

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ



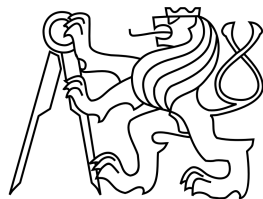
BAKALÁŘSKÁ PRÁCE

Via Sancta - tvorba informačního systému
o poutní cestě

Praha, 2011

Ondřej Pospíšil

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ
obor Geoinformatika



BAKALÁŘSKÁ PRÁCE

**Via Sancta - tvorba informačního systému
o poutní cestě**

Via Sancta - information system of pilgrim way

Vedoucí práce: Ing. Jindřich Hodač, Ph.D.

Katedra mapování a kartografie

Praha, 2011

Ondřej Pospíšil

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci na téma „Via Sancta - tvorba informačního systému o poutní cestě“ vypracoval samostatně a použil jsem pouze podklady uvedené na konci práce v oddílu „Literatura“.

V Praze dne _____

podpis

Poděkování

Děkuji zde především vedoucímu bakalářské práce Ing. Jindřichovi Hodačovi, Ph.D., dále všem zúčastněným, kteří se podíleli na projektu a kteří mně pomohli získat informace potřebné k psaní této bakalářské práce. A to hlavně informace o samotné poutní cestě a jejích kaplích. Jmenovitě děkuji Michaelle Valentové z Ústavu pro ekopolitiku, o.p.s., prof. PhDr. Ing. Janu Roytovi, Olze Jankovcové. Děkuji také Národnímu památkovému ústavu (NPÚ) a znalci historie Prahy 9 a Svaté cesty panu Miroslavu Kurandovi za možnost bez výjimky použít informace z databáze NPÚ, resp. z knihy[7]. Dále děkuji za spolupráci Petrovi Vaverkovi, který zpracoval bakalářskou práci také související s poutní cestou do Staré Boleslavi. V neposlední řadě také děkuji za trpělivost a nápady Majce a všem příbuzným a přátelům kteří mě podporovali.

Abstrakt

Tématem této bakalářské práce je tvorba internetového informačního systému o poutní cestě do Staré Boleslavi zvané „Via Sancta“. Cílem je tento informační systém vytvořit, veřejně zpřístupnit a vycházet při tom z nápadů od památkářů a lidí, kterých se Via Sancta týká. Dále naplnit systém daty získanými z různých, i vlastních zdrojů.

Webová aplikace je napsána v jazyce PHP a využívá databázi MySQL. Navíc je v systému vytvořena jednoduchá interaktivní mapa v rozhraní Google Maps JavaScript API. Internetový informační systém umožňuje přihlásit se registrovaným uživatelům a administrátorovi celý systém spravovat. Daty byla databáze naplněna jen z části, ale bylo vytvořeno rozhraní pro vkládání nových dat, včetně fotografií.

Klíčová slova

informační systém, poutní cesta, Via Sancta, kaple, internet, databáze, PHP, Google Maps

Abstract

The subject of this thesis is a creation of an internet information system of pilgrim way to Stará Boleslav called „Via Sancta“. The first goal of this thesis is to create an information system and to publish it. The information system should be based on the ideas of conservationist and people who are related to the Via Sancta. The second goal is to fill the system with content obtained from different resources (also own).

Web Applications is implement in PHP and database system MySQL is used. In addition, in the system is created a simple interactive map in the Google Maps JavaScript API interface. The Internet information system allows registered users to log in and the administrator to manage the system. The database has been filled with content only partially, but interface was created for entering new data and photos.

Keywords

information system, pilgrim way, Via Sancta, chapel, internet, database, PHP, Google Maps

vložit originální zadání

(kvůli správnému číslování stránek)

Obsah

1	Úvod	13
2	Historie a současný stav Svaté cesty	17
2.1	Historie	17
2.1.1	Vznik a původ Svaté cesty, svatováclavský kult	17
2.1.2	Mariánský kult, Palladium a jezuité	19
2.1.3	Vznik kaplí	20
2.2	Podoba kaplí	21
2.3	Svatá cesta v mapách	23
2.4	Stručný popis a polohy jednotlivých kaplí	24
3	Informační systém z pohledu uživatele	31
3.1	Popis systému	31
3.1.1	Běžný návštěvník stránek	32
3.1.2	Možnosti registrovaného uživatele	36
3.1.3	Možnosti administrátora	41
3.2	Data	43
4	Informační systém z pohledu vývojáře	45
4.1	Způsob zpracování	45
4.1.1	Počítačové jazyky	45
4.1.2	Programy, software	46
4.2	Architektura	47
4.3	Využití databáze	49
4.3.1	MySQL	49
4.3.2	Schéma databáze	49
4.4	Popis jednotlivých skriptů	53
4.4.1	PHP skripty	53
4.4.1.1	Zahrnované skripty	54
4.4.1.2	kaplicky.php	58
4.4.1.3	detail_kaple.php	59
4.4.1.4	reg.php	61
4.4.1.5	login.php	61
4.4.1.6	logout.php	61
4.4.1.7	insert.php	62

4.4.1.8	foto_insert.php	62
4.4.1.9	foto_zmena.php	67
4.4.1.10	mazani_fotografie.php	67
4.4.1.11	smazfoto.php	68
4.4.1.12	ucet.php	68
4.4.1.13	moje_sql.php	68
4.4.1.14	sprava.php	69
4.4.2	JavaScript	69
4.4.3	Ostatní kódy	73
4.5	Využití Google Maps	74
5	Závěr	77
	Literatura	81
A	Obsah příloženého CD	I

Kapitola 1

Úvod

Pojem Via Sancta, neboli Svatá cesta (omezíme-li se jen na tu z Prahy do Staré Boleslavi) můžeme vnímat ve více různých významech.

