



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Správce prezentací pro SAGE
Student:	Marek Peša
Vedoucí:	Ing. Ji í Chludil
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je realizace systému pro tvorbu, správu a zobrazování prezentací na SAGE.

1. Analyzujte nástroje a prostředí pro tvorbu prezentací v prostředí SAGE.
2. Analyzujte požadavky potenciálních p ednášejících na nástroj pro tvorbu prezentací.
3. Navrhn te webovou aplikaci pro správu a vytvá ení prezentací.
 - Požadovaná funkcionálna správy prezentací: správa autor , sdílení prezentací, kontrola konzistence (dostupnost zdroj), klonování, editace, odebrání.
 - Požadovaná funkcionálna vytvá ení prezentací: management zdroj (obrázky , pdf), skládání prezentace ze zdroj (pozicování, asování) , náhled, export do SAGE skript .
4. Navrhn te úpravy prostředí SAGE tak, aby bylo možné automaticky p ehrávat prezentace.
5. Podrobte ešení vhodným test m a vypracujte uživatelskou dokumentaci.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 10. února 2017

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



Bakalářská práce

Správce prezentací pro SAGE

Marek Peša

Vedoucí práce: Ing. Jiří Chludil

12. května 2017

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce Ing. Chludilovi za jeho vedení a pomoc s vývojem aplikace ve frameworku Nette, se kterým jsem nikdy dříve nepracoval. Dále bych chtěl poděkovat personálu laboratoře SAGElab za pomoc v raných částech projektu. A své rodině 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 12. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Marek Peša. 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

Peša, Marek. *Správce prezentací pro SAGE*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato práce se zabývá vývojem aplikace, která usnadní uživatelům vytváření a přehrávání prezentací v prostředí SAGE. Umožňuje přihlašování uživatelů, správu zdrojů (obrázků), ze kterých se prezentace skládá a v neposlední řadě vytváření a spouštění prezentací

Součástí této práce je také podrobná analýza uživatelských požadavků a současných možností vytváření prezentací, návrh webové aplikace využívající framework Nette včetně návrhu databázového modelu a uživatelského rozhraní. Na závěr práce bude aplikace podrobena sadě jednotkových, uživatelských a akceptačních testů.

Klíčová slova SAGE, SAGElab, prezentace, správce, automatizace, přednášení, výuka, uživatelská přívětivost, C++, PHP, Nette

Abstract

This thesis focuses on developing the application, that will simplify presentation creation and launching using SAGE. The application handles management of users, presentation resources (images), cration and launching.

This thesis contains the analysis of user's requirements and current ways of creating presentations. Design of the new web application using the Nette

framework, including design of the database model and the user interface. In the end, the application undergoes unit, user and acceptance testing.

Keywords SAGE, SAGElab, presentation, manager, automatisisation, lecturing, tuition, user friendly, C++, PHP, Nette

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Laboratoř SAGElab	5
2.2 SAGE	6
2.3 Architektura SAGE	6
2.4 Současná řešení vytváření prezentací v SAGE	8
2.5 Funkční požadavky	11
2.6 Nefunkční požadavky	13
2.7 Případy užití	14
2.8 Mapování funkčních požadavků na případy užití	20
3 Návrh	23
3.1 Procesy v aplikaci	23
3.2 Doménový model tříd	24
3.3 Databázový model	24
3.4 Uživatelské rozhraní	26
4 Realizace	31
4.1 Program Sage Viewer	31
4.2 Webová aplikace pro správu uživatelů, prezentací a dat	33
4.3 Diagram nasazení	37
5 Testování	39
5.1 Jednotkové testy	39
5.2 Uživatelské testování použitelnosti	40
5.3 Uživatelské akceptační testování	42

Závěr	43
Zhodnocení výsledků	43
Možnosti dalšího vývoje	44
Literatura	45
A Seznam použitých zkratk	47
B Obsah přiloženého CD	49

Seznam obrázků

2.1	Telestěna během přednášky Grafické čtvrtky, Zdroj:[1]	6
2.2	Architektura SAGE, Zdroj:[2]	7
2.3	Ukázka programu Sage UI. Zdroj:[3]	9
2.4	Ukázka jednoduchého scriptu spouštějícího obrázky. Zdroj:[4]	10
2.5	Model případů užití	14
2.6	Případ užití registrace	15
3.1	Diagram vytvoření sekce	24
3.2	Diagram spuštění prezentace	25
3.3	Doménový model	26
3.4	Databázový model	27
3.5	Wireframe přehledu prezentací	28
3.6	Wirefram edetailu prezentace	28
3.7	Wireframe detailu sekce	29
4.1	Ukázka skládání prezentace 1.	33
4.2	Ukázka skládání prezentace 2.	34
4.3	Ukázka spuštění prezentace v prohlížeči.	35
4.4	Ukázka přihlášení Shibboleth.	35
4.5	Diagram nasazení	38

Seznam tabulek

Úvod

Prezentace jsou pro členy akademické obce na denním pořádku. Laboratoř SAGELab dává kantorům a studentům k dispozici velkou plochu s vysokým rozlišením, na které lze zobrazovat obrázky, videa a další multimédia. Laboratoř SAGELab se často zapojuje přímo do výuky, nebo se v ní konají přednášky na téma počítačová grafika a další.

Aplikace, která z této práce vzejde, je určena pro studenty, zaměstnance FIT ČVUT v Praze a externisty, kteří potřebují prezentovat zvolené téma a chtějí použít nejmodernější technologie zobrazování v ultra vysokém rozlišení.

Téma jsem si zvolil, neboť problém vytváření prezentací na SAGELab nebyl doposud uspokojivě vyřešen. Současná řešení jsou velice neefektivní a vyžadují, aby uživatel měl základní znalosti práce se počítačem skrze příkazovou řádku v jazyce Shell.

Cíl práce

Cílem této práce je analyzovat současné možnosti vytváření prezentací v prostředí SAGE.

Analyzovat požadavky budoucích uživatelů aplikace, kteří jsou převážně kolegové z FIT ČVUT, kteří přednášejí v rámci některých z akcí, které jsou v laboratoři SAGElab pořádány.

Následně navrhnout, naimplementovat a nasadit webovou aplikaci, která bude usnadňovat a automatizovat vytváření a přehrávání prezentací. Úkolem této aplikace bude také správa uživatelských účtů a zdrojů.

Dále analyzovat a navrhnout úpravy v prostředí SAGE, aby tyto prezentace bylo možné zobrazovat na obrazovce v laboratoři.

Nakonec bude tato aplikace podrobena sadě testů, a to jednotkovým testům, uživatelským testům a akceptačním testům, na kterých se budou podílet skuteční uživatelé.

Analýza

V této kapitole se budu zabývat analýzou, která je součástí každého většího projektu. Jejím úkolem je analyzovat prostředí a současná řešení vytváření prezentací na SAGE. Zjistit požadavky uživatelů na novou aplikaci a vytvořit případy užití.

2.1 Laboratoř SAGElab

Laboratoř SAGElab (obrázek 2.1), jinak řečeno Síťová a multimediální laboratoř, byla založena ve spolupráci sdružení CESNET, Fakulty informačních technologií ČVUT a Fakulty elektrotechnické ČVUT[5].

Jedním z účelů laboratoře je podpora výuky studentů. Pořádání popularizačních přednášek organizovaných skupinou GRAFIT, mezi které patří například Grafické čtvrtky. A přímé zapojení do výuky. To se týká například předmětů BI-MGA, kde studenti prezentují své semestrální práce ve vysokém rozlišení, nebo BI-PGA, kde studenti vyvíjí přímo software pro SAGE a SAGE2, a mnohé další.

Dalším účelem laboratoře je výzkum a vývoj technologií přímo související s laboratoří SAGElab. Na tomto vývoji se podílejí především studenti v rámci svých bakalářských a magisterských prací, zaměstnanci CESNET a v neposlední řadě studenti v rámci týmových projektů. Často se pak jedná o vývoj aplikací pro konkrétní události či experimenty[6].

2.1.1 Technické vybavení laboratoře

Laboratoř je vybavena velkoplošným vizualizačním zařízením s vysokým rozlišením a vysokorychlostním připojením k počítačové síti[5].

Telestěna je tvořena dvaceti FullHD LCD displeji o rozlišení 1920x1080, s tenkým rámečkem, uspořádanými v mřížce 5x4. Celkové rozlišení stěny je tedy 9600 x 4320 px. Aby bylo možné dodávat monitorům dostatečně rychle obraz v patřičném rozlišení, je telestěna připojena k pěti serverům se 24jádrovými



Obrázek 2.1: Telestěna během přednášky Grafické čtvrtky, Zdroj:[1]

procesory pomocí optických linek s rychlostí 10Gb/s. Celá sestava je dále připojena optickou linkou do zařízení CESNET[6].

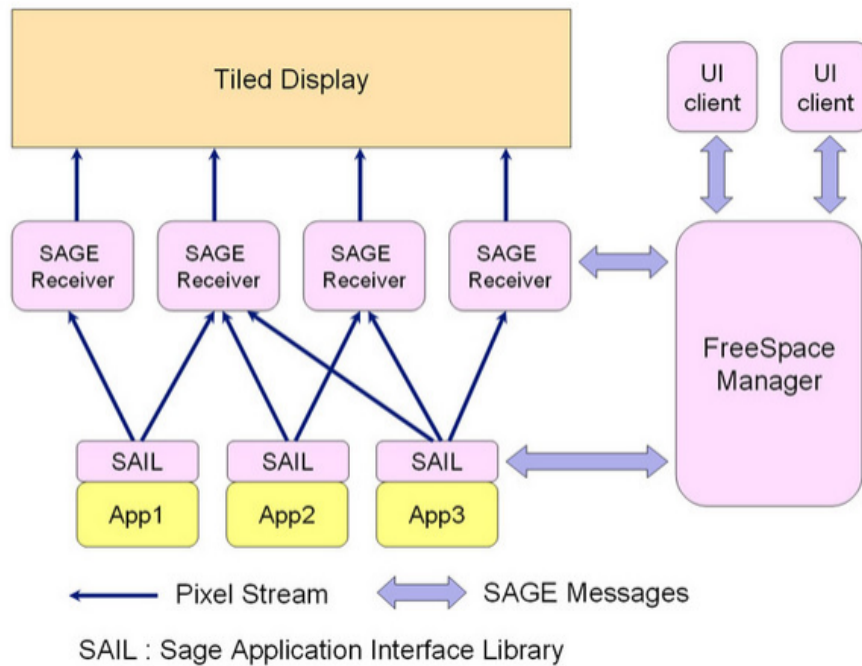
Tuto sestavu ovládní softwarové systémy SAGE a SAGE2. Technické vybavení bylo pořízeno z prostředků sdružení CESNET a Fakulty informačních technologií ČVUT[5].

2.2 SAGE

Scalable Adaptive Graphics Environment (SAGE)[3] je původní verze prostředí implementována v jazyce C++. Zobrazení základního uživatelského prostředí, jako je například zobrazení oken a reprezentace kurzoru, je vytvořeno v jazyce Python. Prostedí SAGE je spustitelné pouze na linuxových operačních systémech a vyžaduje znalost jejich správy. SAGE umožňuje zobrazování multimédií ve vysokém rozlišení, sdílení vlastní plochy na teletěnu a další[7].

2.3 Architektura SAGE

Následující segment, až po sekci 2.3.4. *User Interface klient* včetně, je převzat ze zdroje [8]. Tento zdroj jsem si dovilil citovat, neboť aplikace SAGE je zde velmi dobře popsána v českém jazyce. Autor čerpal z anglického textu [2].



