

Sem vložte zadání Vaší práce.

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



Bakalářská práce

SAGELab - Plánovač prezentací

Vojtěch Štecha

Vedoucí práce: Ing. Jiří Chludil

12. května 2015

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Jiřímu Chludilovi za cenné rady, konzultace a trpělivost. Další díky patří také ostatním členům týmu SAGELab a mé rodině za všechnu podporu při studiu.

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 12. května 2015

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2015 Vojtěch Štecha. 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

Štecha, Vojtěch. *SAGELab - Plánovač prezentací*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

V této práci se zabývám aplikací SAGE2TM, přesněji se zaměřím na část pro tvorbu prezentačního materiálu, který budu rozšiřovat o nové funkce. Analyzuji tento plánovač prezentací a porovnám ho s funkcionalitou, která je očekávaná od uživatelů aplikace SAGE2TM. Na základě výsledků analýzy navrhnou nejlepší způsob řešení pro přidání nových možností do plánovače a toto řešení následně implementuji do aplikace SAGE2TM. Výsledek bude plně funkční aplikace s nově implementovaným plánovačem prezentací.

Klíčová slova SAGE2, vysoké rozlišení, prezentace, JavaScript

Abstract

This thesis is about the application SAGE2TM, focused on its ability to create presentations, which I will be upgrading with new functions. I will analyze this presentation planner and compare its functions, with needs of its users. Based on the analysis, I will design the best solution for adding the new functions into the presentation planner and implement this solution. The result will be a fully functional SAGE2TM with the new extended presentation planner.

Keywords SAGE2, high resolution, presentation, JavaScript

Obsah

Úvod	1
1 Analýza	3
1.1 Laboratoř SAGELab	3
1.2 SAGE2™	4
1.3 Specifikace zadání	5
1.4 Současný stav prezentátoru	6
1.5 Požadavky studentů a lektorů	6
1.6 Funkční požadavky pro Editor	6
1.7 Funkční požadavky pro Přehrávač	7
1.8 Nefunkční požadavky	8
1.9 Úskalí a rizika implementace požadavků	8
1.10 Rozbor různých prezentačních systémů	9
1.11 Aplikace v SAGE2™	10
1.12 Ovládání SAGE2™	12
1.13 Komunikace	12
1.14 JavaScript	12
1.15 Node.js	12
1.16 HTML	13
1.17 CSS	13
1.18 WebSocket	13
1.19 JSON	13
2 Návrh	15
2.1 Případy užití	15
2.2 Technologie	19
2.3 Editor	21
2.4 Přehrávač	24
2.5 Prototyp	27

3	Realizace	29
3.1	Editor	29
3.2	Přehrávač	29
3.3	Instalační příručka	29
3.4	Uživatelská příručka	30
3.5	Návod na budoucí rozšíření	30
4	Testy	31
	Závěr	35
	Literatura	37
A	Seznam použitých zkratk	39
B	Obsah příloženého CD	41

Seznam obrázků

1.1	Zobrazovací stěna v laboratoři SAGELab[1]	4
1.2	SAGE2 TM rozhraní klienta s otevřenými aplikacemi	11
2.1	Diagram případů užití	16
2.2	Diagram případů užití možných akcí	18
2.3	Fyzické nasazení technologií SAGE2 TM	20
2.4	Wireframe Editoru prezentací	23
2.5	Wireframe Přehrávače prezentací	26
4.1	dotazník pro otestování rozšíření část 1	32
4.2	dotazník pro otestování rozšíření část 2	33

Seznam tabulek

2.1	Funkční požadavky a případy užití	19
2.2	Mapování případů užití na funkční požadavky	21

Úvod

V moderní době je výměna informací běžnou záležitostí. Informace se sdílí různými způsoby, jako např. : mluvenou formou v diskusích, čtením dokumentů nebo sledováním obrázků. K vysoké efektivitě přenosu informací se používají různé techniky.

Ve své práci se budu věnovat rozšíření prezentačního nástroje pro zobrazování dat ve vysokém rozlišení. Pro prezentaci je nezbytné si nejprve připravit speciální osnovu, do které je možné zahrnout mnoho různých elementů v závislosti na druhu prezentace. Prezentace se rozlišují nejen obsahem, ale i formou, a to v závislosti na složení jejích elementů. V nástroji, který zde budu vytvářet se používají prvky, které jsou součástí softwaru SAGE2TM.

SAGE2TM je projekt, který se zaměřuje na maximální využití velkých zobrazovacích ploch pro usnadnění týmové spolupráce a společné řešení různých problémů. Tento produkt vznikl a je dále vyvíjen na Illinoiské univerzitě v Chicagu.

Cílem mé práce je porovnání současné funkčnosti prezentačních nástrojů s požadavky uživatelů. Následně provedu jejich analýzu a navrhnu způsob možného rozšíření systému, který odladím a nasadím do současné verze SAGE2TM.

Analýza

1.1 Laboratoř SAGELab

SAGELab[10] je síťová multimediální laboratoř vytvořená ve spolupráci se sdružením CESNET, Fakulty informačních technologií ČVUT a Fakulty elektrotechnické ČVUT. Tato laboratoř se nachází v místnosti A-1452 v budově A Fakulty informačních technologií ve 14. patře. Laboratoř slouží k výuce předmětů zaměřených na využití sítí nebo pro práci s grafikou ve vysokém rozlišení. Lze zde vizualizovat rozsáhlá data a pracovat s nimi. Laboratoř je zajištěna 5 výkonnými servery pro zpracovávání náročných dat.

Hlavní zobrazovací plocha je tvořena 20 HD monitory, které jsou sestaveny do mřížky o šířce 5 monitorů a výšce 4 monitorů. Dohromady je vytvořena úctyhodná zobrazovací plocha s rozlišením 9600 x 4320 pixelů. Pro provoz tohoto vizualizačního systému je potřeba mnoho specializovaného softwaru, včetně SAGETM a SAGE2TM.

Hardwarová konfigurace zobrazovací plochy je následující:

- 20 x monitor NEC X463UN
- FullHD displej (1920 x 1080), velikost 46", 5 mm mezi obrazy
- 6 x server Dell PowerEdge T620
- 72 výpočetních jader
- grafické karty NVIDIA GeForce GTX 680
- přepínač HP 5900
- 48x porty SFP+ 1G/10G
- 4x porty QSFP+ 40G



Obrázek 1.1: Zobrazovací stěna v laboratoři SAGELab[1]

Laboratoř lze obsadit 24 studenty ve výukové konfiguraci a pro přednášky, kde nejsou potřeba stoly, se vyskytne až 50 míst pro sezení.

1.2 SAGE2™

SAGE2™[2] je technologie zaměřená na spolupráci a sdílení dat v reálném čase, které je realizováno pomocí webové prohlížeče. Uživatelé mohou využívat SAGE2™ jak lokálně, tak i přes internet. SAGE2™ je použitelná od ovládnání několika displejů na jednom počítači po nasazení na zobrazovací stěny složené z desítek displejů. K ovládnání se lze připojit přes internet ve webovém prohlížeči. Všechny instance se navzájem synchronizují a vytvářejí jednotné pracovní pole pro uživatele.

SAGE2™ je postavená na server-klient architektuře. Serverová část je implementovaná v JavaScriptu na platformě NodeJS. Klientům je poskytnuto rozhraní v podobě HTML5 stránek.

SAGE2™ poskytuje několik webových stránek, které poskytují různá rozhraní pro uživatele. Jedna z nich poskytuje zobrazování obsahu na displejích. Další je pro ovládnání zobrazovaného obsahu. Třetí z nich slouží pro příjem audio vstupu a poslední je pro sdílení kurzoru myši.

Projekt SAGE2™ byl vyvinut v Electronic Visualization Laboratory (EVL), která sídlí na University of Illinois at Chicago (UIC).

1.3 Specifikace zadání

1. Analyzujte plánovač prezentací, který je součástí nově vydané aplikace SAGE2TM, a zhodnoťte jeho funkcionalitu ve vztahu k požadavkům studentů a zaměstnanců ČVUT.
 - Nejdříve je potřeba se seznámit s již implementovaným plánovačem prezentací a jeho funkcemi. Dále je třeba sepsat, jaké možnosti studenti a ektoři očekávají, a vyhodnotit jejich požadavky. Ze srovnání požadavků a funkcí lze vyvodit chybějící funkcionality a usoudit, jestli prezentátor je dostatečným řešením.
2. Podrobně popište funkcionalitu plánovače a možnosti jeho rozšíření.
 - Sepíši všechny informace ohledně plánovače a vysvětlím jak se dají využít jeho funkce. Analyzuji plánovač a rozhodnu jestli je možné jeho funkce rozšířit. Pokud to možné je, tak zformuluji postup rozšíření.
3. Formulujte požadavky na novou funkcionalitu.
 - Na základě zpětné vazby od současným uživatelů SAGE2TM budou sestaveny požadavky na novou funkčnost, které budou vyhodnoceni na základě jejich priorit.
4. Implementujte a formou tutoriálu demonstруйте rozšíření plánovače o novou funkcionalitu.
 - Nejprve posoudím, zda je možné upravit již existující plánovač, za účelem přidání nových funkcí. Pokud to bude možné, vytvořím podrobný popis způsobu přidání nových funkcí.
5. Plánovač vylepšený o novou funkcionalitu nasadte v prostředí SAGElab.
 - Zajistím, aby nově upravený plánovač fungoval v prostředí SAGE2TM a zároveň, aby nové funkce nenarušily ostatní dovednosti aplikace SAGE2TM.
6. Řešení podrobte vhodným testům.
 - Nově naimplementované funkce podrobím testům současně s vývojem. Po dokončení implementace budou vybraní uživatelé testovat rozšíření pomocí připravených scénářů.