Prvním je samotná poutní cesta, nehmotná, symbolická, ale po staletí využívaná ke každoročním poutím. Samotné propojení Prahy (tedy Pražské katedrály) se Starou Boleslaví můžeme sledovat již v počátku české státnosti, tedy v 10. století. Teprve později začaly vznikat poutě do Staré Boleslavi, spojené nejprve se svatováclavským kultem, poté i s mariánským kultem a s Palladiem země české. Takové poutě vedly jednou ze Staroměstského náměstí, jednou z Pražského hradu od katedrály svatého Víta (dnes i Václava a Vojtěcha), jednou od sochy svatého Jana Nepomuckého, mučedníka a jednoho z českých zemských patronů, která je na Karlově mostě a jednou například od Lorety. Vše záviselo na okolnostech poutě a souviselo s katolickou symbolikou. Cíl této poutě byl vždy společný - mariánský kostel Nanebevzetí Panny Marie ve Staré Boleslavi, kde je Palladium uloženo. Trasa této poutě se také různila.

Druhým významem může být fyzická barokní poutní cesta, kterou jezuité vystavěli v podobě čtyřiačtyřiceti kapliček. Z této cesty celkem přesně víme kudy vedla a kde kapličky stály - začínala v Karlíně na místě dnešní křižovatky ulic Šaldova a Sokolovská, kde stála první kaplička a končila před Starou Boleslaví. Známý je fakt, že vzdálenost mezi jednotlivými kapličkami je 400-600 m, což přibližně odpovídá délce Karlova mostu (500 m), a že počet 44 je dán počtem mariánských invokací loretánské litanie. Podle některých zdrojů je ale invokací hned 50. V každém případě je tato „matematika“ typickým příkladem barokní krajinně - kompoziční symbolické architektury.

Navíc je ve světě pojem „Via Sancta“ často uváděn ve spojitosti poutní cesty obecně.

Když se porozhlédneme po internetu, který je dnes bezesporu alespoň pro laiky největším a nejsnáze dosažitelným zdrojem informací, mohlo by nás zarazit, jak málo je zde o Svaté cestě do Staré Boleslavi. A pokud člověk na nějaké informace narazí, většinou jsou dost roztroušené po stránkách obcí, občanských sdruženích nebo on-line průvodcích, často příliš stručné, neúplné nebo i zkreslené a nepřesné. Jako příklad by mohl posloužit fakt, že se téměř všude člověk dočte o špatném počtu dochovaných kaplí. Nejen na některých internetových stránkách obcí nebo na informační tabuli u 28. kapličky ve Vínorí za průmyslovým areálem, ale i v brožurce „Národní svatováclavská poutě“ z roku 2005 se píše o 29 dochovaných kaplích. Omyl se předává dál a dál. Ve skutečnosti kaplí stojí přibližně 16 - 17, podle toho které do souboru započítáme. U některých je totiž

nejasné zda do souboru ze 70. let 17. století patří či nikoliv. Také například na oblíbené internetové encyklopedii Wikipedie, která slouží nejen jako zdroj informací pro studenty, Via Sancta nemá žádný článek. Ale poutní cesta od pražské Lorety do Hájku, jejíž některé kapličky byly v minulých letech restaurované, článek má, a to poměrně kvalitní a obsáhlý. A to je Via Sancta zcela jistě pro Česko poutní cestou významnější než cesta do Hájku a než jakákoliv jiná poutní cesta - především díky své historii a svázáním s Českou zemí.

Ono se možná není tak úplně čemu divit, vždyť krajina a okolí Vii není na rozdíl od zmiňované poutní cesty zrovna atraktivní. Zbývající kaple jsou buď v průmyslových areálech, uprostřed polí, kde by zemědělci asi neradi viděli davu poutníků, nebo u rušné hlavní silnice z Prahy do Brandýsa nad Labem/Staré Boleslavi. A tak představa poutě nebo i jen výletu či prostě pěší vycházky po kapličkách do Staré Boleslavi nevypadá příliš lákavě. Z toho důvodu byla (alespoň ta fyzická, „kapličková“) Svatá cesta možná trochu pozapomenuta.