Obrázek 2.2: Architektura SAGE, Zdroj:[2]

Architektonicky se SAGE [2] skládá z několika komponentů - Free Space Manager (FSM), Sage Application Interface Library (SAIL), SAGE Receiver a User Interface (UI) klient (Obrázek 2.1). UI klient posílá příkazy k řízení FSM, zpětně dostává zprávy s informacemi o aktuálním stavu SAGE. FSM přijímá uživatelské požadavky přicházející od UI klientů a řídí streamování pixelů mezi SAIL a SAGE Receivery (přijímače). SAIL odchyťává výstupní pixely z klientských aplikací a streamuje je, podle informací získaných od FSM, do příslušné SAGE přijímače. SAGE přijímače pak streamované pixely mapuje na stěnu dělených displejů. O renderování obrazů z přijatých nastreamovaných pixelů a jejich zobrazení na stěnu se stará rendering cluster, rozdělený do několika uzlů, z nichž každý spravuje určitou podmnožinu displejů stěny. Na každý uzel clusteru připadá právě jeden přijímač.

2.3.1 Free Space Manager

Free Space Manager (FSM) je komponenta v SAGE, která poskytuje intuitivní rozhraní pro přemísťování a měnění velikosti grafických prvků (zpravidla aplikační okna) na stěně dělených displejů. Když pohybem na stěně přechází

grafický prvek z jednoho displeje na druhý, FSM prostřednictvím SAIL informuje rendering cluster o nové destinaci pro streamování přelévajících pixelů. Díky tomu působí stěna dělených displejů jako jednolitě plátno, přestože jsou displeje řízeny separátními zdroji.

2.3.2 SAGE Application Interface Library

SAGE Application Interface Library (SAIL) je knihovna využívaná aplikacemi pro komunikaci s FSM a streamování pixelů do SAGE Receiverů (přijímačů). Každá aplikace, která se spustí pod SAGE, vytvoří a inicializuje pro každý uzel clusteru, na kterém aplikace běží, jeden SAIL objekt. Vytvořený SAIL objekt se prostřednictvím SAIL knihovny připojí na FSM a dostává od něj informace o tom, jak má mezi displeji rozčlenit obraz příslušné aplikace a kam (do jakého přijímače) má streamovat rozčleněné obrazy.

2.3.3 SAGE Receivers

SAGE Receivers (SAGE přijímače) získávají od SAIL objektů streamované pixely určené displejům napojeným na uzel, na kterém se příslušný přijímač nachází. Přijímače jsou schopné přijímat streamy pixelů od více SAIL objektů, z logiky věci je to nutná funkce. Přijaté pixely pak přijímače mapují tak, aby jejich výsledný obraz odpovídal rozlišení stěny.

2.3.4 User Interface klient

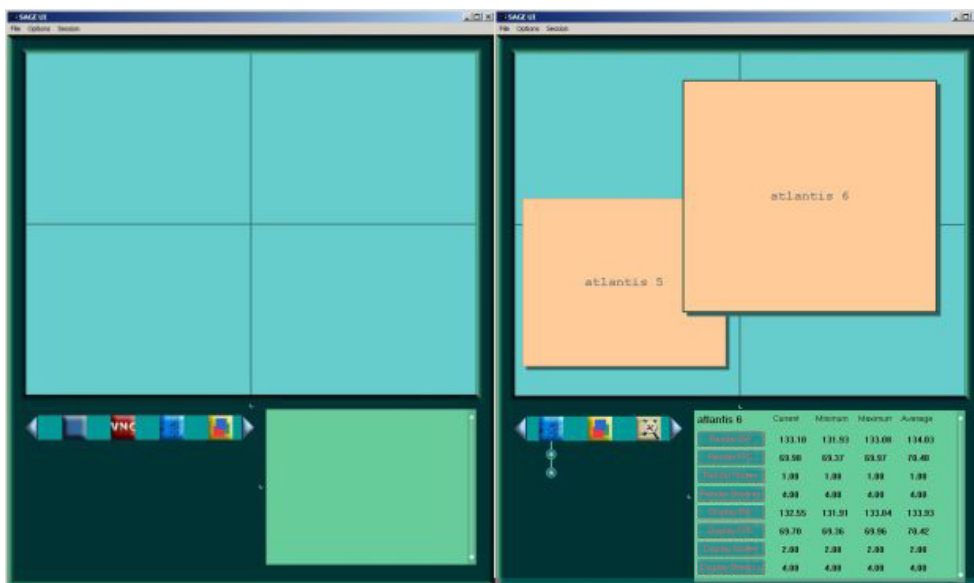
User Interface (UI) klienti posílají požadavky Free Space Manageru a zobrazují stav SAGE systému uživatelům (skladba displejů, rozložení aplikačních oken na stěně apod.). UI klient může být grafické uživatelské rozhraní nebo textově-orientovaná konzole. UI klient může běžet na jakémkoli zařízení, které je schopné navázat spojení s FSM prostřednictvím sítě. Více UI klientů se může napojit na FSM současně a navzájem interagovat. To umožňuje současné spojení více uživatelů s FSM, prostřednictvím jejich osobních počítačů či tabletů, a jejich vzájemná interakce při ovládání SAGE.

2.4 Současná řešení vytváření prezentací v SAGE

V současné době je možné ovládat SAGE buď pomocí grafického rozhraní, a nebo pomocí scriptů v jazyce Shell. Oba tyto způsoby mají své výhody a nevýhody, které budou rozebrány v následujícím segmentu.

2.4.1 Program SAGE User Interface

SAGE User Interface[3] (obrázek 2.3) je grafické rozhraní k ovládání prostředí SAGE. Tato aplikace znázorňuje rozložení oken různých aplikací na telestěně



Obrázek 2.3: Ukázka programu Sage UI. Zdroj:[3]

a umožňuje uživateli s nimi pohybovat, měnit jejich rozměry, nebo je i otevírat a zavírat. Toto prostředí dovolí uživateli "naklikat" si zvolené rozložení a konfiguraci uložit, ale všechna tato činnost se děje v reálném čase a není možné ji ve velké míře měnit uprostřed přednášky, kdy se uživatel potřebuje soustředit na vlastní obsah prezentace.

2.4.2 Pomocí příkazové řádky

Jednotlivé programy, používané pro zobrazování různých multimédií se dají spouštět i z příkazové řádky. To uživateli umožňuje vytvářet skripty, které mohou tyto programy zobrazovat a skrývat v čase, a tím si předem připravit dynamičtější prezentace, než kterých by docílel pouze s využitím SAGE User Interface. Tato činnost ale vyžaduje, aby uživatel znal základy scriptovacích jazyků a práci se soubory v příkazové řádce Shell (obrázek 2.4).

2.4.3 Program Image viewer

Image viewer je zobrazovač obrázků v prostředí sage. Jako vstup očekává jeden parametr a to adresu obrázku, který bude zobrazovat. Všechny potřebné údaje o připojení k obrazovce a výsledné pozici a velikosti obrázku získává Image viewer z konfiguračního souboru `image_viewer.conf`, který se musí nacházet ve stejném adresáři, ze kterého je program Image viewer spouštěn.

```
script.png /home/_sage_data/modely_tech/modely.sh:
            #!/bin/bash

            pushd /storage/.../modely\_tech
            for i in 01 02 03 04 05 06; do
                pushd ./\${i}
                smedia \${i}.png &
                popd
                sleep 1
            done

            read -p "Okey"
            popd
            pkill -9 smedia
```

Obrázek 2.4: Ukázka jednoduchého skriptu spouštějícího obrázky. Zdroj:[4]

2.4.4 Program SAGE media

SAGE media, dále jen Smedia, je program, který slouží k zobrazování obrázků na obrazovce v prostředí SAGE. Jeho autorem je Ing. Jiří Melnikov, správce laboratoře SAGElab. Jedná se o komplexnější rozšíření programu Image viewer. Smedia dovede kromě prostého zobrazování obrázků ještě navíc zobrazovat více obrázků ve smyčce s různými nastaveními pro časování. Od nejnovější verze Smedia podporuje i ovládání pomocí presenteru, který umožňuje zvolenou sadou obrázků libovolně listovat. Stejně jako Image viewer je Smedia závislé na konfiguračním souboru `smedia.conf`, který funguje stejně, jako v programu Image Viewer. Následuje ukázka konfiguračního souboru `smedia.conf`.

```
fsIP 192.168.122.1
fsPort 20002
masterIP 192.168.122.1
nwID 1
msgPort 23010
syncPort 13010
nodeNum 1
appID 0
pixelBlockSize 64 64
blockThreshold 0
winX 0
winY 0
winWidth 1920
winHeight 1080
streamType SAGE_BLOCK_HARD_SYNC
#streamType SAGE_BLOCK_NO_SYNC
nwProtocol TCP
asyncUpdate false
```

2.5 Funkční požadavky

V této sekci popíšeme požadavky na funkcionalitu aplikace. Tyto požadavky vycházejí z potřeb vedoucího práce a ostatních kolegů, které byly získány uživatelským průzkumem.

2.5.1 Uživatelská část

F1: Vytváření účtu pomocí registračního formuláře Uživatel si bude moci vytvořit účet v aplikaci přes formulář, který se zobrazí po stisknutí tlačítka „Registrovat.“

F2: Vytváření účtu přihlášením přes systém Shibboleth Uživateli bude automaticky vytvořen účet v aplikaci, pokud se přihlásí pomocí systému Shibboleth.

F3: Přihlášení pomocí přihlašovacího formuláře Uživatel se bude moci přihlásit ke svému účtu přes formulář, který se zobrazí po stisknutí tlačítka „Přihlásit pomocí jména a hesla.“

F4: Přihlášení přes systém Shibboleth Uživatel se bude moci přihlásit ke svému účtu přes systém Shibboleth po stisknutí tlačítka „Přihlásit pomocí účtu Shibboleth.“

F5: Udržení přihlášení Aplikace bude automaticky přihlašovat uživatele i po opuštění a následném návratu na web.

F6: Obnovení hesla V případě, že uživatel zapomene své heslo, bude mu nové heslo zasláno na email, který vyplnil při registraci. Popřípadě tento email může uživatel kdykoliv změnit.

F7: Odhlášení Uživatel se bude moci odhlásit stisknutím tlačítka „Odhlásit.“

F8: Editace vlastního účtu Uživatel bude moci změnit své heslo a telefonní číslo.

F9: Smazání vlastního účtu Uživatel bude moci zrušit svůj účet v aplikaci.

2.5.2 Práce s prezentacemi

F10: Vytvoření objektu prezentace Uživatel vytvoří prázdnou prezentaci připravenou k další práci.

2. ANALÝZA

F11: Smazání prezentace Uživatel bude moci odstranit prezentaci ze systému.

F12: Zobrazení detailu prezentace Uživatel si zobrazí detail prezentace, kde bude moci provádět další akce

F13: Vložení sekce Uživatel bude moci vkládat a následně časovat existující sekce do prezentace.

F14: Odebrání sekce Uživatel bude moci odebírat existující sekce z prezentace.

F15: Kontrola konzistence Systém zkontroluje, zda jsou všechny zdroje k dispozici, po kliknutí na tlačítko „Kontrola dostupnosti zdrojů“.

F16: Náhled prezentace Uživatel si bude moci spustit prezentaci přímo v prohlížeči

F17: Spuštění prezentace na SAGE Uživatel bude moci spustit prezentaci na obrazovce SAGE.

F18: Sdílení prezentací Uživatel bude moci sdílet své prezentace s jinými uživateli kliknutím na tlačítko „Sdílet.“

F19: Klonování prezentací Uživatel bude moci klonovat prezentace pomocí tlačítka „Klonovat.“

2.5.3 Práce se sekcemi

F20: Vytvoření objektu sekce Uživatel vytvoří prázdnou sekci připravenou k další práci.

F21: Smazání sekce Uživatel bude moci odstranit sekci ze systému.

F22: Zobrazení detailu sekce Uživatel si zobrazí detail sekce, kde bude moci provádět další akce

F23: Vložení obrázku Uživatel bude moci vkládat obrázky do sekce. Dále jim musí nastavit časování, souřadnice na obrazovce a požadovanou velikost.

F24: Odebrání obrázku Uživatel bude moci odebírat existující obrázky ze sekce.

