

Na tomto místě bude oficiální zadání vaší práce

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačové grafiky a interakce



Bakalářská práce

Virtuální galerie v kostele Sv. Petra a Pavla u Líšné

Daniel Šrám

Vedoucí práce: prof. Ing. Jiří Žára, CSc.

Studijní program: Softwarové technologie a management, Bakalářský

Obor: Web a multimedia

11. května 2010

Poděkování

Velmi rád bych poděkoval a vyslovil uznání všem, kteří mi pomáhali při vzniku této práce. Především prof. Ing. Jiřímu Žárovi, CSc., vedoucímu mé bakalářské práce, za trpělivé vedení, ochotu a množství praktických rad. Dále Ing. Zdeňkovi Trávníčkovi, který mi vždy ochotně pomohl cennými informacemi.

Mé další poděkování patří zadavateli práce, paní Kláře Dvořákové, která byla vždy ochotná a stála u vzniku celé práce, od jejího počátku, až do konce. Chtěl bych také poděkovat paní malířce, Adrianě Skálové, za poskytnuté informace a obrazy, kterými je kostel vyzdoben.

Nakonec bych chtěl poděkovat rodičům za poskytnuté zázemí, podporu a trpělivost.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v přiloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Brandýse nad Labem dne 10. 5. 2010

.....

Abstract

This bachelor thesis deals with the design and implementation of a virtual gallery for visitors of St. Peter and Paul church in Líšná village near Zbiroh. The work consists of two parts, a virtual model of the church, along with pre-prepared gallery and web application. The virtual model was created in the VRML language and can be integrated directly into the web site of the church. The creator of an exhibition have a real chance to offer his / her art work in digital form presented via Internet to the public trough this application.

The benefit of this project is not only managing galleries, but also an effort to help and rescue the church, which is by approaching the object to the public and various organizations via the Internet.

Abstrakt

Tato bakalářská práce se zabývá návrhem a realizací virtuální galerie pro návštěvníky kostela Sv. Petra a Pavla u Líšné na Zbirožsku. Práce se skládá ze dvou částí, virtuálního modelu kostela, spolu s předem připravenou galerií a dále z webové aplikace. Virtuální model byl vytvořen v jazyce VRML a lze jej integrovat přímo do webových stránek kostela. Prostřednictvím této aplikace mají tvůrci výstav možnost nabídnout svá výtvarná díla prostřednictvím internetu široké veřejnosti v digitální podobě.

Přínosem tohoto projektu ale není pouze správa galerií, ale také snaha o pomoc a záchranu kostela, která spočívá právě v přiblížení tohoto objektu veřejnosti a různým organizacím pomocí internetu.

Obsah

1	Úvod	1
2	Analýza projektu	3
3	Zvolené technologie a pracovní postupy	5
3.1	Zvolené technologie	5
3.2	Pracovní postupy	6
4	Realizace	7
4.1	Vytváření 3D modelů	7
4.2	Exportování scény	12
4.3	Tvorba galerie a stanovišť	14
4.4	Optimalizace VRML	15
4.5	Technický popis a parametry modelu	16
4.6	Rozdělení aplikace do tří vrstev	19
4.7	Použité frameworky a návrhové vzory	20
4.8	Popis funkčnosti	21
4.9	Diagram nasazení	25
4.10	Diagram případů užití	25
4.11	Diagram tříd	26
4.12	Dokumentace systému	27
4.13	Propojení s 3D scénou	34
4.14	Uživatelské role	34
4.15	Grafický návrh aplikace	35
4.16	Model versus realita	35
5	Testování	39
6	Závěr	41
	Literatura	45
A	Příručka pro návštěvníky virtuálního kostela	47
A.1	Návštěva kostela	47
A.2	Aktivní prvky	48

B	Příručka pro tvůrce virtuálních výstav	49
B.1	Úvod	49
B.2	Uživatelské role	49
B.3	Uživatel	50
B.3.1	Přihlášení uživatele	50
B.3.2	Vytvoření nové galerie	50
B.3.3	Editace galerie	51
B.3.4	Nastavení výstavy	52
B.3.5	Prohlídka výstavy	53
B.3.6	Smazání galerie	54
B.3.7	Tikety - co jsou, k čemu slouží	54
B.3.8	Osobní údaje	54
B.3.9	Uživatelé	55
B.3.10	Nápověda	55
B.3.11	Historie	55
B.4	Administrátor	55
B.4.1	Přihlášení administrátora	55
B.4.2	Registrace nového uživatele	55
B.4.3	Smazání uživatele	55
B.4.4	Osobní údaje	56
B.4.5	Uživatelé	56
B.4.6	Nápověda	56
B.4.7	Historie	56
C	Instalační příručka	57
D	Seznam použitých zkratk	59
E	Obsah přiloženého CD	61

Seznam obrázků

2.1	Blokové schéma komunikace	4
4.1	Modelování kostela	8
4.2	Modelování interiéru kostela	9
4.3	Virtuální modely oltáře a kazatelny	10
4.4	3D model místních varhan	11
4.5	Modelování exteriéru	12
4.6	Kompletní virtuální scéna	13
4.7	Objekt vytvořený křivkami nurbs a objekt exportovaný do VRML	14
4.8	Pohled na levou stěnu galerie	15
4.9	Graf měření FPS v závislosti na LOD varhan	18
4.10	Graf měření FPS jednotlivých stanovišť	18
4.11	Rozdělení aplikace	20
4.12	Implementace MVC	21
4.13	Formulář přihlášení	22
4.14	Hlavní menu, sekce editace	24
4.15	Diagram nasazení aplikace	25
4.16	Diagram případů užití	26
4.17	Diagram tříd	27
4.18	Sekce tiketů uživatele	29
4.19	Sekce pro změnu údajů uživatele	33
4.20	Grafický návrh aplikace	36
4.21	Porovnání předního pohledu na objekt	36
4.22	Porovnání z pohledu od vstupních dveří kostela	37
4.23	Porovnání při pohledu z balkonu na oltář	38
A.1	Kompletní virtuální scéna kostela	48
B.1	Vytvoření nové galerie	51
B.2	Editace galerie	52
B.3	Nastavení výstavy	53
B.4	Prohlídka výstavy	53
B.5	Smazání galerie	54

Seznam tabulek

4.1	Velikosti a rozměry použitých textur	17
4.2	Počty polygonů objektů ve scéně	19
4.3	Časové rozložení tvorby virtuální scény	19
5.1	Tabulka testování aplikace uživateli	40

Kapitola 1

Úvod

Na úvod bych rád zmínil několik slov o projektu, který jsem si vybral pro svoji bakalářskou práci, potažmo pro semestrální projekt, který je z ní odvozen. Tento projekt v sobě obsahuje hned několik různých technologií, které se zde propojují. Jedná se o modelování a práci s virtuální realitou a v další části poté vytváření aplikace, která bude uživatelům umožňovat umísťování obrazů na stěny virtuálního kostela.

Projekt jsem si vybral, protože mne zaujalo toto bohaté zadání a zajímavý projekt. Za dobu svého studia jsem se již mnohokrát setkal s virtuální realitou, pouze však jako její uživatel či návštěvník. Tento obor, či dalo by se říci oblast zájmu, mne velice zaujala a jako programátor jsem se rozhodl jí vytvářet a produkovat.

Kapitola 2

Analýza projektu

Z hlediska bakalářské práce

Bakalářská práce se skládá z více částí. Nejedná se pouze o kostel jako stavbu, což je ze zadání patrné, ale také o aplikaci, která bude uživatelům umožňovat správu a vytváření svých virtuálních výstav. Projekt je tedy rozdělen do dvou fází, přičemž každá fáze je detailněji rozepsána a vysvětlena. Tyto fáze jsem realizoval postupně. Nejprve vznikl kompletní 3D model kostela včetně interiéru a exteriéru a následně jsem implementoval aplikaci a vytvářel pro virtuální model funkční logiku. Blokové schéma, popisující kompletní komunikaci mezi klientem a serverem, na kterém se aplikace i 3D model nachází, popisuje obr. č. 2.1.

Z hlediska aplikace

Cílem této aplikace je umožnit laickým uživatelům vytvářet a spravovat své 3D galerie ve virtuálním světě. Pro tento projekt byl vybrán 3D model kostela, aplikace je však přenositelná na kterýkoli virtuální svět.

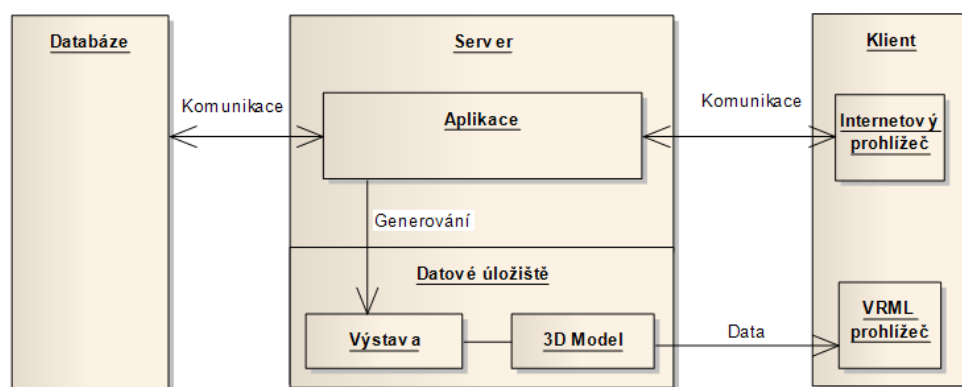
Pro vytváření galerie je třeba uživatele. Jeho práva jsou nejlépe definována uživatelskými rolemi. Aplikace proto rozděluje uživatele do několika skupin podle uživatelských rolí. Dále má aplikace umožňovat kurátorům výstavy realizaci jimi vytvořených galerií přímo ve virtuálním světě. K tomuto účelu slouží generátor VRML souborů, které jsou poté nahrávány virtuálním světem a dále zpracovávány. Tato aplikace umožňuje všechny tyto prvky správy uživatelských galerií plus některé další prvky, jako uzamykání generovaného souboru tikety, změnu uživatelských údajů, různé další výpisy a nastavení.

Z hlediska modelu

V realizaci virtuálního modelu bylo potřeba zaměřit se na část interiéru a exteriéru kostela jako dva samostatné celky, které však se sebou úzce souvisí. Z počátku jsem nutně potřeboval zajistit potřebné zdroje informací a získat současné stavební plány, fotografie a další důležité zdroje. Nezbytná byla samozřejmě i návštěva onoho objektu, který jsem tak analyzoval

osobně. Získal jsem všechny potřebné podklady včetně konzultace s panem Jansou, který již léta o kostel pečuje a zná tedy mnohé detaily.

Dalšími částmi projektu bylo například exportování všech částí do jazyka VRML, ve kterém toto virtuální prostředí vytvářím. Následovala nutná optimalizace z hlediska úpravy a přehlednosti výsledných souborů, i z hlediska co nejmenší náročnosti pro prohlížeče, které tento svět zobrazují uživatelé. Všechny uvedené části popisují detailněji níže. Jazyk VRML jsem zvolil pro svoji práci z několika důvodů. Oproti jiným jazykům, ze kterých bych uvedl např.: X3D má totiž VRML větší podporu v prohlížečích i z toho důvodu, že je to starší technologie, která se vyvíjela už od roku 1994. V bakalářské práci používám kompletní verzi jazyka VRML - VRML97. VRML je nejvíce rozšířeným jazykem v této oblasti, i když ho již postupně nahrazuje jeho nástupce, právě jazyk X3D.



Obrázek 2.1: Blokové schéma komunikace

Kapitola 3

Zvolené technologie a pracovní postupy

3.1 Zvolené technologie

Tvorba projektu je, jak již bylo řečeno, postavena na více technologiích. Pro modelování je důležitý výběr modeláře, který umožňuje vypracovat detailní a věrohodné 3D modely skutečných objektů. Samozřejmě jednoduché křivky lze modelovat i v samotném jazyce VRML. Pro složitost mého projektu jsem však zvolil raději práci s profesionálním modelářem, který nabízí všechny dnes potřebné moderní technologie pro profesionální tvorbu 3D modelů.

Po osobních zkušenostech s open-source programy jsem se rozhodl použít 3D modelář Blender [2] ve verzi 2.49b. Součástí tohoto programu je i možnost exportování do různých formátů, přičemž jsem využil pouze export do souborů s příponou wrl. Pro tento program existuje i mnoho pluginů. Pro moji práci byl nezbytný BS Exporter [4] pro program Blender, verze 1.22 od společnosti Bitmanagement, který umožňuje exportovat i některé texturovací souřadnice. Bohužel textury tento program ani plugin exportovat neumí.

Pro prohlížení virtuálního světa jsem jako primární prohlížeč zvolil program Cortona3D Viewer 6.0 [5] společnosti Parallel Graphics. Samozřejmostí je funkčnost i na ostatních prohlížečích. Jak je ovšem známo i ze světa internetových stránek, každý prohlížeč obsahuje oproti ostatním rozdílné interpretace a chyby a to platí navíc i pro různé verze jedné značky. Co se týče prohlížečů virtuálních scén a světů, platí bohužel to samé pravidlo. Uživatelům je tedy doporučena instalace nejnovější verze Cortony pro správnou a bezproblémovou funkčnost.

Modely, které jsou již ve výsledném souboru wrl, je třeba upravovat a optimalizovat. K tomuto účelu jsem používal program VRML Pad [17] od společnosti Parallel Graphics ve verzi 3.0. Vzhledem k autentické náročnosti projektu jsem musel textury upravovat na základě získaných fotografií. K tomuto účelu jsem použil open-source program Gimp 2 [8].

Vývoj aplikace jsem se rozhodl realizovat v jazyce PHP z důvodu nasazení na kterémkoli současném hostingu a serveru. Tato aplikace bude součástí stránek sdružení pro zachování kostela u Líšné a správcem této aplikace bude právě toto sdružení. Aplikace bude tak přenositelná a bude spolehlivě fungovat kdekoli na světě.

Rozhodl jsem se pro programování pomocí objektově orientovaného přístupu, tedy při programování a následném běhu aplikace vytvářím objekty, které mají atributy, jsou provázány

vazbami a vzájemně spolupracují. Cílem je stabilní aplikace s využitím moderních programovacích technik. Při programování jsem využil rozdělení aplikace do tří vrstev s využitím několika frameworků a návrhových vzorů pro potřeby aplikace.

Jako databázi, která ponese data aplikace jsem zvolil běžně používanou a na většině serverů přístupnou databázi MySQL verze 5.1.37. S touto databází v aplikaci pracuje ORM framework, který může být nasazen na kteroukoli jinou databázi. Aplikace tedy není závislá přímo na tomto typu databáze.

Dalším použitým systémem je Smarty, který přináší komplexní šablonovací systém pro PHP [13]. Jedná se tedy o velice vhodnou komponentu pro vývoj webových aplikací. Umožňuje nám čistý výstupní kód od programátora pro grafika, který tak jednoduše a čistě implementuje na jakýkoli grafický návrh bez nutné znalosti business logiky.

3.2 Pracovní postupy

Při práci na modelu jsem postupoval následujícím způsobem. Nejprve jsem se snažil získat co nejvíce informací o objektu, detailech, v jakých se má projekt uskutečnit a co všechno má být obsahem interiéru. V dalších krocích jsem již přistoupil k samotnému modelování stavby a později objektů. Konkrétní postupy modelování jednotlivých objektů popisují v kapitole níže. Od získávání informací jsem tedy přistoupil k samotnému modelování a poté exportování vymodelovaných hotových částí do VRML souborů. Dalším krokem bylo po každém vymodelovaném objektu scénu otestovat a zjistit a opravit případné chyby v realizaci. Po finálním exportování bylo třeba všechny objekty nějakým způsobem texturovat, aby vytvořily dojem opravdových reálných objektů. Texturování jsem prováděl pomocí získaných obrázků dřeva [15]. Naprostou většinu textur jsem ale musel vytvořit přesně na míru projektu.

V následujícím případě bylo tedy potřeba vytvořit ze získaných fotografií textury, protože dle mého názoru nejlépe působí opravdová fotografie a ve spojení se 3D modelem je zážitek o to umocněn.

Fotografie jsem upravoval v open-source programu Gimp 2 [8], který je jednoduchý na ovládání a velice dobře se s ním pracuje. Velice nutné bylo vytvořit efekt průhledných oken a k tomu bylo zapotřebí oddělit část okna od zbytku fotografie, oříznout okolí a poté do fotografie přidat alfa kanál klíčovaný na bílou barvu. Podobným způsobem jsem postupoval i při vytváření ostatních textur pro střechu [14], veškeré zdi kostela, dveře a ostatní objekty.

Následovalo umístění obrazů do předem vytvořených obrazových rámců, kterých je v kostele několik typů. Pro vkládané obrazy jsem použil rám s deskou, na kterou mapuji jako texturu, zmenšeninu skutečného obrazu. Jako poslední následovalo vytvoření stanovišť, což popisují v samostatné kapitole níže.

Kapitola 4

Realizace

4.1 Vytváření 3D modelů

Část bakalářské práce se skládá ze 3D modelů a tvoří virtuální scénu, neboli virtuální svět. Hlavním modelem pro celou scénu je stavba kostela. Tuto hrubou stavbu jsem modeloval od základů a k jejímu vytvoření jsem použil několik známých a používaných metod.

Prvotní věcí před samotným modelováním bylo nahrát na pozadí modelovacího prostředí fotografii, podle které jsem provedl půdorys budovy. Tento plánek mi byl zapůjčen jako zdroj informací pro vytváření autentického modelu. Po umístění takového pozadí jsem začal postupovat vytvořením půdorysu budovy z křivek, později spojených a vyplněných, tak, jak popisují jednotlivé podkapitoly s příloženou obrazovou dokumentací postupů modelování.