A toto by pro mne měla být výzva hned dvakrát. Za prvé vytvořit poutní cestě funkční a obsáhlý internetový informační systém, který by byl připravený na přidávání a aktualizování informací a ve kterém by byly všechny podstatné a dosud roztroušené informace především o kapličkách, trase cesty, včetně jejích dřívějších a historických verzí, a o zajímavostech (nejen památkách) z jejího okolí. Druhou výzvou je postupné vybudování nebo zprovoznění a především „oživení“ cesty v terénu. Pravda, ta se mě již tolik netýká. Pěkná je představa pěší (ne nutně asfaltové, například šotolinové nebo hliněné) cesty, případně cyklostezky, vedoucí kolem kapliček nebo k některým hůře přístupným odbočujícími. U silnice by cesta vedla například na druhé straně příkopu (tedy na straně pole), v poli spojovala kaple a průmyslové areály šikově obešla a vyhnula se jim. Dále několik kapliček by viditelně potřebovalo opravu; u jedné se zdá že není ohrožena jen její existence ale také existence kolem projíždějících řidičů. To vše by ale mohlo být široké téma na nejednu závěrečnou práci, a v této bakalářské práci se již věnuji zmiňovanému informačnímu systému.

Nastává ale otázka - zda je nyní více potřeba rekonstrukce cesty, kaplí a vybudování chodníků a stezek, nebo právě tvorba uceleného informačního systému. Mohlo by se zdát, že k cestě, která zatím není příliš „použitelná“ a zkulturnovaná nemá ani cenu psát informační systém, protože by neměl kdo ho využít. Podle mého ale naopak k rekonstrukci a fyzické opravě cesty je zapotřebí především to, aby se cesta dostala více do povědomí. A toto by měl vyřešit právě informační systém (nebo jen internetová databáze či webové stránky - záleží jak to nazveme), protože, pokud to přeženu, dnes „co není na internetu, jako by nebylo“.

A tak pro mne bylo a stále je výzvou vytvořit webové stránky, které by byly připravené na změny a jejichž vývoj by rozhodně s odevzdáním bakalářské práce neskončil. Měly by umožňovat po přihlášení odpovědných osob (například z Národního památkového ústavu) vkládat, editovat a rozšiřovat informace. Měly by umožňovat sledovat v mapách (i starších - mapách Rakouského stabilního katastru nebo mapách vojenského mapování) trasu cesty historické (případně vedlejších větví), současné a také polohu kaplí nebo jiných zajímavostí z okolí. Celý tento systém by tedy měl napomoci k jakési rekultivaci cesty, což ještě může být také proces relativně dlouhý.

Volba byla v technickém řešení způsobu tvorby webových stránek. Sestavit pěkné stránky s pomocí speciálních programů by bylo sice pěkné a snadné, ale jednak by

mi to neumožňovalo tolik svobody jako stránky programovat, jednak, a především, by bylo dobré nějak zužitkovat základy objektově orientovaného programování, které by měl každý geoinformatik v podobě semestru s jazykem C++ mít. Již v některém z předchozích ročníků jsme psali CGI skript, který generoval webové stránky zpřístupňující nějakou databázi, v našem případě databázi vodních toků. Pro podobné případy ale existuje velmi užitečný jazyk PHP. Výzvou tedy bylo naučit se tento jazyk a v PHP stránky vytvořit. Co se týká syntaxe, ke zmiňovanému C++ jazyk PHP tolik daleko nemá - nezačínal jsem tak od úplné nuly. Ovšem k tvorbě podobného informačního systému je třeba znát také alespoň základy jazyků HTML (+ kaskádové styly), JavaScript a také SQL pro práci s databázemi. Základy SQL byly také náplní jednoho z předmětů z předchozích ročníků.

Jako největší zdroj informací o PHP sloužila a slouží publikace Mistrovství v PHP 5 (viz [11]) a práce probíhala v populárním editoru PSPad. Všechna data byla při vývoji uložena na pevném disku mého počítače, kde je nainstalovaný server Apache 2.2, server MySQL a PHP pod operačním systémem Windows - tedy kombinaci které se někdy říká „WAMP“.

Při tvorbě jsem se snažil zaměřit spíše na tvorbu samotného systému připraveného na vkládání informací - tedy ne na samotné shromažďování těchto informací. To bude možné i později, po odevzdání práce. Kromě toho jsem s kolegou několikrát cestu projel (autem či na kole) a kapličky prošel, poznamenal si které skutečně existují, zakreslil do mapy (kopie ZM 1:10 000) a především je nafotil. Fotografie potom využil v informačním systému. V každém případě těžištěm práce bylo spíše získávání poznatků o tom, jak by měl informační systém vypadat, co by měl obsahovat, co neměl, a pro koho by měl být určen spíše než samotné hromadění informací o kaplích.

Kapitola 2

Historie a současný stav Svaté cesty

2.1 Historie

Historie této poutní cesty by mohla být tématem na celou závěrečnou práci. Vlastně už byla před několika lety napsána studentkou Fakulty humanitních studií Univerzity Karlovy, Lucií Strolenou (viz [6]). Tato práce byla základem pro tuto kapitolu, pomohla mi také zařadit si určité věci do souvislostí.

2.1.1 Vznik a původ Svaté cesty, svatováclavský kult

Původ a důvod poutní cesty z Prahy do Staré Boleslavi bychom mohli hledat již v desátém století. Počátek (myšleno časově) této Svaté cesty je totiž spojen zejména se svatováclavským kultem. *Svatý Václav* (obr. 2.1, zřejmě 907 - 28. září 935) byl jak známo podle legend ve Staré Boleslavi zavražděn svým mladším bratrem *Boleslavem I.* K činu došlo před dveřmi tehdejšího kostela sv. Kosmy a Damiána, na jehož místě dnes stojí bazilika sv. Václava. Tu založil spolu s kolegiátní kapitulou po roce 1039 kníže *Břetislav I.* Ten podnikl totiž v roce 1039 vojenské tažení do Polska a přivezl do Prahy ostatky druhého pražského biskupa (sv. Vojtěcha). Papež ho za to ale pokáral a jako pokání mu uložil založení kláštera. Namísto toho ale Břetislav I. založil baziliku sv. Václava a kapitolu, čímž napomohl zvýraznění svatováclavského kultu.