2.5.4 Práce se obrázky

F25: Nahrání obrázku na web Uživatel bude moci nahrávat obrázky do svého úložiště na serveru. Odtud je bude moci dále používat pro tvorbu prezentací.

F26: Smazání obrázku Uživatel bude moci odstranit obrázek ze systému.

2.5.5 Administrační část

F27: Editace účtu Administrátor bude moci změnit libovolné parametry zvoleného uživatelského účtu.

F28: Smazání účtu Uživatel bude moci smazat kterýkoliv účet.

2.6 Nefunkční požadavky

V této sekci popíšeme technologické požadavky na aplikaci, které tvoří její funkcionalitu.

N1: Webová Nette aplikace Výstupem bude webová aplikace využívající framework Nette. Toto byl jeden z požadavků vedoucího práce. Framework Nette zvolil na základě snadné orientace v kódu a osobní zkušenosti s tímto frameworkem.

N2: Uživatelská přívětivost Systém bude intuitivní pro uživatele a nebude potřebovat návod k použití.

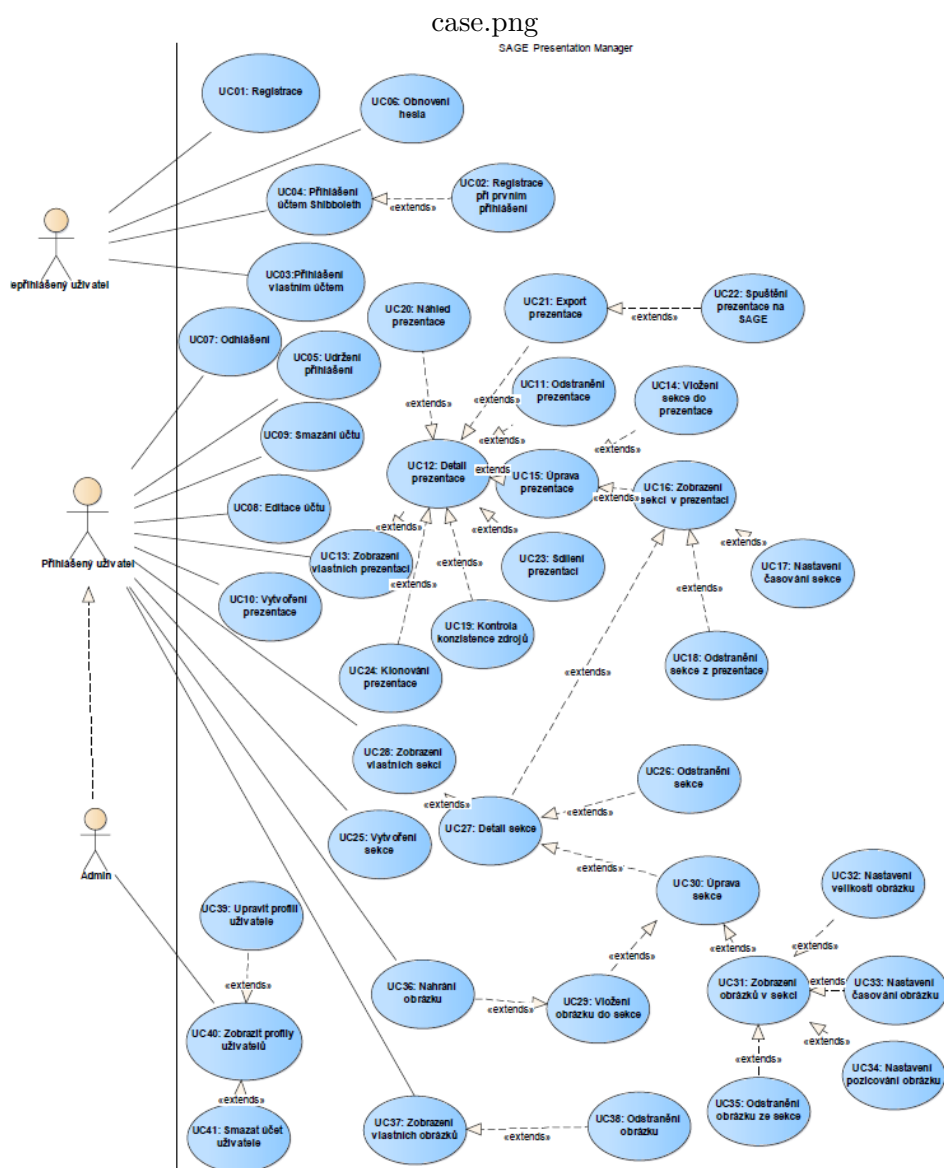
N3: Odolnost proti nestandardním situacím Systém bude spolehlivě ošetřovat případné chyby a nedostane se kvůli nim do nekonzistentního stavu.

N4: Jazyky rozhraní Systém bude s uživatelem komunikovat v českém jazyce.

N5: Rozšiřitelnost Systém bude napsán objektově a strukturovaně tak, aby byl snadno rozšiřitelný.

N6: Odolnost vůči webovým útokům Systém bude spolehlivě ošetřovat vstupy, aby nedocházelo k útokům jako například SQL injection, nebo cross-site request forgery.

2. ANALÝZA

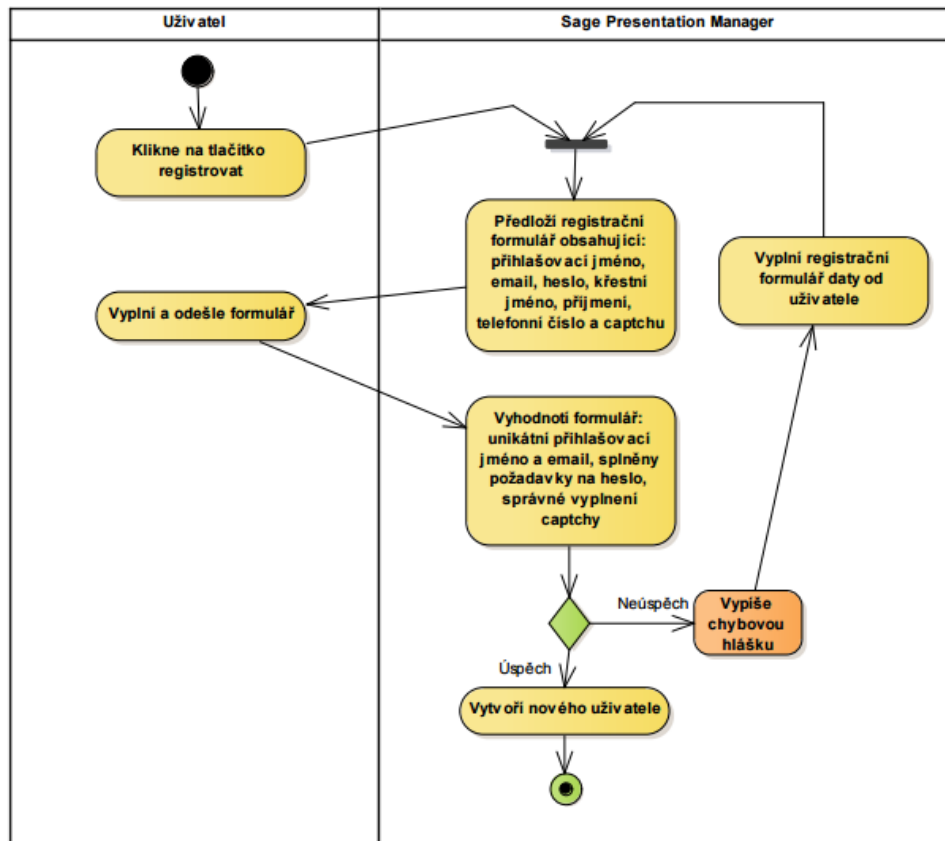


Obrázek 2.5: Model případů užití

2.7 Případy užití

S pomocí nasbíraných požadavků jsem vytvořil případy užití (obrázek 2.5), které popisují typické scénáře používání aplikace.

Všechny případy užití shrnuje následující obrázek. Protože výsledná dokumentace je velmi rozsáhlá, podrobně rozeberu pouze jeden významný a ostatní případy užití budou obsahovat pouze stručný popis.



Obrázek 2.6: Příklad užití registrace

UC1: Registrace Při kliknutí na tlačítko "registrovat" bude uživatel proveden procesem vytvoření nového účtu v systému. Tento případ užití jsem se rozhodl popsat detailně

Registrace (obrázek 2.6) začíná tím, že nepřihlášený uživatel klikne na tlačítko registrovat. Aplikace následně předloží uživateli prázdný registrační formulář, kde uživatel vyplní své přihlašovací jméno, heslo, email, křestní jméno, příjmení a jednoduchou captchu. Zároveň bude moci vyplnit své telefonní číslo, jako nepovinný údaj a formulář odešle. Aplikace formulář zpracuje a vyhodnotí, zda nejsou přihlašovací jméno a email používány již existujícím účtem, zda byla captcha vyplněna správně a jestli zadané heslo splňuje minimální požadavky na heslo. Pokud tato kontrola proběhne bez problémů, aplikace vytvoří uživateli nový účet a přesměruje ho na přihlašovací formulář. Pokud tato kontrola dopadne neúspěšně, aplikace sdělí uživateli, jakých chyb se dopustil a předloží mu registrační formulář k přepracování.

UC2: Registrace při prvním přihlášení Tento případ nastane pouze u uživatelů, kteří se budou přihlašovat pomocí školního systému Shibboleth. Těmto uživatelům bude automaticky vytvořen účet v aplikaci při prvním přihlášení na základě dat poskytnutých systémem Shibboleth.

UC3: Přihlášení vlastním účtem Aplikace předloží uživateli přihlašovací formulář, kde bude třeba vyplnit platné přihlašovací jméno a heslo. Při úspěšném přihlášení bude mít uživatel přístup ke svému uložení a bude moci pokračovat v další činnosti v aplikaci. Při neúspěšném přihlášení bude vypsána chybová hláška a aplikace předloží uživateli přihlašovací formulář k přepracování.

UC4: Přihlášení účtem Shibboleth Aplikace přesměruje uživatele na přihlašovací formulář k systému Shibboleth. Při úspěšném přihlášení bude mít uživatel přístup ke svému uložení a bude moci pokračovat v další činnosti v aplikaci.

UC5: Udržení přihlášení Aplikace bude automaticky přihlašovat uživatele i po opuštění a následném návratu na web. Tento proces bude probíhat na základě obsahu souborů cookie.

UC6: Obnovení hesla V případě, že uživatel zapomene své heslo, bude moci kliknout na tlačítko "obnovení hesla". Následně mu aplikace změní heslo na nové náhodně vygenerované heslo a to bude uživateli zasláno na email, který vyplnil při registraci.

UC7: Odhlášení Po kliknutí na tlačítko "Odhlásit" bude uživatel odhlášen z aplikace a bude mít přístup pouze k funkcionalitám nepřihlášených uživatelů.

UC8: Editace účtu Uživatel bude mít možnost měnit některé údaje svého účtu. Uživatelé přihlášení přes systém Shibboleth si budou moci měnit pouze telefonní číslo. Uživatelé s vlastním účtem si budou moci navíc změnit ještě heslo.

UC9: Smazání účtu Po kliknutí na "Smazání účtu" bude uživateli smazán celý účet včetně obsahu jeho uložení na serveru. Tato akce je nevratná a její potvrzení bude vyžadovat zadání hesla.

UC10: Vytvoření prezentace Uživatel bude moci vytvořit prázdnou prezentaci. Tato akce bude vyžadovat pouze zadání názvu prezentace. Další práce s prezentací budou realizovány pomocí okna detailu.

UC11: Odstranění prezentace Tato akce smaže prezentaci ze systému. Její původní sekce a obrázky však zůstanou uchovány pro další využití.