Modelování exteriéru stavby

Metodou extrudování jsem vytvářel ze základní desky, která již měla tvar půdorysu kostela, vnější stěny a později opět metodou extrudování a zmenšení do několika málo bodů jsem z těchto stěn tvořil střechu kostela. Následovalo několik desítek menších úprav, poměry stěn a všech ostatních objektů jsou však zachovány. Po vytvoření korpusu kostela, viz obr. 4.1 jsem přešel ke tvorbě stropu, který je v současnosti z části renovován, ve 3D modelu však tato renovace zohledněna není, protože na pohled se o žádnou významnou změnu nejedná.

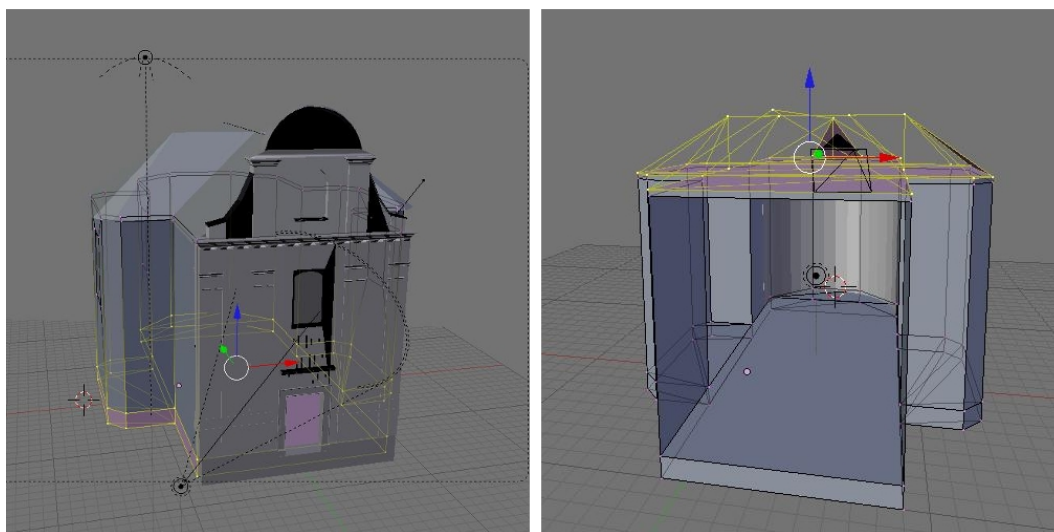
Kostel obsahuje okna, která jsou zasazená do rámců ve zdi. Tyto rámy jsou modelovány zvlášť, protože jsem okna potřeboval mít kompletně vytvořena dříve, než vnitřní stěny kostela, které se tak modelovaly daleko snáze. Do rámců poté přišla okna. Ta jsou v modelu průhledná a celý tento trik se mi podařilo zhotovit pomocí fotografií, které jsem pořídil v kostele. Jedná se tedy o desku, zasazenou do rámu a na ní nanesenou texturu, které jsem přidal alfa kanál a zprůhlednil bílou barvu. Okna jsou tak originální, přímo z kostela.

Tyto rámy by ale nemohly být umístěny, pokud ve zdech kostela nejsou příslušné otvory. Ty jsem vytvářel pomocí metody extrude a scale. Modelář takto umožňuje jednoduše vytvořit jakýkoli otvor. Poté již nebylo problém rám i s oknem použít. Po opětovných detailních úpravách jsem dosáhl výsledku hotové hrubé stavby, ovšem bez přední stěny kostela.

Podle plánek a fotografií jsem se snažil o co nejpřesnější vytvoření modelu, samozřejmě s ohledem na výpočetní složitost. Proto jsou i zde zachována pravidla optimalizace, jako je

odstranění spodních podstav sloupů a podobně. Přední stěna se tedy skládá celkem ze 6 sloupů, které jsou různě zapuštěné směrem dovnitř kostela. Prostřední část byla vytvořena samostatně metodou extrude z vytvořené obyčejné desky. Do této stěny byly poté vymodelovány otvory na míru dveřím, u kterých byl vytvořen masivní kamenný rám, a oknu, které se v čele této stěny nachází. Z primitivních těles, jako je kostka, byly vytvořeny římsy – nad dveřmi a hlavní římsa na vrchu sloupů. U kostela byl modelován i okap úpravami primitivních těles a použitím kovové textury [11] pro jeho kovový vzhled. Opět jsem postupoval metodou osobních zkušeností a snažil se o modelování přesné kopie tamního objektu.

Přední stěna obsahuje dále ozdobné prvky, jako oblouky, střechu či menší ozdoby na krajích přední stěny. Modelování těchto objektů probíhalo stejnými metodami, jako modelování přední stěny s tím rozdílem, že je zde náznak okna, není to však skutečný otvor, ale pouze náznak s římsou.



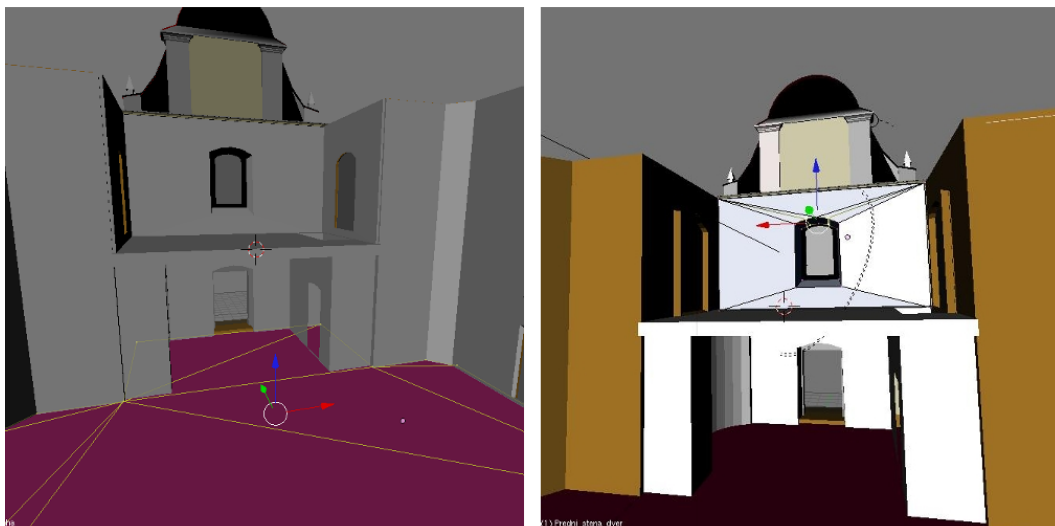
Obrázek 4.1: Modelování kostela

Modelování interiéru stavby

Po kompletaci kostela zvenčí jsem se přenesl na modelování interiéru, viz obr. 4.2. Stále jsem však ještě pracoval na vnitřku kostela. Bylo zapotřebí vymodelovat vnitřní stěny, protože kostel zvenku a zevnitř má jisté odlišnosti. V místě kazatelny je uvnitř zaoblená stěna do půlkruhu, zatímco zvenku je tato stěna ze třech rovných ploch. Toto jsem vyřešil umístěním kružnice, označením poloviny jejích bodů a následně jejich odstraněním. Zbylé body jsem poté extrudoval v ypsilonové souřadnici do výšky stropu a zarovnal. Podobným způsobem jsem vytvářel vnitřní zaoblenou stěnu pod balkonem. Vnitřní rovné stěny jsem modeloval vložím a upravením desky. Pokud se stěny opakovaly (typicky protější stěny byly většinou stejné), použil jsem funkci duplikace objektů.

Následovalo vytvoření samotného prvního podlaží, neboli balkonu, který je opřený na dvou masivních stěnách. Jednu z nich jsem již zmínil. Druhá, protější stěna je rovná a obsahuje otvor do prostoru točitých schodů. Podlahu balkonu jsem modeloval způsobem

vložení krychle a její úpravou. Později jsem odstranil neviditelné, avšak stále přítomné a tudíž zpomalující strany. Ohledně interiéru bylo ještě potřeba vytvořit vnitřní podlahu, respektive její přední odlišnou část, která se nachází pod oltářem. Tuto podlahu jsem vytvořil metodou duplikace části celkové podlahy a její úpravou.



Obrázek 4.2: Modelování interiéru kostela

Modelování objektů

Kostel obsahuje několik památek, které se v současné době budou opravovat. Z důvodů autentičnosti celého projektu je nutné ukázat uživatelům i vnitřek kostela s jeho objekty tak, jak to doopravdy je.

Modely jednotlivých objektů jsem vytvářel na základě vlastního pozorování objektů a na základě vlastních i cizích fotografií a dat. Objekty jsem se snažil tvořit podle původních originálů, což jsem splnil. Některé objekty kostel neobsahuje v takovém detailu. Toto je způsobené přílišnou náročností objektů. Pokud se tedy vyskytl takovýto problém, postupoval jsem nahrazením objektu jeho fotografií. Věřím, že v mnohém je to lepší, než skutečný složitý model.

V nadcházejícím odstavci popisuji vytváření několika objektů v kostele. Jedná se o varhany, oltář a kazatelnu. Ostatní objekty, jako je zábradlí balkonu, lavice či schody jsem modeloval přibližně stejnými metodami, jako níže popisované objekty.

Základem byla vždy fotografie. Tento základ je důležitý pro neustálé vnímání originálu a modelování virtuálního objektu podle něj. Probíhalo tak modelování kazatelny i oltáře. Opět vždy těleso vznikalo ze základního tvaru. U oltáře, viz obr. 4.3 jsem nejprve pomocí spojených bodů naznačil jeho půdorys a poté metodou extrude jsem body vytáhl do výšky oltáře. Opětovným použitím metod, jako je změna velikosti, rotace, duplikace bodů a dalších jsem dosáhl požadovaného tvaru. Poté následovalo vytvoření horních kulových ozdob, které jsem modeloval z válců a horního ozdobného dřeva, které jsem tvořil postupně z křivek kružnice,

jejím vyplněním a extrudováním. Podobným způsobem vznikla i kazatelna 4.3, pouze jsem zde duplikoval dolní sloupky a ozdoby u schodů. Na kazatelnu jsem také výsledně umístil textury dřeva [15] desky s texturami, které představují ozdobné, bohaté erby na stěnách kazatelny. Obrázky jednotlivých modelů se nalézájí níže.



Obrázek 4.3: Virtuální modely oltáře a kazatelny

Tvorba modelu varhan

Model varhan, viz obr. 4.4 vznikl po etapách, přičemž v první etapě byla vymodelována spodní deska, stolička a spodní skříň s ozdobnými rámy. Následovala tvorba části s klávesami, skříňe na tuto část a samotných kláves. Všechny tyto objekty jsem modeloval ze základních typů, jako je deska, či krychle metodou extrude. Následovalo vytvoření horních menších skříňek, otvorů a umístování píšťal jako válců se zúženou dolní podstavou. Poté jsem tvořil ozdobné prvky a upravoval detaily modelu.

Ostatní výřezy či jiné nápadité modelové prvky jsem prováděl prostřednictvím modelovacích nástrojů programu Blender. Mimo jiné jsem hojně využil například kníže tool – řezací nůž či bewel - zaoblení.

V prvním kroku jsem si tedy vložil opět obrázek na pozadí jako základ reálného objektu, který jsem poté modeloval. Model je tak co nejvíce přiblížen reálnému vzhledu. O jeho chování je řečeno později. Nejprve vznikaly základy tohoto hudebního nástroje, jako samotné spodní skříňe, sedátko, piano a nakonec menší horní skříňky s píšťalami a ozdobami.

Varhany, stupně detailu a dynamika

Tento model je věrnou kopií modelu originálního, je tedy modelován s vysokými detaily. Potíž nastává v tom, že člověk, neboli jeho avatar, ne vždy vidí plný detail modelu. Je třeba



Obrázek 4.4: 3D model místních varhan

tedy využít uzlu jménem LOD, který nám ve scéně nastavuje různé modely (většinou jeden model a jeho různé detailové odnože). Můžeme tak dosáhnout dobrých FPS výsledků, pokud již u modelu přímo nestojíme a jsme dál (přepne se model do stavu s nižším lod) a my téměř nevidíme změnu. I tento model je tak vytvořen. Má celkem 4 stupně detailu, přičemž 4. uzel je již prázdný, tudíž se model nezobrazí vůbec. Jednotlivé lod modely jsem vytvářel zjednodušováním originálu.

Model obsahuje i několik ovládacích prvků, které zajišťují funkčnost modelu a přibližují ho uživateli ze stránky, jak model doopravdy funguje. Popis jednotlivých senzorů událostí naleznete níže. Shrnutí, co všechno model umí je v následujících bodech.

- Manipulátor pro stoličku
- Manipulátor pro zavírání / otvírání víka kláves
- Animace kláves

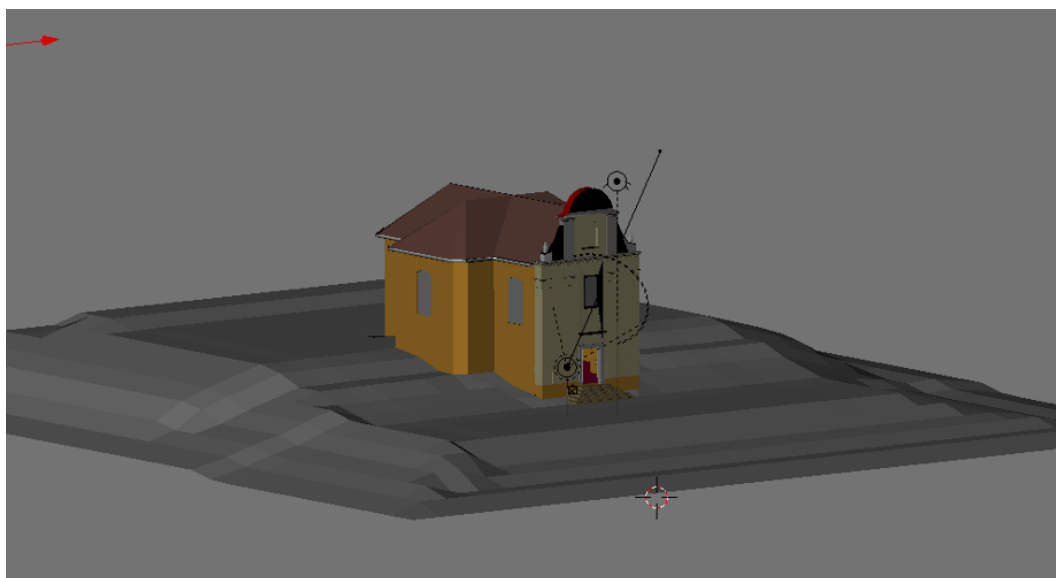
Modelování exteriéru kostela

K modelování externí části krajiny jsem musel přistoupit, protože se kostel nachází ve svahu a je tedy z části mírně zapuštěný v zemi.

Krajinu v okolí kostela jsem modeloval pomocí vložení mřížky a jejím postupným upravováním do požadovaného tvaru, viz obr. 4.5. V okolí se také nachází několik stromů. Jejich modelování je však složité a vytváří problém náročnosti scény versus výpočetní schopnost počítače. Tento problém jsem vyřešil umístěním dvou na sebe kolmých desek v poměru

stran 1:0.7 umístěných na výšku a následným přidáním textury s alfa kanálem [16]. Vznikl tak přijatelný a velice jednoduchý výsledek.

Pozadí krajiny modelováno není. Předpokládám totiž, že návštěvníka bude nejvíce zajímat objekt, jeho stav a jeho vnitřek a galerie. Proto jsem jako okolí kostela použil pět fotografií, představujících kompletní okolní krajinu i s horní fotografií nebe [12]. Kompletní virtuální scénu popisuje obr. 4.6.



Obrázek 4.5: Modelování exteriéru

4.2 Exportování scény

Scénu a samotné modely vytvořené v modeláři je nutné nějakým způsobem převést do souborů wrl jazyka VRML, a k tomuto účelu jsem použil již zakomponovaný exporter v programu Blender. Pro exportování souřadnic textur jsem použil plugin od firmy bitmanagement, popisovaný v kapitole „technologie“. Tento plugin však umožňuje exportovat pouze UV texturovací souřadnice, proto jsem tento nástroj mohl použít jen v ojedinělých případech.

Ne vždy se však export povedlo provést bez komplikací a proto bylo nutné vždy model kompletně otestovat a zkontrolovat. Mnohdy se závada projevila ve špatném nastavení parametrů u těles, jako je „solid“, což určuje, zda-li má těleso zobrazovat i odvrácené strany plošek či nikoli, nebo „ccw“, určení, jakým směrem se mají plošky vytvářet. Tyto dvě chybná nastavení se v mém modelu vyskytovaly zdaleka nejčastěji.

Exportování s sebou nese i další úskalí a tím je zanesení textur skrze modelovací program až do jazyka VRML. Bohužel toto program Blender neumožňuje, a proto jsem jako řešení zvolil ruční umístování a nastavení textur přímo v souborech wrl, což vedlo k lepšímu seznámení se se strukturou dokumentu a obecným povědomím o nastavení všech detailů. Po kompletaci hlavního souboru, který obsahuje stavbu kostela, okna a okenní rámy jsem se



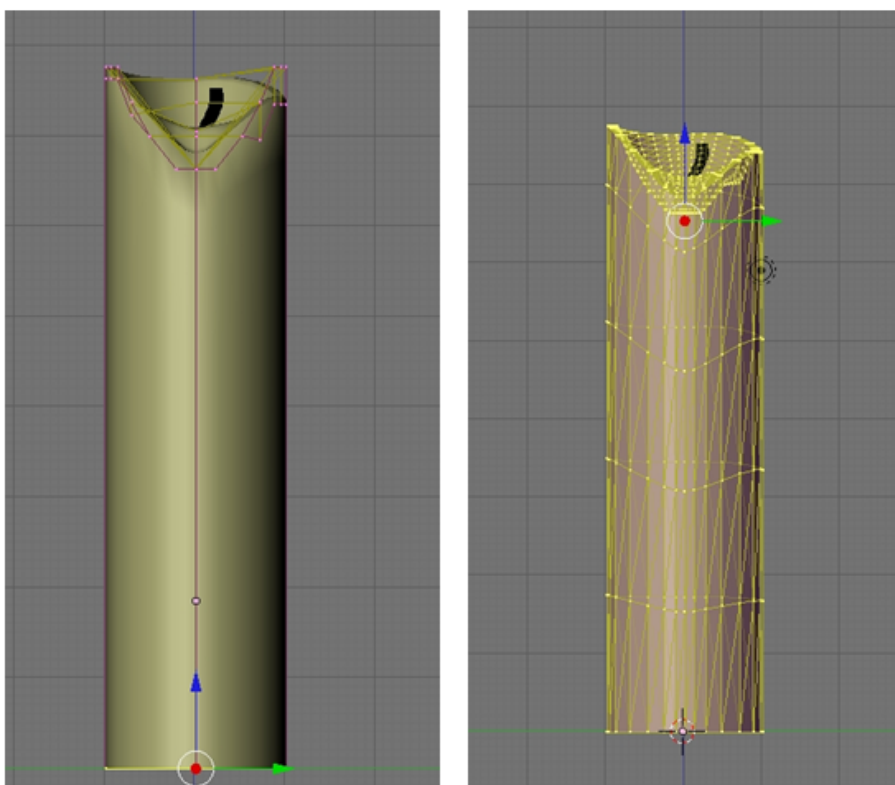
Obrázek 4.6: Kompletní virtuální scéna

setkal s problémem úpravy pouze některých částí stavby. Řešení takového dílčího problému se nacházelo v úpravě tělesa v modeláři a následný export pouze té části, která byla chybná. Následovalo vložení či přepsání upraveného objektu do již hotového souboru, nebo celé scény. Z mého pohledu se tak jedná o nejrychlejší variantu řešení, pokud takovýto problém nastane.