Svatý Václav se za svého života mimo jiné zasadil i o založení pražské Rotundy svatého Víta. Ta se nacházela na místě dnešní katedrály sv. Víta, Václava a Vojtěcha (do roku 1997 jen katedrála sv. Víta, viz obr. 2.2). Rotunda byla sice až třetí stavbou na Pražském hradě (po kostelíku Panny Marie a baziliky sv. Jiří), ale přesto byla později významnou sakrální stavbou na území Čech. Za prvé zde byla uložena relikvie rámě svatého Víta (cca 290 - 303), navíc se rotunda ve 2. polovině 11. století stala součástí nově vybudované baziliky. Především sem ale bylo Boleslavem I. ze Staré Boleslavi převezeno a zde pochováno tělo svatého Václava (v roce 938, podle D. Třeštíka až v 60. letech 10. století).

Tím jsou vytyčené dva krajní body Svaté cesty, která získávala v následujících stoletích mimo jiné právě kvůli růstu svatováclavském kultu na důležitosti. Svatý Václav byl a je patron české země a postava, na níž se v historii často odkazovali a mnohdy jej i „využívali“ čeští panovníci jako symbol zbožného života, šíření křesťanské víry a statečnosti. K této úctě a kultu se obrátil i několikrát zmiňovaný Boleslav I, který jej za-

2.1. HISTORIE



Obrázek 2.1: Svatý Václav v katedrále sv. Víta, Václava a Vojtěcha



Obrázek 2.2: Katedrála sv. Víta, Václava a Vojtěcha

vraždil. *Karel IV.* (1316 - 1378), český, lombardský, římský král a římský císař například rozvinul svatováclavský kult politickým pojetím. Především věnoval korunu a symbolicky i správu české země do vlastnictví sv. Václavovi. Byl dokonce křtěný jménem Václav, jakožto svatým jménem. Svatováclavskou kapli na pražském hradě v katedrále sv. Víta, která byla vysvěcena roku 1367, Karel IV. bohatě vyzdobil.

2.1.2 Mariánský kult, Palladium a jezuité

Zároveň ale Karel IV. také rozvinul mariánský kult, jehož ctitelem byl. To se projevilo mimo jiné v množství staveb a kostelů vysvěcených právě Panně Marii. Mezi nimi také jeden, již čtvrtý, staroboleslavský kostel který byl budovám přibližně v letech 1375 - 1385. Na počátku 17. století na jeho místě vznikla poutní bazilika Nanebevzetí Panny Marie, spojená s Palladiem země České a známá jako nejstarší české mariánské poutní místo.

Poutníci v této době sledovali především cestu, po níž podle Legendy jel kníže Václav v den smrti do Boleslavi a zároveň po které později jeho bratr slavnostně převezl jeho ostatky do pražské rotundy. Odtud „Svatá cesta“, neboli „Via Sancta“¹. Otázkou je kudy přesně cesta tehdy vedla. Kapličky tehdy ještě nestály, ty nechali jezuité vystavět až ve druhé polovině 18. století. Důležitými záchytnými body cesty podle legend ale musel být Prosek s kostelem sv. Václava² a potok Rokytka. Do Staré Boleslavi vedly tedy pravděpodobně dvě cesty. Severní větev, někdy zvaná svatováclavská, vedla přes Libeň a zmiňovaný kostel sv. Václava, kde bývala významná zastávka všech procesí. Dále pokračovala přes Popovice³ k brodu u Staré Boleslavi. Druhá větev, mladší, někdy zvaná Svatá cesta neboli Via Sancta, vedla více jižně. Od Poříčské brány pokračovala přes Karlín, Vysočany, Klíčov, Kbely, Vinoř, Podolánku, Dřevčice, Vrábí a Brandýs nad Labem k brodu u Staré Boleslavi. Právě podél této cesty později byly vystavěny kaple.

Koncem šestnáctého století se začal dostávat do popředí mariánský význam poutí do Staré Boleslavi. V letech 1596 i 1600 a později se konaly poutě do Staré Boleslavi v oktávu mariánského svátku Nanebevzetí Panny Marie⁴, nejprve ke kostelu sv. Václava a později i k mariánskému kostelu. Tím poutě nabývaly čím dál více charakteru mariánského [2]. Poutě většinou pořádali jezuité a jejich obliba vrcholí ještě před rokem 1620, tedy před bitvou na Bílé hoře. Velké poutě se odehrály například v letech 1607 a 1609. V roce 1611 například jezuité uspořádali pouť, u které si ve svých záznamech uvádějí účast více než tři tisíce poutníků [2]. Obliba je také čím dál více spojována se zázračným medailonkem, Palladiem⁵, ochranným obrazem země. Mezi lety 1613 - 1625 byl postaven velký

¹[1], str. 140

²Kostel svatého Václava na Proseku, který je tamní nejstarší historickou památkou, podle Hájkovy kroniky založil Boleslav II., protože se mu na tomto místě při návratu ze Staré Boleslavi zjevil svatý Václav. Udává se vznik kostela na přelomu 11. a 12. století.