1.4 Současný stav prezentátoru

V projektu SAGE2TM se nacházejí aplikace, které se dají spustit a ovládat, každá z nich tvoří samostatný modul se specifickým rozhraním v projektu SAGE2TM. Každá z těchto aplikací má svůj zaměřený účel. Prezentátor je jednou z těchto aplikací, která spouští program PowerPoint od společnosti Microsoft. Při spuštění se automaticky nahraje prezentace pomocí HTTPS odkazu z Windows Live SkyDrive. Tento odkaz je napevno zadán v kódu aplikace. Ovládání prezentace lze provést pouze pomocí myši na systému, kde je spuštěn výstup zobrazovaných dat, ale protože tato část běží na serverech SAGELabu, kde myš připojena není, je ovládání prakticky nepřístupné. Z těchto důvodů shledávám aplikaci nepoužitelnou. Je tedy potřeba navrhnout úplně nové řešení, které pokryje potřebné požadavky.

1.5 Požadavky studentů a lektorů

Po opakovaných jednáních, která se uskutečnila v období únor až duben 2015 se skupinou 5 studentů a 2 odborných pracovníků laboratoře, kteří pokrývají většinu lidí zkušených v SAGE2TM technologiích, jsem zformuloval níže uvedené požadavky. Tyto požadavky byly touto reprezentativní skupinou uživatelů laboratoře SAGELab odsouhlaseny.

Pro přehlednost jsou požadavky rozděleny do 2 kategorií. Požadavky pro editor prezentace, neboli část ve které se prezentace vytvoří, a na přehrávač, ve kterém se prezentace bude přehrávat. Dále je potřeba určit, s jakými datovými typy nebo aplikacemi se mohou prezentace tvořit. SAGE2TM implementuje okenní systém, kde každé z oken je samostatná aplikace. V prezentaci se bude klást důraz na ovládání těchto oken. Druhořadě se bude analyzovat a navrhovat ovládání přímo těchto aplikací.

1.6 Funkční požadavky pro Editor

F1 Vytvoření nové prezentace

- Uživatel vytvoří novou prezentaci, pojmenuje ji a bude mu automaticky uložena do jeho úložného prostoru. Uživatel bude moci přidávat různé akce do prezentace. Spuštění nových aplikací a samozřejmě také jejich zavírání. Možnost změny pozice a velikostí okna.

F2 Úprava existující prezentace

- Možnost nahrání již uložené prezentace, jako kdyby byla vytvořená od začátku. Její následné úpravy a zpětné uložení do úložiště SAGE2TM.

F3 Duplikace prezentace

- Možnost vytvoření kopie již existující prezentace.

F4 Rozlišení verzí prezentace

- Možnost úpravy prezentace a její následné uložení takovým způsobem, že si uživatel bude moci vybrat, jakou verzi prezentace bude chtít přehrát.

F5 Odstranění prezentace

- Umožňuje odstranění prezentace z úložného prostoru. Toto odstranění je nevratné, bez možnost obnovení smazaného souboru.

F6 Možnost spuštění akce uživatelem

- Příkladem jedné akce je otevření nové aplikace. V rámci jednoho kroku je možné uskutečnit několik akcí. Prezentace je rozdělena do kroků, které na sebe navazují. K změně kroku je potřeba přijmou uživatelský vstup. Prezentace bude pokračovat na základě příkazů uživatele.
- Další možnost zahrnuje pokročilé akce, kdy bude možné přímo ovládat funkce aplikace. Neboli zasílání rozkazů dovnitř aplikace.

F7 Akce po určitém časovém úseku

- Uživatelský vstup může být nahrazen časovanou událostí. Tato událost se spustí buď uživatelským vstupem, nebo jinou časovou událostí, a po daném čase se provede.

F8 Nahrávání prezentace

- Uživatel bude mít možnost vytvořit prezentaci nahráváním svých akcí v SAGE2TM. Z jeho akcí se vyberou ty stěžejní a tyto akce vytvoří jádro prezentace.

1.7 Funkční požadavky pro Přehrávač

F9 Spuštění prezentace

- Prezentaci lze spustit vybráním zdrojového souboru prezentace na úložišti SAGE2TM.

F10 Zastavení prezentace

- Prezentace skončí a všechny aplikace, které byly její součástí, se uzavřou.

F11 Možnost kombinovat prezentaci s uživatelským vstupem

- Možnost při přehrávání přímo ovládat aplikaci. Klient při spuštění prezentaci není nijak limitován v jeho akcích a stále funguje stejným způsobem, jako když prezentace není spuštěná.

1.8 Nefunkční požadavky

Protože se jedná o upravování již stávající aplikace, tak některé z nefunkčních požadavků jsou převzaty ze SAGE2TM.

N1 Přístup z moderních webových prohlížečů

- Software se využívá přes webový prohlížeč po přístupu na server. Některé funkce se provádí na serveru a některé lokálně ve webovém prohlížeči.

N2 Integrace do SAGE2TM.

- Nebude se potřeba připojovat mimo SAGE2TM nebo využívat externí programy.

N3 Licence pro nekomerční využití

- Lze nasadit v laboratoři SAGELab a využívat nové rozšíření.

N4 Responzivní webové rozhraní

- Webové stránky je možné zobrazit na různých zařízeních a stránky se přizpůsobují specifikaci zařízení.

1.9 Úskalí a rizika implementace požadavků

Software SAGE2TM je omezen licencí, která umožňuje použití a upravení SAGE2TM pro akademické účely. Tato práce je určena pro akademické využití a není předpoklad pro jejího využití pro komerční účely. Dále přikládám nutnou citaci z licence, „SAGE2TM middleware was developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago, and the Laboratory for Advanced Visualization and Applications at the University of Hawai'i at Manoa.“ [9] Překlad do češtiny je následující: SAGE2TM middleware (specializovaný software poskytující služby nad rámec operačního systému za účelem jednoduššího vývoje aplikací) byl vyvinut elektronické vizualizační laboratoři na Illinoiské univerzitě v Chicagu, a laboratoří pro pokročilou vizualizaci a aplikace na Havajské univerzitě v Manoa. Vzhledem k tomu, že bakalářská práce má akademické zaměření, je tedy možné tuto práci uskutečnit bez porušení licenčních podmínek.

Dokumentace[3] projektu SAGE2TM není kompletní, hlavně popisuje nasazení a instalaci softwaru, jeho spuštění a jakým způsobem je možné se připojit na spuštěný projekt. Druhá část se zabývá popisem tvorby nové aplikace v prostředí SAGE2TM a možnosti jejího nasazení. Jsou zde popsány povinné funkce a vysvětlení jejich rolí. Lze nalézt i ukázkou implementace jedné z aplikací. Ovšem dokumentace, která by popisovala fungování samotných funkcí, z kterých se SAGE2TM skládá neexistuje v dostatečné míře, aby se dala použít. Samotný kód není samopopisující a komentáře uvnitř zdrojových kódů se vyskytují jen zřídkka.

1.10 Rozbor různých prezentačních systémů

Pro inspiraci a poučení jsem se rozhodl se podívat a vyzkoušet jiné již používané prezentátory. Tento průzkum mi pomůže k přesnějšimu náhledu do problematiky a pevnějšimu uchopení základu. Nejpre rozeberu PowerPoint, který se hojně používá a mnoho uživatelů s ním umí pracovat. Následně popíši program Sony Vegas.

1.10.1 PowerPoint

S PowerPointem se setkala většina počítačových uživatelů a mnoho z nich prezentaci vytvářelo. Často se s PowerPointem setkávají také studenti základních a středních škol. PowerPoint je součástí kancelářského balíčku Microsoft Office. Tento produkt zajišťuje více programů od psaní dokumentů po úpravu fotek. Díky tomu je velmi rozšířený a osobně jsem v něm prezentace také vytvářel.