UC12: Detail prezentace Detail prezentace umožní uživateli provádět další operace s prezentací jako celkem. Tyto operace jsou popsány v dalších případech užití. Jedná se o:

- Spouštět prezentace na SAGE
- Přehrávat prezentace v prohlížeči
- Export scriptu pro Sage Viewer
- Provádění kontroly konzistence zdrojů
- Úpravu prezentace
- Smazání prezentace
- Sdílení prezentace
- Klonování prezentace

Dále detail umožňuje zobrazovat, filtrovat, vkládat a upravovat sekce.

UC13: Zobrazení vlastních prezentací Uživatel bude mít možnost si zobrazit vlastní prezentace. Pomocí tohoto seznamu bude přistupovat k detailům prezentací.

UC14: Vložení sekce do prezentace Uživatel bude moci vložit již existující sekci do prezentace. To bude dělat přímo v detailu prezentace.

UC15: Úprava prezentace Uživateli bude předložen formulář, kde bude možné změnit některé z parametrů prezentace. V současné době se jedná jen o přejmenování, ale bylo myšleno na případné budoucí rozšíření. Úprava prezentace se vztahuje pouze na prezentaci jako celek. Vkládání a odebírání sekcí spadá pod vlastní případy užití.

UC16: Zobrazení sekcí v prezentaci Jako součást detailu prezentace bude výpis všech sekcí, ze kterých se daná prezentace skládá, a jak následují chronologicky po sobě. Z tohoto výpisu bude dále možno přejít na detaily jednotlivých sekcí.

UC17: Nastavení časování sekce Uživatel bude moci nastavit, v jakém časovém okamžiku od startu prezentace se zvolená sekce spustí. Tento údaj bude zadán buď pouze v sekundách, nebo v hodinách, minutách a sekundách.

UC18: Odstranění sekce z prezentace Tato akce odstraní sekci z prezentace. Její původní obsah však zůstane uchován pro další využití.

UC19: Kontrola konzistence zdrojů Jedná se kontrolu, která ověří dostupnost všech obrázků a sekcí v uložení aplikace. Tato operace vypíše pouze hlášku o výsledku kontroly. Je čistě informativní.

UC20: Náhled prezentace Jedná se o případ, kde se uživatel rozhodne spustit prezentaci v prohlížeči. Tato akce vyžaduje vyexportování scriptu pro Sage Viewer.

UC21: Export prezentace Data uložená v databázi budou exportovány do textového souboru, který bude sloužit jako vstup pro program Sage Viewer, který pak zobrazí zvolenou prezentaci v prohlížeči, nebo na obrazovce SAGE.

UC22: Spuštění prezentace na SAGE Jedná se o případ, kde se uživatel rozhodne spustit prezentaci v učebně SAGELab. Tato akce vyžaduje vyexportování scriptu pro Sage Viewer.

UC23: Sdílení prezentací Uživatel bude moci sdílet libovolnou ze svých prezentací s jiným uživatelem. Při tomto procesu se překopírují všechny zdroje zvolené prezentace do úložiště cílového uživatele.

UC24: Klonování prezentací Uživatel bude moci klonovat libovolnou svou prezentaci. Systém vytvoří kopii prezentace, včetně kopií všech sekcí, které jsou v prezentaci používány. Obrázky z jednotlivých sekcí kopírovány nebudou

UC25: Vytvoření sekce Uživatel bude moci vytvořit prázdnou sekci. Tato akce bude vyžadovat pouze zadání názvu sekce. Další práce se sekcí bude realizována pomocí okna detailu.

UC26: Odstranění sekce Tato akce smaže sekci ze systému. Její původní obrázky však zůstanou uchovány pro další využití.

UC27: Detail sekce Detail sekce umožní uživateli provádět další operace se sekcí jako celkem. Tyto operace jsou popsány v dalších případech užití. Jedná se o úpravu sekce a smazání sekce. Dále detail umožňuje zobrazovat, filtrovat, vkládat a odstraňovat obrázky.

UC28: Zobrazení vlastních sekcí Uživatel bude mít možnost si zobrazit vlastní sekce. Pomocí tohoto seznamu bude přistupovat k detailům sekcí.

UC29: Vložení obrázku do sekce Uživatel bude moci vložit již existující obrázek do sekce. To bude dělat přímo v detailu sekce.

UC30: Úprava sekce Uživateli bude předložen formulář, kde bude možné změnit některé z parametrů sekce. V současné době se jedná jen o přejmenování, ale bylo myšleno na případné budoucí rozšíření. Úprava sekce se vztahuje pouze na sekci jako celek. Vkládání a odebírání obrázků spadá pod vlastní případy užití.

UC31: Zobrazení obrázků v sekci Jako součást detailu sekce bude výpis všech obrázků, ze kterých se daná sekce skládá, společně s údaji o pozicování a časování. Z tohoto výpisu bude dále možno upravovat parametry jednotlivých obrázků v sekci.

UC32: Nastavení velikosti obrázku Při vložení obrázku do sekce bude třeba nastavit výšku a šířku, se kterými se bude obrázek zobrazovat. Nejedná se o vlastní rozměry obrázku, ale doporučuje se vlastní rozměry obrázku zachovávat, aby se předešlo deformaci, nebo ztrátě kvality. Tento údaj bude zadáván v pixelech.

UC33: Nastavení časování obrázku Při vložení obrázku do sekce bude třeba nastavit údaje začátek a konec zobrazení. Jedná se o uplynulou dobu od startu mateřské sekce. Tyto údaje budou zadány buď pouze v sekundách, nebo v hodinách, minutách a sekundách.

UC34: Nastavení pozicování obrázku Při vložení obrázku do sekce bude třeba nastavit souřadnice levého spodního rohu obrázku. Tento údaj bude zadáván v pixelech.

UC35: Odstranění obrázku ze sekce Tato akce odstraní obrázek ze sekce. Vlastní obrázek však zůstane uchován v uložišti uživatele.

UC36: Nahrání obrázku Uživatel bude nahrávat obrázky do svého uložště v aplikaci. Následně k těmto obrázkům bude mít přístup z výpisu vlastních obrázků.

UC37: Zobrazení vlastních obrázků Uživatel bude mít možnost si zobrazit vlastní obrázky. Pomocí tohoto seznamu bude moci zvolené obrázky mazat a přidávat.

UC38: Odstranění obrázku Tato akce smaže obrázek ze systému.

2. ANALÝZA

UC39: Upravit profil uživatele Administrátor bude moci upravovat všechny parametry libovolného uživatele. Mezi tyto parametry patří:

- přihlašovací jméno
- heslo
- křestní jméno
- příjmení
- email
- telefonní číslo
- uživatelskou roli
- zda se jedná o účet Shibboleth

UC40: Zobrazit profily uživatelů Administrátor si bude moci zobrazit výpis všech uživatelských profilů. Z tohoto výpisu bude mít přístup k úpravě a mazání uživatelských účtů.

UC41: Smazat účet uživatele Administrátor bude moci smazat libovolný uživatelský účet. Kliknutím na "Smazat účet" bude uživateli smazán celý účet včetně obsahu jeho úložiště na serveru. Tato akce je nevratná a její potvrzení bude vyžadovat zadání hesla.

2.8 Mapování funkčních požadavků na případy užití

Následující tabulky popisují, jak funkční požadavky odpovídají konkrétním případům užití. Tento segment je rozdělen na dvě části, protože celá tabulka by se na stránku nevešla.

2.8. Mapování funkčních požadavků na případy užití

2.8.1 Část 1

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
UC01	x													
UC02		x												
UC03			x											
UC04				x										
UC05					x									
UC06						x								
UC07							x							
UC08								x						
UC09									x					
UC10										x				
UC11											x			
UC12												x		
UC13												x		
UC14													x	
UC15													x	
UC16													x	
UC17													x	
UC18														x

2. ANALÝZA

2.8.2 Část 2

	F15	F16	F16	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28
UC19	x													
UC20		x												
UC21			x											
UC22			x											
UC23				x										
UC24					x									
UC25						x								
UC26							x							
UC27								x						
UC28								x						
UC29									x					
UC30									x					
UC31									x					
UC32									x					
UC33									x					
UC34									x					
UC35										x				
UC36											x			
UC37												x		
UC38												x		
UC39													x	
UC40													x	
UC41														x

Návrh

3.1 Procesy v aplikaci

Ze všech procesů, které budou v aplikaci probíhat, jsem si zvolil dva nejvýznamější. A to vytváření sekcí a spouštění prezentací. Tyto procesy jsem se rozhodl podrobně rozebrat.

3.1.1 Vytvoření sekce

Proces vytvoření sekce (obrázek 3.1) začíná tím, že uživatel vytvoří prázdný objekt sekce. S takovou sekcí již bude moci pracovat pomocí editační stránky.

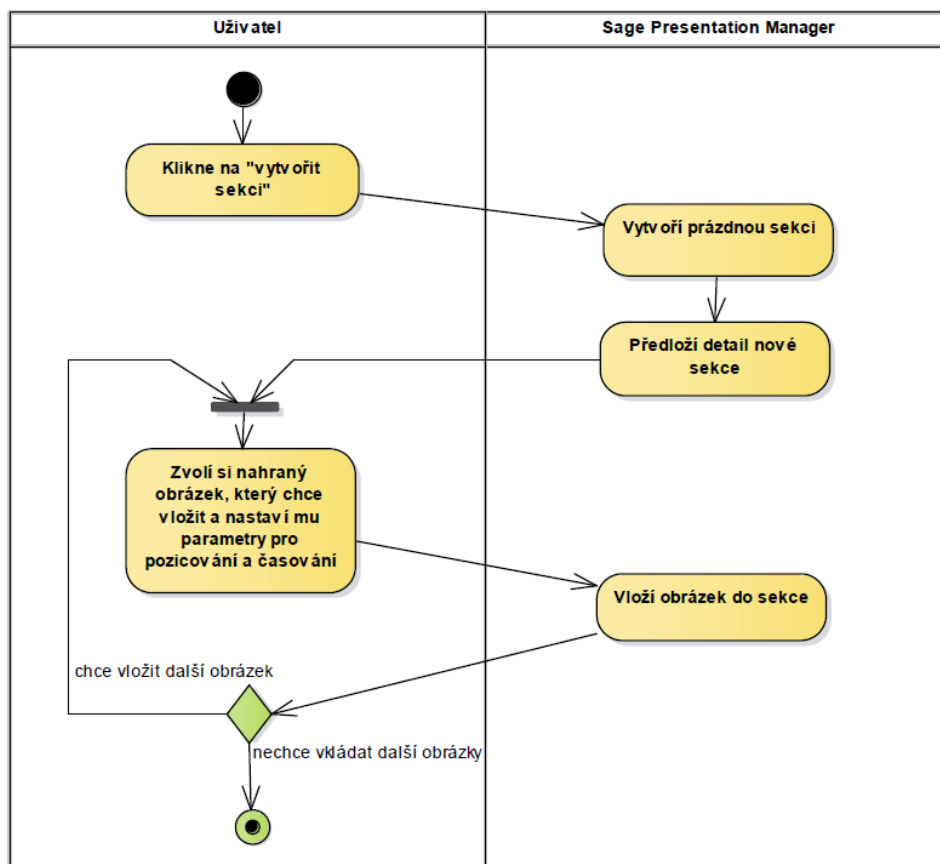
Uživatel bude vkládat obrázky, které má ve svém uložišti, a bude jim nastavovat parametry pro pozicování a časování v rámci sekce. Proces končí, když se uživatel rozhodne, že už další obrázky přidávat nechce.

Z takto vytvořených sekcí uživatel následně poskládá prezentaci, kterou bude moci spustit.

3.1.2 Spuštění prezentace

Spouštění prezentací (obrázek 3.2) se skládá z několika kroků. Nejprve je třeba přejít na detail prezentace. Další, dobrovolný, krok je provedení kontroly konzistence zdrojů. Tato akce je pouze informativní pro uživatele, ale bez jejího úspěchu nelze zaručit, že se v prezentaci zobrazí všechny snímky. Následně se vygeneruje script pro program Sage Viewer. Tento script lze spouštět jak na obrazovce SAGE, tak přímo v prohlížeči. Spuštěním prezentace proces končí.