³Popovice jsou dnes městská část Brandýsu nad Labem/Staré Boleslavi

⁴15. srpna římskokatolická církev slaví slavnost Nanebevzetí Panny Marie

⁵*Palladium země české* je posvátný milostný obraz, kterému se přikládá zvláštní ochranná moc nad Českou zemí. Jeho původ je spojován s řadou legend. Známa je ta o oráčovi, který medailonek našel na poli. Jeho koně se údajně před medailonkem zastavili, nechtěli dál. Oráč odnesl obraz domů, ale ráno jej opět našel na stejném místě. Když se to několikrát opakovalo, donesl jej kanovníkům staroboleslavské kapituly, kteří jej umístili do kostela sv. Václava. K překvapení všech se situace zase několikrát opakovala, medailonek se opět objevil na stejném místě. Posléze kanovníci pochopili, že musí postavit kapli na místě, kam se medailonek vrací. Na tomto místě dnes stojí chrám Panny Marie. Jiné legendy jsou spojeny s Ludmilou a Cyrilem a Metodějem. A skutečně - podle dnešních výzkumů je medailonek vytvořen ze slitiny pocházející z Balkánu (z tzv. korintské mědi). Reliéf byl dne 18. srpna 1609 prohlášen Palladiem země české, ochranným obrazem vlasti. Od té doby prošel mnohými útrapami. Nejprve v roce 1632 reliéf ukradli saští vojáci, později byl ukryt kvůli obavám před Švédy, kteří ho ale také nakonec získali. Zpět do Staré Boleslavi se reliéf slavnostně vrátil v roce 1650. Další nebezpečí v letech 1662 a 1774, 1778 Palladium přečkalo schované v Praze a Berouně. Ve 20. století bylo Palladium také úspěšně uchráněno před fašisty (dokonce zazděno) a posléze komunisty. Pokaždé, když se vrátil na své místo

2.1. HISTORIE

poutní kostel Nanebevzetí Panny Marie ve Staré Boleslavi (viz obr. 2.3), který nahradil starší gotický kostelík a ve kterém je dodnes Palladium uchovááno. Během třicetileté války (1618 - 1648) byli dokonce dočasně vyhnáni jezuité⁶.



Obrázek 2.3: Kostel Nanebevzetí Panny Marie ve Staré Boleslavi

Po bitvě na Bílé Hoře, tedy po porážce českých stavů, na český trůn nastoupil Ferdinand II.⁷, který nastolil tvrdou rekatolizaci země. V roce 1623 sám podnikl k Palladiu pouť, čímž zavedl tradici dodržovanou i jeho nástupci. Zároveň se do země vrátili jezuité, kteří obnovili tradici poutí do Staré Boleslavi. A to hned 15. srpna 1621 na svátek Nanebevzetí Panny Marie. Jezuité obnovili mnohá poutní místa a dokonce poutní cesty dále rozvíjeli a rozšiřovali. Po roce 1622 Týnský chrám na Staroměstském náměstí připadl zpět katolíkům⁸, a tak mnohé pouti začínaly právě u něj. V průběhu století se velmi dostává do popředí mariánský kult, byť stále postavený na svatováclavských základech. S touto změnou se také srdcem a nejdůležitějším místem českého království stává Stará Boleslav s Palladiem. Do této doby jím byla kaple svatého Václava na Pražském hradě.

2.1.3 Vznik kaplí

V roce 1655 se jezuitský řád trvale usídlil ve Staré Boleslavi. Druhá polovina 17. století je již spjatá čistě s generací jezuitů. Ve velkém se budovaly a stavěly mariánské sloupy, sochy, kaple a také celé soubory kaplí - poutní cesty, vše v duchu typické barokní symboliky. V 70. letech přišla na řadu také Svatá cesta do Staré Boleslavi.

do staroboleslavského chrámu Panny Marie, pomohl k několika zázračným uzdravením. [6]

⁶[3], s. 50

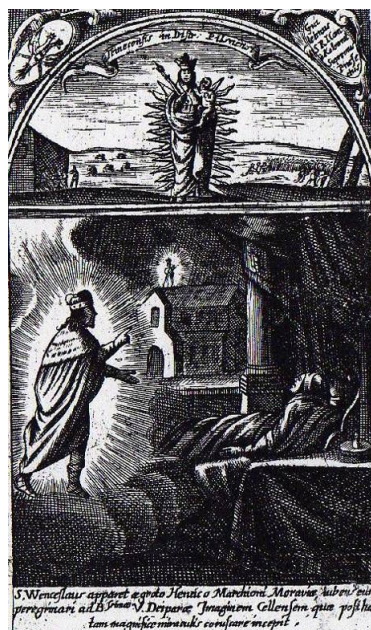
⁷*Ferdinand II. Štýrský* byl římský císař, český, uherský a chorvatský král (v letech 1619 - 1637), arcivévoda rakouský a štýrský vévoda, který pocházel z rodu Habsburků. Po porážce českých stavů se ujal vlády v zemi, a po řadě zatýkání a konfiskací 21. června 1621 nechal Ferdinand II. na Staroměstském náměstí popraviti 27 českých pánů.