Tvorba komponent pomocí křivek nurbs

Domnívám se, že modelování tímto způsobem přinese do budoucna mnohé výhody. Již dnes se jedná o velice hojně využívanou metodu v modelování. Umožňuje vytvářet přirozenější tvary, práce s těmito křivkami je snadná a výsledný efekt případných animací je velice plastický a vjemově velice pěkný.

Problém, se kterým jsem se setkal, však pramení z nedostatku současných programů pro modelování 3D objektů exportovat tento typ křivek do jazyka VRML. Po pokusech s 3D modelem svíček, které jsou pro kostel jakousi nezbytností se mi nepodařilo zdárně tyto modely exportovat do výsledného souboru, jak bych si představoval. Program nejprve tyto křivky převede na soustavu bodů v prostoru a poté každý bod samostatně uloží do souboru. Ve výsledku se tedy nejedná o poměrně jednoduché křivky typu nurbs, viz obr. 4.7, ale o množinu bodů známou jako „indexed faced set“, viz obr. 4.7. Velikost takové svíčky v souboru přesáhla hranici 400kB, což je nežádoucí, především z důvodů datových přenosů mezi koncovým uživatelem a zdrojem scény.



Obrázek 4.7: Objekt vytvořený křivkami nurbs a objekt exportovaný do VRML

4.3 Tvorba galerie a stanovišť

Jak již bylo zmíněno, v kostele se bude nacházet virtuální galerie, viz obr. 4.8, do které budou návštěvníci vstupovat. Budou si tak moci prohlížet obrazy, ať už sami, nebo z předem určených stanovišť.

Umístění obrazů bylo předem konzultováno s paní malířkou Adrianou Skálovou, která se rozhodla, kde bude mít své obrazy vystavené, v jaké velikosti a jaké výšce. Obrazy tedy jsou téměř po všech stěnách kostela a vytvářejí tak postupnou cestu skrze jednotlivá umělecká díla.

V budoucnu se pro další malíře, či kurátory výstavy počítá s přemisťováním obrazů. Nastává zde problém, jak umožnit zanesení změny laikem, který nezná strukturu, souřadnice a další potřebné informace, do wrl souboru. Jako možné řešení je umístění více ploch pro případné zobrazení obrazů, které by se generovali až s pomocí aplikace, pokud by byla tato místa žádoucí.

Realizace galerie bude probíhat automatickým generováním v pozdější části práce, nyní je však důležité obrazy umístit a vytvořit jim předem daná stanoviště. K tomu jsem použil tzv. head up display, který ke zjednodušení určování pozice obrazů a stanovišť ukazuje aktuální přesnou polohu avatara a jeho orientaci v prostoru. Dosáhl jsem tak dobrých a rychlých výsledků.

Obrazy jsou na stěny umísťovány vždy do dřevěných ráků, které mají ale rozdílné parametry. Jedná se především o tloušťku ráku a jeho velikost. Ráky jsou na stěnách umístěny napevno a v budoucnu bude tedy možné si zvolit, na kterou pozici chce uživatel umístit svůj obraz a poté si ve virtuálním kostele prohlédnout, jak to bude vypadat. Ke vkládání ráků jsem použil vytvořený prototyp, kterému měním parametry, jako je textura (url obrazu, který se má v ráku zobrazovat), velikost, pozice či rotace.



Obrázek 4.8: Pohled na levou stěnu galerie

4.4 Optimalizace VRML

Jazyk VRML obecně je značkový jazyk, který umožňuje i mnoho dalších zajímavých funkcí [1]. Ve virtuálním světě využívám možnosti tvorby prototypu k částem, které se ve scéně opakují a zároveň mění nějakou svoji vlastnost. Nejčastěji pozici, velikost či zvolenou texturu. Obecně prototyp používám pro vkládání varhan, které se nacházejí na balkonu uvnitř kostela a stromů, kterých je typicky několik druhů a jsou umístěny různě kolem kostela, jako doplněk pro příjemnější vnímání celého prostředí. Dále prototypy využívám hojně při tvorbě galerie a obrazů, popsané v předchozí kapitole. Dalo by se říci, že bez prototypů se ve VRML neobejdeme, pokud chceme dělat složitější světy.

Z hlediska optimalizace souborů se jedná o elegantní řešení, při němž se zmenšuje velikost souboru, tedy zmenšuje se počet řádků, soubor se více zpřehlední a umožní se mnohonásobné užití jednoho objektu, pouze s jinými parametry.

Optimalizačních hledisek je hned několik a použití prototypu je jedno z nich. Další používanou metodou, jak zjednodušit a zpřehlednit wrl soubory je použití definovaných uzlů – „DEF“ a jejich znovu užití – „USE“. Tuto metodu jsem použil jako řešení u mnoha desítek komponent v kostele a to ať již na samotné stavbě, nebo poté u každého jednotlivého modelu, který scéna obsahuje. Několik příkladů užití DEF a USE lze popsat například na místním modelu varhan, které jsou věrnou kopií varhan originálních. Varhany obecně fungují

na principu proudění vzduchu píšťalami. Mnou modelovaný typ nástroje obsahuje ve třech skříních v každé osm píšťal. Není tedy žádoucí, aby prohlížeč při každém načítání souboru musel dvaatřicetkrát načíst v podstatě jednu a tu samou věc. Z optimalizačního pohledu je nezbytné definovat jednu z těchto osmi píšťal a poté ji sedmkrát užít na jiném místě. Celý soubor se mi podařilo rapidně zmenšit a to platí zejména u složitějších objektů, které mají rozmanitý povrch. Zjednodušení tímto způsobem tak ušetří neuvěřitelně mnoho paměti.

Metodu DEF a USE jsem použil i u ostatních objektů. Nalezneme ji u hrubé stavby kostela, u lavic a jiných objektů. Především bych vyzdvihl zábradlí, které se nachází v přední části kostela a poté na balkoně u točitých schodů. Toto zábradlí se skládá z několika desítek stejných sloupků, spojených horní a dolní deskou. Sloupek sám o sobě představuje poměrně složitou soustavu uspořádaných bodů. Byl zde totiž kladen důraz na co nejvěrohodnější záznam místního originálu. Použití obyčejného exportování bez následné úpravy a optimalizace by z takového modelu vytvořilo několika MB soubor. S využitím metody DEF a USE se však výsledek vešel do 51kB, což je již z hlediska datového přenosu velice přijatelná velikost.

Zmenšování souborů, které přenášíme přes datové linky však nemusí být nutně práce ve wrl souborech. V modelu jsem již předem počítal s několika částmi, které uživatel neuvidí. Tyto plochy, které se následně triangulují, by také zpomalovaly chod virtuálního světa a zbytečně by zvětšovaly nároky na paměť. V mém případě se jedná např. o spodní plochy objektů tvaru válce, ze kterých jsou tvořeny píšťaly či sloupky u zmiňovaného zábradlí. Tyto spodní a vrchní plochy ani nemohou být vidět, není tedy důvod je vůbec modelovat. Pokud tímto způsobem uvažujeme již dopředu, ušetříme tím mnoho paměti a zjednodušíme výpočetní náročnost. S tímto pravidlem virtuální model kostela pracuje.

Optimalizací nemusí být nutně pouze zmenšování souborů a jejich zpřehledňování, jak zde uvádím. V modelu používám i nastavení pro vykreslování odvrácených ploch, tzv. nastavení „solid false“ např. u oken či zdi, která má jen jednu plochu, zato ji potřebujeme vidět z obou stran. Dovolil bych si proto i takovýto zásah nazvat optimalizací, protože tím dokážeme upravit soubor tak, aby fungoval jak potřebujeme a optimalizujeme tím výsledný vzhled celého objektu.

V tomto odstavci dále popisuji sestavování souborů dohromady. Používám přitom metodu include, tedy vkládání jednoho souboru do druhého. V mém případě se svět sestává z mnoha objektů. Tyto objekty nejsou pro přehlednost umístěny v jednom velkém souboru. Vytvořil jsem klasickou stromovou strukturu, která tyto modely sdružuje do skupin a tím optimalizuje přehlednost a jejich seskupení. Sestavování těchto objektů do jedné scény poté vypadá velice jednoduše a výsledkem je přehledný a programátorsky dobře čitelný soubor, který obsahuje odkazy na jednotlivě vkládané objekty. Ty poté mohou být i ve formě prototypu. V tomto hlavním souboru se navíc globálně nastaví i práva uživatele na pohyb, případné aktivace světél či pozadí celého virtuálního světa.

4.5 Technický popis a parametry modelu

Testování modelu

Testování virtuální scény probíhalo v několika prohlížečích, jednalo se o Cortonu 3D [5], který je doporučen jako primární prohlížeč virtuální reality. Druhým prohlížečem byl BS Contact [3], ve kterém byla scéna testována navíc i pro aktivní stereoskopickou projekci. V případě

druhé varianty se zde vyskytly malé potíže s pluginem tohoto programu v prohlížečích, avšak v samotném programu scéna bez problémů fungovala.

Velikosti textur

Velikosti textur se pohybují v rozmezí od 80 x 80 pixelů až do 1024 x 777 pixelů a to ve formátu JPEG, PNG i GIF. Celková paměťová náročnost všech 39 textur je 4,80MB. Velikost 3D modelu je poté pouhých 1,40MB. Velikost textur je způsobena požadovaným dostatečným detailem pro interiér i exteriér kostela, kde jsem použil textury přímo z pořízené fotodokumentace. Kostel je tak totožnou kopií originálu se zachováním všech prvků (malba, typický povrch či barevné provedení stěn). Velikosti jednotlivých textur popisuje tabulka 4.1.

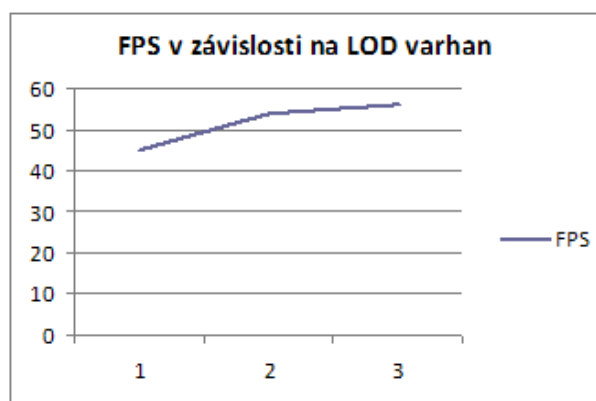
Objekt	Textura	Velikost(kB)	Rozměry
Okap	alu.jpg	10	300x300
Anděl	andel_levy.gif	95	125x192
Strom	bleech2.png	1178	1024x768
Dveře 1. křídlo	dvere.jpg	39	256x768
Dveře 2. křídlo	dvere2.jpg	37	256x768
Dveře levé	dvere_male.jpg	97	512x683
Povrch terénu	grass.jpg	17	160x160
Obraz na oltáři	obraz.png	435	512x683
Textura kazatelny	obrazek.png	242	300x380
Okenní tabule	okno_2.png	138	200x260
Okenní rám	okno_ram.jpg	27	180x245
Podlaha	podlaha.jpg	19	512x540
Podlaha přední	podlaha_predni.jpg	14	373x476
Rímsa	rimsa.jpg	7	480x22
Okolí 1 - 4	rockvalley_(1-4).jpg	45, 41, 26, 32	512x512
Okolí horní pohled	rockvalley_up.jpg	16	512x512
Střecha - taška	roof.jpg	37	225x178
Schody	schod.jpg	32	400x100
Sloup, 2	sloup(2).jpg	19, 20	83x512
Horní sloup, 2	sloup_horni(_1).jpg	13, 12	80x324
Dolné sokl	sokl.jpg	39	650x80
Stěna, stěna 2	stena(2).jpg	48, 43	584x1024
Stěna balkonu	stena_balkon.jpg	22	462x480
Stěna venkovní 1 - 4	stena_venkovni(_2-4).jpg	40, 176, 196, 201	493x600
Strom 2	strom2.png	998	1024x777
Strop	strop.jpg	26	585x563
Dřevo 1	tmave_drevo.jpg	14	200x200
Dřevo 2	tmave_drevo_2.jpg	10	320x200
Zdivo	wall.jpg	15	322x163

Tabulka 4.1: Velikosti a rozměry použitých textur

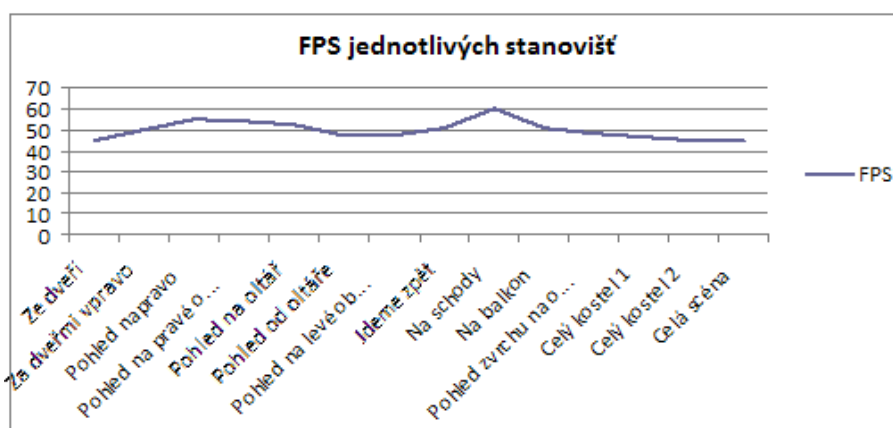
Měření FPS a počty polygonů

Měření FPS a náročnosti scény bylo prováděno na notebooku s následující konfigurací: Procesor: Intel(R) Core(TM)2 Duo CPU T6600 2.20GHz; operační paměť: 4GB; operační systém: Microsoft Windows 7 (x64).

Náročnost scény samozřejmě klesá se vzdáleností, což je způsobeno LOD velice složitého modelu varhan, avšak předpokládám, že uživatel se pohybuje nejvíce v interiéru kostela vzhledem ke galerii a obrazům. Z toho plyne, že jiné modely, než varhany v prvním patře se měnit nemohou, aby uživatel viděl vše v místnosti v dostatečném detailu. Jednotlivá měření popisují grafy 4.9 a 4.10. Počet polygonů ve scéně poté popisuje tabulka 4.2.



Obrázek 4.9: Graf měření FPS v závislosti na LOD varhan



Obrázek 4.10: Graf měření FPS jednotlivých stanovišť

Fotodokumentace

Množství pořízené fotodokumentace čítá 250 fotografií. Fotodokumentaci jsem získal z několika zdrojů, přičemž 107 fotografií kostela jsem pořídil vlastním fotoaparátem za účelem získání

Objekt	Počet polygonů
Terén, střecha	800
Kostel stavba	3400
Varhany	6900
Oltář	2950
Kazatelna	1200
Kříž v kostele	300
Lavice	1700
Doplňky (obrazy, stromy)	1500
Celkem	18750

Tabulka 4.2: Počty polygonů objektů ve scéně

detailů a textur pro modelování a texturování kostela. Ostatní fotografie jsem získal od paní Kláry Dvořákové, která je členkou sdružení, které se o kostel stará a zároveň je také zadavatelkou tohoto projektu.

Časová náročnost

Časová náročnost celého projektu se pohybuje v řádech týdnů a nejlépe ji popisuje tabulka 4.3, ve které jsou zachyceny jak jednotlivé modely, optimalizace i práce s okolní scénou a texturami.

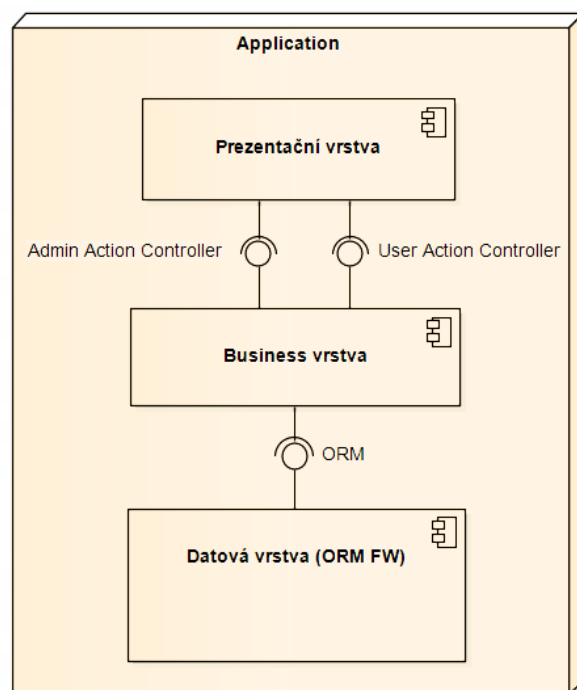
Aktivita	Počet hodin
Modelování kostela	64
Modelování varhan	10
Modelování oltáře	5
Modelování kazatelny	3
Modelování doplňků	10
Modelování scény	5
Optimalizace VRML	12
Práce s VRML	24
Texturování	24
Celkem	157

Tabulka 4.3: Časové rozložení tvorby virtuální scény

4.6 Rozdělení aplikace do tří vrstev

Aplikace je rozdělena do tří vrstev. Toto dělení má jisté výhody, jako oddělení prezentační logiky od výkonného kódu a databáze. Systém je tedy rozdělen do tří vrstev, kterými jsou: prezentační logika, business logika a persistence, neboli datová vrstva, jak popisuje obr.4.11.