Prezentaci v PowerPointu lze rozdělit na jednotlivé prezentační stránky (často nazývané slidy). Každý slide je stránka, na kterou můžeme dát obrázek, video nebo text v určitém formátu. Lze nastavit font, barvu a další parametry. Tyto elementy, ať už obrázek nebo text, je možné zařadit do sekvence. Každý přechod mezi prvky této sekvence může být i animovaný. Tato sekvence prvků vytváří tělo prezentace. Když poté prezentaci spustíme, tak zobrazujeme jednotlivé slidy této sekvence a uživatel určujeme kdy má být uskutečněn přechod na další slide prezentace. To vytváří postupný průchod a odděluje jednotlivé stavy prezentace.

Při tvorbě prezentace je ihned vidět, jak prezentace bude vypadat. To je způsobeno tím, že prvky se už vkládají vykreslené, tak jak se budou i finálně zobrazovat. Uživatelsky příjemné prostředí napomáhá k rychlé orientaci a umožňuje procházet jednotlivé slidy. Pro spuštění prezentace je zapotřebí mít nainstalován PowerPoint nebo si bezplatně stáhnout PowerPoint Viewer z oficiálních stránek Microsoftu.

1.10.2 Sony Vegas

Vegas je software určený pro stříh videa. Není zde přímá souvislost s prezentacemi, ale systém, jak se stříhá video a jak se dává dohromady, má určité podobnosti s vytvářením prezentace. Hlavní důvod analýzy tohoto softwaru spočívá v požadavku F7. V tomto požadavku je navazování akcí v závislosti na čase, stejný princip využívá i software Vegas.

Vegas má zajímavý způsob navazování a spojování videí. Uživatel nasazuje na časovou linii jak videa, tak i jiné prvky. V této časové linii se prvky mohou vyskytovat i paralelně. To přináší inspiraci pro nové možnosti prezentace, kde různé prvky mohou být zobrazeny najednou a spolu interagovat. Časová linie umožňuje využití událostí závislých na čase bez zásahu uživatele, ale je velmi nelehké tuto časovou linii synchronizovat s manuálním postupem v sekvenci prezentace.

1.10.3 Závěr

Na základě získaných faktů jsem toho názoru, že ovládání programu PowerPoint je uživatelsky přívětivější. Časová linie je velmi matoucí a není přehledná. Proto se zaměřím na využití prezentačních stránek (slidů). Poskytují lepší výkon v závislosti na jejich složitosti oproti nepřehledné časové linii.

1.11 Aplikace v SAGE2TM

SAGE2TM využívá vlastní okenní systém. Každé z těchto oken tvoří vlastní aplikaci, podobně jako v moderních operačních systémech. Každá z těchto aplikací musí splňovat určité rozhraní, aby bylo možné ji spustit v SAGE2TM. Tyto aplikace jsou spuštěny na serveru a přes klienta s nimi může uživatel manipulovat. Přes klienta lze také posílat informace přímo do aplikace a tím ovládat její vlastní funkce. Například pokud uživatel otevře textový editor, tak zvětšení nebo zmenšení je vnější manipulace, ale psaní do něj je vnitřní interakce.

1.11.1 Rozhraní aplikace

Aplikace v projektu SAGE2TM jsou napsány v jazyce JavaScript. Tyto aplikace musí splňovat určité rozhraní a v něm implementovat daný set metod. Každá z těchto metod je potom využívána v SAGE2TM. Metoda zavolaná při startu aplikace provede inicializaci a uvede aplikaci do výchozího stavu. Pokud dochází k ukončení aplikace, zavolá se také odpovídající metoda. Metoda která se často využívá je ta, která reaguje na změnu velikosti aplikace. Zde je potřeba provést úpravy, aby se aplikace správně škálovala. Nejpožívanější metoda je pro vykreslení aplikace. Tato metoda se volá v základním nastavení 60 krát za sekundu. V této metodě se odehrává život aplikace, a proto je nejdů-



Obrázek 1.2: SAGE2™ rozhraní klienta s otevřenými aplikacemi

ležitější ze všech. Druhá významná metoda slouží ke zpracování uživatelského vstupu.

Metodu pro zpracování uživatelského vstupu je potřeba analyzovat z důvodu možného využití v prezentaci. Například pokud aplikace přehrává video a podporuje nějaké akce jako pozastavení videa. Je možné, aby se tato funkce objevila v prezentaci. Metoda využívá 5 parametrů, typ, pozici, uživatelská data, data, čas. Typ určuje co za vstup byl přijat. Vstup může být akce myši nebo zmáčknutí klávesy na klávesnici. Pozice určuje místo akce. Uživatelská data nesou informaci o uživateli, který vstup způsobil. Data nesou různorodé informace a čas může být použit pro synchronizaci.

Z těchto informací lze určit, že všechna rozhodnutí o vstupu se nechávají na aplikaci samotné. Pro zjištění o jaké vstupy má aplikace zájem, je potřeba nahlédnout přímo do této aplikace. Klávesové vstupy jsou lehce napodobitelné strojem, protože nezáleží na pozici. Ovšem pokud je potřeba pro nějaké akce kliknout na určité místo, tak není jisté, jestli při různých velikostech zůstane kontrolní prvek na stejném místě. Pokud bych měl zavést ovládání vnitřku aplikací, tak bych musel každou z aplikací zvlášť nastudovat, abych je mohl úspěšně použít.

Z důvodu nedostatku obecnosti je přidání jiných akcí do editoru velmi obtížné a udržování konzistence by bylo přímo nemožné. Při každé úpravě aplikace nebo vydáním nové by se musel editor upravit. Z těchto důvodů nebudu vnitřní ovládání aplikací podporovat a může se jen objevit u některých

aplikací jako bonus.

1.12 Ovládání SAGE2™

Hlavní řídicí jednotkou SAGE2™ je server spuštěný v platformě NodeJS. Komunikace s tímto serverem probíhá pomocí WebSocketů. WebSockets umožňují posílání zpráv. Každá zpráva má určitou funkci. Skrze tyto zprávy se také udržují klienti v aktuálním stavu. Z toho vyplývá, že pro komunikaci se serverem je potřeba využít WebSocket. Všechny možné akce jsou tedy zachyceny v serverové části SAGE2™. Při potřebném rozšíření by se musel upravit server, aby podporoval dodatečné zprávy. Protože Websockets nejsou univerzální, tak můj kolega a člen týmu SAGELab, Ondřej Brém, vytváří ve své bakalářské práci[8] jednotné API pro ovládání SAGE2™. Je tedy možné, že bych toto API mohl také využít, pokud se projeví dosavadní ovládání jako nedostačující.

1.13 Komunikace

Komunikace mezi klienty a serverem probíhá pomocí WebSocketů. Poslané informace se posílají ve dvou formách. Buď v binární podobě, nebo v datovém typu string. Každá zpráva má dohodnutý tvar. První část zprávy obsahuje název zprávy a určuje, co se má provést. Druhá část jsou data. Některé zprávy potřebují poslat binární data. V tomto případě se zpráva převede do binární podoby a připojí se na začátek k datům. Avšak pokud data jsou objektového tvaru, tak se celá zpráva převede na JSON a ten se pošle skrz WebSocket. Samozřejmě toto rozhraní je způsobeno třídami, které obalují základní verzi WebSocketu. Ten posílá všechny data v binární podobě a je na okolních prostředcích, aby se s tím vypořádaly.

1.14 JavaScript

JavaScript[4] je objektově orientovaný skriptovací jazyk, který se zejména využívá pro webové stránky. Tento jazyk se vykonává ve webovém prohlížeči a nezatěžuje server. Ale lze ho použít i mimo prohlížeč, například na platformě Node.js. Podporu pro JavaScript lze nalézt v řadě moderních prohlížečů. Aplikace SAGE2™ je převážně napsaná v tomto jazyku, a proto se v plánovači určitě také vyskytne.

1.15 Node.js

Node.js[5] je platforma běžící na JavaScriptu. Její hlavní přednosti leží v jednoduchém vývoji škálovatelných síťových aplikací. Zvládá obsluhovat více různých

ných spojení s minimální odezvou a proto je vhodná pro vývoj aplikací s potřebou synchronizace. Na této platformě je nasazená serverová část aplikace SAGE2™.

1.16 HTML

HyperText Markup Language, HTML, je jeden z prvním jazyků využívaných na internetu. Jeho jednoduchá konstrukce umožňuje široké řadě programátorů v tomto jazyce psát. V tomto jazyce se klade důraz na zvýraznění důležitosti určitých částí textu. Přiřazování různých záměrů, tagů, je možné napsat formátovaný dokument.

1.17 CSS

Cascading style sheets je jazyk, který popisuje jakým způsobem máme zobrazit části dokumentu, který je napsán v HTML. Tyto dva jazyky se doplňují a s jejich pomocí lze napsat plnohodnotné statické webové stránky.