⁸*Kostel Matky Boží před Týnem* neboli Týnský chrám od husitské doby patřil pod správu husitům.

Autory celé soustavy kaplí jsou tedy jak známo pražští jezuité, především však Jan Tanner⁹. Ten nechal vybudovat v okolí Prahy mnoho různých kapliček. Pro rekonstrukci a představu původní výzdoby kaplí je velmi důležité a nenahraditelné jeho dílo *Svatá cesta z Prahy do Staré Boleslavi...* [8], které vyšlo poprvé česky v roce 1679, poté latinsky 1690 a dále německy v letech 1680 a 1705. Každé zastavení Svaté cesty je zde podrobně popsáno, spolu s rytinou původního obrazu kaple, pocházející od Samuela Dvořáka, pražského rytce. Z této publikace mimo jiné čerpá i jiný autor, současný, Miroslav Kuranda ve svém díle *Z Prahy do Staré Boleslavi svatou cestou podél čtyřiačtyřiceti kapliček a svatováclavskou cestou ze Staré Boleslavi do Prahy* z roku 2009 [7].

2.2 Podoba kaplí

Podoba, resp. výzdoba každé kaple vznikala pravděpodobně společně dohodou konkrétního donátora, který je právě díky zmiňovanému Tannerovu dílu známý, a konkrétního umělce. Je tedy málo pravděpodobné, že by Tannerova publikace sloužila jako předem napsaná příručka ke stavbě a výzdobě kaplí.



Obrázek 2.4: Rytina původní výzdoby 40. kaple - Sv. Václav se zjevuje nemocnému moravskému markraběti Jindřichovi [8]

⁹[4], s. 347-348; Jan Tanner (17. srpna 1623 - 4. listopadu 1694) byl významným pražským jezuitou, studoval v Brně a v pražském Klementinu, byl pedagogem v Pražské a Olomoucké jezuitské koleji, ve svých dílech se věnoval především teologii, hagiografii a genealogii. Pro nás jsou podstatná jeho dvě pravděpodobně nejdůležitější díla, *Život sv. Václava* a *Svatá cesta z Prahy do Staré Boleslavi k nejdůstojnější rodičce Boží panně Marii, čtyřicítí a čtyřmi krásnými stavuňky, podle počtu litanie lauretánské titulům jejími obrazy v Čechách slavnějšími ozdobená, ale předtím dávno svatého milého Václava, dědice českého na smrt odjezdem, a potom těla přenešením posvěcená, nyní také malováním životem a zázraků jeho okrášlená.*

2.2. PODOBA KAPLÍ

Spis obsahuje ke každé kapli (tedy 44-krát) celou rytinu, která zachycuje její původní výzdobu, modlitbu mariánského cyklu a modlitbu cyklu svatováclavského, resp. popis zobrazeného výjevu ze života sv. Václava. Autorem rytin je pravděpodobně slavný pražský rytec Samuel Dvořák, který ryl například i Komenského mapu Moravy.



Obrázek 2.5: Rytina původní výzdoby 44. kaple - Rodokmen sv. Bořivoje s sv. Ludmilou [8]

Kaple byly jednotného vzhledu, dnes se některé nepatrně liší - například tvarem křížku. Jsou výklenkového typu, po stranách mají dva toskánské pilastry, které tvoří edikulu, typickou pro barokní kapličky. Mezi pilastry je nika neboli výklenek zakončený obloukem. Vše završuje trojúhelníkový tympanon, většinou s kruhovým slepým okénkem (ale ne u každé kaple). V zadní straně výklenku byla malba na omítce rozdělená na 2 části (kromě první a poslední kaple). V horní, menší segmentové části v oblouku niky byla zobrazena socha Madony nebo poutní obraz, podle kterého byla daná kaple pojmenována a které byla kaple zasvěcena. V ozdobné stuzce pak byl latinsky napsán název daného poutního místa. V dolní, větší části byl obraz vyjadřující výjev ze života sv. Václava. Po vnitřních bočních stranách niky byly česky vepsány modlitby mariánského (levá strana) a svatováclavského (pravá strana) cyklu a na klenbě zespodu byla napsána latinsky příslušná invokace [1]. Po stranách výklenku v horní části byl ještě vlevo erb stavebníka neboli donátora kaple, vpravo jeho celé jméno [5]. Erb, jméno, a celý obraz v zadní části výklenku je u každé kaple zachycen v Tannerově knize jako jedna rytina. Podle J. Dvořáka *byly obrazy v kapličkách ve shodě s rytinami v knize, jak lze soudit z nepatrných dochovaných zbytků* (č. 23, 25, 30 a 33). Faktem je, že za 38 let se *nepatrné zbytky* nemusejí zachovat. Výzdoba jednotlivých kaplí byla pravděpodobně nejméně jednou obnovována, u kaple č. 14 šlo ve zbytku invokace rozeznat dvě vrstvy omítky. Zbytky výzdoby včetně jediné dosud dochované invokace u 26. kaple umožňují porovnáním s Tannerovou knihou jednoznačně přiřadit pořadová čísla jednotlivým kaplím [1].