Tyto vrstvy mezi sebou vzájemně komunikují, je zde však kladen důraz na rozdělení působnosti každé vrstvy. S výhledem na budoucí využití aplikace se toto rozdělení jeví jako velice důležité a práci usnadňující budoucím programátorům, kteří se tak budou moci soustředit pouze na svoji část.



Obrázek 4.11: Rozdělení aplikace

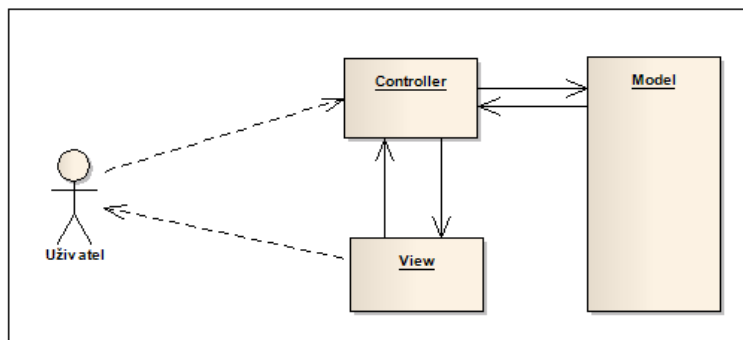
4.7 Použité frameworky a návrhové vzory

Aplikace, která je rozdělena do několika vrstev využije frameworky, které pomáhají ve vytváření aplikace a obsahují určité knihovny pro její stabilní a bezpečnější chod.

Protože aplikace pracuje objektově, je nutné pro ukládání dat použít framework, který dokáže mapovat objekty do relační databáze. Takovýto ORM framework je například Doctrine ORM pro PHP [6], který se nachází nad databázovou abstraktní vrstvou (DBAL). Doctrine nabízí řadu výhod pro programátory, jako je například zápis databázových dotazů v jazyku DQL, což je jakási alternativa k jazyku SQL a zároveň inspirace od frameworků, které pracují s prostředím Java, jako Hibernate s dotazovacím jazykem HQL.

Dalším frameworkem, který jsem v aplikaci využil je MVC Framework 4.12 pro rozdělení aplikace na 3 logické části: model, view a controller. Cílem je rozdělení vrstev tak, aby mohly být upravovány samostatně, a přitom dopad jednotlivých úprav byl na ostatní vrstvy co nejmenší. Model je doménová reprezentace dat a business logiky aplikace. View slouží pro zobrazení dat uživateli, tedy zobrazuje uživatelské rozhraní. Controller obsluhuje reakce na události.

Komunikace mezi uživatelem a aplikací vypadá následovně: Uživatel provede akci v uživatelském rozhraní, controller obdrží akci od uživatele (z objektu) a provede přístup k modelu, který zpracuje získaná data. Poté uživatel opět dostane aktualizovaný model prostřednictvím view a uživatelské rozhraní čeká, než uživatel provede další akci.



Obrázek 4.12: Implementace MVC

Vzor Factory

Návrhový vzor Factory byl implementován z důvodů logického rozdělení formulářů pro jednotlivé uživatelské role. Podle rolí lze navíc rozdělit formuláře podle své konkrétní funkce, o což se v aplikaci dále starají jednotlivé Factory třídy.

Vzor Singleton

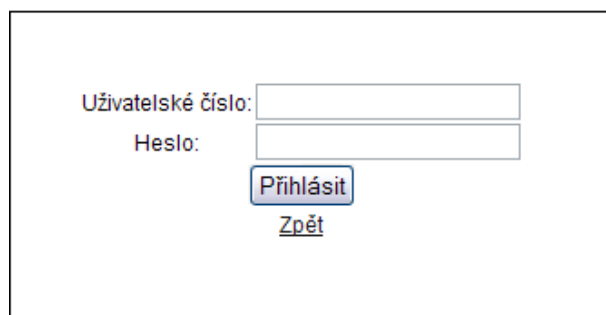
Jako vzor singleton byl implementován Smarty šablonovací systém [13]. Třída SmartyView.php vytváří instanci dané třídy, není však důvod uchovávat více instancí této třídy v paměti, i když je to samozřejmě možné po implementační stránce.

4.8 Popis funkčnosti

Přihlášení uživatele

Uživatel se přihlašuje do aplikace prostřednictvím klasického přihlašovacího formuláře, který zná z ostatních webových aplikací, viz obr. 4.13. Výjimku zde tvoří přihlašování pomocí unikátního čísla uživatele. Toto číslo je automaticky generované systémem a nelze ho změnit. Rozhodl jsem se pro číselnou hodnotu namísto řetězce z důvodu vyššího stupně bezpečnosti a také z důvodu toho, že uživatele může přidávat pouze administrátor. Ten by si musel pokaždé vymyslet uživatelský login, popřípadě se dohodnout s budoucím uživatelem na jeho znění, což by zatěžovalo komunikaci a zdržovalo obě zúčastněné strany. Mnohokrát se navíc administrátor a uživatel nikdy neviděli a ani se neznají.

Uživatel se tedy po vstupu do aplikace dostane na úvodní stránku, zvolí možnost login a přihlásí se do aplikace, kde již může pracovat.



Obrázek 4.13: Formulář přihlášení

Ztráta hesla

Pokud se uživateli přihodí situace, že si nezapamatoval správně své heslo, nebo uživatelské číslo, prostřednictvím kterého se přihlašuje do aplikace, může zvolit možnost Ztráta hesla, která ho přesměruje na stránku s přesnými instrukcemi pro jeho opětovné získání. Na stránce uživatel postupuje dle pokynů, což znamená, vyplní emailovou adresu, na kterou je registrovaný a kterou má nastavenou ve svých osobních údajích jako kontakt. Poté počká na doručení emailu se svým uživatelským číslem a novým heslem, které systém automaticky znovu vygeneruje.

V databázi uživatelů ani na žádném jiném místě totiž aplikace neuchovává z bezpečnostních důvodů hesla uživatelů v otevřeném tvaru. Heslo je vždy uloženo v hashované podobě pomocí hashovacího algoritmu SHA-1. Je tedy nemožné získat zpět jeho původní podobu a proto s musí heslo znovu vygenerovat a nastavit. Uživatel si po přihlášení heslo změní na jakoukoli jemu odpovídající hodnotu podle předepsaných norem.

Uživatelské rozhraní

Uživatelské rozhraní bylo navrženo s ohledem na přehlednost, jasné popisy akcí a elegantní, jednoduchý design. Hlavní menu uživatele popisuje obr. 4.14. Obsahuje tedy sekce pro práci s galerií, dále správu tiketů, osobní údaje a tabulku uživatelů pro snadný kontakt všech uživatelů mezi sebou.

Tikety slouží uživateli jako pojistka, aby žádný jiný uživatel nemohl v době trvání výstavy nastavit svoji galerii jako aktivní a tím desinformovat případné návštěvníky. Vyplnění probíhá v sekci nastavení galerie, kde si uživatel vybere galerii, kterou chce nastavit jako právě probíhající, zvolí příslušný čas a datum, dokdy se výstava koná a poté potvrdí akci. O generování a nastavení galerie se stará třída GallerySetupForm.php, která nejprve zjistí z ostatních tiketů, zdali je možné nastavit galerii a zda nedojde ke kolizi s jinou galerií. Poté vygeneruje VRML soubor s příslušnými daty. Pokud kontrola neprojde, ať již z důvodu nesprávných dat, zadaných uživatelem, či kolize s jinou aktivní výstavou, objeví se uživateli varovná zpráva, dokdy bude muset počkat, než bude moci aktivovat svoji výstavu.

Uživatel má k dispozici změnu svých osobních údajů, které slouží především pro ostatní uživatele, ke zlepšení komunikace mezi správcí, kurátory a dalšími uživateli. Hlavní myšlenkou je realizovat jakýsi otevřenější systém pro uživatele, kde každý bude moci ihned vidět aktuální

data o ostatních. Stejně tak bude moci sledovat i historii všech uživatelů. Tlačítko historie je k dispozici po přihlášení v patičce spolu s tlačítkem pro nápovědu. Do patičky jsem se je rozhodl umístit z důvodu menší priority. V historii má uživatel možnost shlédnout historii vlastních akcí a poté akce všech uživatelů pohromadě. U každé galerie mají uživatelé také možnost shlédnout krátký komentář, či poznámku a to opět realizovanou pomocí nástroje Fancybox [7], kliknutím na odkaz „Informace“.

Sekce nápověda slouží uživatelům pro jakoukoli činnost, se kterou si nevědí rady. Někde se zaseknou, potřebují pomoc. Nápověda jim poskytne kompletní manuál a provede je všemi akcemi od vytvoření galerie až po její odstranění.

Práce s galerií uživateli umožňuje vytvoření, editaci, nastavení galerie a její odstranění. Při vytváření galerie systém vede uživatele jednotlivými kroky, z důvodu rozdělení do sekcí tak, jak jednotlivé pozice pro obrazy v kostele umožňují tato díla rozmístit. Rozdělení do kroků jsem použil také proto, že pokud nastane při validaci odeslaného formuláře s daty od uživatele nějaký problém a formulář se uživateli vrátí, nezobrazí se tak již načtené obrázky a proto by uživatel musel opět všechny své požadované soubory ručně načítat znovu. Tento problém se bohužel nedá vyřešit, neboť protokol neumožňuje z bezpečnostních důvodů přednastavení cesty k souboru. Uživatel tak projde všech 6 kroků vytvoření galerie a poté se mu zobrazí stránka se zprávou o stavu vytvoření. Uživatel má možnost kdykoli operaci přerušit tlačítkem Storno. Mezi jednotlivými kroky se pohybuje tlačítkem „Pokračovat“. Pokud uživatel vykoná neočekávanou akci, musí počítat s tím, že vytváření galerie neproběhlo úspěšně a že tak v galerii budou chybět některé sekce.

Sekce editace galerie, viz obr. 4.14 umožňuje uživateli po výběru ze seznamu svých galerií, editovat obrazy, přemísťovat je mezi sekcemi, mazat je, měnit jim měřítko či je přidávat. Uživatel zde vidí celou galerii najednou a to z toho důvodu, aby nemusel složitě přecházet mezi šesti sekcemi, pokud by si nemohl vzpomenout, kde se nachází jaký obraz. Mazání obrazů probíhá pomocí checkboxu. Ty obrazy, které uživatel zatrhne se smažou, jejich pozice však zůstane zachována. Ve výsledné galerii se však objeví pouze ta díla, která mají nahraný obraz. Uživatel má u každé sekce na výběr z několika možností orientace a velikosti obrazu. Náhledy jednotlivých výsledků jsou mu k dispozici jak v nápovědě, tak přímo v sekci editace v dolní části stránky. Nahrávání nových obrazů probíhá stejným způsobem, jako při vytváření galerie. Uživatel má též možnost náhledu obrázku pomocí nástroje Fancybox [7], po kliknutí na jeho náhled. Tento nástroj je pod licencí GPL a je tedy volně k dispozici pro používání.

Galerie může být uživatelem odstraněna. Tato funkce je jakousi samozřejmostí. Uživatel tedy jednoduše v sekci Smazat vybere ze seznamu zvolenou galerii a jako potvrzení musí zatrnout checkbox pole pro potvrzení, že si je opravdu vědom, co dělá. Po tomto kroku se galerie odstraní. Pokud by byl u galerie nastaven tiket, odstraní se též. Při odstraňování galerie se samozřejmě smaže i složka s externími daty galerie z důvodu velikosti potřebného úložného prostoru.

Administrátor aplikace

Administrátor má k dispozici především akce, které se týkají uživatelů systému. Je jejich správcem, a proto je může odstraňovat a přidávat. Přidat může jak obyčejného uživatele, tak dalšího administrátora. Po vyplnění požadovaných formulářových polí a validaci se novému



Obrázek 4.14: Hlavní menu, sekce editace

uživateli odešle email s informací o jeho přihlašovacím unikátním číslem a heslem. Heslo si může uživatel po přihlášení změnit v sekci osobních údajů.

Pokud se administrátor rozhodne pro odstranění některého uživatele z jakéhokoli důvodu, musí vyplnit textové pole se zprávou pro uživatele, která se mu zašle s informací (důvodem) jeho smazání. Ve zprávě by měl administrátor vždy uvést důvod smazání uživatele, aby nedošlo k nedorozumění.

Další funkci představuje přihlášení se za jakéhokoli uživatele, což administrátor provede v menu „Uživatelé“ přes speciální ikonu. Toto přihlášení řeší případnou neaktivitu uživatele a další problémy. Administrátor tak má možnost v roli uživatele pracovat s jeho daty a jeho galeriemi a vyřešit případné nesrovnalosti.

Data mimo databázi

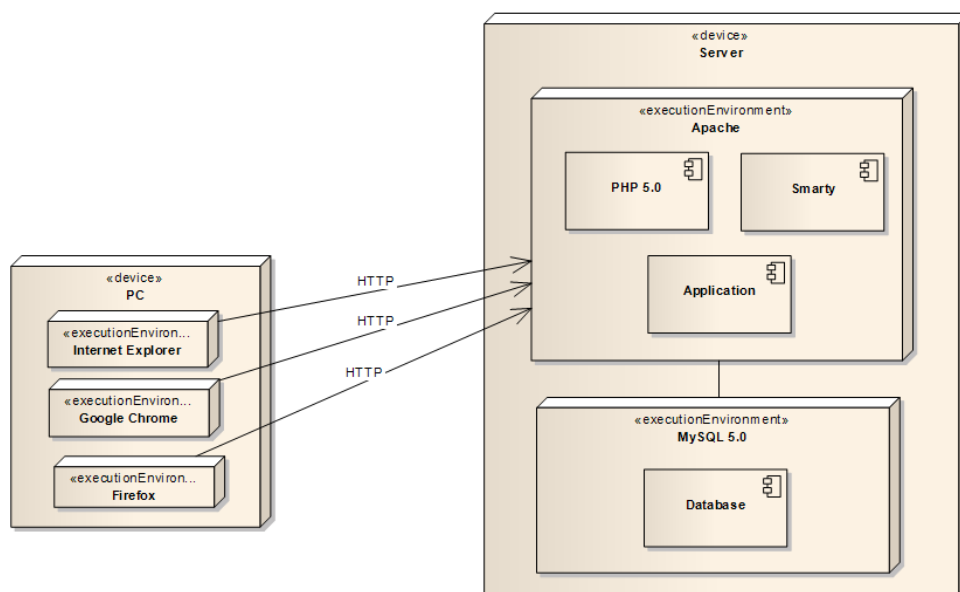
Mimo databázi se ukládají pouze data, které uživatele nahrávají jako obrázky, nebo náhledy svých maleb, které se budou později vkládat do generovaných galerií pro virtuální výstavu. Rozhodl jsem se tak z toho důvodu, aby nebyla databáze zbytečně zatěžována daty a systém nemusel obrázky pokaždé převádět do a z databáze. Do databáze se tak uloží pouze cesta ke zdroji tohoto obrázku. Pro každou novou galerii se vytvoří složka s unikátním názvem obsahující název galerie zvolené uživatelem, podtržítka a identifikační číslo uživatele. Název galerie je také unikátní a uživatel tedy nemůže zadat stejný název u více galerií najednou. Pro klienta s číslem 1 bude složka nést název například **Galerie_1**. V této složce se poté budou nacházet veškeré obrázky, které uživatel nahraje do této galerie. Složka se nachází v

adresáři s názvem „img“, který je umístěn přímo v kořenovém adresáři aplikace. Tyto data jsou svázány s galerií a pokud tedy dojde k jejímu smazání, je důležité aby se smazala i tato složka. Pokud by tomu tak nebylo, nastal by po čase problém se zahlceným datovým prostorem, který je přidělen serverem a aplikace by tak byla naprosto nefunkční.

Při mazání takové složky ovšem nestačí jednoduchý příkaz. Pokud galerie obsahuje obrázky, je nutné nejprve smazat tento obsah složky, aby nenastala chyba neprázdného adresáře a operace se nepřerušila. Po úspěšném smazání složky se tak odstraní záznam v databázi a uvolní se paměť pro další uživatelské výtvoř.