1.18 WebSocket

WebSocket[6] je protokol pro síťovou komunikaci. Tento protokol je součástí verze HTML5. Tento nový protokol poskytuje plně duplexní přenos, neboli je možné posílat a přijímat data ve stejný okamžik. Komunikace probíhá hlavně přes jediné TCP připojení, kromě navázání spojení, které se uskuteční přes HTTP protokol. HTML5 WebSocket specifikace definuje API, které lze využít ve webových prohlížečích. Ze strany serveru se definují 3 speciální funkce (event listeners), které se automaticky volají, pokud dojde k určité změně. Například lze definovat funkce a poté je přiřadit tak, že se zavolají při otevření nebo zavření spojení. Nejdůležitější funkcí je ta, která reaguje na přijmutí nových dat. Zde je potřeba analyzovat přijatá data a poté na ně reagovat korespondující událostí. Z klientské strany jsou k dispozici jen 2 funkce. Jedna, která pošle daná data, a druhá, která zavře navázané spojení.

1.19 JSON

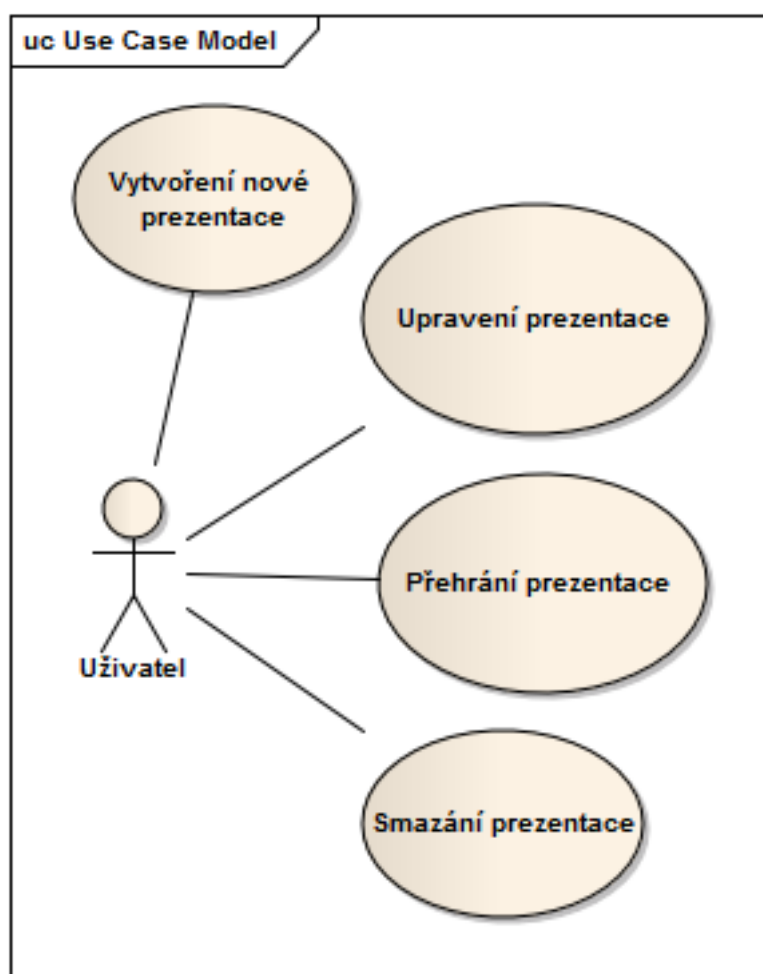
JavaScript Object Notation, JSON[7], je datový jazyk určený pro výměnu dat. Velkou výhodou je jednoduché čtení jak strojem, tak lidmi. Základní koncepce je zápis dobře známých datových struktur. Tyto hlavní struktury jsou: objekt, pole a hodnota. Objekt je seznam dvojic datového typu string spojený s hodnotou. Pole je seznam hodnot. Hodnota je: string, číslo, objekt nebo pole. Jak je zřejmé, tak se struktury mohou do sebe vnořovat.

Návrh

2.1 Případy užití

Případy užití popisují různé scénáře uživatelských aktivit. Je zde popsáno jakým způsobem má uživatel dosáhnout svého cíle a co všechno má k dispozici. V tabulce 2.1 jsou shrnuty funkční požadavky a případy užití. V navazující tabulce 2.2 je zobrazeno, které scénáře využívají jaké funkční požadavky.

1. Scénář „Vytvoření nové prezentace“
 - a) Uživatel se připojí na webového klienta SAGE2TM. Tento klient je volně dostupný na adrese serveru a slouží k ovládní aplikace SAGE2TM.
 - b) V tomto klientovi, lze najít ovládací panel, přes který se ovládají aplikace. V tomto panelu lze nalézt také možnost přejít do nově vytvořeného druhu klienta, do Editoru, v kterém se prezentace tvoří.
 - c) Pomocí nabídky kontrolních prvků začneme tvořit prezentaci. Akce, které lze provádět jsou uvedeny na obrázku 2.2 a dále v této kapitole. Pokud jsme spokojeni z touto obrazovkou, tak ji uložíme tím, že přejdeme na další krok. V dalším kroku zase provedeme další akce a postupně tímto sestavíme celou prezentaci.
 - d) Při ukládání celé prezentace je potřeba zadat jméno, pod kterým se tato prezentace uloží. Jméno se nesmí shodovat se jménem již uložené prezentace.
2. Scénář „Úprava prezentace“
 - a) Uživatel se připojí na webového klienta SAGE2TM.
 - b) V tomto klientovi se nacházejí ovládací prvky pro správu aplikace SAGE2TM a mezi nimi lze nalézt tlačítko pro přechod do Editoru.



Obrázek 2.1: Diagram případů užití

- c) V Editoru je zapotřebí nahrát již existující prezentaci. V prohlížeči souborů je potřeba jít do složky "presentations", kde se ukládají vytvořené prezentace. Zde si uživatel vybere prezentaci kterou chce upravit a potvrdí výběr. Tímto se mu prezentace nahraje do editoru. Přes ovládání si uživatel může vybrat který krok prezentace je zapotřebí upravit nebo uživatel může pokračovat s přidáváním nových kroků.
- d) Po skončení všech úprav se prezentace uloží. Není potřeba psát jméno prezentace, protože už jednou bylo zadáno.

3. Scénář „Přehrání prezentace“

- a) Uživatel se připojí na webového klienta SAGE2™.

- b) V kontrolních prvcích lze nalézt prohlížeč souborů serveru. Zde ve složce “presentations” si uživatel najde požadovanou prezentaci, vybere a otevře ji.
 - c) Otevřením prezentace se přidají nové ovládací prvky. Ty slouží k prohlížení a ovládání prezentace. Při otevření se začíná ihned přehrávat a načte se ihned první krok prezentace.
 - d) Pokud si uživatel přeje přejít na další krok, tak klikne na tlačítko pro další krok prezentace. Druhé tlačítko, které se přidalo při přehrávání prezentace, slouží k zastavení prezentace. Když se prezentace zastaví, tak se automaticky uzavřou všechny aplikace, které byly touto prezentací spuštěny.
 - e) V průběhu prezentace je možné zasahovat do jednotlivých aplikací. Ovládání není nijak omezeno a chová se stejným způsobem jako když prezentace zapnutá není. Ovšem uživatel nesmí provést takové změny v aplikaci SAGE2TM, které poruší prezentaci. Například odstranění aplikace z prezentace, když v dalším kroku má být zvětšena.
 - f) Pokud prezentace je zastavena pomocí ovládacího prvku zastavení, tak se klient vrátí do základní konfigurace. To samé se stane, když prezentace dosáhne konce.
4. Scénář „Smazání prezentace“
- a) Uživatel se připojí na webového klienta SAGE2TM.
 - b) V prohlížeči souboru ve složce “presentations” označíme a odstraníme ho. Po odstranění není žádná možnost obnovení tohoto souboru.

