nebo rozšiřovat.

Logika knihovna je implementována v jazyce JavaScript verze ES6 a je rozdělena do několika tříd. Kód pro produkční nasazení je kompilovaný pomocí nástroje rollup.js³¹, který vytvoří z několika vzájemně provázaných souborů jeden komplexní. Dále je pro lepší podporu starších prohlížečů výsledný soubor zkompilován do starší verze JavaScriptu ES5 nástrojem Babel³².

Vizuální podoba prezentace je přizpůsobena pomocí jazyku SASS³³, který je kompilován do CSS. SASS byl zvolen, protože rozšiřuje kaskádové styly o proměnné, funkce a moduly, které zvyšují přehlednost kódu a usnadňují jeho další vývoj. Obsah prezentace je strukturované HTML5.

Celá knihovna využívá správce balíčku npm³⁴, který automatizuje instalaci, aktualizaci a konfiguraci celé knihovny.

4.3 Testování

Poslední částí implementační fáze bylo testování, které se soustředilo na generování stromové struktury prezentace z HTML dokumentu a správné zobrazení napříč různými rozlišeními obrazovek.

Pro ověření generování prezentace byly vytvořeny vzorové dokumenty s očekávaným výstupem (příloha E). Obsah dokumentů a očekávaný výstup vy-

²⁹Uživatelský manuál byl vytvářen formou ReadMe[10]

³⁰<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter>

³¹<https://github.com/rollup/rollup>

³²<https://babeljs.io/>

³³<http://sass-lang.com/>

³⁴<https://www.npmjs.com/>

chází přímo ze specifikace HTML5³⁵. Vzhledem k poměrně složité automatizaci tohoto procesu bylo výsledné porovnávání prováděno na manuální úrovni.

Správné zobrazení prezentace bylo testováno prostřednictvím vývojářského nástroje Google Chrome³⁶. Ten nabízí otestování na řadě předdefinovaných profilů zařízení s možností vytvoření vlastních. Pro účely Tree Presenteru byla vytvořena sada profilů těch nejvíce používaných rozlišení³⁷, na kterých bylo ověřeno správné rozložení prvků a jejich čitelnost.

³⁵<https://www.w3.org/TR/html5/sections.html>

³⁶<https://developer.chrome.com/devtools>

³⁷Nejvíce používané rozlišení byla vybrána na základě dat z <https://www.projectorpeople.com/resources/resolution-guide.asp>

Závěr

Cílem bakalářské práce byla analýza současných řešení a požadavků na prezentační nástroje a následný návrh a implementace nového nástroje, který umožní generování prezentací se stromovou strukturou ze standardních formátů.

Analýza ukázala, že v současné době není dostupný takový nástroj, přestože uživatelé o něj mají zájem. Tedy na základě návrhu byla vytvořena knihovna generující stromovou prezentaci s derivací obsahu, která umožňuje ovládání všemi směry s podporou externích ovladačů. Prezentaci také lze exportovat do souborů ve formátu HTML, PDF a \LaTeX .

Vzhled a chování výsledné prezentace lze přizpůsobit prostřednictvím nastavení s uživatelským rozhraní. Globální orientace a navigace v prezentaci umožňuje schéma generované ze stromové struktury.

Do práce se podařilo zahrnout již existující službu, do které byl navržen modul, jehož výstupem je strukturovaný HTML dokument akceptovaný vytvořenou knihovnou. V kombinaci s touto službou lze vytvářet stromové prezentace přímo z dokumentů Google a Word.

V budoucnu by bylo možné knihovnu rozšířit o výchozí nastavení prostřednictvím konfiguračního souboru. Ten by kromě aktuálních vlastností mohl umožňovat přemapování klávesových zkratk nebo úpravu vzhledu výsledné prezentace ve formátu PDF a \LaTeX . Další možností rozšíření by mohl být export prezentace do PDF se stromovou strukturou, kde by každá stránka obsahovala navigaci, která by propojovala jednotlivé slajdy prezentace.

Literatura

- [1] Spáčil, J.: 6 kroků, jak vytvořit dotazník [online]. *Quanda*, [cit. 2017-01-04]. Dostupné z: <https://www.quanda.cz/blog/online-dotazniky/jak-vytvorit-dotaznik>
- [2] Bradner, S.: Key words for use in RFCs to Indicate Requirement Levels [online]. *Harvard University*, 1997, [cit. 2017-02-02]. Dostupné z: <https://www.ietf.org/rfc/rfc2119.txt>
- [3] Dobiáš, R.: Jak na prioritu informací v textu – obrácená pyramida v akci [online]. *O psaní*, 2011, [cit. 2017-04-13]. Dostupné z: <http://www.o-psani.cz/jak-na-prioritu-informaci-v-textu-obracena-pyramida-v-akci>
- [4] W3C: HTML5 - W3C Recommendation [online]. 2014, [cit. 2017-04-20]. Dostupné z: <https://www.w3.org/TR/html5/sections.html#outlines>
- [5] Podhrázký, M.: Procházení stromů [online]. 2002, [cit. 2017-03-25]. Dostupné z: <http://www.alg.webzdarma.cz/diplomka/kap4/pruchod.html>
- [6] Michálek, M.: Media Queries [online]. *Vzhůru dolů*, 2017, [cit. 2017-04-17]. Dostupné z: <http://www.vzhurudolu.cz/prirucka/css3-media-queries>
- [7] Žára, O.: DOM události: co o nich možná nevíte [online]. *zdroják.cz*, 2013, [cit. 2017-04-25]. Dostupné z: <https://www.zdrojak.cz/clanky/dom-udalosti-co-o-nich-mozna-nevite/>
- [8] Network, M. D.: Touch events [online]. 2017, [cit. 2017-04-27]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Touch_events
- [9] W3Schools: HTML5 Local Storage [online]. 2017, [cit. 2017-04-27]. Dostupné z: https://www.w3schools.com/html/html5_webstorage.asp

LITERATURA

- [10] Netguru: How to Write a Perfect ReadMe [online]. 2015, [cit. 2017-04-20]. Dostupné z: <https://www.netguru.co/blog/how-to-write-a-perfect-readme>

Pravidla a zásady projektů FIT

Příloha obsahuje kompletní metodiku, podle které byla celá práce vedena a vychází z ní požadavky na kvalitu aplikace.