Připomínám, že pokud uživatel spouští již hotovou prezentaci opakovaně, není třeba pokaždé generovat nový script. To je potřeba pouze, pokud uživatel provede změny v prezentaci. Naopak provádění kontroly konzistence zdrojů se doporučuje před každým spuštěním.



Obrázek 3.1: Diagram vytvoření sekce

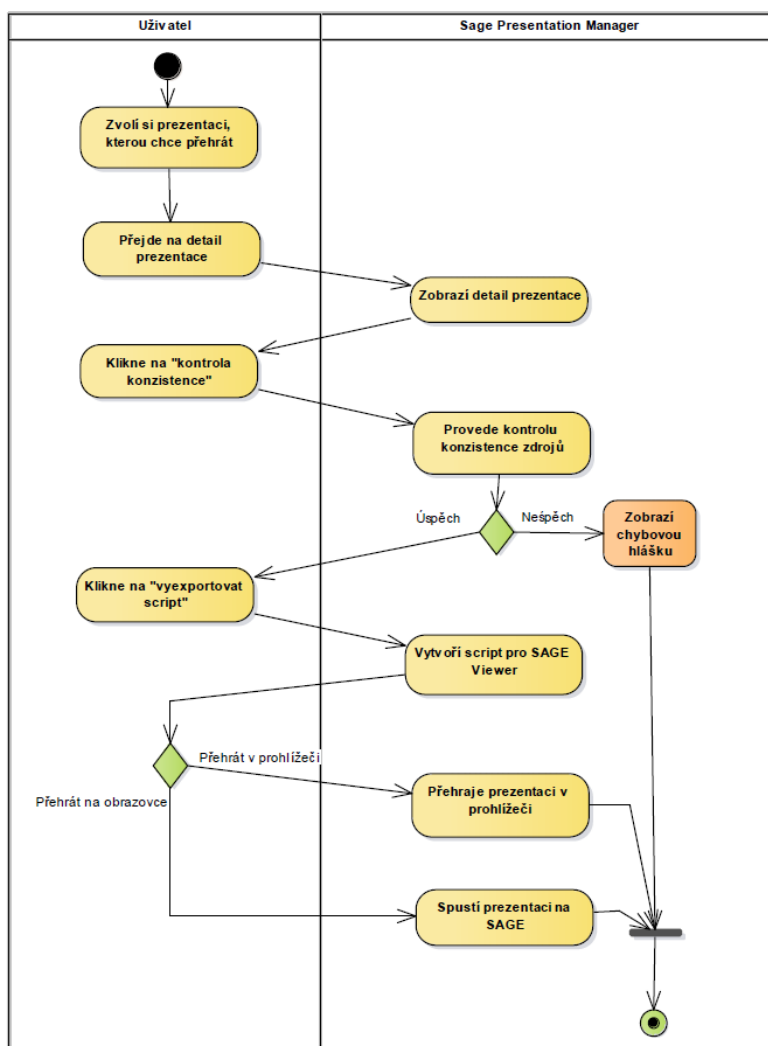
3.2 Doménový model tříd

Navrhl jsem doménový model tříd (obrázek 3.3), od kterého se bude odvíjet výsledná databáze, kterou bude aplikace využívat pro ukládání dat.

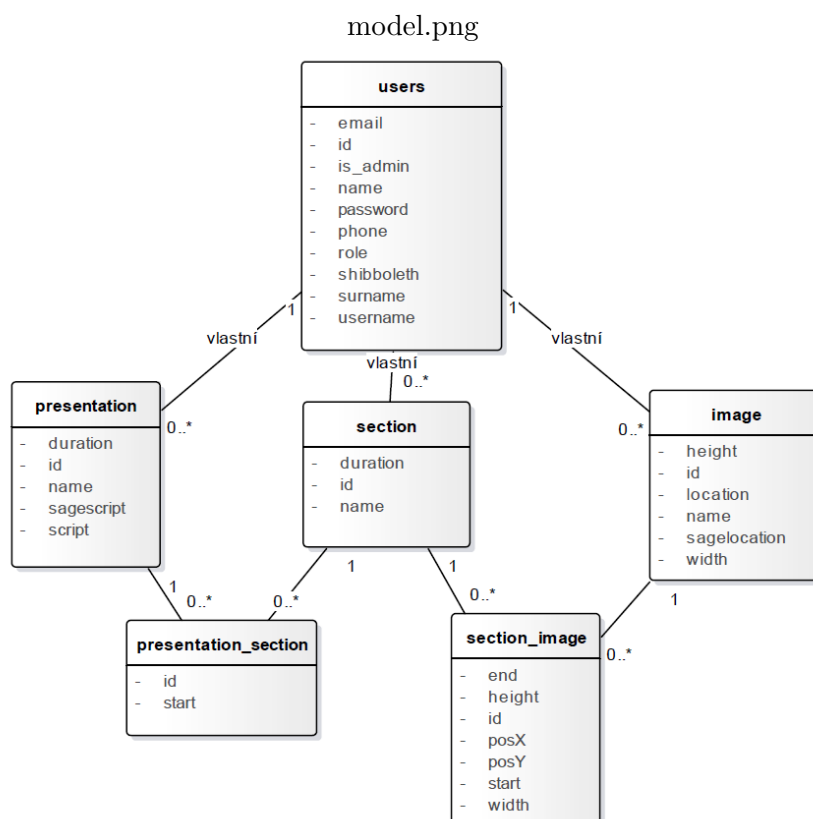
Tento model dovede nezávisle uchovávat informace o prezentacích, sekcích a obrázcích. To nám dovoluje pracovat s jednotlivými částmi prezentací nezávisle na zbytku a umožňuje to například snadné sdílení dat mezi uživateli.

3.3 Databázový model

Na základě analýzy na požadavky aplikace a následně vytvořeného doménového modelu jsem navrhl následující databázový model (obrázek 3.4). Tento model z doménového modelu vychází a je rozšířen o primární a cizí klíče a konstrukce systému PostgreSQL. Model splňuje všechny požadavky na apli-



Obrázek 3.2: Diagram spuštění prezentace



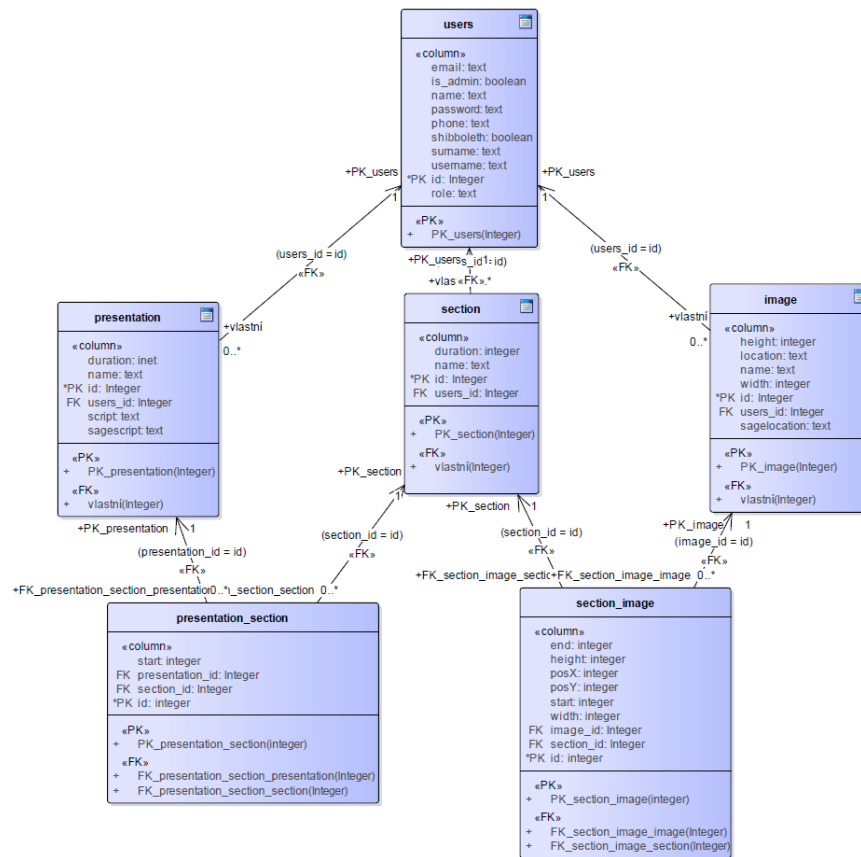
Obrázek 3.3: Doménový model

kaci a je navržen tak, aby bylo možné ho v budoucnu snadno pozměnit, nebo rozšířit.

3.4 Uživatelské rozhraní

Uživatelské rozhraní bylo navrženo tak, aby bylo jednoduché a co nejvíce intuitivní. Na ukázkou jsem připravil sadu wireframů, které by měly vzhled aplikace demonstrovat. Tyto wireframy nepopisují všechny části aplikace. Vybral jsem je tak, aby typově popisovaly celou aplikaci a lišily se od nepopsaných stránek pouze obsahem dat.

V horní části stránky najdeme hlavičku s logem aplikace a přihlašovací částí. Nalevo se nachází jednoduché menu. Tyto části jsou pro všechny stránky stejné. Vlastní obsah stránky se u jednotlivých částí liší.



Obrázek 3.4: Databázový model

3.4.1 Přehled prezentací

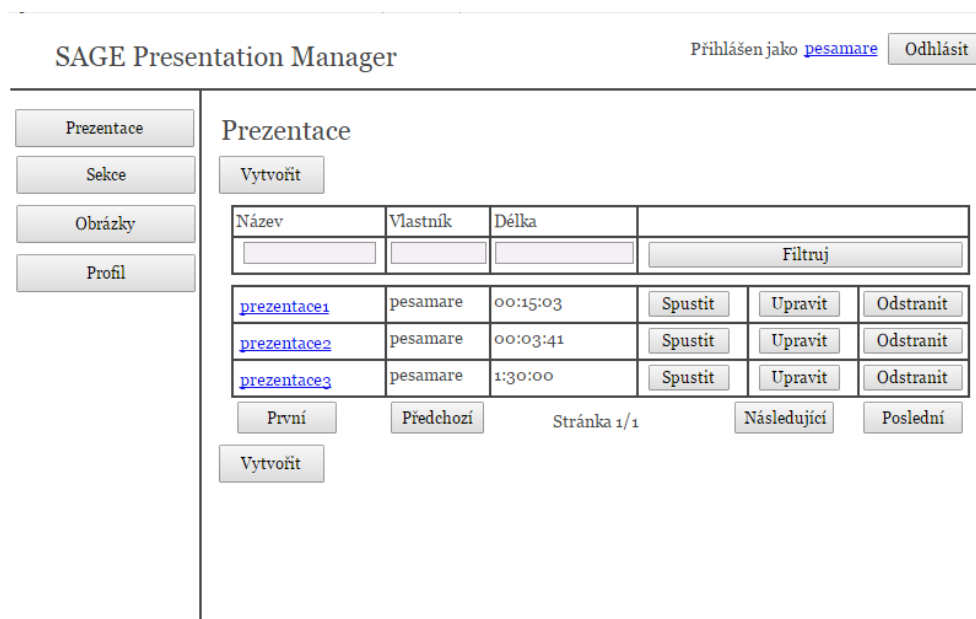
Část přehled prezentací (obrázek 3.5) nám předloží jednoduchý výpis prezentací, možnost vytváření a přechodu na detail prezentace. Dále můžeme výpis filtrovat a nabízí nám i několik praktických tlačítek pro práci s prezentacemi, které můžeme provádět bez přechodu na detail.