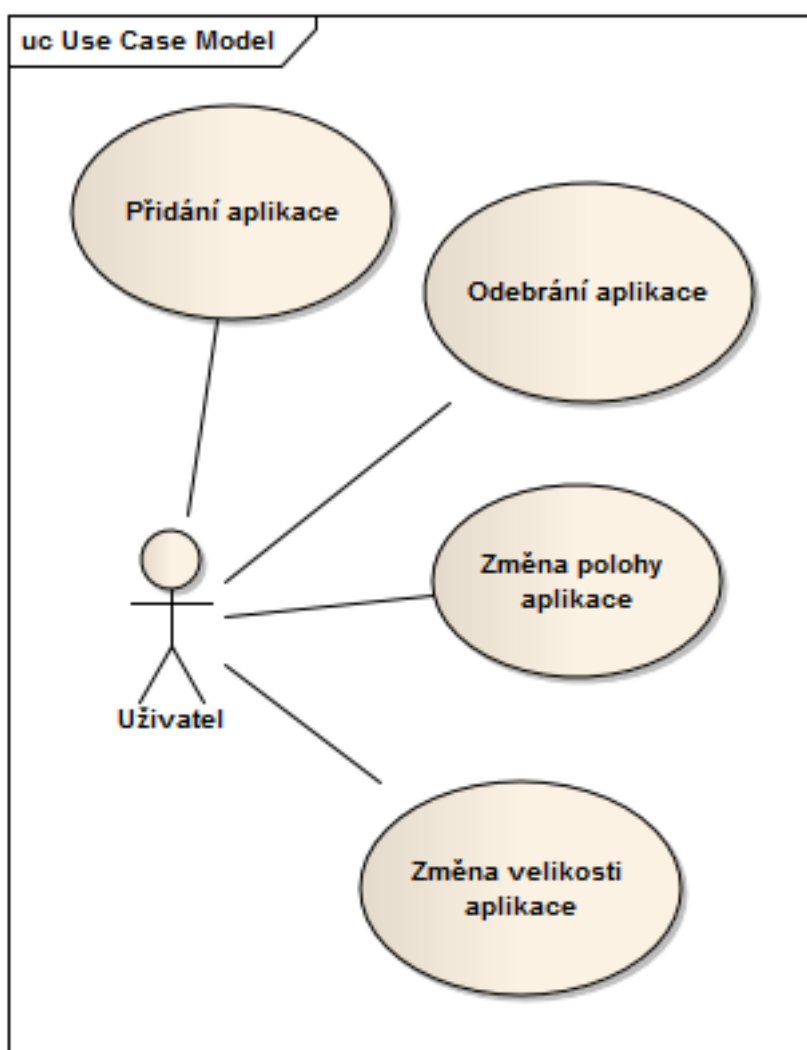
Scénáře pro akce prezentace

5. Scénář „Přidání aplikace“

Otevřeme dialog aplikací a vybereme požadovanou aplikaci. Je možné přidat více druhů stejných aplikací.

6. Scénář „Odebrání aplikace“

Na plátně kliknutím vybereme požadovanou aplikaci a klikneme na ovládací prvek pro odstranění vybrané aplikace. Druhá možnost je kliknout levým tlačítkem myši do horního pravého rohu zobrazené aplikace.



Obrázek 2.2: Diagram případů užití možných akcí

7. Scénář „Změna polohy aplikace“

Označíme aplikaci, kterou chceme zmenšit a tím se vyplní údaje o její poloze do ovládacích prvků. Souřadnice jsou zobrazeny v editovatelných políčkách a jsou napsány číselnými hodnotami, které ukazují vzdálenost aplikace od horního levého rohu. Druhá možnost je označit aplikaci a držením levého tlačítka myši aplikaci posunout.

8. Scénář „Změna velikosti aplikace“

Označíme aplikaci, kterou chceme zmenšit a tím se vyplní údaje

o její poloze do ovládacích prvků. Zde nalezneme políčka s velikostí aplikace a ty můžeme upravit. Druhá možnost je zaměřit kurzor myši nad požadovanou aplikaci a pomocí kolečka myši měnit velikost aplikace.

Tabulka 2.1: Souhrn funkčních požadavků a případů užití

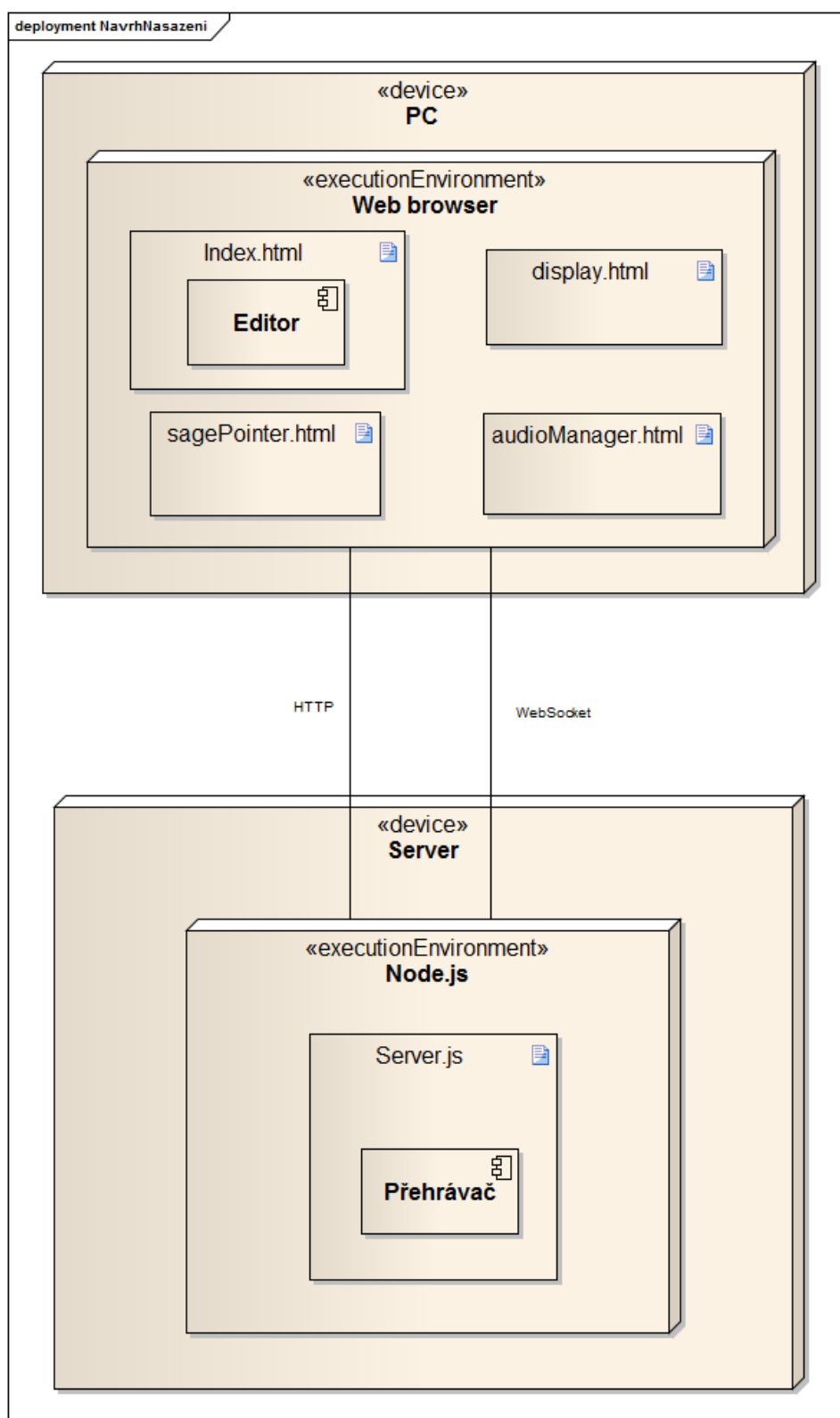
#	Funkční požadavky
F1	Vytvoření nové prezentace
F2	Úprava existující prezentace
F3	Duplikace prezentace
F4	Rozlišení verzí prezentace
F5	Odstranění prezentace
F6	Možnost spuštění akce uživatelem
F7	Akce po určitém časovém úseku
F8	Náhrávání prezentace
F9	Spuštění prezentace
F10	Zastavení prezentace
F11	Možnost kombinovat prezentaci s uživatelským vstupem

#	Případy užití
PU1	Vytvoření nové prezentace
PU2	Úprava prezentace
PU3	Přehrání prezentace
PU4	Smazání prezentace
PU5	Přidání aplikace
PU6	Odebrání aplikace
PU7	Změna polohy aplikace
PU8	Změna velikosti aplikace

2.2 Technologie

Projekt SAGE2TM je založen na webových technologiích. Jednotlivé stránky jsou ve formě statických HTML stránek. Jejich vzhled je upraven pomocí CSS. Veškerá logika aplikací a jejich dynamické fungování zajišťují knihovny napsané ve skriptovacím jazyce JavaScript. Celé schéma nasazení si můžete prohlédnout na obrázku 2.3. Z důvodů N1 je potřeba využít technologie, které se používají ve webovém prohlížeči a pokud k tomu

2. NÁVRH



Tabulka 2.2: Mapování případů užití na funkční požadavky

	PU1	PU2	PU3	PU4	PU5	PU6	PU7	PU8
F1	✓							
F2		✓						
F3								
F4								
F5				✓				
F6					✓	✓	✓	✓
F7								
F8								
F9			✓					
F10			✓					
F11			✓					

přidáme N2, tak jediným způsobem, jak tohoto docílit, je pokračovat ve využívání stejných technologií. Díky tomu se také zachová přehlednost a konzistence projektu. V mé práci budu potřebovat zasáhnout do všech těchto technologií. JavaScript pro hlavní logiku, HTML pro úpravu stránek a CSS pro správné zobrazování obsahu v HTML stránkách.