Pravidla a zásady projektů FIT

Tento dokument popisuje preferovaný způsob řešení projektů FIT. Pokud některá část tohoto dokumentu není vůči konkrétnímu projektu efektivní nebo na ní není dostatek prostředků, je jí snížena úroveň nezbytnosti¹ nebo je od ní zcela odstoupeno. O takovém rozhodnutí musí existovat záznam.

Tento dokument je nedílnou součástí projektové dokumentace jako příloha ke kapitole Požadavky na kvalitu. Veškeré oblasti tohoto dokumentu, které uvedená kapitola nezmíní, se předpokládají a požadují. Výchozí úroveň nezbytnosti požadavků (není-li uvedeno jinak) je [MUST].

Obsah dokumentu

[Dokumenty](#)

[Harmonogram](#)

[Ostatní dokumenty projektu](#)

[Architektura a integrace do infrastruktury](#)

[Webová přístupnost](#)

[Kód aplikace](#)

[Verzování](#)

[Kontrola kvality](#)

[Provoz, údržba a rozvoj aplikace a podpora uživatelů](#)

[Bezpečnost a ochrana osobních údajů](#)

[Další koncepty webových aplikací \(W2.0\)](#)

Dokumenty

Ke každému projektu vzniká sada samostatných dokumentů v čele s harmonogramem. Dokumenty jsou (až na výjimky) psané česky. Výchozím dokumentem je Harmonogram, který podléhá schvalování. Není-li uvedeno jinak, o projektu rozhoduje vedení fakulty (grémium děkana).

¹ [Key words for use in RFCs to Indicate Requirement Levels](#)

Harmonogram

Stěžejním rozcestníkem projektu je iterativní harmonogram. V každém běhu se předpokládá realizace jen takových funkcí a vlastností aplikace, které náleží do daného běhu. Každý běh (iterace) obsahuje všechny uvedené fáze harmonogramu. Pro každou fázi je stanoven termín předpokládaného dokončení. Každá fáze podléhá schvalování vedením písemnou formou. Obsah harmonogramu se s postupem času upřesňuje.

1. Analytická fáze

- Průzkum existujících řešení (dále jen SOTA²).
- Vytvoření hypotéz jako kvalitativním průzkum, typicky formou rozhovorů.
 - i. Uživatelé skupiny
 - ii. Potřeby uživatelů
- Ověření hypotéz jako kvantitativní průzkum, typicky dotazníkem v souladu s explicitně zmíněnou metodikou³.
- Sestavení požadavků na kvalitu (vycházející z tohoto dokumentu).
- Sestavení požadavků na funkce (prioritní seznam rozdělený dle úrovně nezbytnosti).
- Posouzení SOTA vůči požadavkům na funkce.

2. Návrhová fáze

- Procesy (back-end, front-end)
- Prototypování (paralelní).
- Diagramy (procesní, aktivit).
- Scénáře průchodu, user-stories, případové studie
 - i. pro všechny vznikající funkce
 - ii. testování (např. formou storyboarding, inspekce).
- Hi-fi prototypy (grafický návrh, ...).

3. Implementační fáze

- Koncepty řešení dílčích požadavků (funkční / nefunkční).
- Technická specifikace (API, parametry, manuál) jako příloha k projektové dokumentaci nebo přímo u projektu v repozitáři.
- Testování (inspekce, heuristika).

4. Vyhodnocení

- Testování, logování.
- Vyhodnocení (feedback, statistiky, logy).

Ostatní dokumenty projektu

V rámci vývoje vznikají další typy dokumentů s níže uvedenými náležitostmi pro různé skupiny čtenářů. Obsah každého dokumentu je cílený na příslušnou skupinu uživatelů a zohledňuje jejich schopnosti a možnosti.

● Projektová dokumentace

- je množina samostatných dokumentů vznikajících pro jednotlivé iterace agilního vývoje aplikace⁴,
- má strukturu podle fází harmonogramu,
- odkazuje na související legislativní úpravu problémové domény aplikace,
- je určená pro zadavatele, návrháře a vývojáře projektu a případně pro uživatele.

² [State of the Art](#)

³ [6 kroků, jak vytvořit dotazník](#)

⁴ [Agile software development](#)

- **Instalační a provozní příručka**
 - obsahuje kompletní postup pro sestavení (build) a nasazení (deployment) aplikace a nových verzí,
 - popisuje dostupná prostředí (staging/produkční verze) v návaznosti na [kap. Údržba a rozvoj](#),
 - popisuje provozuschopnost v případě nedostupnosti souvisejících služeb,
 - popisuje proces obnovení provozu v případě výpadku.
- **Uživatelská dokumentace**
 - je průběžně udržovaný samostatný dokument,
 - obsahuje informaci, k čemu a komu aplikace souží,
 - je určena koncovým uživatelům frontendové aplikace⁵, resp. aplikačního rozhraní (RESTful API)⁶,
 - je dostupná z webu FIT (stačí odkazem)⁷,
 - zahrnuje CHANGELOG (viz [kap. Verzování](#)).
- **Dokumentace vnitřního API**
 - je sada dokumentů generovaná z kódu průběžně udržovaná společně s kódem aplikace,
 - je psaná anglicky v příslušné standardizované syntaxi⁸, přičemž dokumentace veřejných entit zahrnuje minimálně:
 - souhrnný popis dokumentované entity (funkce, třídy, metody, proměnné, ...),
 - souhrnný popis parametrů (funkce/metody) nebo typových proměnných (generické typy),
 - popis vyřazených výjimek (které výjimky a kdy vznikají),
 - popis návratové hodnoty (a její význam).