4.9 Diagram nasazení

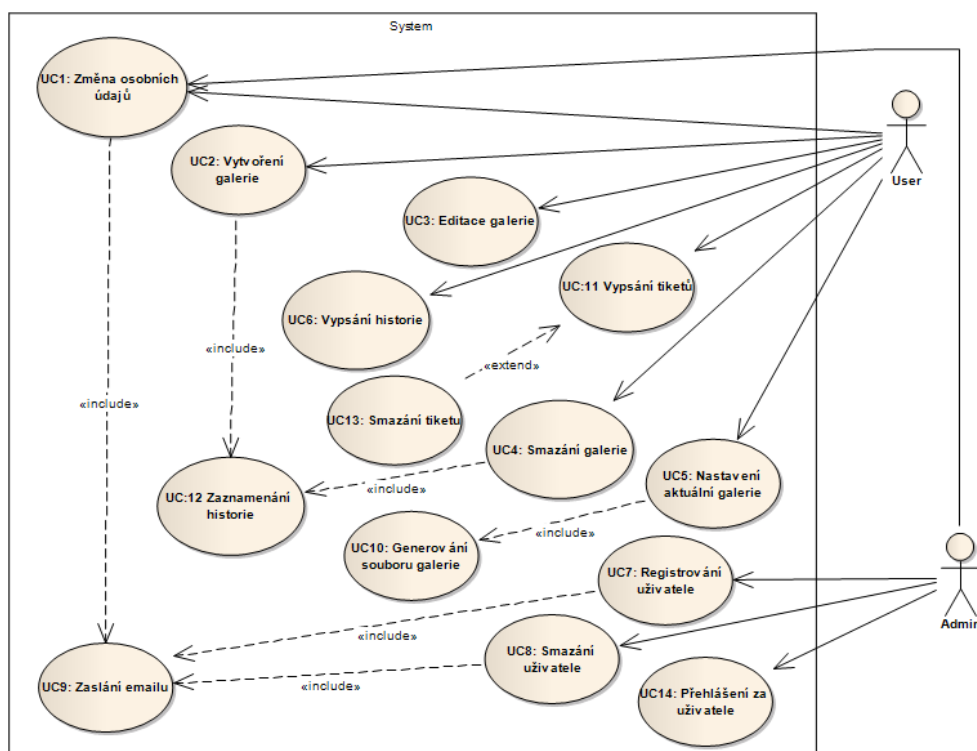
Jak ukazuje diagram nasazení, viz obr. 4.15, aplikace bude přístupná na serveru Apache verze 2.2.14 s databází MySQL verze 5.1.37 a nainstalovaným PHP a Smarty. Na serverové straně se tedy bude vykonávat veškerá business logika a k tenkému klientovi se bude posílat finální HTML/XHTML výstup. Aplikace bude podporovat většinu dnešních moderních prohlížečů, jako jsou Firefox, Safari, Gogole Chrome či Internet Explorer a bude tak přístupná všem operačním systémům.



Obrázek 4.15: Diagram nasazení aplikace

4.10 Diagram případů užití

Na tomto diagramu, viz obr. 4.16 popisují všechny uživatelské role a jejich akce, které mohou být vykonávány. Systém zde obsahuje dvě role, přičemž v budoucnosti není problém s rozšířením systému o role další. Pro tuto aplikaci však postačují tyto dvě. V úvahu se nabízela



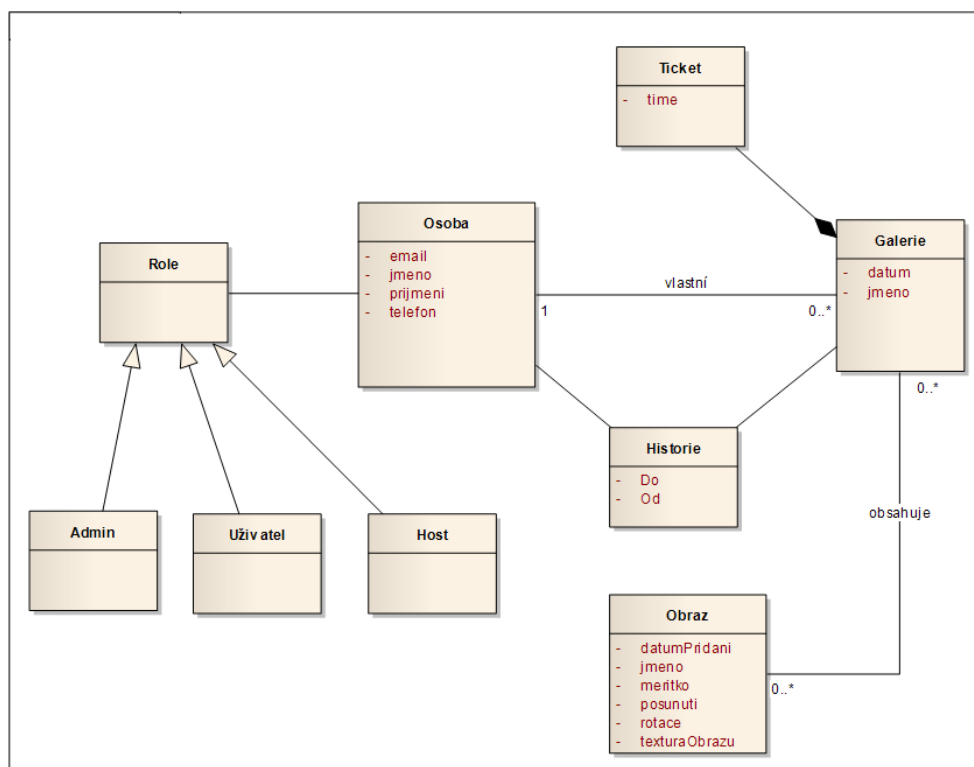
Obrázek 4.16: Diagram případů užití

ještě možnost generalizace rolí a to ve směru od administrátora k uživateli. Administrátor aplikace by tak měl k dispozici všechny akce, které jsou přístupné v systému. Od tohoto jsem však upustil. Důvodů bylo několik. Prvním byl přístup uživatelů, kteří budou aplikaci obsluhovat. Jelikož se bude jednat o laiky v tomto oboru, bylo by pro ně tolik akcí zahlcující a systém by se pro ně stal nepřehledným. Za druhé, vzhledem k přidávání uživatelů a jejich manipulacím jsem se rozhodl roli administrátora oddělit. Pokud tedy bude obě role zastávat jedna osoba, bude muset mít dva uživatelské účty. Pro zmiňovanou osobu bude pohyb v aplikaci jednodušší. Bude tak vždy přesně vědět, jaké akce má v tu dobu k dispozici.

4.11 Diagram tříd

Tento model, viz obr. 4.17 popisuje strukturu aplikace. Jádrem je tedy osoba, která může mít v čase více rolí. V aplikaci se jedná o roli uživatele a administrátora. Pokud není uživatel přihlášen do systému, nemá žádnou roli. Jeho akce jsou však vykonávány prostřednictvím Action Controlleru pro hosta (v diagramu se proto nachází role host). Osoba má jednotné atributy a liší se pouze rolemi v systému. Uživatel, pokud je jeho role „user“, vlastní neomezené množství galerií. Každá galerie však musí mít vlastníka. Mezi osobou a galerií se zaznamenává i historie vazby, a to jejího vytvoření i zrušení (smazání galerie). Galerie může mít 0:N obrazů a 0:N tiketů. Obrácené pořadí vždy dovoluje pouze jednoho vlastníka, tudíž opět vazba 1:N. Obraz má atributy, které uživatel mění v čase, jako měřítko, texturaObrazu

a další a vždy musí nutně patřit nějaké galerii.



Obrázek 4.17: Diagram tříd

4.12 Dokumentace systému

Adresářová struktura

Aplikace obsahuje několik důležitých adresářů obsahujících, jak třídy programované pro konkrétní aplikaci, tak třídy, které jsou zahrnuty do použitého frameworku. Celá struktura vypadá následovně: v kořenovém adresáři aplikace se nachází pouze třída `index.php`, která je volána jako první při přístupu do aplikace a poté třída `createUsers.php`. V tomto hlavním adresáři se dále vyskytují složky `classes`, ve které je obsáhlá veškerá business logika aplikace, složka `doctrine`, což představuje použitý framework a složka `smarty`, šablonovací systém.

Složka `classes` obsahuje dále `Controller`, `Bean`, `Form`, `Factory` a `View`. Každá z těchto složek má své další třídy a složky, se kterými je v aplikaci pracováno. Složka `Controller` obsahuje všechny třídy, které se starají o obsluhování uživatelů či hostů, složka `Bean` poté obsahuje třídy, které jsou naplněny metodami, příslušnými dané oblasti a daným objektům, se kterými je pracováno. `Form` obsahuje veškeré formuláře aplikace. `Factory` je složka pro použitý návrhový vzor a obsahuje třídu `FormFactory.php` s metodou `getForm()` pro přidělování formulářů jednotlivým rolím. Dále tato složka obsahuje adresář `Form`, ve kterém jsou již

konkrétní třídy pro přidělování konkrétních formulářů pro konkrétní role, přidělené ve třídě `FormFactory.php`.

Složka `Doctrine` obsahuje podložku `lib` s dalšími knihovnami. Ty již představují konkrétní práci frameworku. Poté se zde nachází složka `models`, která je velice důležitá, protože právě v ní programátor vytváří a definuje jednotlivé modely, se kterými poté v aplikaci pracuje.

Popis tříd

Třída `index.php` má za úkol nastartování `session`, do které se případně ukládají data pro udržení stavu aplikace, načtení souboru `bootstrap.php`, což je třída, sloužící jako spojení s ORM frameworkem a poté podmínku a jednoduchý přepínač mezi rolemi. Každé roli se tak přiřadí konkrétní `Controller`, který následně obsluhuje akce, které uživateli náleží. Podmínka, která se ve třídě vyskytuje se týká přednastavení databáze a vytvoření prvotních uživatelů. Pokud tedy není nastavena žádná role v `session` ani akce a jedná se tedy o spuštění třídy `index.php`, systém nejprve vytvoří pomocí frameworku databázi, vytvoří základní objekty a ty uloží. Při této první operaci, kterou by měl vykonat administrátor se vytvoří i základní účty uživatelů. Nejdůležitějším účtem je administrátor systému, který může dále uživatele přidávat. Pokud je vše již vytvořeno, podmínka se po neúspěchu odchytí.

Třída `Form.php` se vyskytuje v adresáři `classes` a jedná se o abstraktní třídu, ze které dědí všechny formuláře ve složce `Form`. Obsahuje základní metodu, kterou je `validate`. Po klasickém vyplnění formuláře se tak v příslušném formuláři zavolá tato funkce, která provede kontrolu zadaných formulářových polí a vrátí výsledek, podle kterého je dále rozhodováno. Složka `Form` tak obsahuje všechny potřebné formuláře pro galerii, uživatele, práci s historií či formuláře pro administrátory.

Třída `ActionController` v adresáři `Controller` je rodičovská třída pro ostatní kontrolery, která obsahuje důležité metody pro práci aplikace. Konkrétně `set`y a `get`y pro objekt `Smarty`, formuláře a příslušné akce. Důležitými třídami v aplikaci jsou třídy v adresáři `Controller/Action`. Ty se starají o interakci mezi uživatelem a systémem. Použité třídy jsou konkrétně: `AdminActionController.php`, `GuestActionController.php` a `UserActionController.php` s dostupnými akcemi, jako: `login`, `validate login`, `registerUser`, `setUpGallery` apod. V následujícím kódu se po vyvolání akce „`myTickets`“ vypíše uživateli celkový seznam jím vytvořených tiketů ke galeriím. Výstup takovéto operace, který se předá uživateli je zobrazen na obrázku [4.18](#).

```
case "myTickets": //pokud uživatel vybral tuto akci
    $this->setForm(FormFactory::getForm("myGalleryTickets")); //Factory
    vyrobí nový formulář
    $f = $this->getForm(); // získání vyrobeného formuláře metodou get

    $bean = new GalleryBean(); //zavolání nové beany - obsahující metody
    $result = $bean -> getMyTickets($_SESSION["id"]); //tikety od uživatele
    $this->getSmartyObject()->assign_by_ref('result', $result); //přiřazení
    proměnné do šablonovacího objektu
    $this->setView("myTickets"); // nastavení zobrazení na šablonu "myTickets"
    break; // dále se již nepokračuje
```




Obrázek 4.18: Sekce tiketů uživatele

Třídy, které jsou umístěny ve složce Bean obsahují důležité metody pro získávání a práci s uživatelskými daty. Jedná se o tyto nejdůležitější třídy: GalleryBean.php, UserBean.php a HistoryBean.php. Tyto třídy jsou postupně volány z action kontroleru uživatele, popřípadě administrátora, který vykonává jednotlivé akce a k tomu nutně potřebuje znát a pracovat z takto získanými daty. Mezi nejdůležitější metody patří výpisy uživatelů, práce s galerií, změna textury konkrétního objektu, změna velikosti jeho měřítka či vyhledávání v objektech pomocí některé metody. Na příkladu níže, se v dané metodě po jejím zavolání s příslušnými parametry provede vyhledání požadovaného objektu, změna jeho atributu na požadovanou hodnotu a následné uložení objektu do databáze. Další třídou, která obsahuje nastavení aplikace a do budoucna bude představovat důležitý prvek, je třída SettingsBean.php. Prozatím obsahuje pouze metodu pro získání limitu pro uploadované obrázky, ale v budoucnu se jistě bude rozšiřovat o další metody.

```
// funkce pro obnovení náhledu u obrazu
public function refreshPaintingTexture($id, $texture){
    // nalezne objekt typu Painting s požadovaným id
    $p = Doctrine::getTable('Painting')->find($id);
    // nastaví mu atribut texturaObrazu ze získaného parametru
    $p->texturaObrazu = $texture;
    $p->save(); // uloží objekt
}
```

Metody v ostatních třídách jsou této obdobné, pouze vykonávají jiné požadavky. Některé metody vracejí celá pole, která se poté upravují v příslušném formuláři či kontroleru a vypisují položku po položce. Tímto způsobem se například vypisují všechny seznamy aplikace, či se tak vytváří textové řetězce pro následný zápis do souboru při akci generování galerie.

Třída SmartyView, která se nachází ve složce Classes/View obsluhuje šablonovací systém. Nejprve načte soubor Smarty.class.php, vytvoří novou třídu SmartyView, která dědí vlastnosti

od rodičovské třídy Smarty. Poté načte soubory, potřebné pro správné fungování a vytvoří jednu instanci SmartyView, protože není potřeba vytvářet pokaždé novou instanci vytvořené třídy.

Řešené problémy

Během programování aplikace jsem narazil na problém s výpisem dat od galerií uživatele a to konkrétně tiketů, které se galerií týkají. Pro uživatele je v aplikaci dostupný seznam všech tiketů galerie, se kterými uživatel může dále pracovat. Původně model Ticket.php neobsahoval atributy user_id a gallery_id, které se logicky měly získat propojením vzájemnými vazbami mezi modely. Prostřednictvím frameworku je možné pro spojování tabulek a výpisy takovýchto dat využít tzv. DQL – Doctrine Query Language, který bohužel v tomto případě nepracoval tak, jak jsem v aplikaci potřeboval. Problém jsem vyřešil přidáním atributů, podle kterých lze ve výsledku filtrovat vstup do tabulky, která se uživateli zobrazí, s tím, že vazba mezi galerií a tiketem zůstala zachována pro další operace. Tou je např. mazání galerie, při které se smažou i všechny její tikety, či vytváření nových tiketů.

Definování modelů

Modely definují jak objekty, se kterými pracuji v systému, tak i třídy, které se pomocí frameworku mapují do relační databáze. Definování modelů probíhalo na začátku programování, protože je důležité si po návrhu aplikace a její funkčnosti namodelovat i základní objekty, se kterými se bude pracovat.

Tyto modely jsou vytvořeny ve složce classes/doctrine/models a jedná se o třídy: User.php, Painting.php, Gallery.php, Tickets.php, Settings.php a History.php. Nejdůležitějšími modely jsou první tři třídy. Tyto modely si lze představit jako budoucí tabulky v databázi, se kterými framework pracuje. Každá z těchto tříd dědí ze třídy Doctrine_Record, což je rodičovská třída pro každý model. Každý potomek této třídy obsahuje dvě metody – setTableDefinition() a setUp(). První metoda definuje položky, které model obsahuje a se kterými potřebujeme pracovat. Druhá metoda setUp() slouží pro definování závislostí mezi modely.

Třída User.php, tak obsahuje definice jako jsou: name, surname, id, password, email, role, phoneNumber a má 1:N vazbu k modelu Gallery.php. Tím definujeme, že uživatel může mít N galerií, ve smyslu 0 – nekonečno, na druhou stranu ale každá galerie, která je v systému zanesena nutně musí patřit nějakému uživateli. Eliminuje to problém s pozdějším chaotickým uspořádáním a jakýmsi nepořádkem v aplikaci. Pokud by existovalo N nastavených galerií bez svých vlastníků či kurátorů, nebylo by to pro ostatní uživatele příliš dobré. Na této úrovni při vytváření modelů navíc máme možnost definovat i jiné nastavení atributů, jako jejich datový typ či tzv. mutator, kterým vyjadřujeme požadavek ke kódování příslušného atributu. V aplikaci jsem využil funkci hasMutator k atributu password, které v aplikaci hashují funkcí SHA-1. O toto hashování se poté již programátor nemusí starat. Je nastavené v modelu, se kterým se pracuje a použitý ORM framework již zastane operace typu: „před porovnáním hesla v databázi a získaného hesla od uživatele je nutné na získané heslo nejprve aplikovat hashovou funkci.“ Tyto operace již dále v programování business logiky neřešíme.

Třída Gallery.php popisuje objekt galerie, kterou uživatel během svého působení vytváří a se kterou také dále pracuje. Objekt má atributy: name, date a user_id. Atribut user_id je

důležitý pro použití vztahu mezi galerií a uživatelem. Každá vytvořená galerie tak bude mít tento atribut nastavený podle uživatele, který ji vytvořil a tímto polem se v pohledu níže do databáze bude na uživatele odkazovat. Třída obsahuje v metodě `setUp()` nastavení `hasOne`, které vyjadřuje právě jednu stranu vazby 1:N a poté zde má nastavení `hasMany()`. Galerie totiž obsahuje jak malby, které jsou také typu objekt, tak i tikety, které chrání galerii proti přepsání. Galerie má zajisté více obrazů a může mít i více tiketů. Pokud jednomu tiketu uplyne jeho doba působení, vytvoříme další atd. Případů je mnoho a proto se i k tiketům i k obrazům jasně vztahuje také vazba 1:N vůči galerii. V metodě `setUp()` tak budou integritní omezení `hasMany` pro `Ticket` i pro `Painting`.

Modely `Ticket.php` a `Painting.php` jsou v podstatě stejné, liší se ve struktuře pouze názvy atributů, kterých má každý model několik. V metodě `setUp()` poté musí být nastavení na galerii, ke které tento objekt náleží, tudíž `hasOne` s odkazem na model `Gallery.php`.