2.3 Editor

Základní návržení editoru se bude velmi podobat nynějšímu ovládání. Jedním z hlavních důvodů je, že uživatelé nebudou mít problém s použitím editoru, protože na toto rozhraní budou již uživatelé zvyklí. Prvky, které je potřeba přidat, tvoří hlavně panely pro zobrazení informací. Hlavní z panelů bude informovat uživatele o právě upravovaném kroku v sekvenci a bude umožňovat procházení mezi jednotlivými body sekvence. Další rozhraní, které je potřeba poskytnout jsou možnosti pro ovládání jednotlivých aplikací. Naposledy budou prvky pro obecnou kontrolu.

2.3.1 Návrh implementace editoru

Editor vznikne úpravami webového klienta. To poskytne výhodu pro uživatele, kteří jsou z klientem zvyklí pracovat. Pokud uživatel využívá klienta, tak každá jeho akce je přeložena do WebSocketových zpráv a posílána serveru ke zpracování. Editor některé z těchto zpráv nebude posílat na server, ale bude z nich vytvářet prezentaci. Kvůli tomu, že nebude využívat server pro zpravování aplikací, tak nebude blokován od jeho normální funkce. Avšak server stále bude potřeba pro různé operace například: načítání souborů na serveru nebo načtení ikony aplikace. Nevýhodou tohoto přístupu je potřeba implementování určité logiky serveru přímo v editoru. Tato logika se hlavně zabývá správou otevřených aplikací a jejich pohyb po plátně (canvasu).

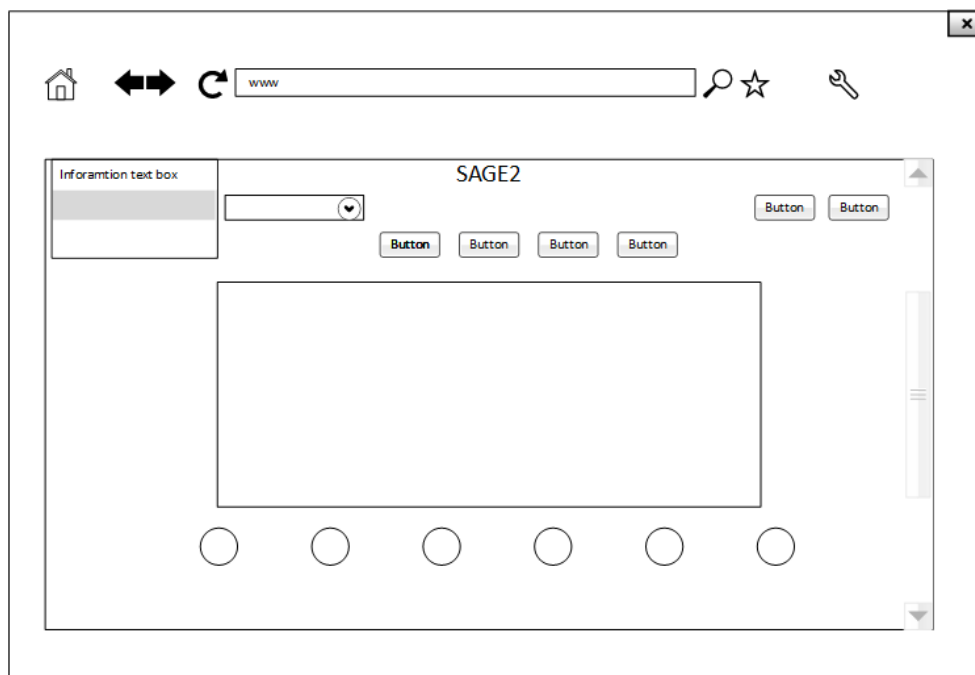
Bude tedy potřeba vytvořit třídu pro správu těchto aplikací. Pomocí souřadnic kurzoru je možné identifikovat aplikaci, na kterou uživatel kliknul. Tyto souřadnice lze porovnat se souřadnicemi a pozicí otevřených aplikací. Pokud se zde nachází více aplikací (jsou na sobě), tak se vybere ta v popředí. Pohyb aplikací se bude provádět pohybem kurzoru při držení levého tlačítka myši.

2.3.2 Ovládání editoru

Hlavním prvkem je plocha kde se zobrazují otevřené aplikace. Na této ploše jsou neměnné náhledy na otevřené aplikace a je možné s nimi pohybovat. Při výběru jedné z otevřených aplikací se zobrazí v informačním dialogu informace o této aplikaci a akce které jsou k dispozici. Z informací o aplikaci uživatel bude vědět co se s touto aplikací v tomto kroku prezentace stane. Po nastavení akcí s jednotlivými aplikacemi, uživatel uloží krok prezentace a může začít připravovat krok další. Po vyhotovení celé prezentace může ji poté uživatel uložit na server. Editor bude poskytovat možnost otevřít již uloženou prezentaci a upravit ji.

2.3.3 Uživatelské rozhraní

Náčrt rozhraní je vidět zde 2.4. Uživatelské rozhraní je tvořeno třemi skupinami prvků. Jedny jsou informačního řádu a slouží k zobrazování informací. Další poskytují ovládání pro uživatele. A poslední jsou kombinované, kde zároveň slouží k zobrazení informací a ovládání aplikace. V prostředku se bude nacházet plátno, kde se zobrazují spuštěné aplikace a uživatel s nimi na tom plátně může manipulovat. Tímto si uživatel nastaví jejich pozice a velikost. V levém horním rohu se budou vyskytovat informace o jednotlivých aplikacích a v pravém horním rohu se budou



Obrázek 2.4: Wireframe Editoru prezentací

nacházet prvky pro obecné nastavení editoru. Pod již zmíněným plátně se budou nacházet prvky pro manipulaci s aplikacemi.

2.3.4 Integrace kódu

Základ editoru bude tvořit webový klient používaný pro ovládání aplikací v SAGE2TM. Tento klient obsahuje plátno na kterém jsou schématicky vykresleny otevřené aplikace. Pod tímto plátnem se vyskytují kontrolní prvky. Přes ně je možné například spustit novou aplikaci nebo zahájit sdílení kurzoru do displeje SAGE2TM.

Jedny úpravy se především budou zaměřovat na přidání nových potřebných ovládacích prvků a informačních panelů. Tyto prvky poskytnou nové funkce, přes které bude možno vytvořit, upravit a naposledy uložit vytvořenou prezentaci. Bude tedy potřeba upravit zdrojovou HTML stránku. Z grafického hlediska pak bude potřeba upravit přiložené stylové příkazy v souborech typu CSS.

Zadruhé je potřeba upravit logiku. Zakázat odesílání zpráv serveru a lokálně zpracovávat příkazy. Server nemůže být ovlivňován klientem, který pracuje na tvorbě prezentace a musí pracovat normálně.

2.4 Přehrávač

Zde nebude potřeba vytvářet složité ovládání. Je potřeba hlavně spustit a zastavit prezentaci. Tohoto lze docílit pomocí páru ovládacích prvků zasazených přímo do klientského ovládání SAGE2TM. Hlavní práce bude načíst již vytvořenou prezentaci z paměti. Ovládání bude poskytnuto skrze WebSocketové zprávy. Průchod prezentace bude probíhat čtením prezentačního souboru postupně po jednotlivých krocích.

2.4.1 Umístění jádra přehrávače

Server

Upravování serveru přidává na jeho složitosti a může ovlivnit jeho výkonnost. Výhodou, ale je kompaktnost kódu a větší přehlednost. Protože by se logika odehrávala na serveru, tak by klienti nemusely nic dělat a každý z těchto klientů by měl možnost interagovat s prezentací. Informace by byly pohromadě a nemuseli by přicházet po síti. tento způsob by přidal na spolehlivosti. Také je možné podporovat více různých klientů v budoucnu.

Klient

Implementace na straně klienta povede k menším úpravám serverové části. Docílí se omezení, kde jen někteří klienti budou moci využívat prezentaci. Logika by se prováděla lokálně a nezatěžovala by server. Ovšem každý krok prezentace se může skládat z mnoha akcí. Všechny tyto akce je potřeba odeslat serveru. To vede k většímu nároku na komunikaci a zatížení sítě. V případě, že by síť nebyla stabilní nebo s velkou odezvou, tak narušení prezentace bude zjevné. Další omezení je obtížnější ovládání prezentace pomocí více klientů. Jen ten klient který spustil prezentaci by ji mohl ovládat.

Závěr

Každé z řešení má své klady a zápory. Po vyhodnocení vychází serverové řešení jako lepší varianta z následujících důvodů:

- Server se v obou variantách musí upravit a je lepší uchovávat kód blíž u sebe pro přehlednost
- Sníží se potřebná komunikace a je menší riziko neočekávaného chování kvůli problémům se sítí.

- Jakmile by se něco stalo s klientem, stačí se jen znovu připojit a neovlivní to prezentaci na serveru.