Architektura a integrace do infrastruktury

Projekt je webovou aplikací, která efektivně využívá existující technologie a služby FIT. Aplikace je členěná na nezávislé části, které je možné vyměnit a je provozuschopná i v případě výpadků souvisejících služeb.

- Projekt je webovou aplikací s
 - uživatelským rozhraním (UI) pro webový prohlížeč, nebo
 - RESTful API s upřesněním standardu vč. formátu⁹.
- Architektura aplikace
 - striktně odděluje frontend a backend,
 - správa uživatelů (user-management) je zajištěna fakultním IDM,
 - využívá maximum dostupných služeb (např. notifikace),
 - [SHOULD] podporuje použití pro více fakult na jediné instanci.
- Aplikace
 - je součástí katalogu služeb FIT¹⁰ od počátku práce na projektu (stav „připravuje se“),
 - využívá mezipaměť pro urychlení obsluhy požadavků,

⁵ [10 Examples of Great End User Documentation](#)

⁶ Doporučené nástroje pro dokumentaci RESTful API: [RAML](#), [Swagger](#) / [OpenAPI](#)

⁷ [Návod ke psaní dokumentace ICT FIT](#)

⁸ Např. JavaDoc nebo DoxyGen

⁹ Doporučujeme vycházet ze standardu [JSON API](#)

¹⁰ [Katalog služeb ICT FIT](#)

- je provozuschopná i v případě nedostupnosti (zpomalení) souvisejících služeb¹¹.
- [MAY] Aplikace je implementovaná na platformě/jazyku:
 - Ruby,
 - JavaScript, resp. izomorfní JavaScript¹²,
 - Groovy/Java na Spring Frameworku,
 - Python,
 - PHP na frameworku Symfony.

Webová přístupnost

Aplikace je přístupná pro uživatele bez ohledu na jejich omezení a zařízení, kterým k aplikaci přistupují.

- Aplikace respektuje požadavky WCAG 2.0 AA¹³, zejména
 - sémantické značkování výstupu HTML,
 - jednoznačné perzistentní URL jednotlivých stránek¹⁴,
 - podpora tisku,
 - *progressive enhancement*¹⁵.
- Výstup aplikace (HTML) je v souladu s principem *mobile-first*¹⁶, *media-first*¹⁷, zejména
 - přizpůsobivé uživatelské rozhraní¹⁸,
 - použitelnost ovládacích prvků pro manipulaci prsty,
 - nenáročnost s ohledem na výkon CPU a spotřebu baterie,
 - minimalizace přenesených dat.
- [MAY] Aplikace je odolná vůči výpadkům připojení a funkčnost bez připojení k Internetu.¹⁹
- [MAY] Aplikace podporuje *Web App Manifest*²⁰ a integraci do operačního systému²¹ zahrnující
 - podporu push notifikací²²,
 - synchronizaci na pozadí přes *Service Workers*²³.

Kód aplikace

Veškerý kód je psaný kompletně v angličtině s prioritou udržitelnosti a čitelnosti. Vývoj kódu přehledně odděluje provozní větve od vývojových. Před nasazením prochází každý nově vzniklý kód kontrolou kvality na několika úrovních.

¹¹ Např. bez datového, resp. autentifikačního, zdroje, informace z mezipaměti, resp. zobrazí jen veřejné informace (s příslušným upozorněním).

¹² Viz [Isomorphic JavaScript](#).

¹³ [Web Content Accessibility Guidelines \(WCAG\) 2.0](#)

¹⁴ Viz [Cool URIs](#) a [Why JavaScript web applications should embrace traditional URLs](#).

¹⁵ Poskytnout klientovi úplnou funkcionalitu i v případě, že nepodporuje dynamické technologie; viz [článek na Gov.UK](#).

¹⁶ Viz [kniha Mobile First \(Luke Wroblewski\)](#).

¹⁷ Společná definice zobrazení od sémantického obsahu pro čtečky a textové interprety, přes tisk a malé obrazovky až po velké obrazovky.

¹⁸ Viz [Responsive Web Design](#).

¹⁹ Tzv. [Offline-First](#).

²⁰ Viz [Web App Manifest](#).

²¹ Viz články [Progressive Web Apps: Escaping Tabs Without Losing Our Soul](#) a [Getting started with Progressive Web Apps](#).

²² Viz [Push Notifications on the Open Web](#).

²³ Viz [Introduction to Service Worker](#).

- Aplikace respektuje *best practices* pro psaní udržitelného a čitelného kódu²⁴; zejména
 - logické členění kódu do modulů podle funkcionality,
 - specifikace konvencí používaných technologií (např. CSS²⁵, JavaScript²⁶, Java²⁷),
 - dodržování stylu autora při editaci cizího kódu,
 - minimalizace importů²⁸ (import, include, using, atd.),
 - používání existujících knihoven²⁹, kdykoli je to efektivní a smysluplné,
 - používání návrhových vzorů³⁰,
 - komentování potenciálně nejasných částí.
- Veškerý kód je psaný
 - v UTF-8 s unixovým koncem řádek (řídící znak LF / 0x0A),
 - anglicky (názvy funkcí a proměnných, komentáře, systémová a jiná hlášení).
- Veškeré výstupy (texty pro uživatele) podporují lokalizaci a internacionalizaci.
- Pro aplikaci existuje kompletní česká lokalizace.