Obrázek 2.6: 14. kaple - jedna z těch zachovalejších; zdroj: vlastní

Náměty na hlavní obrazy jsou různé, vždy však související se svatým Václavem. První dva obrazy jsou spojené s Palladiem země České, další výjevy zachycují různé důležité epizody života sv. Václava, stejně tak z doby po jeho smrti. Například jak posílá posly do Říma, aby zjistil jaké církevní slavnosti se v Římě konají na den sv. Kosmy a Damiána a archanděla Michaela (14. kaple), jeho zavraždění (30. kaple), nebo zjevení sv. Václava Boleslavovi, aby mu nakázal razit mince s jeho obrazem. Poslední obraz (který není rozdělený, viz obr. 2.5) zachycuje rodokmen sv. Ludmily a Bořivoje - rodina sv. Václava.

Počet všech kaplí odpovídá počtu invokací v loretánských litaniích a délka mezi jednotlivými zastaveními také není náhodná. Má odpovídat délce Karlova mostu v Praze (tedy přibližně 500 m). Toť typický příklad barokní symboliky - vše do sebe zapadá.

2.3 Svatá cesta v mapách

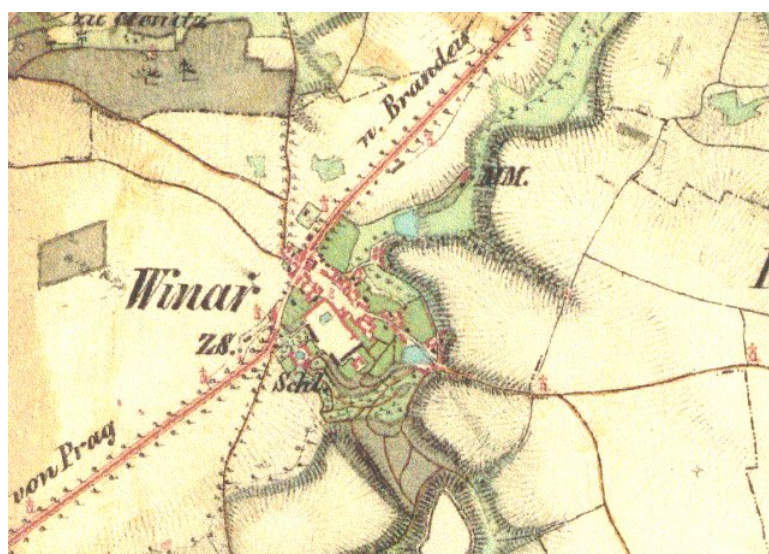
Jednotlivé kaple jsou kartograficky zachyceny nejprve v mapách I. vojenského mapování (Josefské, 1763 - 1787, zde zachycující území z let 1780 - 1783 [1]). Na Müllerově mapě Čech (1708 - 1720, měřítko cca 1:132 000) kaple ještě zachyceny nejsou. Pro mapy I. voj. mapování bylo zvoleno měřítko 1:28 800 [9]. Zde jsou kaple vyznačeny relativně přesně a zřetelně.

Dalšími mapami, kde jsou kaple dohledatelné, jsou mapy rakouského stabilního katastru a indikační skici, které byly součástí stabilního katastru a které vznikaly jako kolorované tisky určené pro polní měření a určení (neboli indikování) změn. Mapy stabilního katastru včetně indikačních skic jsou mapami daleko přesnějšími než mapy vojenského mapování, neboť již vznikaly triangulací. Bohužel zde ale kaple nejsou zachyceny zdaleka

2.4. STRUČNÝ POPIS A POLOHY JEDNOTLIVÝCH KAPLÍ

všechny, dokonce ani některé z těch, které dnes stojí. Navíc se z mapové značky nedá poznat, zda kaple patří do Svaté cesty nebo ne. Je zajímavé sledovat, že již v této mapě silnice, někdy ani polní cesta, mnohdy nevedla kolem kaplí, ale hranice zřetelně naznačují původní průběh poutní cesty.

V mapách druhého vojenského mapování (Františkovo, 1807 - 1869 v našem prostoru z let 1850 - 1852, měřítko 1:28 800 [9]) kaple vyznačeny jsou, někde více a někde méně zřetelně (viz obr. 2.7). To se dnes ostatně každý může přesvědčit na mapách na serveru www.mapy.cz¹⁰, kde je mapa z II. vojenského mapování označena jako *historická*. Opět ale není jasné například to, která kaple do Svaté cesty patří a která ne. Z těchto map vznikl ve 30. letech 19. století plán okolí Prahy, na němž kaple vyznačeny jsou [1].



Obrázek 2.7: Ukázka mapy II. vojenského mapování z Víně. Na mapě jsou zřetelně vidět u cesty kaple. Zdroj: Mapy.cz

Třetí vojenské mapování proběhlo v letech 1870 - 1883 [9], zde bylo mapováno v roce 1877 a v měřítku 1:25 000. Zde kaple zakreslené jsou, až na některé kaple uvnitř obcí jsou snad zakreslené správně. V dnešních turistických mapách a státních mapových dílech (Základní mapě 1:10 000 a Státní mapě) jsou zakreslené jen některé kaple, většinou v extravilánu.