2.4.2 Návrh implementace přehrávače

I když prezentaci nebudou tvořit již využívané WebSocketové zprávy, je potřeba využít podobný přístup. Prezentace nabízí různé operace, které se postupně budou vykonávat. Lze tedy navrhnout každou z těchto zpráv do daného záznamu. Tento záznam musí obsahovat dostatečně hodnot, aby se akce mohla úspěšně provést. Každá akce se od sebe různí, ale některé hodnoty jsou pro všechny nutné. Je potřeba znát typ akce. Pokud máme tuto informaci, můžeme rozhodnout jaké hodnoty je potřeba dále načíst. V tomto poskytuje formát JSON neskutečnou výhodu. Není potřeba definovat všechny druhy proměnných předem a není ani potřeba dodržovat jejich počet.

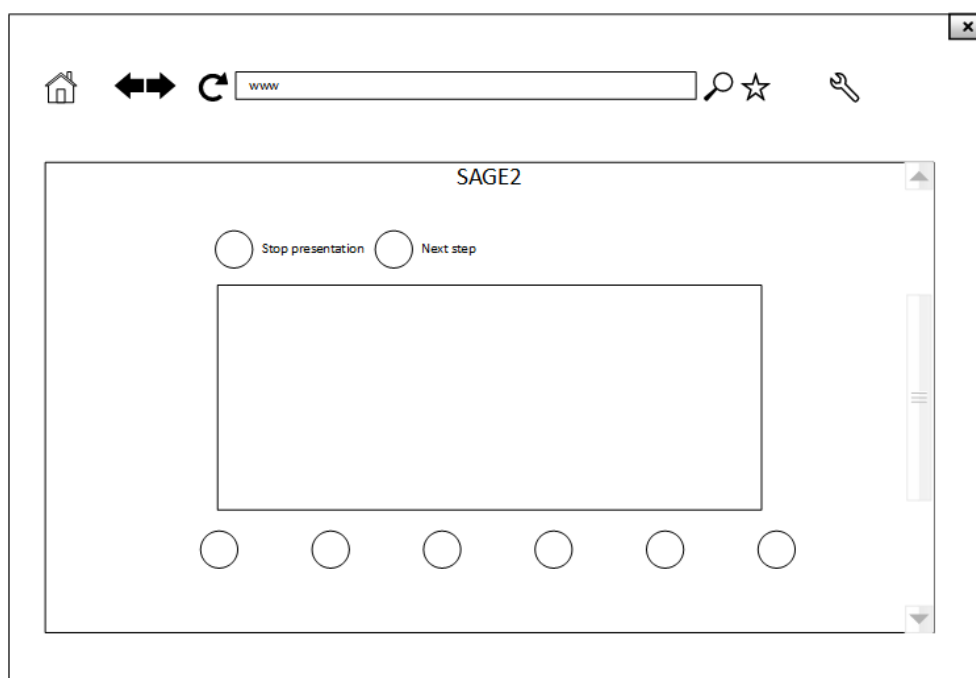
Každá ze zpráv bude mít vlastní funkci. Pokud se vyskytne problém ve formátu nebo v obsahu zprávy, tak se tato zpráva neprovede. Pro budoucí možná rozšíření je potřeba přidat záznam o novém typu zprávy a nasměrovat ji na správnou funkci. Všechna logika se vykonává na serveru a klient jen posílá zprávy pro pokračování, otevření a zavření prezentace.

2.4.3 Integrace kódu

Nový kód bude součástí hlavního souboru `server.js`. Je potřeba zachovat stávající funkce a neovlivnit ostatní chování serveru. Pokud to půjde, tak se budu vyhýbat úpravám již existujících funkcí a radši budu duplikovat tyto funkce. Tím se minimalizuje dopad nového kódu na předchozí funkce. Duplicita kódu neovlivní výkonnost, jen lehce zhorší orientaci uvnitř kódu. Hlavním úkolem bude přidat další možné WebSocketové zprávy. Díky nim potom můj upravený klient využívat další nové funkce. Pro přidání těchto zpráv se využije již hotových knihoven, které implementují základní JavaScript WebSokety.

2.4.4 Forma Prezentace

Komunikace se serverovou částí SAGE2™ probíhá pomocí zpráv přes websocket rozhraní. Prezentace bude probíhat podobným způsobem. Avšak nebudou použity přímo již použité WebSocketové zprávy, ale využiji podobný systém. Vytvořím vlastní typ zpráv, který se nebude posílat přes internetové spojení, ale rovnou číst a provádět na straně serveru. Při čtení zdroje prezentace se každá uložená zpráva přečte a logika přehrávače vykoná požadovanou akci. Normální WebSocketové zprávy nebudou nijak



Obrázek 2.5: Wireframe Přehrávače prezentací

blokovány, a proto bude možné v průběhu prezentace ovládat SAGE2™ normálně. Pokud ovšem uživatelé hrubě naruší tok prezentace, například zavřením aplikace, nebude moci prezentace pokračovat. Prezentace se bude ukládat ve formátu JSON na úložišti SAGE2™.

2.4.5 Uložená prezentace

Jak už je v kapitole 1.19 popsáno je JSON ideálním formátem pro uložení informací, zvláště pro informace využívané v jazyce JavaScript. Knihovna JSON poskytuje 2 funkce pro jednoduché převádění objektu na JSON a obráceně. Objekt je převeden na text a ten se potom uloží do souboru. Jádro prezentace tvoří jeden hlavní objekt s mnoha vlastnostmi. Nejdůležitější bude vlastnost, kde se uloží jednotlivé kroky prezentace. Každý krok je pole zpráv, kde každá zpráva je objekt. Jedna zpráva vykoná jednu akci.

2.4.6 Uživatelské rozhraní

Náčrt je dostupný na obrázku 2.5. Uživatelské rozhraní se bude vyskytovat na klientské straně v souboru “index.html”. Nové uživatelské prvky zasadím do stávajícího menu. Z pohledu přehrávače jsou potřeba 2 ovlá-

dací prvky. První z nich je potřeba pro zastavení prezentace. Druhý prvek bude pro posun prezentace na další krok. Přes tento prvek se bude kontrolovat průchod prezentací. Uživatel bude spouštět prezentaci vybráním prezentačního souboru v prohlížeči mediálních souborů.

2.5 Prototyp

Protože se jedná o upravování již existujícího projektu, tak se těžko odhaduje proveditelnost a pracnost všech požadavků. V implementaci se proto zaměřím hlavně na základní funkce, které by umožnily používání mého rozšíření. Skrze používání od různých uživatelů se zjistí jaké další funkce jsou potřeba a naopak které funkce jsou zbytečné nebo v praxi nepoužitelné.

Pro editor jsou určitě nutné základní funkce pro vytvoření nové prezentace a pro její úpravu. To zahrnuje požadavky F1 a F2. Když se dá prezentace vytvořit, tak je dobré ji umět i smazat, to zahrnuje požadavek F5. Tyto funkce tvoří začátek editoru. Na to je možné navázat tvorbou prezentace a přidáváním nových aplikací. Manipulaci s jejich velikostí a polohou. A možnost zrušení aplikace. Každá aplikace má tyto možnosti a díky nim se dá vytvořit základní prezentace. Tento požadavek lze nalézt v F6. Prezentace se neukládá na u klienta, ale je poslána na server kde se uloží. V přehrávači budou implementovány všechny požadavky (F9, F10, F11), protože jsou všechny nezbytné k fungování prezentace.

Ostatní požadavky nebudou zatím implementovány a bude se o nich uvažovat až po úspěšném nasazení a vyzkoušení prototypu.

Realizace

3.1 Editor

Upravené soubory jsou různě rozděleny po složkách v projektu. Rozdělení je z důvodů přehlednosti a využití různých technologií. Většina souboru je rozdělena pospolu podle jeho typu. Většina úprav které jsem provedl je k nalezení v novém souboru „SAGE2_editor.js“, kde se vyskytuje hlavní logika. Ovládací prvky se nacházejí v souboru „editor.html“. Základ tohoto souboru byl převzat z „index.html“, který vlastně tvoří klienta. Stejně platí i pro stylovací soubor „style_ui_editor.css“, který upravuje grafiku zobrazených prvků, kde se jedná o rozšířenou verzi souboru „style_ui.css“. Využívány jsou i ostatní JavaScriptové knihovny, které byli použity v klientovy.

3.2 Přehrávač

Přehrávač lze nalézt přímo v souboru „server.js“, který je hlavní součástí SAGE2. To je z důvodu, protože funkce, které reagují na přijímání WebSocketových zpráv, jsou přímo v tomto souboru. Přehrávač také podporuje určité tyto zprávy a je tedy spolu s ostatními v tomto souboru. Každý typ zprávy uložený v prezentačním souboru ve formátu JSON reprezentuje určitou funkci, která vykoná potřebné události.