Verzování

Vývoj kódu je organizovaný s přehledným oddělením provozní a vývojové větve. Umožňuje operativní opravy kritických chyb (hotfix) a nezávislý vývoj nových funkcí. Podporuje bezpečný model nasazování nových verzí³¹ pro účely testování a ladění (akceptační testy).

- Kód aplikace je vyvíjen na revizním systému Git³² využívající
 - repozitář na službě GitLab provozované fakultou³³ nebo oddělením ICT³⁴,
 - standardní branching model Git Flow³⁵ (nástroje OMGF³⁶ nebo Git-Flow Cheatsheet³⁷) a
 - sémantické verzování³⁸.
- Používání revizního systému se řídí pravidly *commitování*, zejména
 - *commit* každé dílčí změny funkcionality,
 - zachování funkcionality celku přes jednotlivé *commity*,
 - používání rozkazovacího tvaru v přítomném čase³⁹.
- Součástí vývoje je udržování aktuálního souboru CHANGELOG⁴⁰ dostupného z webu na úrovni
 - nových funkcí (či inovací) vždy při jejich začlenění do vývojové větve,
 - nových MINOR verzí vždy při začlenění vývojové větve do provozní.
- Na společných (sdílených) větvích není povoleno přepisování historie.
- Veškeré texty verzování jsou anglicky.

²⁴ Viz [Best Practices](#) a kniha [The Pragmatic Programmer](#).

²⁵ Konkrétně [konvenci SUIT CSS](#).

²⁶ [JavaScript Quality Guidelines and Recommendations](#)

²⁷ [Code Conventions for the Java Programming Language](#)

²⁸ Např. neimportovat celý balíček, když z něj bude použita jen malá část.

²⁹ Pod svobodnými nebo open-source softwarovými licencemi a respektovat podmínky těchto licencí.

³⁰ [Gang of Four Design Patterns](#) či [Design Patterns na Wiki](#)

³¹ [Deployment environment](#)

³² [Jak na Git](#)

³³ [GitLab FIT ČVUT](#)

³⁴ [GitLab ICT](#)

³⁵ [Git Flow](#)

³⁶ [OMGF](#)

³⁷ [Git-Flow Cheatsheet](#)

³⁸ [Semantic Versioning 2.0.0](#)

³⁹ [How to Write a Git Commit Message](#)

⁴⁰ [Keep a Changelog](#)

Kontrola kvality

Veškerý kód se před nasazením patřičně kontroluje na úrovni automatizovaných nástrojů a dílčích (jednotkových a dalších) testů. Součástí kontroly kódu je (jednoduchý) schvalovací proces. Alternativně se kód vyhodnocuje, zda splňuje stanovené kvalitativní metriky.

- Vývoj kódu se opírá o kontrolní nástroje jako zejména
 - příslušný *linter*⁴¹.
- Veškeré nasazování změn kódu (merge) procházejí kontrolním procesem⁴² s následujícími pravidly.
 - Veškeré merge jsou prováděny formou požadavků na začlenění⁴³ (dále PR).
 - Veškeré PR (bez ohledu na svou podstatu a závažnost) budou potvrzované minimálně druhým členem týmu – programátorem, alternativně nadřízeným.
 - PR kritického požadavku si může jeho řešitel sám akceptovat. O takovém úkonu neprodleně vyrozumí členy týmu. Povinnost potvrzení podle předchozího bodu zůstává, však může být učiněno dodatečně (bez zbytečného prodlení).
 - Součástí kontrolního procesu nasazování je continuous integration⁴⁴ na GitLabu⁴⁵.
- Kód obsahuje automatické testy, mezi které patří zejména
 - jednotkové testy,
 - integrační testy (API, resp. automatizované průchody).
- [SHOULD] Vývoj kódu je řízený testy⁴⁶.
- [SHOULD] Minimální požadované hodnoty metrik⁴⁷ pomocí fakultní služby Sonar⁴⁸ jsou stanoveny následujícím způsobem:
 - Method Total Length (< 30 lines)
 - Class Total Length (< 300 lines)
 - Unit Tests Line coverage (> 70 %)
 - Unit Tests Branch coverage (> 70 %)
 - Density of duplicated lines (< 5 %)
 - Lack of cohesion of methods (< 3)
 - Average complexity by method (< 5)
 - Rules compliance index (žádné závady úrovně „blocker“ ani „critical“)

Provoz, údržba a rozvoj aplikace a podpora uživatelů

Veškerá (nově vznikající) funkcionalita je uživatelům dostupná přehledně a jednoduše. Aplikace (nová verze) se nasazuje do provozu po splnění akceptačních testů. Součástí údržby a dlouhodobého rozvoje aplikace je sběr informací o používání a jejich pravidelné vyhodnocování následované patřičným zapracováním do aplikace.

⁴¹ Platí zejm. pro dynamické a značkovací jazyky; viz např. [doporučení pro JavaScript](#) a [CSSLint](#).