Posledním definovaným modelem je model `History`. Z názvu plyne funkce tohoto modelu. Zde zaznamenávám události typu vytvoření či smazání galerie. Popřípadě jakékoli jiné události. Každý záznam je nový objekt, který má nastavené atributy a který je poté uložen prostřednictvím frameworku do databáze. S objektem lze klasicky pracovat, upravovat ho a mazat. V aplikaci ale taková možnost není. Galerie vznikají a zanikají, ale historie se pouze vytváří. Uživatelé tak mají přehled o všech galeriích, které se v aplikaci vyskytují, nebo se zde vyskytovaly.

Práce s modely

S vytvořenými modely lze pracovat jako s jakýmkoli objekty. Výhodou jsou operace typu přístup k provázaným objektům. Z aplikace jsou to například objekty, které obsahuje galerie, či galerie, které patří danému uživateli. Pro příklad bych uvedl práci s vytvořením objektu a jeho provázáním. Příklad níže obsahuje třída s umístěním `classes/Form/Gallery/SetupGalleryForm.php`. Ta má za úkol nastavování aktuální galerie, přiřazování tiketu a generování galerie.

```
$gal = $bean -> getGallery($this->gallery_id);//zjištění galerie, vybrané
uživatelem
$ticket = new Ticket(); // vytvoření objektu podle modelu Ticket
$ticket->time = $time; //přiřazení objektu atribut time jako časové známky
$ticket->date = date("j.n.Y G:i:s", ($time)); //přiřazení data
$ticket->user_id = $_SESSION["id"]; //nastavení pole uživatele ze session
$ticket->name = $gal->name; //nastavení jména galerie
$ticket->save(); //uložení objektu

//nastavení tiketu ke galerii
$gal->Tickets[] = $ticket; //přidání tiketu do pole
$gal->save(); //uložení galerie
```

Důležité je také získávání objektů a jejich výpisy. S objekty se poté dále pracuje a je možno v nich měnit různé vlastnosti. Zpět do úložiště se opět uloží metodou `save()`. Takovýto přístup je používán v celé aplikaci, dalo by se říci ve většině tříd, které s objekty nějakým způsobem pracují. Nejvíce se takového operace s objekty využijí ve třídě `GalleryEditForm.php`, která

obsahuje úpravu uživatelem vytvořené galerie. Každý obraz, který je zde vytvořen je objekt, kterému uživatel mění jeho atributy. S využitím frameworku se tak práce stává mnohem jednodušší. Pro změněný objekt, kterému se při validaci formuláře nastaví atributy se použije metoda `save()` a framework se postará o ostatní operace a proces uložení. Práce mezi business logikou a úložištěm tak případně jiné vrstvě, což usnadní a zrychlí práci při programování funkčnosti aplikace.

Udržování stavu aplikace

Stav aplikace lze udržovat několika způsoby. Nejbezpečnější je samozřejmě práce se session, kterou aplikace využívá. Po přihlášení uživatele se tak do session nastaví několik proměnných, se kterými se pracuje. Jedná se především o informace o uživateli, jako jeho jméno, příjmení, role a uživatelské číslo – id.

Session využívám i v jiných případech, jako je proces vytváření galerie. Při vstupu do sekce vytvoření se nejprve odstraní ze session proměnná `gallery`. Je tomu tak z toho důvodu, aby se nemíchaly dvě galerie do jednoho objektu, pokud uživatel nečekaně akci ukončí. Po vyplnění prvního kroku galerie se do session nastaví právě proměnná nově vytvořeného a uloženého objektu, do kterého se v dalších krocích pouze přidávají objekty typu `Painting` – malba. Přidávání probíhá takovým způsobem, že se nejprve zjistí ze session identifikační číslo galerie, zavolá se třída s metodami pro galerii označená jako `GalleryBean.php` a dále se v této třídě zavolá metoda `getGallery($id)`. Výsledná hodnota se vrátí a s objektem poté pracujeme stejně jako v předchozích příkladech. Výsledný kód poté vypadá následovně:

```
$gallery = $x->getGallery($_SESSION["gallery"]); //získání objektu(gal.)
$gallery->Paintings[] = $p1; //přidání obrázu ke galerii
$gallery->Paintings[] = $p2;
...
$gallery->save(); //uložení objektu
```

Při akci odhlášení uživatele se ze session hodnoty odstraní. Pokud není role nastavena, defaultní hodnota pro rozeznání uživatele je účet hosta – `guest`. Host poté může pracovat prostřednictvím svého controlleru.

Bezpečnost

Bezpečnost aplikace je řešena na několika vrstvách z níž jednotlivé vrstvy by se daly rozdělit na databázovou a formulářovou.

Vzhledem k možným útokům na dnešní aplikace, ať již plánovaným či způsobených pouhou nepozorností či zvědavostí uživatele, jsem se rozhodl tento obecný bezpečnostní problém řešit tak, že všechny formuláře, jakékoli odesílání dat a komunikace mezi uživatelem a aplikací je zajištěna formuláři, které posílají data metodou `POST`. V aplikaci nikde nelze data zaslat metodou `GET`, která se obvykle přenáší v url – adrese v prohlížeči. Jediná výjimka je samozřejmě samotná adresa url s názvem akce, kterou chce uživatel vykonat. I toto je však dostatečně zabezpečeno a pokud není uživatel přihlášen a nemá nastaveny potřebné informace v session, kontroler mu akci nepovolí.

Dalším důležitým bezpečnostním opatřením je zabezpečení vstupních dat proti SQL Injection. Tento problém však již v základu řeší zvolený framework, který neumožní z databáze útočníkem požadovaná data získat. V aplikaci navíc neuchovávám žádná důležitá data o jejích uživateli. Předpokládám však, že i kontaktní údaje by mohly být zajímavou informací. Hesla v aplikaci pro bezpečnost nejsou ukládána. Každý objekt typu uživatel má své heslo, které je však do databáze ukládáno v hashované formě, konkrétně funkcí SHA-1, která je jednosměrná a nelze proto z databáze získat otevřená hesla i v případě, že by došlo ke získání databázových dat.

Pokud uživatel požaduje změnu údajů, musí zároveň s novými údaji vyplnit i své stávající heslo, což je další bezpečnostní prvek aplikace. Pokud požaduje změnu stávajícího hesla, vyplní navíc i pole pro nové heslo a vypíše i jeho duplikát do pole „kontrola“, viz obr. 4.19. Bez znalosti současného hesla tedy nemohou být údaje jakkoli změněny.

The screenshot shows a web application interface for changing personal data. At the top, there is a navigation bar with links: Domů, Galerie: Vytvořit - Editovat - Nastavit - Smazat, Tikety, Osobní údaje, and Uživatelé. The main heading is 'Změna osobních údajů'. Below it, there are several input fields: 'Jméno:' (Testovací), 'Email:' (dan.sram@seznam.cz), 'Telefon:' (000 123 456), and 'Heslo:' (masked with dots). There is also a 'Přijmení:' field (Ucet). Below these, there is a section for changing the password: 'Pro změnu hesla vyplňte nové heslo:', 'Nové Heslo:' (masked), and 'Kontrola:' (masked). A 'Změnit' button is located below the password fields. At the bottom, there is a 'Zpět' link. The footer contains copyright information: '© Copyright 2010 Daniel Šrám - sramdani@fel.cvut.cz' and links for 'Historie' and 'Nápověda'.

Obrázek 4.19: Sekce pro změnu údajů uživatele

Každý formulář v aplikaci má funkci validace, která provádí ověření vstupních dat, zadaných uživatelem. Mezi nejdůležitější vstupní data můžeme zařadit email, datum a čas, jméno, či heslo. Všechny tyto položky musí být určitého formátu, musí být platné a musí dosahovat příslušné délky, což platí především pro zadávané heslo. Validace dat se vztahuje i na soubory, které předpokládám, že bude uživatel v této aplikaci velice často nahrávat. V případě textur pro obrazy, či jiná umělecká díla jsem se rozhodl uživateli povolit vstup pouze ve formátu jpeg a gif. Z důvodu kompatibility mezi nahranými formáty a následným zobrazením v jazyce VRML. Velikosti jednotlivých nahrávaných textur jsem se rozhodl nechat na uživateli, je zde však určitý strop velikosti nahrávaného souboru, prvotně nastavený na 3MB. Tato velikost může být samozřejmě měněna. Změna je nyní možná pouze prostřednictvím zásahu do databáze, nebo do souboru Data.php v položce „Settings ID 1“ a to proto, že u aplikace není třeba sekce pro rozsáhlejší nastavování parametrů aplikace.

Uživatel při vytváření galerie musí projít několika kroky. Toto je elegantní způsob, jak umožnit uživateli základní znalost rozdělení sekcí galerie a zároveň, jak uživateli postupně předkládat jednotlivé požadavky na nahrávání souborů. Vzhledem k omezení, jaké jazyk

HTML má, není možné přednastavit do vstupního formulářového pole cestu k souboru. Bezpečnostní nastavení to nedovolují a proto, pokud by uživatel nastavil mnoho cest k souborům a udělal jednu chybu, všechny přednastavené cesty by musel opět zadávat ručně znovu. Proto je výhodnější postupovat po menších krocích a zjednodušit tak uživateli práci.

4.13 Propojení s 3D scénou

Aplikace byla vytvořena pro správu virtuální výstavy. Hlavním cílem tedy je spravovat a vytvářet soubory a galerie, ze kterých se výsledná 3D výstava bude vytvářet. Scéna ve virtuální realitě ovšem pracuje poněkud rozdílně, než tomu je u programovacího jazyku, a to s uzly a body, které mají souřadnice a další atributy. Aplikace tedy umožňuje uživateli práci poněkud jiným - jednodušším způsobem. Nepracuje již s 3D objekty a jeho souřadnicemi, ale s objekty v programátorském slova smyslu. Vytvořením uživatelské galerie aplikace tyto objekty sama vytvoří a uživatel jim poté může přiřazovat atributy, například obraz (jeho náhled), velikost apod. Objekt se po nastavení galerie, jako výsledné výstavy vygeneruje podle prototypu. 3D model obrazu je tedy namodelován předem a do virtuálního světa se vkládá jako prototyp, u kterého má uživatel možnost měnit jeho orientaci, velikost, či zobrazovanou malbu.

Aplikace po nastavení galerie sama vygeneruje soubor, na který je odkazováno přímo z toho souboru, jenž obsahuje všechny části 3D scény. Výsledek, generovaný v aplikaci, je pouze jeden soubor, který obsahuje celou galerii a je tedy nežádoucí, aby jej různí uživatelé aplikace přepisovali svými virtuálními výstavami v době již probíhající výstavy, či jiné akce. Pokud tedy jako uživatel potřebujeme, aby naše výstava zůstala okolnímu světu přístupná po nějakou dobu, vytvoříme tiket, nastavíme datum jeho platnosti a potvrdíme. Po této akci máme jistotu, že námi nastavená galerie a vygenerovaná výstava nebude nikým přepsána po celou dobu zvoleného trvání.

Struktura výsledného souboru je již v jazyce VRML. Obsahuje definování příslušného prototypu se všemi nastaveními a odkazy na externí prototyp, který představuje skutečný 3D model, vytvořený v modelovacím programu. Poté již obsahuje potomky tohoto uzlu, což jsou uživatelem vytvořené obrazy na jednotlivých stěnách virtuální galerie. Celkový počet prototypů v souboru je čtyři. Jedná se o rozdílný prototyp pro obraz vpravo za dveřmi, prototyp na pravé a levé stěně, horní i dolní.

Pokud se uživatel rozhodne nahrát v aplikaci do galerie obrazy pouze na některá místa, bylo by nežádoucí, aby se vždy vygenerovala celá databáze možných umístění, i s těmi obrazy, které nejsou nastavené a uživatel je v reálné výstavě mít nechce. Proto se vždy generují jen takové objekty, které mají nastavené textury obrazu a uživatel tedy vidí jejich náhled. Ostatní objekty jsou ignorovány, aby nedošlo k efektu prázdných ráků na zdech virtuální výstavy.

4.14 Uživatelské role

Aplikace umožňuje uživatelům rozdělení práv podle uživatelských rolí. Jedná se o rozdělení na dvě základní úrovně: administrátor a uživatel. Obě role mají přístupné pouze svoje akce, přičemž některé akce jsou přístupné oběma rolím. Aplikace nabízí tyto uživatelské role, aby

logicky oddělila práce s uživateli a práce s výstavami a galeriemi. Pro uživatele se tak aplikace stává přehlednější a lépe pochopitelná.

Administrátor se stará o správu uživatelů daného systému. Má právo uživatele odstranit, musí však uvést důvod, proč tak jedná. Ten je poté zaslán uživateli na jeho uvedenou emailovou adresu. Má také právo uživatele přidat, a to jak roli běžného uživatele, tak dalšího administrátora.

Uživatel systému má k dispozici podstatně více akcí, než administrátor. Může pracovat s galeriemi, přidělovat jim tikety či je mazat. Uživatelská role byla do systému zanesena pro konečnou správu galerií. Jedná se tedy o hlavního uživatele, který s aplikací aktivně pracuje. Pokud se vyskytne jakýkoli problém, má možnost kontaktovat administrátora, či ostatní uživatele. Jejich detaily si zobrazí v přehledu všech uživatelů aplikace.

Další uživatelské role systém nenabízí. Byly by to role redundantní a nepotřebné. Aplikace je však otevřená pro další vývoj a není tedy problém do systému jednoduše zanést další uživatelské role a následně s nimi pracovat a programovat jim další rozšířenou funkčnost.

4.15 Grafický návrh aplikace

Grafický návrh byl vytvořen tak, aby co nejjednodušší formou umožňoval uživatelům orientaci v celé aplikaci. Hlavní stránka, viz obr. 4.20 obsahuje především fotografie z digitálního kostela, zatímco další odvětví již obsahují důležitá nastavení a umožňují uživatelské akce. Celý návrh je tak přehledný a nenáročný pro uživatele. Byl také kladen důraz na jakousi vkusnost a „čistotu“ designu. Nejsou tedy použity rozmanité barvy, toolbary a nadpisy. Vše je tvořeno s ohledem na tyto prvky, jak ukazuje obr. 4.20.

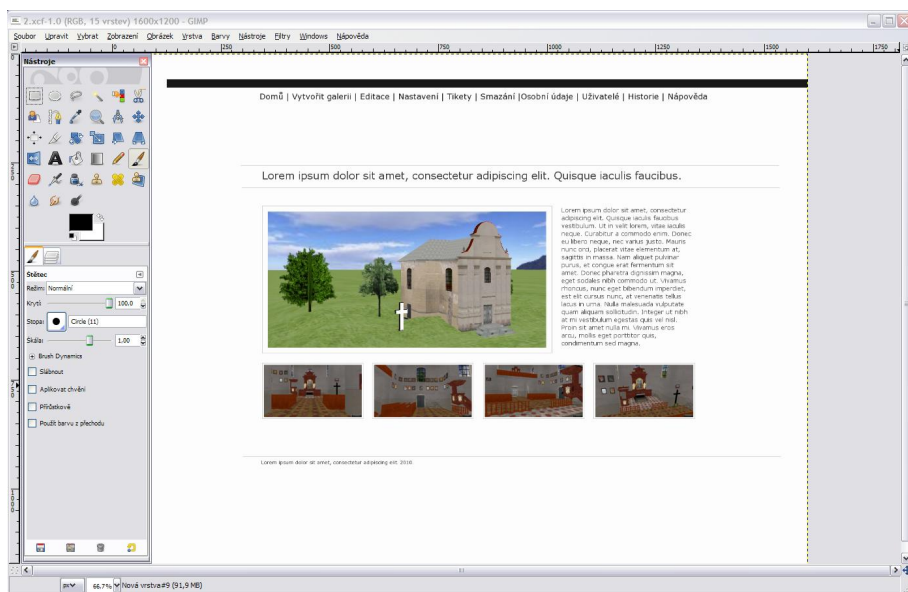
Návrh byl tvořen z vrstev, které se vzájemně překrývají. První vrstvou je pozadí, které jsem zvolil bílé, na něj jsou poté umístěny vrstvy s obrázky kostela, jeho vnitřku a celkového pohledu na scénu. V dalších několika vrstvách se nacházejí texty a linky, které od sebe oddělují důležité části (hlavička, nadpis, hlavní část, patička).

Tento zpracovaný návrh bylo poté třeba sloučit a rozřezat podle potřeb pro sestavení designu pro aplikaci. Ten je tvořen pomocí kaskádových stylů. Veškerý grafický obsah aplikace, vyjma několika obrázků v sekci nastavení a editace, je tedy oddělen do samostatného souboru a zpřehledňuje tak prezentační logiku aplikace.

Návrh byl vytvářen v open-source programu Gimp 2 [8].

4.16 Model versus realita

Tento odstavec slouží především pro porovnání virtuální reality a reálného objektu ve formě kostela. Při vytváření virtuálního modelu byl kladen důraz na vytvoření co nejvěrnějšího modelu, připomínajícího opravdový reálný kostel, nacházející se u Líšné na Zbirožsku. Následující fotografie tedy slouží k porovnání reálné stavby a její digitální podoby.



Obrázek 4.20: Grafický návrh aplikace



Obrázek 4.21: Porovnání předního pohledu na objekt



Obrázek 4.22: Porovnání z pohledu od vstupních dveří kostela



Obrázek 4.23: Porovnání při pohledu z balkonu na oltář

Kapitola 5

Testování

Aplikace byla testována na všech zmíněných prohlížečích, to je: Internet Explorer verze 8, Mozilla Firefox verze 3.5.9, Google Chrome a Apple Safari. Aplikace by měla však být funkční i v ostatních prohlížečích. Dále testování proběhlo na operačních systémech: MS Windows verze XP a verze 7, MAC OS X a Linux. Model byl textován ve VRML prohlížečích Cortona 3D [5] ve verzích 5.1 a 6 a BS Contact [3] verze 7.2. Ve druhém prohlížeči proběhlo i testování pomocí 3D brýlí a speciálního projektoru, díky němuž se uživatel mohl v kostele projít s realistickým 3D vjemem. Pro testování jsem využil osob, které nemají s vývojem webových aplikací ani virtuální reality žádné zkušenosti a jedná se tedy o laiky v tomto oboru. Cílem bylo namodelovat situaci, běžnou pro budoucí uživatele systému ve všech možných detailech.