3.4.2 Detail prezentace

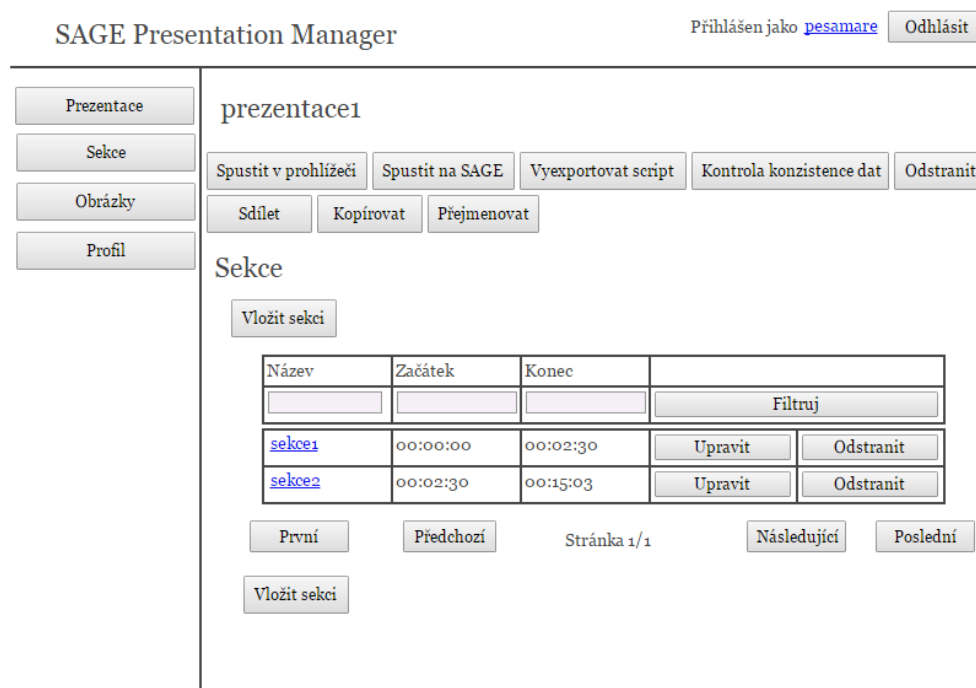
V detailu prezentace najdeme všechny akce, které můžeme se zvolenou prezentací, jako celkem, provádět.

Dále zde najdeme seznam sekcí (obrázek 3.6), ze kterých se daná prezentace skládá. Tyto sekce opět můžeme filtrovat, vkládat, odebírat a upravovat.

3. NÁVRH



Obrázek 3.5: Wireframe přehledu prezentací



Obrázek 3.6: Wirefram edetailu prezentace

SAGE Presentation Manager Přihlášen jako [pesamare](#)

sekce1

Obrázky

Název	Začátek	Konec	Výška	Šířka	Souřadnice X	Souřadnice Y		
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Filtruj"/>	
obrázek1	00:00:00	00:02:30	1920	1080	1920	3240	<input type="button" value="Upravit"/>	<input type="button" value="Odstranit"/>
obrázek2	00:02:30	00:15:03	1920	1080	5760	1080	<input type="button" value="Upravit"/>	<input type="button" value="Odstranit"/>
obrázek3	00:00:00	00:02:30	1920	1080	1920	2160	<input type="button" value="Upravit"/>	<input type="button" value="Odstranit"/>
obrázek4	00:02:30	00:15:03	1920	1080	3840	2160	<input type="button" value="Upravit"/>	<input type="button" value="Odstranit"/>
obrázek5	00:02:30	00:15:03	1920	1080	1920	1080	<input type="button" value="Upravit"/>	<input type="button" value="Odstranit"/>

Stránka 1/1

Obrázek 3.7: Wireframe detailu sekce

3.4.3 Detail sekce

Detail sekce (obrázek 3.7) se v mnohém podobá detailu prezentace, ale rozhodl jsem se ho tu uvést zvlášť, neboť zde bude probíhat hlavní tvůrčí činnost. Do sekce bude uživatel vkládat obrázky a nastavovat jim jejich časování a pozicování. Toto je nejnižší vrstva aplikace, kde se bude uživatel pohybovat.

Realizace

Tato práce se skládá ze dvou různých programů, které společně tvoří aplikaci Správce prezentací pro SAGE. Konkrétně se jedná o program Sage Viewer, jehož úkolem je zobrazovat vytvořené prezentace na stěně SAGE. A webové aplikace, jejíž úkolem je spravovat účty uživatelů a umožnit jednoduché vytváření prezentací. Obě tyto části budou podrobně rozebrány v následujících sekcích.

4.1 Program Sage Viewer

Tento program slouží k zobrazování prezentací na telestěně nacházející se v laboratoři SAGElab. Jedná se o krátký program napsaný v jazyce C++11. V prostředí SAGE je spouštěn z příkazové řádky s jediným argumentem, tím bude adresa vstupního souboru, který reprezentuje prezentaci k přehrání.

4.1.1 Vstup programu

Vstupem tohoto programu je textový soubor, který tvoří celou prezentaci. Každý řádek je reprezentací jednoho zobrazovaného obrázku a musí dodržovat následující formát:

```
adresa_obrazku souřadnice_X souřadnice_Y šířka výška start trvání.
```

Adresa obrázku je absolutní adresa, na které se daný obrázek nachází v uložišti, všechny ostatní údaje jsou číselné a oddělené mezerou. Souřadnice X a Y jsou souřadnice levého spodního rohu obrázku v pixelech. Výška a šířka jsou rozměry, na které se bude obrázek promítat. A údaje start a trvání jsou časové údaje, kdy se bude obrázek zobrazovat během prezentace, v sekundách. Konkrétní řádek v prezentaci pak může vypadat takto:

```
/users/pesamare/data/img/obr1.jpg 0 0 1920 1080 0 30.
```

4.1.2 Výstup programu

Výstup programu je sada ovládacích příkazů v jazyce Shell. Jejichž úkolem je vytvářet adresáře a konfigurační soubory (více viz. sekce Pozicování obrázků) a spouštět program Smedia (více viz. sekce Zobrazování obrázků).

4.1.3 Časování obrázků

Časování obrázků je řešeno tak, že si program napřed seřadí obrázky vzestupně podle jejich údaje start a předá si je do zásobníku. Program si dále pamatuje počet sekund, kolik už běží. V cyklu, který se opakuje každou sekundu, potom zobrazuje všechny obrázky z vrcholu zásobníku, u kterých se shoduje údaj start s počtem uplynulých sekund. Program skončí, jakmile se zásobník vyprázdní.

4.1.4 Pozicování obrázků

Program využívá pro zobrazování obrázku zobrazovač Smedia, od něhož také přebírá způsob, jakým pozicuje obrázky. Pozicování obrázků se děje pomocí konfiguračních souborů, které jsou ve stejném adresáři, ze kterého je program Smedia spouštěn. Proto program Sage Viewer vytváří adresáře společně s konfiguračními soubory na základě údajů poskytnutých vstupním souborem. Následně do těchto adresářů přechází a odtud spouští program Smedia.

4.1.5 Zobrazování obrázků

Validní zobrazování obrázků byl jedním z největších problémů této aplikace. Aplikace využívá již hotový program Smedia. Tento program ale nebyl navržen pro účely této práce a vyskytly se problémy s validním ukončováním. Program Smedia zobrazuje obrázek ve smyčce, až dokud není ukončen vstupem z klávesnice. To však značně ztěžuje paralelizaci a automatizaci. Předchozí řešení tohoto problému byly založeny na programu `pkill`. To ale pořád nebylo dostačující a navíc to není programátorsky správně.

Další alternativou bylo používání programu Image Viewer. Ten ale způsoboval fatální chybu, kdykoliv byl spuštěn paralelně na pozadí, takže jsem ho okamžitě zamítl.

Nakonec jsem se rozhodl pozměnit program Smedia, aby vyhovoval potřebám této práce. Odstranil jsem některé z funkcionalit, jako jsou zobrazování více obrázků ve smyčce a přidal jsem časovač, po jehož uplynutí se program automaticky ukončí. Tento nový program Smedia je využíván pouze aplikací SAGE Presentation Manager a není součástí instalace SAGE.

Detail prezentace prezentace1

Zpět

Spustit na SAGE Spustit v prohlížeči Vyexportovat sript Sdílet Kopírovat Kontrola konzistence zdrojů Upravit Smazat

Sekce

Vložit sekci

	Sekce	Začátek	Konec	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Filtruj"/>
sekce1	0	10		<input type="button" value="Uprav"/> <input type="button" value="Vymaž"/>
sekce2	5	35		<input type="button" value="Uprav"/> <input type="button" value="Vymaž"/>

« First « Previous 1 / 1 Next » Last »

Obrázek 4.1: Ukázka skládání prezentace 1.

4.2 Webová aplikace pro správu uživatelů, prezentací a dat

Tato webová část je naprogramována v jazyce PHP s pomocí frameworku Nette. Jako Nette aplikace dodržuje architekturu Model View Presenter (MVP) a využívá návrhový vzor dependency injection. To je společné pro většinu aplikací v Nette, proto tyto části nebudu již dále rozebírat. Místo toho se zaměřím na ty části aplikace, které jsou specifické pro tuto práci.

4.2.1 Skládání prezentací

Skládání prezentací (obrázky 4.1 a 4.2) je řešeno pomocí sady výpisů a formulářů. Uživatelé budou samostatně vytvářet prezentace, sekce a nahrávat obrázky. Poté po přechodu na detail budou moci vkládat jednotlivé obrázky do sekcí a sekce do prezentací. Formuláře jsou řešeny pomocí továrniček (factories) a o výpisy se stará Nextras Datagrid.

4.2.2 Generování scriptů

Stisknutím tlačítka **Generovat script** v detailu prezentace se spustí funkce pro generování scriptů. Tato funkce prochází veškeré sekce ve zvolené prezentaci. Ze všech těchto sekcí získá data o jednotlivých obrázcích a začne je zapisovat do dvou různých textových souborů. Jeden tento soubor bude sloužit pro zobrazování prezentací v prohlížeči. Jeho název bude složen z ID a názvu prezentace. Druhý soubor bude sloužit jako vstupní soubor pro program Sage Viewer v prostředí SAGE. Jeho název bude stejný, jako u předchozího souboru, ale bude mít suffix "(sage)".

Kromě samotného generování scriptů bude tato funkce i updatovat údaje o délce prezentace, která se do té doby bude zobrazovat nulová.

Detail sekce sekce1

[Zpět](#)
[Upravit](#) [Smazat](#)

Obrázky

[Vložit obrázek](#)

Obrázek	Šířka	Výška	Pozice X	Pozice Y	Začátek	Konec	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	Filtruj
CMS_Creative_164657191_Kingfisher.jpg	3840	2160	3840	2160	0	10	Uprav Vymaz
desktop-year-of-the-tiger-images-wallpaper.jpg	3840	2160	0	0	0	5	Uprav Vymaz
littlepony.jpg	1000	1000	3000	0	0	10	Uprav Vymaz

[« First](#)
[« Previous](#)
[1 / 1](#)
[Next »](#)
[Last »](#)

Obrázek 4.2: Ukázka skládání prezentace 2.

4.2.3 Vzdálené spouštění programu Sage Viewer

Aplikace ovládá SAGE pomocí vzdáleného připojení pomocí programu a zároveň protokolu SSH (Secure Shell). Pokud chce uživatel zobrazit prezentaci na obrazovce SAGE, aplikace nahraje na server SAGE všechny potřebné zdroje a script, pomocí zmíněného protokolu. Následně prezentaci spustí. Je důležité zmínit, že tento proces by měl uživatel spouštět pouze, pokud je přítomen v laboratoři SAGElab, aby tím omylem nenarušoval chod laboratoře.