⁴² [Best Practices for Code Review](#) a [Code reviews v praxi](#)

⁴³ [Code Review Via GitLab Merge Requests](#)

⁴⁴ [Continuous integration](#)

⁴⁵ [GitLab Continuous Integration](#)

⁴⁶ [Test-Driven Development](#)

⁴⁷ [Sonar Metric Definitions](#)

⁴⁸ [Sonar FIT ČVUT](#)

- Vzhled uživatelského rozhraní (UI) aplikace je moderní, přehledný, vzdušný a tvořený obsahem.⁴⁹
- Sada akceptačních testů je specifikovaná pro účely testování všech dostupných a nově vznikajících funkcí.
- Sběr dat se provádí na základě
 - zpětné vazby uživatelů prostřednictvím
 - funkce issue tracking v rámci GitLabu a
 - e-mailu na helpdesk,
 - logování⁵⁰ a integrace s monitorovacími službami na úrovni
 - chyb (fatal, warning),
 - informačních zpráv o používání⁵¹ (používanost funkcí, doby trvání, přístupy) a
 - systémových zpráv a dalších výstupů.
- Proces vyhodnocování dat zahrnuje
 - podporu uživatelů a
 - opravy chyb včetně klasifikace jejich závažnosti.
- [SHOULD] Proces rozvoje od návrhu po realizaci za účelem
 - vylepšování stávajících funkcí (optimalizace chodu a procesů),
 - přidávání nových funkcí.
- [SHOULD] Proces testování nových verzí aplikace zahrnuje
 - provoz nezávislé (beta) verze,
 - podporu AB testování,
 - provádění inspekci a heuristik,
 - pozorování.

Bezpečnost a ochrana osobních údajů

Aplikace je standardně zabezpečená; zejména nepracuje s hesly uživatelů a veškerá komunikace probíhá přes šifrovaný protokol. Aplikace také respektuje nařízení rektora o ochraně osobních údajů.

- Aplikace respektuje principy bezpečných webových aplikací⁵², jmenovitě
 - veškerá komunikace (S2S, S2C) probíhá přes HTTPS,
 - jako API poskytuje různé úrovně oprávnění pomocí *scopes*⁵³,
 - nepracuje s hesly uživatelů; autentizace, resp. autorizace, probíhá přes Shibboleth (není-li potřeba autorizace), resp. autorizační server FIT (protokol OAuth 2.0)⁵⁴.
- Aplikace respektuje nařízení rektora o ochraně osobních údajů⁵⁵, zejména dokumentace definuje,
 - jaké informace jsou citlivé/osobní,
 - jaká data jsou dostupná kterým uživatelům v závislosti na autorizaci uživatele (např. anonymní uživatel, přihlášený uživatel, student, vyučující, administrátor),
 - které citlivé/osobní informace jsou přístupné v rozporu s nařízením.

⁴⁹ [UXMyths: You don't need the content to design a website.](#)

⁵⁰ [Logging Best Practices](#)

⁵¹ [Google Analytics s využitím událostí \(events\)](#)

⁵² Viz [principy OWASP](#).

⁵³ [Securing Access with OAuth2: How to deal with OAuth Scopes](#)

⁵⁴ Viz [Autorizační server FIT \(OAuth 2.0\)](#).

⁵⁵ [Příkaz rektora č. 5/2015 Ochrana osobních údajů na ČVUT v Praze](#)

Další koncepty webových aplikací (W2.0)

Aplikace explicitně zohledňuje možnosti využití níže uvedených webových konceptů a případně dalších. Vzhledem k omezení jednotlivých projektů mohou být využití konceptů pouze součástí projektové dokumentace – byť jen jako potenciální rozšíření funkcionality s uvedenými přínosy a konkrétními příklady.

- Dokumentace (např. v příloze) popisuje možnosti a úroveň nezbytnosti využití všech následujících konceptů:
 - RSS,
 - personalizace,
 - customizace,
 - folksonomie (tagování),
 - social networking,
 - real-time web⁵⁶,
 - crowdsourcing⁵⁷,
 - kolaborace,
 - průvodce (wizardy),
 - konfiguratory (rozšířených dotazů vyhledávání, parametrů služby),
 - gamifikace⁵⁸,
 - mikrodata⁵⁹.

⁵⁶ [Real-time web \(Wiki\)](#)

⁵⁷ [Crowdsourcing \(Wiki\)](#)

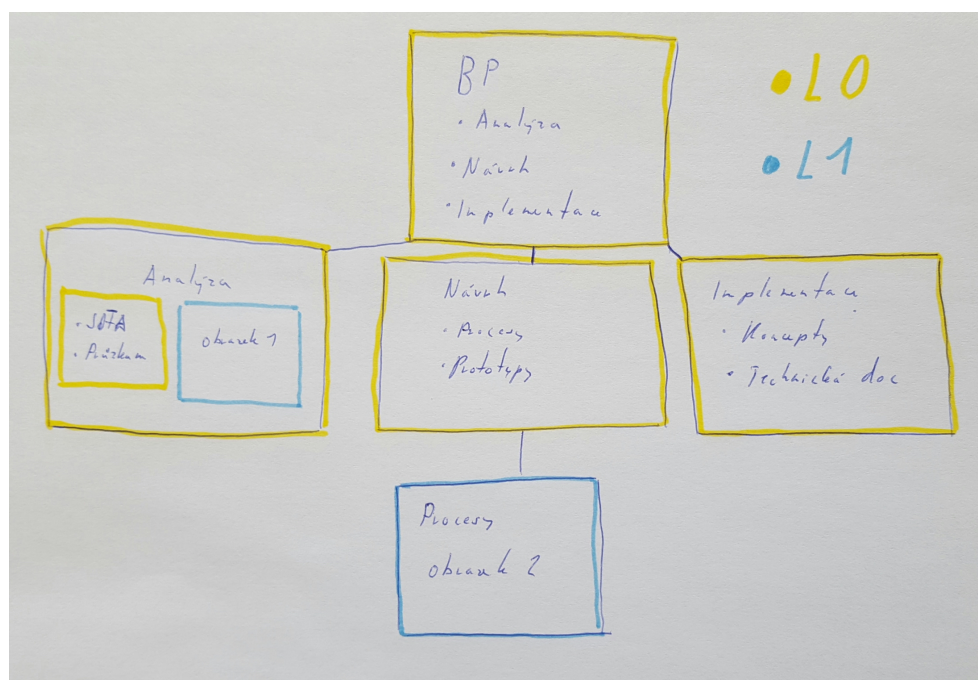
⁵⁸ [Gamification \(Wiki\)](#)