Testování aplikace

Pro ověření, zda-li aplikace správně funguje jsem se rozhodl udělat uživatelské testování s přesně daným scénářem, podle kterého jsem sledoval uživatelské akce a vyhodnocoval chyby a případné změny v aplikační logice. Při testování byl virtuální kostel obrazy zaplňován, aby se zjistilo, jaký vliv má jeho úplné zaplnění na datový přenos. Datový přenos není z hlediska modelu velký a proto největší vliv měly právě nahrávané obrázky, přičemž se testovaly obrázky velké (jednotky MB) a malé (jednotky kB). Z tohoto testování poté vyplynulo několik doporučení a omezení na nahrávané obrázky, které se na chází v příručce tvůrce virtuálních výstav.

Scénář testování

Po dohodě s administrátorem Vám byl vytvořen uživatelský účet v aplikaci pro virtuální výstavy. Pomocí informací, které jste dostali na emailovou adresu se přihlaste do aplikace, vytvořte svoji galerii s libovolným názvem a naplňte ji libovolnými obrázky (JPEG nebo GIF). Plnění proveďte tak, aby v každé sekci byly minimálně tři obrázky různé velikosti. Po vytvoření galerie vstupte do sekce Editace, kde galerii upravte. Budete upravovat měřítko obrazu, změnu jeho pozice či změnu obrazu, protože jste se rozhodli pro změnu. Poslední obraz ve vaší galerii vymažte. Po dokončení editace nastavte galerii do libovolného data konání výstavy. Vytvořenou výstavu si prohlédněte ve svém prohlížeči. V sekci Tikety poté

zkontrolujte, zda li jste konečné datum konání zadali opravdu tak, jak jste chtěli a poté tiket smažte. V dalším kroku změňte své osobní údaje (jakékoli) a nahlédněte do historie akcí, zda li přibyl řádek o Vaší galerii a galerii, kterou jste vytvořili smažte. Na konec se bezpečně z aplikace odhlaste.

Zjištěné chyby

Při testování dělalo uživatelům potíží zjistit detaily, např. které obrázky mohou nahrávat a jaké velikosti. Tyto informace se již nacházejí v nápovědě. Bohužel málokterý uživatel nápovědy využije. Rozhodl jsem se proto do uživatelské příručky umístit informace o formátech, podporovaných aplikací a dalších detailech, které uživateli poskytnou dodatečné informace. Uživatelé dále neměli s aplikací žádné větší problémy. V sekci editace je k dispozici tabulka velikostí, kde se uživatelé dozvědí informace o jednotlivých měřítkách pro každou sekci, co které znamená a jak bude výsledná obraz velký, popřípadě, jak velký obrázek nahrát, aby bylo zobrazení co nejlepší.

Tabulka uživatelů

V následující tabulce č.5.1 se nachází seznam všech testujících uživatelů. Tabulka popisuje zjištěné chyby a problémy, čas strávený při práci s aplikací a stav zjištěných chyb. Ty byly po testování odstraněny.

Číslo	Čas (h)	Zjištěné nedostatky	Opraveno?
1	2	gramatické nedostatky, špatný popis tiketů	ANO
2	1,5	chybí odkaz na plugin, proč je potřeba	ANO
3	1	gramatické chyby, chybí odkaz na prohlížeč	ANO
4	1,5	nefunkční formulář na ztrátu hesla, gram. chyby	ANO

Tabulka 5.1: Tabulka testování aplikace uživateli

Kapitola 6

Závěr

Na závěr bych rád připomněl přínos celého projektu, který je určen pro občanské sdružení na záchranu kostela u Líšné. Projekt je tedy primárně určen ku pomoci tomuto sdružení s cílem propagace kostela široké veřejnosti i státním institucím, které spravují veřejné finance pro památky a obnovu památkových objektů. Virtuální model má umožnit prohlížení reálného kostela v digitální podobě odkudkoli, kde mají uživatelé možnost přistoupit k internetu. Celý tento model i s příloženou aplikací bude přesunut na webové stránky kostela [9].

Dalším přínosem tohoto projektu je otevřít dveře příznivcům umění. Ze zadání bakalářské práce je patrné, že v kostele bude umístěna virtuální výstava se skutečnými uměleckými díly (jejich digitální podobou), která kostel oživí a dodá mu přátelskou a živoucí atmosféru. Kostel je nejen místo pro bohoslužby a kázání, ale je to i místo pro setkávání lidí. Propagací pomocí internetu a virtuální reality může dosažení tohoto cíle velice pomoci.

V bakalářské práci jsem nejprve vytvářel virtuální model kostela, který se mi podařilo vymodelovat jako poměrně věrnou digitální podobu reálného objektu. Virtuální model je funkční a přitom není extrémně složitý. Počet polygonů výsledné scény je 18 750, přičemž plně obsazený kostel všemi obrazy má na toto číslo zanedbatelný vliv. Vypočítat takovou scénu tak bez problémů dokáže běžný počítač poměrně rychle a i hodnoty FPS jsou v rozmezí kolem čísla 60. Datový přenos závisí z velké části na uživateli, tedy jím nahrávaných obrázcích do galerie, protože samotný 3D model má i s texturami pouze 5,9MB.

Dále jsem se v bakalářské práci zaměřil na programování aplikace, která v tomto virtuálním kostele umožňuje interaktivně pracovat s obrazy a vytvářet tak výstavy. Tato aplikace je zcela funkční a byla programována s ohledem na uživatele a je tedy uživatelsky nanáročná. Poskytuje rozdělení uživatelů podle rolí na administrátora a běžného uživatele, který může spravovat své galerie a poté z nich vytvářet výstavy. Výsledné propojení 3D modelu, aplikace a společné nasazení na server se podařilo a aplikace je tak připravena na běžný provoz.

Možná vylepšení do budoucna

Nové funkce

Jedním z možných vylepšení projektu v jeho budoucnosti jsou funkce pro automatické rozpoznávání velikosti a orientace uživatelem nahrávaných obrázků. Systém by mohl vše

automaticky nastavit podle vstupu, který mu uživatel poskytne. Uživatel by do systému zanesl pouze přesné velikosti svých obrazů, které budou v kostele umístěny a nahrál by příslušný náhled obrazu. Systém by tyto jednotky přepočítal a automaticky nastavil velikost prototypu obrazu a jeho textury v podobě nahraného obrázku, čímž by se daly odstranit některé funkce v nynější editační sekci, jako je právě volba orientace a velikosti obrazu.

Virtuální ovládací panel

Virtuální ovládací panel, neboli tzv. "Head Up Display"(HUD), by mohl sloužit kurátorovi výstavy i jejím návštěvníkům pro nastavení různých typů a intenzit osvětlení, případně i denní doby celé virtuální scény. Návštěvníkovi by se virtuální kostel zobrazoval v souladu s denní dobou, např.: večer by byla všechna světla rozsvícena a kostel nasvícen a naopak. Prostřednictvím HUD panelu by návštěvník mohl přepínat různé volby zobrazení, aby si tak i při své večerní návštěvě mohl virtuální kostel prohlédnout za denního světla.

Sekce nastavení

Aplikace v nynější podobě postrádá sekci pro nastavování jejích parametrů, jako je např.: volba velikosti nahraného obrázku. V této době není třeba jiného nastavování, avšak později, při dalších vylepšování aplikace v budoucnu bude třeba této sekce, ve které se budou dále shromažďovat i ostatní důležité informace pro chod aplikace. Mezi nejdůležitější parametry bude patřit např.: volba parametrů pro automatickou změnu velikostí obrázků atd.

Použité vývojové prostředí

Pro vývoj aplikace bylo použito vývojové prostředí NetBeans IDE 6.7.1 [10] s přidaným pluginem verze 1.6.2 (zdroj: NetBeans IDE 6.7.1 (Build 200907230233)) pro vývoj aplikací v jazyku PHP. V tomto prostředí byla vyvíjena jak business logika, tak i prezentační logika a stylování výsledného zobrazení.

Vývoj probíhal na lokálním serveru XAMPP 1.7.3 for Windows [18] s nainstalovaným:

- Apache 2.2.14 (IPv6 enabled) + OpenSSL 0.9.8l
- MySQL 5.1.41
- PHP 5.3.1
- phpMyAdmin 3.2.4

Zdroje

Zdrojů, ze kterých jsem při mé práci čerpal, je mnoho a mezi ty nejdůležitější patří samotná návštěva kostela a pořízení fotografií, které později sloužily pro tvorbu textur. Samotné fotografie ovšem nejsou postačujícím materiálem a proto bylo třeba použít stavební nákresy budovy. Bohužel kostel je natolik starý, že originální plány již neexistují. Byly mi však

zapůjčeny plány novodobé – současné, podle kterých jsem 3D model vytvářel. Měření délek stěn, rozlohy a detailů jsem provedl v budově osobně. Vznikla tak věrohodná a téměř přesná kopie stávajícího kostela u Líšné.

Zdrojem mnoha důležitých informací byly také konzultace se zadavatelkou práce, paní Klárou Dvořákovou a s paní malířkou, Adrianou Skálovou, která se projektu účastní a vytváří díla do plánované galerie.

Je nezbytné zmínit internetové stránky kostela [\[9\]](#), na kterých jsem našel fotografie z okolí stavby, koncertů a mší, které tvorbě velice pomohly.

Literatura

- [1] Žára Jiří. Laskavý průvodce virtuálními světy : Vrm1 97. 1999.
<http://www.cgg.cvut.cz/LaskavyPruvodce/>, stav z 22.12.2009.
- [2] Blender.
<http://www.blender.org/download/get-blender/>, stav z 1.3.2010.
- [3] Bitmanagement Software GmbH - BS Contact.
<http://www.bitmanagement.com/en/products/interactive-3d-clients/bs-contact>,
stav z 22.12.2009.
- [4] Bitmanagement Software GmbH - Blender Exporter.
[http://bitmanagement.com/developer/contact/
sdk-prev/doc/tools/blenderexporter/index.html](http://bitmanagement.com/developer/contact/sdk-prev/doc/tools/blenderexporter/index.html), stav z 22.12.2009.
- [5] Cortona 3D.
<http://www.parallelgraphics.com/products/cortona3d/>, stav z 1.3.2010.
- [6] Doctrine ORM.
<http://www.doctrine-project.org/>, stav z 15.12.2009.
- [7] Fancybox online - webové stránky.
<http://www.fancybox.net>, stav z 19.3.2010.
- [8] Gimp 2.
<http://www.gimp.org/windows/>, stav z 1.3.2010.
- [9] Kostel sv. petra a pavla - webové stránky.
<http://petrapavel.czweb.org>, stav z 11.4.2010.
- [10] NetBeans IDE 6.7.1.
<http://netbeans.org>, stav z 20.11.2009.
- [11] Textury – aluminium.
<http://www.brightshine.co.nz/images/swatches/ALUMINIUM38.jpg>, stav ze
22.12.2009.
- [12] Ametyst 7 – Rockvalley Env Box.
<http://amethyst7.gotdoofed.com/rockyvalleypan.htm>, stav z 22.12.2009.

- [13] Smarty šablonovací systém.
<http://www.smarty.net>, stav z 19.2.2010.
- [14] Textura – střecha.
http://farm4.static.flickr.com/3508/3943796118_34ba662838.jpg, stav ze 22.12.2009.
- [15] Textury – z7server.com.
<http://textury.z7server.com/nSave2.php?id=10>, stav ze 22.12.2009.
- [16] Blender for Architecture.
<http://blender-archi.tuxfamily.org/Textures#Trees>, stav ze 22.12.2009.
- [17] VrmI Pad.
<http://www.parallelgraphics.com/products/vrmlpad/>, stav z 1.3.2010.
- [18] XAMPP 1.73 for Windows.
<http://www.apachefriends.org/en/xampp.html>, stav z 15.12.2009.

Příloha A

Příručka pro návštěvníky virtuálního kostela

A.1 Návštěva kostela

Pro návštěvu virtuálního kostela budete potřebovat nainstalovaný přídatný modul neboli plugin do svého internetového prohlížeče. Pro tento virtuální svět se doporučuje plugin Cortona [5], na kterém je primárně scéna testována a je tak zaručena její bezproblémová funkčnost. Použít samozřejmě lze i jiný prohlížeč, např.: BS Contanct. Po nainstalování takového pluginu již můžete přes odkaz, který Vám bude sdělen příslušnou osobou, vstoupit do virtuálního světa – 3D modelu kostela Sv. Petra a Pavla u Líšné. Samotné internetové prohlížeče, ve kterých byla scéna stestována jsou: Internet Explorer verze 8, Mozilla Firefox verze 3.5.9, Google Chrome a Apple Safari. Aplikace by měla však být funkční i v ostatních prohlížečích.

Jako návštěvníci virtuálního kostela, viz obr. A.1 máte možnost pohybovat se po objektu, sledovat ho a prohlížet jeho jednotlivé části. Model je vytvořen podle originálních autentických plánů a fotografií. Stěny kostela, podlaha a okrasné prvky, včetně 3D modelů oltáře, kazatelny, varhan a ostatních objektů jsou věrnou kopií místních unikátů, které tvoří nedílnou součást kostela.

Ve virtuálním kostele máte možnost pohybovat se několika způsoby. Prvním, nepříliš plynulým způsobem, je pohyb pomocí směrových kláves. Takto se můžete projít kolem kostela a shlédnout ho ze všech stran. Uvnitř je doporučeno pohybovat se způsobem druhým, a to pomocí tzv. Viewpoits, neboli prohlížečích stanovišť. Tyto stanoviště jsou pevně určeny a uživatel se mezi nimi přepíná - prochází pomocí kláves „Page up“ a „Page down“. Takto ho 3D scéna provede postupně všemi galeriemi a poté návštěvníka přesune na balkon, do prvního patra. Na konec se ukáže návštěvníkovi pohled na kostel ze předu a pohled na kompletní scénu i s okolím. Pokud se stane, že se kdekoli ve scéně dostanete do situace, že bude pomocí kláves obtížné se vrátit či jít vpřed, použijte opět klávesu pro pohyb mezi stanovišti. Takto vás scéna přesune zpět na výchozí pozici některého ze stanovišť a vy se budete opět moci bez problémů pohybovat.



Obrázek A.1: Kompletní virtuální scéna kostela

A.2 Aktivní prvky

Ve virtuálním kostele se nachází několik aktivních prvků pro obohacení scény a dodání realističtějšího efektu uživateli. Pokud se po scéně budete pohybovat pomocí směrových kláves na Vaší klávesnici, je potřeba vědět, že vstupní dveře kostela se otevírají kliknutím na dveře levým tlačítkem myši. Dveře se tak na krátkou dobu otevřou a návštěvník může projít dovnitř. Otevírání se tak provádí stejným způsobem zevnitř i zvenku kostela.

Dalším aktivním prvkem jsou místní varhany, ke kterým se návštěvník dostane vstupem do prvního patra – na balkon. Dojít tam lze pomocí kláves, avšak schody jsou modelovány tak, jak tomu je v opravdovém kostele a tudíž se po nich špatně chodí. Návštěvník tedy nejlépe dosáhne cíle pomocí kláves „Page up“ a „Page down“, které ho po průchodu kostelem přesunou plynule na balkon. Tam již může manipulovat s varhany. Návštěvník má možnost přesouvat stoličku, zavírat kryt kláves, či si na varhany zahrát. Vše se ovládá pomocí myši a levého tlačítka.

Příloha B

Příručka pro tvůrce virtuálních výstav

B.1 Úvod

V tomto dokumentu Vám bude popsán průchod všemi dostupnými akcemi, od přihlášení až po nastavení své virtuální výstavy. Prostřednictvím virtuálního kostela a výstav máte možnost oslovit širokou veřejnost a upoutat zájem o vaši expozici, aniž by lidé museli cestovat přímo na místo skutečného kostela. Máte tak možnost zpřístupnit tuto památku mnohem širší veřejnosti rychleji a efektivněji.

Aplikace Vám, jako uživatelům umožňuje vytvářet, upravovat a kompletně spravovat svoje virtuální galerie a nastavovat je jako výstavy. Veškeré informace pro vaši práci naleznete v dokumentaci níže.

Před samotnou prací s aplikací by se měl každý uživatel s virtuálním kostelem seznámit, projít si ho a získat tak povědomí o umístění jednotlivých sekcí obrazů, které bude později obsazovat svými vybranými malbami. Do virtuálního kostela lze vstoupit přímo z aplikace, či prostřednictvím jiných webových stránek. Název stránek Vám rovněž sdělí zodpovědná osoba. Po objektu kostela se doporučuje pohybovat prostřednictvím kláves "Page UP" a "Page Down", čímž ovládáte jednotlivá pozorovací místa. Přechody mezi těmito místy jsou plynulé. Samozřejmostí je pohyb směrovými klávesami.

Mnoho úspěchů při vykonávání kurátorské práce Vám přeje Daniel Šrám (tvůrce aplikace).

B.2 Uživatelské role

Aplikace umožňuje uživatelům rozdělení práv podle uživatelských rolí. Jedná se o rozdělení na dvě základní úrovně: administrátor a uživatel. Obě role mají přístupné pouze svoje akce, přičemž některé akce jsou přístupné oběma rolím. Aplikace nabízí tyto uživatelské role, aby logicky oddělila práce s uživateli a práce s výstavami a galeriemi. Pro uživatele se tak aplikace stává přehlednější a lépe pochopitelná.

Administrátor se stará o správu uživatelů daného systému. Má právo uživatele odstranit, musí však uvést důvod, proč tak jedná. Ten je poté zaslán uživateli na jeho uvedenou emailovou adresu. Má také právo uživatele přidat, a to jak roli běžného uživatele, tak dalšího administrátora.

Uživatel systému má k dispozici podstatně více akcí, než administrátor. Může pracovat s galeriemi, přidělovat jim tikety či je mazat. Uživatelská role byla do systému zanesena pro konečnou správu galerií. Jedná se tedy o hlavního uživatele, který s aplikací aktivně pracuje. Pokud se vyskytne jakýkoli problém, má možnost kontaktovat administrátora, či ostatní uživatele. Jejich detaily si zobrazí v přehledu všech uživatelů aplikace.