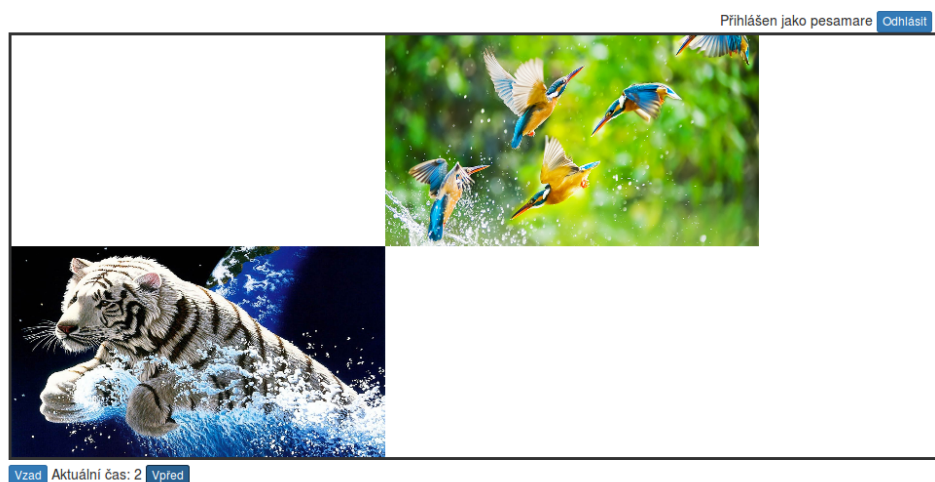
4.2.4 Spouštění prezentace v prohlížeči

Přestože tento program má stejnou funkci, jako má program Sage Viewer na serveru SAGE, jeho kód je značně odlišný. Je to způsobeno tím, že zobrazovací funkce pro HTML5 je fundamentálně odlišná od programu Smedia, který využívá program Sage Viewer.

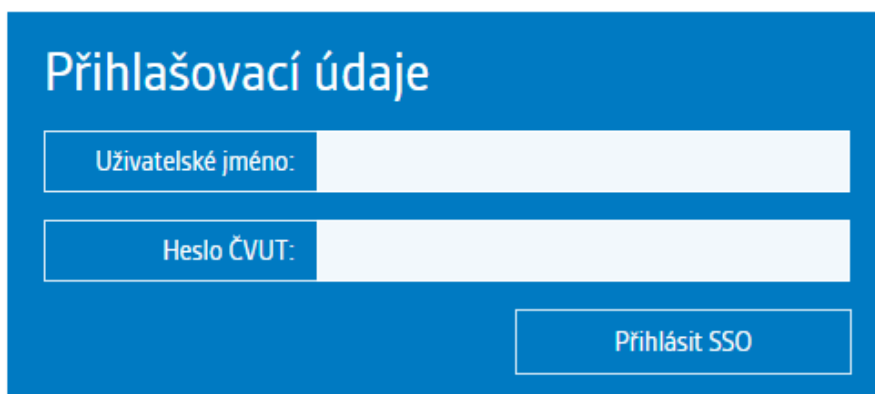
Sage Viewer na základě aktuálního času spouští program Smedia a o zobrazené obrázky se již dále nestará. Funkce pro zobrazování prezentací v prohlížeči si musí pro každý časový úsek (1 sekundu) vyhledat v prezentaci obrázky, které mají být právě zobrazeny, a vykreslit je. O samotné vykreslování se stará HTML5 tag canvas (obrázek 4.3). Tato zobrazovací funkce je napsaná v jazyce Javascript, zatímco Sage Viewer je napsán v jazyce C++11.

4.2.5 Integrace do systému Shibboleth

Shibboleth (obrázek 4.4) je technologie umožňující SSO (single sign-on). Jedná se o způsob přihlášení do informačního systému ČVUT. Po přihlášení v systému Shibboleth mají zaměstnanci a studenti přístup do vybraných subsys-



Obrázek 4.3: Ukázka spouštění prezentace v prohlížeči.



Obrázek 4.4: Ukázka přihlášení Shibboleth.

témů ČVUT [9]. Pro úspěšnou integraci aplikace do systému Shibboleth je potřeba vykonat následující kroky.

4.2.5.1 Authentiátor v aplikaci

Jednoduchá třída authenticator, která získá od systému Shibboleth přihlašovací jméno. Pokud je toto jméno platné jméno některého z uživatelů, byl uživatel úspěšně přihlášen. Pokud se uživateli nepovedlo přihlásit, metoda `getUsername` vrací `false`.

4. REALIZACE

```
<?php
class ShibbolethAuth extends Nette\Object{
    //can be used to check if shibboleth authentication has taken
    place

    public static function getShibIdentityProvider() {
        if (isset($_SERVER['REDIRECT_Shib-Identity-Provider']) &&
            $_SERVER['REDIRECT_Shib-Identity-Provider'] !== '') {
            return $_SERVER['REDIRECT_Shib-Identity-Provider'];
        }
        return false;
    }

    public static function getUsername() { //used by login.php
        if (isset($_SERVER['REDIRECT_uid']) && $_SERVER['REDIRECT_uid']
            !== '')
            return $_SERVER['REDIRECT_uid'];
        return false;
    }
}
```

4.2.5.2 Nastavení protokolu HTTPS v Nette

Je třeba zajistit, aby přihlašování skrz systém Shibboleth probíhalo na zabezpečeném připojení pomocí protokolu HTTPS. Aby server podporoval zabezpečené připojení, je třeba mít aktivovaný modul ssl na Apache serveru. Návod na aktivaci modulu ssl v systému Ubuntu je k nalezení na tomto zdroji [10].

V souboru `SagePresentationManager/App/bootstrap.php` se vytvoří route `Sign:in`. To je stránka, na které se přihlašování do systému Shibboleth děje. Zajistění HTTPS protokolu se provede zadáním fráze `Route::SECURED`.

```
$router[] = new Route('login', 'Sign:in', Route::SECURED); // login
je vždy sifrovany
```

4.2.5.3 Zajištění přesměrování na službu Shibboleth

Následně je třeba zajistit přesměrování do systému Shibboleth v momentě, kdy se bude chtít uživatel přihlásit. To se dá zařídit na několika úrovních. Například v metodě `render` v patřičném presenteru aplikace. Nebo přímo na úrovni serveru Apache ještě dříve, než se začne vykonávat kód v aplikaci. Tuto druhou možnost jsem si na doporučení vedoucího práce zvolil. Pro zajištění správného přesměrování do systému Shibboleth provedeme následující změny v souboru `httpd.conf`

Zajistíme používání HTTPS protokolu. Pokud by se někdo připojil k aplikaci pomocí HTTP protokolu (port 80), bude přesměrován na HTTPS (port 443).

```
<VirtualHost *:80>
    ServerName manager.fit.cvut.cz
    Redirect 301 / https://manager.fit.cvut.cz
</VirtualHost>
```

Dále je třeba zajistit přesměrování do systému Shibboleth, pokud uživatel přistoupí na přihlašovací stránku `manager.fit.cvut.cz/www/sign/in`.

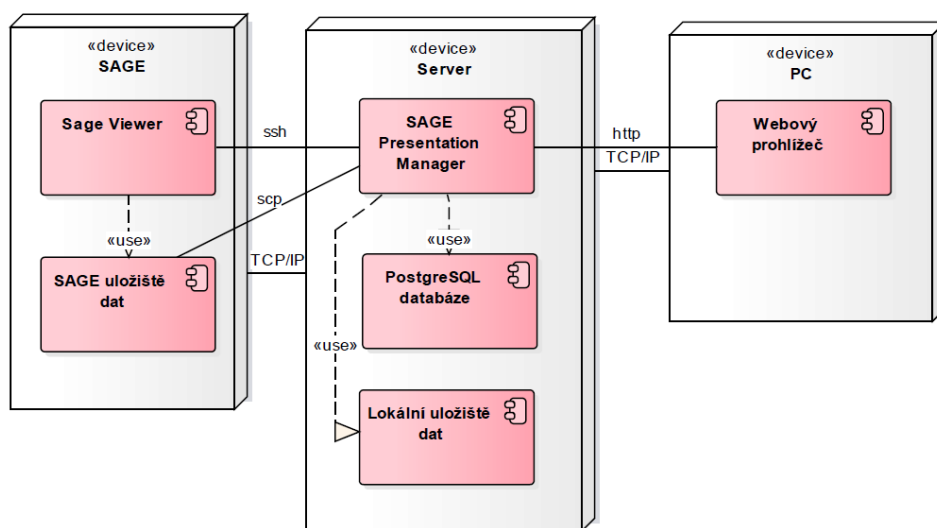
```
<VirtualHost *:443>
    ServerName manager.fit.cvut.cz

    <Location /www/sign/in>
        AuthType shibboleth
        ShibRequireSession On
        require valid-user
    </Location>
</VirtualHost>
```

4.3 Diagram nasazení

Hlavní aplikace je nasazena na serveru společně s PostgreSQL databází, ve které budou uloženy záznamy o uživatelích a jejich prezentacích, a dále bude na serveru využíváno úložiště dat, kde budou ukládány obrázky a skripty prezentací. Tento server mi byl přidělen vedoucím práce. Přistupovat k aplikaci je možno z osobních počítačů pomocí webového prohlížeče. Následně se aplikace spojí přes protokol SSH k serveru SAGE, kde je nainstalován program Sage Viewer, který obstarává zobrazování prezentací na obrazovce (obrázek 4.5). I na serveru SAGE má aplikace vlastní úložiště, kam jsou nahrávány prezentace a jejich zdroje. Prezentace se zde následně přehrávají.

4. REALIZACE



Obrázek 4.5: Diagram nasazení

Testování

V této kapitole se budu zabývat testy, které bude aplikace Správce prezentací pro SAGE podstupovat. Jedná se o jednotkové testy, uživatelské akceptační testy a uživatelské testování použitelnosti. Všechny tyto testy budou podrobně popsány a navrženy v následujících sekcích.

5.1 Jednotkové testy

Jednotkové testy neboli unit testy jsou automatické testy, které ověřují správnost implementace jednotky. Jednotkou se rozumí dílčí část programu, kterou je možné testovat nezávisle na zbytku aplikace. U oběktově orientovaného programování se jedná o jednotlivé třídy a jejich funkce. Tyto testy ze zpravidla zapisují v podobě programového kódu, takže je provádí sám programátor [11].

Pro tuto aplikaci, napsanou ve frameworku Nette, má cenu jednotkově testovat pouze jednotlivé modely.

5.1.1 PresentationModel

Z důvodu nedostatku času jsem prováděl jednotkové testy pouze na třídě `PresentationModel`. To ovšem není velký nedostatek, neboť ostatní tři modely jsou pouze podmnožiny třídy `PresentationModel`. Všechny třídy v sobě zahrnují základní práci s databází a vytváření Nextras datagridu. `PresentationModel` navíc ještě obsahuje jednorázové akce pro práci s prezentací, jako je například vytvoření hluboké kopie prezentace, nebo vygenerování scriptů pro prezentaci. Jediné chyby v ostatních modelech, které by tyto testy neodhalily jsou případné překlepy.

Pro testování používám třídu Nette Tester. Testy jsou realizovány jako PHP scripty, které jsou následně spouštěny z příkazové řádky pomocí příkazu `tester tests/Název_testu.phpt`, a nebo je možné spustit všechny testy najednou pomocí příkazu `tester tests`. Pro demonstraci je jeden z testů uveden přímo v textu práce. Ostatní testy jsou součástí přílohy na CD.

5. TESTOVÁNÍ

```
<?php
function testCheckConsistency(){
    Assert::same(null,$this->presentationModel->checkConsistency($idOkay[0]));
    Assert::same(null,$this->presentationModel->checkConsistency($idOkay[1]));
    $badresult=$this->imageModel->getImage($idBadImage)
    Assert::same($badresult,$this->presentationModel->checkConsistency($idFail[0]));
}
```

5.2 Uživatelské testování použitelnosti

Skrze uživatelské testování použitelnosti může programátor získat vstupy od opravdových uživatelů, kteří aplikaci vidí poprvé, a následně aplikaci upravit na základě jejich zpětné vazby. Testování spočívá v tom, že se vzorku potenciálních uživatelů předloží aplikace a scénář průchodu aplikací. Následně je třeba pozorovat uživatele při plnění úkolů a testování řádně vyhodnotit. Uživatelské testování použitelnosti může odhalit chyby, které zůstaly vývojáři skryté.