⁵⁹ [Microdata \(Wiki\)](#)

Modul derivace obsahu

Příloha obsahuje původní návrh derivace obsahu, který byl navrhován jako další modul do CMS. Výstupem modulu je derivace obsahu, která bude následně využita pro generování PDF, \LaTeX a HTML souboru pro prezentaci. Derivace je realizována pomocí níže popsané transformace, která se skládá z $\{L0\}$ a $\{L1\}$. Pro lepší pochopení transformací uvádím následující příklad, kde pro danou strukturu obsahu dokumentu, nad kterým je provedena $\{L0\}$ a $\{L1\}$ se vygeneruje prezentace na obrázku B.1.

```
<h1>BP</h1>
<p>Souhrn..</p>
<h2>Analyza</h2>
<figure>
  
</figure>
<h3>SOTA</h3>
<h3>Průzkum</h3>
<h2>Návrh</h2>
<p>Souhrn..</p>
<h3>Procesy</h3>
<p>Souhrn..</p>
<figure>
  
</figure>
<h3>Prototypy</h3>
<h2>Implementace</h2>
<h3>Koncepty</h3>
<p>Souhrn..</p>
<h3>Technická doc</h3>
<p>Souhrn..</p>
```



Obrázek B.1: Struktura vytvořená modulem derivace obsahu

B.1 Logika 0 {L0}

{L0} je založena na metodě obrácené pyramidy tedy, že informace s nejvyšší prioritou se nachází na nejvyšší úrovni dokumentu a se zvětšujícím se zanořením jejich důležitost klesá. Tedy nadpis kapitoly nejvíce vypovídá o jejím obsahu a lze ho považovat za stručné shrnutí. Nadpisy v dokumentu představují jednotlivé slajdy prezentace neboli uzly stromu, přičemž při derivaci rozlišujeme vnitřní a koncové uzly (listy).

V případě {L0} pracujeme pouze s vnitřními uzly, u kterých zachováváme hlavní nadpis a obsah je tvořen nadpisy jeho potomků, které jsou transformovány do neseřazeného seznamu tak, že každý nadpis tvoří položku seznamu.

B.2 Logika 1 {L1}

Tato logika ve srovnání s {L0} pracuje se všemi uzly stromu a umožňuje rozdělení slajdu prezentace do několika dalších v rámci jednoho uzlu. Rozdělení je prováděno vizuálně na straně {K} a dojde k němu, pokud je výsledný obsah slajdu tvořen více blokovými elementy. {L1} je definována konfiguračním souborem. Ten se skládá z CSS selektoru a k němu podporovaných operací, které definují chování v případě nalezení obsahu, jenž odpovídá příslušnému selektoru.

Tabulka B.1: Výchozí nastavení derivace

Selektor	Operace
p	odstraněno
figure	zachováno
ol, ul, dl	odstraněno
table	odstraněno
img	odstraněno

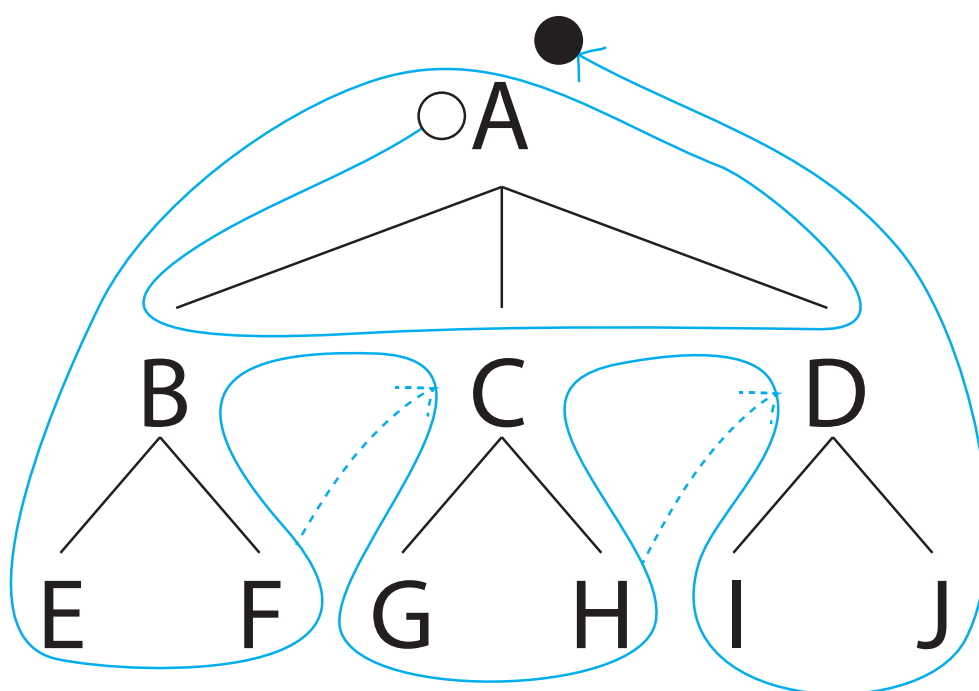
Pro účely této práce byl konfigurační soubor zjednodušen a bylo nastaveno jeho výchozí chování B.1. Avšak do budoucna se nabízí široké možnosti rozšíření, kdy například definiční seznam (dl) je transformován na neseřazený seznam tak, že pojmy (dt) se stanou položky seznamu (li) a popisy pojmů (dd) jsou ignorovány.