3.3 Instalační příručka

Nejdříve je zapotřebí nainstalovat aplikace SAGE2. Všechny potřebné informace jsou k nalezení na stránkách <https://bitbucket.org/sage2/sage2/wiki/Home>. Pro instalaci nového rozšíření lze využít instalačního skriptu. Tento skript naleznete v příloženém mediálním nosiči. Přesné

pokyny k instalaci jsou přiloženy do README, které naleznete spolu s instalačním skriptem.

3.4 Uživatelská příručka

Uživatelskou příručku naleznete na přiloženém mediálním nosiči. V této příručce je popsáno, jak se z rozšířením zachází. Jsou zde popsány všechny dovednosti a vysvětleno, jak je možné je využít.

3.5 Návod na budoucí rozšíření

Přidání podpory pro nové funkce se dá implementovat následujícím způsobem. Začneme u editoru. Pokud jsou potřeba další ovládací prvky, tak se musejí přidat v souboru „editor.html“. Pokud tento prvek není přímo button, lze na něj navázat funkcionalitu ve funkci „handleClick“ v knihovně „SAGE2_UI_editor“. Přes „addPresentationEvent“ z knihovny „SAGE2_editor.js“ přidejte akci do logu tohoto kroku. Na serverové části upravte funkci „wsNextStep“ o novou akci a navažte na to vaší funkci. Tímto způsobem byste měli úspěšně přidat vámi navrženou a implementovanou novou funkcionalitu.

Testy

V průběhu implementace jsem vše řádně vyzkoušel a ujistil se, že vnitřní funkce pracují správně. Z mého pohledu jsem nenarazil na žádný problém.

Další test proběhne po dohodě s vedoucím práce o letních prázdninách roku 2015. Tento test je uživatelského typu a je zaměřený na ovladatelnost a použitelnost. Podstata spočívá v tom, že test se skládá z mnoha menších úkolů, které má uživatel provést. Uživatel po skončení každého z těchto úkolů provede ohodnocení. Testy se podají jak osobám neznalých ovládání aplikace SAGE2TM, tak i osobám kteří mají s tímto softwarem zkušenost. Od těchto testu očekávám zjištění jestli je rozšíření v praxi použitelné a jaké další funkce jsou žádané.

4. TESTY

SAGE2 - Prezentace

Tento dotazník slouží k otestování všech funkcí a návrhu uživatelského rozhraní přídatku pro tvorbu a přehrávání prezentací v aplikaci SAGE2. Prosím postupně se snažte o splnění úkolů a napište jestli se vám to povedlo nebo pokud jste narazili na nějaký problém. Děkuji za vaší snahu.

Otevřete media browser a vyberte zde ve složce "presentation" soubor s názvem "Presentation_example.prs". Prohlédněte si celou prezentaci.

- Uspěl jsem a neměl jsem žádný problém
- Měl jsem problém s otevřením prezentace
- Měl jsem problém s nalezením tlačítka pro ovládání prezentace
- Jiné:

Přepněte klienta do editoru prezentací, vytvořte novou prezentaci a pojmenujte ji.

- Uspěl jsem a neměl jsem žádný problém
- Měl jsem problém s přepnutím do editoru
- Měl jsem problém s vytvořením nové prezentace
- Měl jsem problém s pojmenováním prezentace
- Jiné:

Přidejte do okna 2 aplikace, nastavte jejich polohu jak uvážíte za vhodné a potvrďte tento krok prezentace.

- Uspěl jsem a neměl jsem žádný problém
- Měl jsem problém s přidáním aplikací
- Měl jsem problém s označením aplikace
- Měl jsem problém s pohnutím aplikace
- Měl jsem problém s přejitím na další krok prezentace
- Jiné:

Obrázek 4.1: dotazník pro otestování rozšíření část 1

Změňte polohu aplikací a uložte krok prezentace.

Uspěl jsem a neměl jsem žádný problém

Měl jsem problém s označením aplikace

Jiné:

Změňte velikost aplikací a uložte krok prezentace

Uspěl jsem a neměl jsem žádný problém

Měl jsem problém se změnou velikosti aplikace

Jiné:

Odstraňte jednu z aplikací a uložte krok prezentace.

Uspěl jsem a neměl jsem žádný problém

Měl jsem problém s odstraněním aplikace

Jiné:

Uložte celou prezentaci a pojmenujte ji.

Uspěl jsem a neměl jsem žádný problém

Měl jsem problém s uložením prezentace

Měl jsem problém s pojmenováním prezentace

Jiné:

Přepněte z editoru zpět do prohlížeče, spusťte a prohlédněte si vaši nově vytvořenou prezentaci.

Prezentace byla taková jaká měla být

Jiné:

Všechny připomínky, nápady a jiné komentáře prosím napište sem.

Obrázek 4.2: dotazník pro otestování rozšíření část 2

Závěr

Cílem této bakalářské práce bylo analyzovat nově vydanou aplikaci SAGE2, se zaměřením na její prezentační schopnosti. Při analýze jsem zjistil, že předpokládané dovednosti prezentační části nejsou dostatečné. Po získání vyžadovaných funkcí od většiny uživatelů této aplikace v laboratoři SAGELab, jsem se rozhodl navrhnout a implementovat nový způsob vytvoření a přehrání prezentace. Tento způsob zahrnoval úpravy přímo do jádra aplikace SAGE2.

Současnou implementaci jsem rozšířil jen o část požadavků a to hlavně z důvodů složitosti vnitřního návrhu, architektury aplikace a neexistující dokumentace. Pro splnění určitých požadavků bych musel provést mnoho úprav do hlavních součástí SAGE2 a na to jsem neměl dostatek času. Ovšem finální verze je plně funkční a lze v ní vytvořit prezentace. Tyto prezentace je možné i spustit.

Z této práce jsem si odnesl hodně zkušeností s nasazením webových technologií a jejich nespočetné možnosti využití. Dozvěděl jsem se mnoho informací ze světa, kde se používá velmi vysoké rozlišení. Tuto zkušenost velice kladně hodnotím. Poprvé jsem měl také možnost zasáhnout do již vytvořeného projektu a připomněl jsem si, jak je vypracovaná dokumentace důležitá.

V budoucnu má aplikace SAGE2 mnoho potenciálu. Stěny složené z více obrazovek jsou častější, a proto bude větší poptávka po softwaru, který by plně využil onoho vysokého rozlišení. Prezentace by se mohla rozšířit o další možné akce. Pokud by se přidala podpora takovýchto akcí přímo do požadovaného rozhraní pro aplikace v SAGE2. Bylo by je možné jednotlivě načítat a i využívat v prezentacích. Tímto by byla kontrola těchto aplikací zaručena.

Literatura

- [1] Fotogalerie. *Sagelab* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <http://sagelab.cesnet.cz/fotogalerie-2>
- [2] *SAGE2* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <http://sage2.sagecommons.org>
- [3] *SAGE2 dokumentace* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <https://bitbucket.org/sage2/sage2/wiki/Home>
- [4] JavaScript. *Mozilla developer network* [online]. ©2005-2015 [cit. 2015-05-11]. Dostupné z: <https://developer.mozilla.org/cs/docs/Web/JavaScript>
- [5] *Node.js* [online]. ©2015 [cit. 2015-05-11]. Dostupné z: <https://nodejs.org/>
- [6] *WebSocket* [online]. 2013 [cit. 2015-05-11]. Dostupné z: www.websocket.org
- [7] *JSON: JavaScript Object Notation* [online]. 1999 [cit. 2015-05-11]. Dostupné z: www.json.org
- [8] Brém, Odnřej. *SAGELab - WWW manager zařízení SAGE*. Bachelářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.
- [9] Beta release. *SAGE2* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <http://sage2.sagecommons.org/beta-release/>
- [10] O laboratoři. *SAGELab – Síťová multimediální laboratoř* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <http://sagelab.cesnet.cz/o-laboratori/>

Seznam použitých zkratk

- JSON** JavaScript Object Notation
- HTML** HyperText Markup Language
- SAGE** Scalable Adaptive Graphics Environment
- SAGE2** Scalable Amplified Group Environment
- CESNET** Czech Education and Scientific Network
- ČVUT** České Vysoké Učení Technické
- HD** High Definition
- EVL** Electronic Visualization Laboratory
- UIC** University of Illinois at Chicago
- HTTP** Hypertext Transfer Protocol
- HTTPS** Hypertext Transfer Protocol Secure
- API** Application Programming Interface
- TCP** Transmission Control Protocol
- CSS** Cascading Style Sheets

Obsah přiloženého CD

readme.txt	popis obsahu CD
└─ src	
├─ impl	zdrojové kódy implementace
├─ help	příručky
├─ thesis	zdrojová forma práce ve formátu \LaTeX
├─ thesis.pdf	text práce ve formátu PDF
└─ assignment.pdf	zadání práce