Já jsem se rozhodl provádět uživatelské testování již na hotové aplikaci. Alternativou by bylo provádět toto testování v dřívější části vývoje. Výhodou testování na konci vývojového procesu je, že uživatelé mohou otestovat i funkčnost aplikace a ne jen uživatelské rozhraní. Nevýhodou je, že pokud se najde nějaká zásadní chyba, nemuselo by být možné tuto chybu opravit.

Cílem uživatelského testování je vyhodnotit uživatelskou přívětivost aplikace. K tomu budou brány v potaz faktory, jako je chybovost při provádění úkonů, rychlost s jakou bude uživatel tyto úkony provádět, nejistotu uživatele při pohybu v aplikaci a jeho celkový dojem.

5.2.1 Vstupní dotazník

Vstupní dotazník bude předložen každému testovacímu subjektu před začátkem samostatného testování. Tento dotazník by měl obsahovat otázky, které by mohly potenciálně ovlivnit výsledek testování. Ptám se zde subjektů například na věk, otázky, které objasní technickou zdatnost uživatele a zda má uživatel již nějakou zkušenost s vytvářením prezentací. Tyto znalosti mi dovolí kategorizovat jednotlivé testovací subjekty. Kompletní vstupní dotazník je součástí přílohy této práce.

5.2.2 Testovací scénář

Testovací scénář je seznam úkolů, který reprezentuje ideální průchod aplikací. Cílem je simulovat běžné používání aplikace. Uživatelé budou mají na počítači připravenou sadu obrázků, které budou potřebovat pro prezentování. V

aplikaci je vytvořen jeden cizí uživatelský účet, ke kterému nemají uživatelé přístup.

1. Zaregistrujte se

Vyplňte libovolné registrační údaje, ale je třeba si je zapamatovat. Není třeba vyplňovat nepovinné údaje.

2. Přihlašte se do aplikace.

3. Zobrazte vlastní profil a změňte svoje telefonní číslo na 123456789.

4. Nahrajte do aplikace obrázky "obr1.jpg", "obr2.jpg" a "obr3.jpg", které jsou připravené na testovacím počítači.

5. Vytvořte novou prezentaci s názvem "prezentace1".

6. Vytvořte dvě nové sekce s názvy "sekce1" a "sekce2".

7. Vložte obrázek "obr1.jpg" do sekce "sekce1" tak, aby se zobrazil na souřadnicích 0,0 byl velký 1920x1080 px a zobrazil se na začátku sekce po dobu 5 sekund.

8. Vložte obrázek "obr2.jpg" do sekce "sekce2" tak, aby se zobrazil na souřadnicích 0,0 byl velký 5760x2340 px a zobrazil se na začátku sekce po dobu 10 sekund.

9. Vložte obrázek "obr3.jpg" do sekce "sekce2" tak, aby se zobrazil na souřadnicích 1920,1080 byl velký 3840x2160 px a zobrazil 5 sekund po začátku sekce na dobu 5 sekund.

10. Vložte sekci "sekce1" do prezentace "prezentace1" tak, aby se zobrazila na začátku prezentace.

11. Vložte sekci "sekce2" do prezentace "prezentace1" tak, aby se zobrazila po skončení sekce "sekce1".

12. Proveďte kontrolu konzistence zdrojů pro prezentaci "prezentace1"

13. Vyexportujte script.

14. Přehrajte prezentaci v prohlížeči.

15. Spusťte prezentaci na SAGE.

16. Sdílejte prezentaci "prezentace1" s uživatelem "user2".

17. Vytvořte kopii prezentace "prezentace1".

18. Přejmenujte kopii prezentace1 na "prezentace2".

5. TESTOVÁNÍ

19. přejmenujte obě sekce v prezentaci "prezentace2" na "sekce3" a "sekce4".
20. odstraňte obrázek "obr3.jpg" ze sekce "sekce2".
21. Smažte prezentaci "prezentace1".
22. Odhlašte se.

5.2.3 Průběh testování

Testování proběhne v laboratoři SAGElab. Testovací subjekty, které obstará vedoucí práce, napřed vyplní vstupní dotazník. Po vyplnění vstupního dotazníku jsou uživatelé připuštěni k testovacímu počítači a testovacímu scénáři, jehož úkoly musí splnit. Je důležité, aby uživatelé nebyli během plnění úkolů rušeni a od organizátora dostávali pouze nutné instrukce. Během plnění úkolů jsou subjekty natáčeny kamerami, ze kterých bude možné později vyhodnotit jejich počínání. Následně uživatelé vyplní výstupní dotazník, který by měl objasnit, jaké pocity z aplikace uživatelé mají.

5.2.4 Výstupní dotazník

Výstupní dotazník je předložen testovacímu subjektu na konci testovacího sezení. Uživatel by se měl podělit se svými dojmy a postřehy z testování. Je důležité dát uživateli prostor k rozepsání, aby se uživatel mohl podělit o své postřehy, ale zároveň klást konkrétní otázky, aby dotazník nebyl moc obecný.

Konkrétní znění dotazníku je součástí přílohy práce.

5.2.5 Vyhodnocení testování

Z důvodu nedostatku času bude na doporučení vedoucího práce prováděno testování v mezidobí mezi odevzdáním a obhajobou práce. Proto bude prezentace výsledků uživatelského testování obsahem obhajoby práce.

5.3 Uživatelské akceptační testování

Akceptační testy, nebo také user acceptance test (UAT), jsou poslední testy, které v praxi provádí zákazník po nasazení na svůj server. Tyto testy jsou často prováděny testovacím týmem zákazníka podle scénáře, který společně vytvořili zákazník a programátor. Účelem těchto testů je zjistit, zda výsledná aplikace obsahuje veškerou funkcionalitu a objevit případné chyby, které vývojář neobjeví [11].

V případě této práce provedu akceptační testy já po finálním nasazení na deploy server, který mi byl přidělen vedoucím práce. Projdu všechny případy užití rozebrané v tomto textu a budu je aplikovat na Správce prezentací pro SAGE.

Závěr

Úkolem této práce bylo analyzovat možnosti vytváření a zobrazování prezentací v prostředí SAGE. Analyzovat požadavky potenciálních uživatelů na aplikaci, která by sloužila k vytváření a zobrazování prezentací. Navrhnout a naimplementovat webovou aplikaci, která bude obstarávat vytváření, úpravy a spouštění obrázkových prezentací. Navrhnout úpravy v prostředí SAGE, které umožní zobrazování prezentací. A celou aplikaci řádně otestovat.

Zhodnocení výsledků

V rámci analýzy současných prostředků zobrazování prezentací jsem rozebral funkci prostředí SAGE a dvou základních zobrazovačů Smedia a Image viewer. Dále jsem analyzoval postupy, jakými jsou prezentace v současné době vytvářeny, program SAGE UI a vytváření skriptů v jazyce Shell.

Pro analýzu požadavků uživatelů jsem vytvořil funkční a nefunkční požadavky, které vycházejí z potřeb kolegů pracovníků laboratoře SAGElab. Sestavil jsem případy užití, ke kterým bude v aplikaci docházet.

Navrhl jsem webovou aplikaci s použitím metod softwarového inženýrství. Navrhl jsem Doménový model tříd, ze kterého následně vzešel návrh databázového modelu. Analyzoval jsem procesy, které budou v aplikaci probíhat a některé z nich jsem podrobně rozepsal. Nakonec jsem navrhl vzhled uživatelského rozhraní aplikace a realizoval ho pomocí wireframů.

Tuto aplikaci jsem následně naimplementoval a nasadil na serveru, který mi byl přidělen vedoucím práce.

V prostředí SAGE jsem vytvořil nový program Sage Viewer a modifikoval stávající program Smedia tak, aby společně dovedly zobrazovat prezentace na obrazovce v laboratoři SAGElab.

Závěrem jsem navrhl a vytvořil sadu testů a těm jsem celou aplikaci podrobil. Jednalo se o jednotkové testy, uživatelské testy a akceptační testy.

Všechny body zadání jsem naplnil a aplikace je nasazená a připravená k používání.

Možnosti dalšího vývoje

Existuje několik hlavních směrů, kterými by se dala tato aplikace dále rozšiřovat. Jednou možností by bylo rozšířit aplikaci o podporu dalších multimédií, než jsou jen obrázky. Například soubory pdf a videa. Nebo přidat další funkcionality navíc, jakou by mohlo být například sdílení prezentací přes sociální sítě. Druhým hlavním směrem vývoje by bylo rošíření této aplikace do prostředí SAGE2. Třetí větví rozvoje by bylo rošíření na další platformy, jako jsou chytré telefony.

Literatura

- [1] CESNET: *SAGELab síťová multimediální laboratoř - fotogalerie [online]*. [cit. 2017-04-23]. Dostupné z: <https://sagelab.cesnet.cz/cz/fotogalerie/>
- [2] The University of Illinois Board of Trustees: *About SAGE [online]*. [cit. 2017-04-26]. Dostupné z: <http://sage.sagecommons.org/project/about/>
- [3] The University of Illinois Board of Trustees: *SAGE Documentation [online]*. [cit. 2017-04-23]. Dostupné z: http://sage.sagecommons.org/wp-content/uploads/downloads/2013/10/SAGEDocumentation_9.27.2013.pdf
- [4] RICHTR, R.: Triviální zobrazování prezentací na zařízení SAGE, [cit. 2017-05-04].
- [5] CESNET: *SAGELab síťová multimediální laboratoř - o laboratoři [online]*. [cit. 2017-04-23]. Dostupné z: <https://sagelab.cesnet.cz/cz/o-laboratoři/>
- [6] BRÉM, O.: *SAGELab - REST rozhraní zařízení SAGE*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií.
- [7] SEDLÁČEK, L.: *SAGELab - Bezdotykové ovládání*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií.
- [8] PHAM, D. T.: *SAGELab - Android Manager zařízení SAGE*. Bakalářská práce, České vysoké učení technické v Praze, Fakulta informačních technologií.
- [9] České vysoké učení technické v Praze: *Single sign on (SSO) – Shibboleth [online]*. [cit. 2017-05-11]. Dostupné z: <https://portal.cvut.cz/pristup-do-is/uzivatele-a-pristup-do-is-cvut/single-sign-on-sso-shibboleth/>

LITERATURA

- [10] DigitalOcean™ Inc.: *How To Create a SSL Certificate on Apache for Ubuntu 14.04 [online]*. [cit. 2017-05-11]. Dostupné z: <https://www.digitalocean.com/community/tutorials/how-to-create-a-ssl-certificate-on-apache-for-ubuntu-14-04>

- [11] Testování softwaru: *Fáze a úrovně provádění testů [online]*. [cit. 2017-05-06]. Dostupné z: <http://testovanisoftwaru.cz/metodika-testovani/druhy-typy-a-kategorie-testu/faze-testu/>

Seznam použitých zkratk

SAGE Scalable Adaptive Graphics Environment

SAGE2 Scalable Amplified Group Environment

SAGE UI SAGE User Interface

ČVUT České vysoké učení technické

IS ČVUT Informační systém ČVUT

SSO Single Sign-on

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

MVP Model View Presenter

SSH Secure Shell

SCP Secure Copy

TCP/IP Transmission Control Protocol/Internet Protocol

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
bin	
_ smedia	spustitelná verze programu Smedia
_ sage_viewer.....	spustitelná verze programu Sage Viewer
src	
_ SAGE Presentation Manager.....	zdrojové kódy webové aplikace
_ Smedia.....	zdrojové kódy upraveného programu Smedia
_ sage_viewer.cpp	zdrojový kód programu Sage viewer
_ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
_ thesis.pdf	text práce ve formátu PDF
tests	
_ PreTestForm.pdf	Vstupní dotazník
_ Instructions.pdf	Testovací scénář
_ PostTestForm.pdf	Výstupní dotazník
_ Unit.....	Zdrojové kódy jednotkových testů