Stromová struktura prezentace, jež je výsledkem {L0} také může být rozšířena o další koncové uzly (listy) a to za předpokladu, že listy vstupního stromu, které byly ignorovány {L0} obsahují nějaký obsah po aplikování operací definovaných konfiguračním souborem.

Linearizace průchodu

Příloha obsahuje alternativní průchod stromem, který by mohl být využit pro linearizaci obsahu pro export do PDF a \LaTeX nebo průchod webovou prezentací pouze pomocí ovladače s tlačítka předcházející a další.

Linearizace průchodu stromem prezentace spočívá v průchodu do hloubky s inkrementálně narůstajícím limitem hloubky. Každý cyklus končí návratem na kořen stromu. Pohyb vpravo (mimo hranu) je možný pouze za předpokladu přímých sourozenců. Do úvahy připadají dvě varianty průchodu: úplný a zkrácený (na obrázku C.1 tečkovaně). Rozepsání obou variant z obrázku (úplný v závorce): A — B — C — D — A — B — E — F — (B) — C — G — H — (C) — D — I — J — A.



Obrázek C.1: Schéma linearizace průchodu

Uživatelský manuál

Vzhledem k tomu, že knihovna Tree Presenter je samostatně funkční celek a její kód je verzován pomocí systému Git, tak byla umístěna na školní server GitLab³⁸. Na GitLabu byl vytvořen uživatelský manuál, jako README.md³⁹, který poskytuje potřebné informace k použití nebo dalšímu rozšíření knihovny. Zbývající část přílohy obsahuje daný uživatelský manuál exportovaný přímo z repozitáře.

³⁸<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter>

³⁹README.md slouží k popisu repozitářů na službách typu GitHub, GitLab a dalších

Tree Presenter

Tree Presenter is a JavaScript library that allows you create web presentations with hierarchical structure.

Try it at <https://cernym43.github.io/Tree-Presenter/#root> (<https://cernym43.github.io/Tree-Presenter/#root>)!

Features

Here is a brief summary of the Tree Presenter features which you can try in the example above.

Tree structure

The library allows you to create a two-level tree structure presentation. The first level is automatically generated from the HTML5 content. Second level is the content of the presentation itself like pictures, texts and lists.

Customisation

It is possible to customize:

- Presentation aspect ratio,
- Color theme,
- Visibility of navigation,
- Tree traversal.

Navigation

Three navigation elements are available:

1. Arrows navigation – shows the closest possible nodes.
2. Map of presentation – shows the entire presentation structure including the current position.
3. Icons to open settings, schema etc.

Download

The first level of presentation can be downloaded in HTML, PDF or LaTeX format.

Usage

You can download fully working example and just edit content from project folder dist (<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter/tree/master/dist>) or you can start from scratch by following instructions.

Necessary requirements

The following requirements are necessary for the presentation to work correctly.

- Importing JS libraries

```
<script src="./js/outliner.min.js"></script>
<script src="./js/pdfmake.min.js"></script>
<script src="./js/vfs_fonts.js"></script>
<script src="./js/raphael.js"></script>
<script src="./js/Treant.js"></script>
<script src="./js/TreePresenter.js"></script>
```

- Importing CSS

```
<link rel="stylesheet" href="./css/style.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awes
ome/4.7.0/css/font-awesome.min.css">
```

- Importing styles for browsers without JS support

```
<noscript><link rel="stylesheet" type="text/css" href="TreePresenter_NoJS.cs
s"></noscript>
```

- HTML document that respects the source data syntax.

Source data syntax

Below is a list of rules which allows the creation of the correct tree presentation. Compatibility is not guaranteed in case of non-compliance.

- All presentation content is part of one HTML file [MUST].
 - The library ignores the links.
- HTML5 language [SHOULD].
 - When using another standard, proper functionality is not guaranteed.
- Valid HTML syntax [SHOULD].
 - Ensures the functionality and correct generation of the presentation.
- Existence of heading id [SHOULD].
 - Each headline should contain an id attribute with a valid value.
- The content of the presentation in the element body [MUST].
 - The tree structure of the presentation is created based on this content by an HTML5 outliner (<https://github.com/h5o/h5o-js>) algorithm.
 - Examples of valid content structures are available at w3.org#sample-outlines

(<https://www.w3.org/TR/html5/sections.html#sample-outlines>) and in example below.

Example

```
<h1 id="slide1">Slide 1</h1>
<ul>
  <li>Root of the tree</li>
</ul>

<section>
  <h1 id="slide2">Slide 2</h1>
  <ul>
    <li>Slide with depth 1</li>
  </ul>

  <h2 id="slide3">Slide 3</h2>
  <ul>
    <li>Slide with depth 2</li>
  </ul>

  <h2 id="slide4">Slide 4</h2>
  <ul>
    <li>Slide with depth 2</li>
  </ul>
</section>

<h2 id="slide5">Slide 5</h2>
<ul>
  <li>Slide with depth 1</li>
</ul>

<h3 id="slide6">Slide 6</h3>
<ul>
  <li>Slide with depth 2</li>
</ul>
```

The resulting presentation structure generated from the previous code:



Controls

During presentation, you can use following shortcuts, mouse and touch gesture if the device allows it.