B.3 Uživatel

B.3.1 Přihlášení uživatele

Jako uživatel - tvůrce, či kurátor, máte k dispozici své identifikační číslo a heslo. Tyto informace zadáváte klasickým způsobem do formuláře na první - přihlašovací stránce aplikace. Pokud tyto údaje doposud nemáte, kontaktujte prosím zodpovědnou osobu, která Vám tuto příručku poskytla.

Pokud se Vám stane, že své uživatelské číslo, nebo heslo ztratíte, máte možnost využít formuláře, ke kterému se dostanete z přihlašovací stránky. Po vyplnění příslušných informací Vám bude na registrovaný email odesláno nové heslo, pomocí kterého se přihlásíte. Toto heslo si později můžete změnit v sekci "Osobní údaje". Po prvním přihlášení, prosím, vyplňte v sekci "Osobní údaje" vaše dodatečné informace.

B.3.2 Vytvoření nové galerie

Pro vytvoření nové virtuální galerie nejprve stisknete v menu odkaz "Vytvořit". Dostanete se do prvního kroku vytváření. V každém ze 6 kroků máte možnost na příslušné oddělení v kostele nahrát své obrazy. Pokud se tedy rozhodnete, že v kroku č.1 nahrajete obraz na pozici 2 (Pozice vidíte na obrázku), vyvoláte tlačítkem výběr obrázků, vyberete obrázek v podporovaném formátu a potvrdíte. V prvním kroku musíte zadat jméno vaší galerie (či název výstavy).

POZOR! Podporované formáty aplikace jsou pouze JPEG a GIF o MAXIMÁLNÍ velikosti 3MB. Tuto původní hodnotu může administrátor aplikace změnit, o čemž by jste měli být včas informováni. Doporučený poměr stran pro nahrávané obrázky je 4:3 či 3:4 (na výšku). V případě úzkých či širokých obrázků doporučuji přidat paspartu, aby byl doporučený formát zachován a vaše obrazy se zobrazily ve virtuální výstavě správně. Dále také není třeba vkládat do galerie obrázky s příliš vysokým rozlišením. V galerii bohatě stačí rozlišení 600x800 pixelů, což sníží nároky na přenos dat mezi uloženou výstavou a návštěvníkem a přitom se výsledné zobrazení nepozná.

Při vytváření galerie postupujete krok za krokem. Mezi jednotlivými kroky se pohybujete tlačítkem "Pokračovat". Mezi kroky se nelze vracet zpět! Pokud tedy zjistíte, že potřebujete obrázek změnit, provedete tak v sekci "Editace galerie". Pokud si přejete vytváření galerie přerušit, stisknete v jakémkoli kroku tlačítko "Zrušit". Galerie se tak nevytvoří. Při vytvoření galerie se Vám vytvoří všechny možnosti, kam můžete obrazy nahrávat. To poté můžete uskutečnit v sekci editace. Pozice, které nemají žádný obraz se v kostele nezobrazí!

Obrázek B.1: Vytvoření nové galerie

B.3.3 Editace galerie

Po vytvoření galerie je třeba nastavit vlastnosti jednotlivých obrazů. V sekci "Editace" si vyberte galerii, kterou chcete měnit a potvrďte tlačítkem. Poté se vám zobrazí samotná galerie s náhledy obrázků a dalšími možnostmi.

Změna obrazu: Pro změnu obrazu nějaké pozice stiskněte u příslušné pozice tlačítko, vyberte obrázek a potvrďte. Zobrazí se Vám cesta k souboru, který jste vybrali. Pro dokončení editace stiskněte tlačítko "Editovat galerii", které je umístěno u každého oddělení dole.

Změna velikosti: Protože vámi nahraný obraz může být jinak veliký, či může být jiné orientace, než je výchozí nastavení, je třeba tuto orientaci a velikost nastavit správně. Z tabulky velikostí (ve spodní části obrazovky) si vyberte tu velikost příslušné sekce, které se Váš obraz nejvíce blíží a podle toho poté vyberte přímo u obrazu jeho velikost z výběrového menu, viz obr. B.2. Celou editaci potvrdíte tlačítkem "Editovat galerii", které je umístěno u každého oddělení dole.

Smazání obrazu: Pokud chcete obraz odstranit (již se na pozici nehodí, chcete jeho pozici změnit), zaškrtněte u obrazu políčko s názvem "Smazat". Po celkovém potvrzení tlačítkem "Editovat galerii", které je umístěno u každého oddělení dole, se obraz odstraní.

Velký náhled: Při editaci máte možnost velkého náhledu obrazu. Pokud kliknete na obrázek, zobrazí se Vám velký náhled.



Obrázek B.2: Editace galerie

B.3.4 Nastavení výstavy

Po vytvoření galerie a její editaci je nutný ještě jeden krok k tomu, aby se Vaše galerie zobrazila ve virtuálním kostele a byla tak dokončena virtuální výstava. Je tedy nutné nastavit některou z Vašich galerií jako aktuální výstavu. Tím se vygeneruje vaše výstava a nahraje se do virtuálního kostela, kde si ji poté budete moci prohlédnout. Pro nastavení galerie nejprve vstupte do sekce "Nastavení". Zde vyberte podle tabulky požadovanou galerii, kterou chcete do kostela umístit. Poté vyplňte údaje o datu a času, do kterého bude výstava aktuální. Akci potvrďte tlačítkem. Pokud je v tu dobu nastavena již jiná výstava, zobrazí se Vám červené hlášení. To znamená, že již jiný uživatel (kurátor výstavy) má v kostele svoji výstavu a nelze ji tedy přehrát tou vaší. Musíte se s uživatelem domluvit, nebo počkat, až její platnost vyprší.

uživatel Dan Šrám, uživatel | Odhlásit

Domů | Galerie: Vytvořit - Editovat - **Nastavit** - Smazat | Tikety | Osobní údaje | Uživatelé

Nastavení galerie

Číslo galerie	Název galerie	Datum vytvoření
5	Galerie	26.4.2010 13:46:19

* Vyberte galerii:

* Datum: 2010

Čas:

© Copyright 2010 Daniel Šrám - sramdani@fel.cvut.cz

[Historie](#) | [Nápověda](#)

Obrázek B.3: Nastavení výstavy

B.3.5 Prohlídka výstavy

Po nastavení výstavy je možné se přesunout do virtuálního kostela a výstavu prohlížet. Vstoupit do virtuálního kostela lze několika způsoby. Po nastavení vaší výstavy se na stránce objeví potvrzení o úspěšném nastavení a odkaz, prostřednictvím kterého se můžete do kostela podívat. Další možností, jak do kostela vstoupit je přímo z hlavní stránky aplikace, přes odkaz: Vstup do virtuálního kostela (viz obrázek nápovědy).

Vítejte v sekci galerií 3D modelu kostela u Líšné



Vítejte v aplikaci pro správu virtuálních galerií v kostele Sv. Petra a Pavla u Líšné. Prostřednictvím virtuálního kostela a výstav máte možnost oslovit širokou veřejnost a upoutat zájem o vaši expozici, aniž by lidé museli cestovat přímo na místo skutečného kostela. Máte tak možnost zpřístupnit tuto památku mnohem širší veřejnosti rychleji a efektivněji.

Tato aplikace Vám, jako uživateli, umožňuje vytvářet, upravovat a kompletně spravovat svoje virtuální galerie. Veškeré nástroje pro vaši práci naleznete v hlavním menu aplikace (nachází se nahoře). Pro větší bezpečnost doporučujeme, abyste se po každé, po ukončení práce, z galerie bezpečně odhlásili.

Veškeré informace o fungování aplikace, jak vytvořit galerii a jak ji spravovat, naleznete v **nápovědě**. Odkaz na nápovědu se nachází v pravém dolním rohu obrazovky.

Mnoho úspěchů při vykonávání kurátorské práce Vám přeje Daniel Šrám (tvůrce aplikace).






[VSTUP do virtuálního kostela](#)

Obrázek B.4: Prohlídka výstavy

B.3.6 Smazání galerie

Smazat můžete libovolnou ze svých galerií. V sekci "Smazat" si nejprve vyberete z tabulky vašich galerií a níže poté vyberete tu, kterou požadujete smazat. Zatrhnete políčko, ve kterém souhlasíte, že jste si opravdu jisti, že galerii nenávratně smažete a akci potvrdíte. Při smazání galerie se tak smažou i všechny tikety a obrazy s galerií spojené. Doporučujeme staré galerie mazat, uvolníte tak místo pro ostatní, nové galerie.



Obrázek B.5: Smazání galerie

B.3.7 Tikety - co jsou, k čemu slouží

Tikety slouží pro správu nastavené galerie. Každá galerie, kterou jste vytvořili, může mít několik tiketů. Vyplněný a aktivní tiket představuje ochranu Vámi nastavené aktuální výstavy proti jejímu odstranění, nebo přepsání některou jinou výstavou od jiného uživatele. Nastavená galerie (obsahující aktuální aktivní tiket) je aktuální výstava, kterou můžete vidět ve virtuálním kostele. Sekce "Tikety" Vám tedy umožňuje spravovat tikety od jednotlivých galerií, které jste dříve vytvořili. Pokud máte nastavenou nějakou výstavu a chcete ji změnit, musíte nejprve v sekci "Tikety" vymazat příslušný tiket, který výstavu chrání proti přepsání. Vymazání tiketu probíhá tak, že z tabulky vyberete, který tiket chcete smazat, zvolíte ho níže (vyberete ho), zatrhnete potvrzení, že jste si vědomi této operace a potvrdíte. Smazat můžete jakýkoli vlastní tiket.

B.3.8 Osobní údaje

Pokud si přejete změnit své osobní údaje. Klikněte na položku v menu "Osobní údaje". Poté uvidíte před-vyplněná pole se stávajícími údaji. Pokud si některý záznam přejete změnit, přepište ho na Vámi požadovanou hodnotu. Pro úspěšné dokončení změny je potřeba vyplnit Vaše aktuální heslo do systému.

Pokud si přejete změnit heslo, vyplňte své stávající heslo, poté do kolonky "Nové heslo" a "Kontrola" vepište Vámi zvolené nové heslo a potvrďte.

B.3.9 Uživatelé

Do sekce uživatelé se dostanete prostřednictvím odkazu v menu "Uživatelé". Zde máte možnost vidět seznam všech registrovaných uživatelů aplikace, včetně administrátorů. Toto místo slouží pro rychlé zjištění informací, pokud se potřebujete spojit s ostatními uživateli.

B.3.10 Náповěda

Do sekce historie se vstupuje prostřednictvím odkazu v pravém spodním rohu obrazovky. V této sekci se Vám zobrazí nápověda pro tvorbu výstavy, podobná té, kterou právě čtete.

B.3.11 Historie

Do sekce historie se vstupuje prostřednictvím odkazu v pravém spodním rohu obrazovky. Historie obsahuje informace o vytvořených a smazaných galeriích po celou dobu fungování této aplikace.

B.4 Administrátor

B.4.1 Přihlášení administrátora

Jako uživatel - administrátor, máte k dispozici své identifikační číslo a heslo. Tyto informace zadáváte klasickým způsobem do formuláře na první - přihlašovací stránce aplikace. Pokud tyto údaje doposud nemáte, kontaktujte prosím zodpovědnou osobu, která Vám tuto příručku poskytla. Pokud jste v aplikaci prvním administrátorem, máte již vytvořený účet a Vaše identifikační číslo je 2, heslo: admin. Po prvním přihlášení, prosím, vyplňte v sekci "Osobní údaje" vaše dodatečné informace a změňte si heslo.

Pokud se Vám stane, že své uživatelské číslo, nebo heslo ztratíte, máte možnost využít formuláře, ke kterému se dostanete z přihlašovací stránky. Po vyplnění příslušných informací Vám bude na registrovaný email odesláno nové heslo, pomocí kterého se přihlásíte. Toto heslo si později můžete změnit v sekci "Osobní údaje".

B.4.2 Registrace nového uživatele

Pokud si přejete zaregistrovat nového živatele, klikněte na položku v menu "Registrace". Vyberte, jakou roli bude uživatel mít (běžný uživatel / administrátor). Dále vyplňte požadované údaje a potvrďte akci. Pokud se ve formuláři vyskytla chyba, zobrazí se Vám červeně její popis. Chybu opravte a odešlete formulář znovu.

B.4.3 Smazání uživatele

Pokud si přejete uživatele smazat, klikněte v menu na položku "Smazat uživatele". Vyberte ze seznamu uživatelů, koho si přejete ze systému odebrat a vyplňte důvod, proč se tomu tak děje. Akci potvrďte. Poté se na email odstraněné osoby odešle informace o jejím odebrání ze systému i s uvedeným důvodem.

B.4.4 Osobní údaje

Pokud si přejete změnit své osobní údaje. Klikněte na položku v menu "Osobní údaje". Poté uvidíte před-vyplněná pole se stávajícími údaji. Pokud si některý záznam přejete změnit, přepište ho na Vámi požadovanou hodnotu. Pro úspěšné dokončení změny je potřeba vyplnit Vaše aktuální heslo do systému.

Pokud si přejete změnit heslo, vyplňte své stávající heslo, poté do kolonky "Nové heslo" a "Kontrola" vepište Vámi zvolené nové heslo a potvrďte.

B.4.5 Uživatelé

Do sekce uživatelé se dostanete prostřednictvím odkazu v menu "Uživatelé". Zde máte možnost vidět seznam všech registrovaných uživatelů aplikace, včetně administrátorů. Toto místo slouží pro rychlé zjištění informací, pokud se potřebujete spojit s ostatními uživateli. Zde máte možnost se přihlásit za jakéhokoli uživatele. Tato funkce slouží pro případ, že by některý uživatel blokoval výstavu, nebyl k zastížení, či by nebyl schopen činnosti z jakéhokoli důvodu. Jako administrátor získáte přihlášením za uživatele jeho práva a v aplikaci se tak pohybujete pod jeho uživatelským účtem. Přihlášení se za uživatele administrátor provede stiskem ikony s obrázkem u požadovaného uživatele. Poté bude přihlášen jako vybraný uživatel. Pro návrat do administrační sekce je třeba se odhlásit a přihásit se opět jako administrátor.

B.4.6 Náповěda

Do sekce historie se vstupuje prostřednictvím odkazu v pravém spodním rohu obrazovky. V této sekci se Vám zobrazí náповěda pro tvorbu výstavy, podobná té, kterou právě čtete.

B.4.7 Historie

Do sekce historie se vstupuje prostřednictvím odkazu v pravém spodním rohu obrazovky. Historie obsahuje informace o vytvořených a smazaných galeriích po celou dobu fungování této aplikace.

Příloha C

Instalační příručka

Pro správnou funkčnost aplikace je třeba mít k dispozici hosting na serveru s nainstalovaným Apache. Dále budete potřebovat databázi MySQL, která by měla být již součástí Vámi vybraného hostingu. V případě nejasností se obraťte na správce hostingové služby.

Do přidělené složky (obvykle pře protokol FTP) nahrajte do kořenového adresáře složku s aplikací (získáte rozbalením přiloženého archivu zip, ve kterém se nachází kompletní aplikace vč. virtuálního modelu). Složce "img" musíte nastavit práva zápisu na 777. Pokud nevíte, jak tento krok realizovat, zeptejte se svého administrátora. V případě, že používáte Total Commander, klikněte na složku a v horním menu zvolte "změna atributů". Zde zaškrtněte políčka: čtení, zápis, vykonání ve všech skupinách.

Na webovém serveru si vytvořte databázi. V případě, že ji již máte vytvořenou pokračujte dalším krokem.

V souboru, který se nachází ve složce ./doctrine/bootstrap.php změňte následující údaje:

```
$dsn = 'mysql:dbname=NAZEV DATABASE;host=ADRESA';  
$user = 'JMENO UZIVATELE';  
$password = 'HESLO';
```

Soubor uložte.

Nyní již můžete zadat do prohlížeče internetovou adresu. Při prvním spuštění si aplikace vytvoří tabulky v databázi a vytvoří i základní uživatelské účty:

- Účet administrátora - už. číslo 2, heslo: admin
- Účet uživatele - už. číslo 3, heslo: test

V administrátorské sekci můžete registrovat ostatní uživatele i administrátory.

Hesla doporučuji ihned ZMĚNIT!!!

Pokud heslo zapomenete, máte ho možnost získat přes odkaz na úvodní stránce s přihlášením.

To je vše, můžete začít pracovat s aplikací.

Příloha D

Seznam použitých zkratk

3D Three-Dimensional

VRML Virtual Reality Modeling Language

X3D Extensible 3D

LOD Level Of Detail

FPS Frames Per Second

HUD Head-up Display

GIF Graphics Interchange Format

JPEG The Joint Photographics Experts Group

PNG Portable Network Graphics

HTML HyperText Markup Language

PHP Hypertext Preprocessor

FTP File Transfer Protocol

CPU Central Processing Unit

ORM Object-relational mapping

MVC Model-view-controller

DQL Doctrine Query Language

HQL Hibernate Query Language

DBAL Database Abstraction Layer

GPL General Public License

Příloha E

Obsah příloženého CD

1. Instalace.txt - postup instalace aplikace
2. README.txt - popis, co se nachází ve které složce, hw nároky na model a aplikaci
3. text - obsahuje bakalářskou práci ve formátu pdf
 - (a) BP.pdf
4. aplikace - zdrojové soubory aplikace a virtuálního modelu
 - (a) src.zip
 - (b) model-src.zip
5. prirucky - příručky pro tvůrce a návštěvníky virtuálního kostela
 - (a) Prirucka pro navstevniky virtualniho kostela.pdf
 - (b) Prirucka pro tvurce virtualnich galerii v kostele Sv. Petra a Pavla u Lisne.pdf
6. data - data související s BP
 - (a) data.zip
 - (b) fotografie.zip - fotografie kostela
7. html - dokumentace aplikace - výstup z programu PHP Documentor
 - (a) dokumentace.zip