Key	Action
Arrow keys	Navigation between slides

Key	Action
Swipe :arrow_left: :arrow_right: :arrow_down: :arrow_up:	Navigation between slides
Enter	Zoom into slide (if possible)
M	Slides overview
n/N	Next/previous presentation (if exist)
S	Settings
D	Download
H	Help/Shortcuts

Development

Want to contribute or just customize your presentation? No problem!

Clone the repository:

```
git clone https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter.git
```

Install dependencies:

```
npm install
```

The development code is located in the folder `/src` (<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter/tree/master/src>)

After you done run following command in root of the folder which compile ES6 -> ES5 and SASS -> CSS. Final presentation is located in `dist/index.html`.

```
./script.sh
```

Testing

The folder `tests` (<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter/tree/master/tests>) contains two sets of sample files with:

1. valid HTML 5 content,
2. expected output structure of presentation based on HTML 5 file.

Technology

The library uses the following technologies and libraries:

- HTML5 (https://www.w3schools.com/html/html5_intro.asp)
- ECMAScript 2015 (<http://www.ecma-international.org/ecma-262/6.0/>)
- npm (<https://www.npmjs.com/>)
- SASS (<http://sass-lang.com/>)
- Babel (<https://babeljs.io/>)
- Rollup (<https://github.com/rollup/rollup>)
- Treant.js (<http://fperucic.github.io/treant-js/>)
- h5o-js (<https://github.com/h5o/h5o-js>)
- pdfmake (<http://pdfmake.org/index.html#/>)

Vzorové dokumenty

Příloha obsahuje sadu vzorových dokumentů, které jsou zároveň součástí repozitáře s knihovnou Tree Presenter⁴⁰. Každý dokument obsahuje vstupní obsah v jazyce HTML5 a očekávanou výstupní strukturu. Dokumenty slouží k ověření správné funkčnosti generování stromové struktury z HTML5 a jejich obsah je převzat ze specifikace W3C⁴¹.

```
<h1>The Tax Book</h1>
<h2>Earning money</h2>
<p>Earning money is good.</p>
<h3>Getting a job</h3>
<p>To earn money you typically need a job.</p>
<h2>Spending money</h2>
<p>Spending is what money is mainly used for.</p>
<h3>Cheap things</h3>
<p>Buying cheap things often not cost-effective.</p>
<h3>Expensive things</h3>
<p>The most expensive thing is often not the most cost-effective
  either.</p>
<h2>Investing money</h2>
<p>You can lend your money to other people.</p>
<h2>Losing money</h2>
<p>If you spend money or invest money, sooner or later you will
  lose money.
<h3>Poor judgement</h3>
<p>Usually if you lose money it's because you made a mistake.</p>
```

1. Bachelor thesis
 1. Analysis
 1. Getting a job
 2. Spending money
 1. Cheap things

⁴⁰<https://gitlab.fit.cvut.cz/cernym43/Tree-Presenter/tree/master/tests>

⁴¹<https://www.w3.org/TR/html5/sections.html#outline>

2. Expensive things
 3. Investing money
 4. Losing money
 1. Poor judgement
-

```
<h1>The Tax Book</h1>
<section>
  <h1>Earning money</h1>
  <p>Earning money is good.</p>
  <section>
    <h1>Getting a job</h1>
    <p>To earn money you typically need a job.</p>
  </section>
</section>
<section>
  <h1>Spending money</h1>
  <p>Spending is what money is mainly used for.</p>
  <section>
    <h1>Cheap things</h1>
    <p>Buying cheap things often not cost-effective.</p>
  </section>
  <section>
    <h1>Expensive things</h1>
    <p>The most expensive thing is often not the most
      cost-effective either.</p>
  </section>
</section>
<section>
  <h1>Investing money</h1>
  <p>You can lend your money to other people.</p>
</section>
<section>
  <h1>Losing money</h1>
  <p>If you spend money or invest money, sooner or later you will
    lose money.
  <section>
    <h1>Poor judgement</h1>
    <p>Usually if you lose money it's because you made a
      mistake.</p>
  </section>
</section>
</section>
```

1. Bachelor thesis
 1. Analysis
 1. Getting a job
 2. Spending money
 1. Cheap things
 2. Expensive things

-
3. Investing money
 4. Losing money
 1. Poor judgement
-

```
<h1>Apples</h1>
<p>Pomaceous.</p>
<h1>Bananas</h1>
<p>Edible.</p>
<h1>Carambola</h1>
<p>Star.</p>
```

1. Apples
 2. Bananas
 3. Carambola
-

```
<section>
  <h1>Apples</h1>
  <p>Pomaceous.</p>
  <h1>Bananas</h1>
  <p>Edible.</p>
  <h1>Carambola</h1>
  <p>Star.</p>
</section>
```

1. (untitled page)
 1. Apples
 2. Bananas
 3. Carambola
-

```
<section>
  <h1>A plea from our caretakers</h1>
  <p>Please, we beg of you, send help! We're stuck in the server
    room!</p>
</section>
<h1>Feathers</h1>
<p>Epidermal growths.</p>
```

1. (untitled page)
 1. A plea from our caretakers
 2. Feathers
-

Seznam použitých zkratk

- BP** Bakalářská práce
- DOM** Document Object Model
- FIT** Fakulta informačních technologií
- FS** File system
- GD** Google Docs
- HTML** HyperText Markup Language
- HTTP** Hypertext Transfer Protocol
- JS** JavaScript
- JSON** JavaScript Object Notation
- PDF** Portable Document Format
- QA** Quality assurance
- SVG** Scalable Vector Graphics
- TP** Tree Presenter
- URL** Uniform Resource Locator

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	example.....	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis.....	zdrojová forma práce ve formátu \LaTeX
	text.....	text práce
	thesis.pdf.....	text práce ve formátu PDF