2.4 Stručný popis a polohy jednotlivých kaplí

Via Sancta tedy začínala asi jeden a půl kilometru za Poříčskou bránou, v místě kde se dnes kříží ulice Sokolovská a Šaldova a končila přímo u hranic Staré Boleslavi. V následujícím seznamu je vždy pořadí, jméno kaple (odvozené od místa madony, které je kaple zasvěcená) a stručný popis polohy. Informace jsou čerpány z díla J. Dvořáka[1], publikace J. Stibora[5], částečně také z diplomové práce L. Strolené[6], dále z dosud

¹⁰V tomto informačním systému bude využito právě této mapy a bude možnost si kaple na mapě II. vojenského mapování prohlédnout přímo ze stránek poutní cesty.

pravděpodobně nejnovější knihy o cestě od M. Kurandy[7] a v neposlední řadě z vlastní zkušenosti, kdy jsem poutní cestu osobně prošel, zdokumentoval a vyfotil. Tak ale zjistím jen jak konkrétní kaple vypadá, zda stojí a kde stojí, ne však jestli patří do původního souboru a jaké je její původní pořadí. To se liší i u jednotlivých zdrojů. Největší nesrovnalosti jsou mezi kaplemi 26 a 32. Dnes zde stojí kaple 3 - za cukrovarem hned další před Podolankou a nad Podolankou nad vsí, z nichž poslední navíc díky vzhledu nezapadá do souboru. Podle Dvořáka je kaple za cukrovarem 29. kaplí, ta u Podolanky tedy 30. To ostatně jasně vyplývá i ze znázornění v mapách II. vojenského mapování kde jsou všechny zakresleny. Podle Kurandy a zároveň podle současných zdrojů (včetně informační tabule naučné stezky přímo u kaple) ale je kaple za cukrovarem v pořadí 28., ta před Podolankou tedy 29. Je zde tedy poněkud nejasné číslování nejméně dvou kaplí. Podle Dvořáka existují také nejasnosti s polohou kaplí 6 a 7.

1. kaple - *Staroboleslavská* - dnes neexistuje, stála pravděpodobně vlevo, na místě dnešního křížení ulic Sokolovská a Šaldova a není jasné ani kdy zanikla - zda v 18., 19. nebo až 20. století.
2. kaple - *Aitlická (Údlická)* - dnes neexistuje, ale na jejím místě je kaple novodobá, menší. Ta stojí vlevo naproti Invalidovně. Původní pravděpodobně stála o kousek blíže řece, ale musela ustoupit železničnímu náspu.
3. kaple - *Bechyňská* - neexistuje a není známo ani kdy zanikla. Stála pravděpodobně před současným hotelem Olympic.
4. kaple - *Brněnská* - neexistuje, pravděpodobně stála v místech bývalých dílen Dopravního podniku.
5. kaple - *Budějovická* - neexistuje, stála pravděpodobně vlevo u křižovatky na Palmovce.
6. kaple - *Chlumecká* - neexistuje, stála nejspíš v místech dnešního železničního mostu přes ulici Sokolovskou, v místech víceúrovňové křižovatky s ulicemi Českomoravská a Spojovací.
7. kaple - *Chomutovská* - neexistuje, stála poblíž křižovatky Sokolovská a U Svobodárny, v dnešní Čihákově ulici. Tato kaple (jako první) dokonce existuje na fotografii, musela ustoupit kulturnímu středisku Gong a okolní výstavbě.
8. kaple - *Družecká* - neexistuje, stála údajně v místě křížení dnešních ulic Sokolovská a U Skládky.
9. kaple - *Jičínská* - neexistuje, stála pravděpodobně vlevo u Rokytky, na místě dnešní sochy *Muž*.
10. kaple - *Kladská* - neexistuje, podle Dvořáka stála na východním konci dnešní Zbuzkovy ulice, není o ní však žádný doklad.
11. kaple - *Hejnická* - neexistuje, stála údajně v dnešním jižním ohybu ulice Pod Krocínkou. V blízkosti dříve také stál památný kříž.

2.4. STRUČNÝ POPIS A POLOHY JEDNOTLIVÝCH KAPLÍ

12. kaple - *Horažd'ovská* - původní kaple neexistuje, stála pravděpodobně na severním konci ulice Pod Krocínkou, po pravé straně. Pamětníci nejistě vzpomínají na kapli v parku na Klíčově. Není také úplně jasné, zda se nejedná o (přestavěnou, nebo od počátku se odlišující) kapli, která se dnes nachází pod Čakovickou ulicí. Ta se odlišuje jak vzhledem, tak velikostí, je totiž mohutnější než ostatní.
13. kaple - *Hradčanská* - dnes neexistuje, byla zbourána v zimě 1970-1971, kdy musela pravděpodobně ustoupit výstavbě. Stála na dvoře garáží ČSAD (ze kterých dnes už také nic nezbylo).
14. kaple - *Jeníkovská* - první dnes existující, leží daleko v poli severně od dnešní silnice na Brandýs nad Labem. Nedávno opravená.
15. kaple - *Kájovská* - nachází se vlevo od silnice v poli, za rohem plotu areálu *PSS Knauf*. Je zachovaná, ale ve viditelně horším stavu než předchozí.



Obrázek 2.8: 15. kaple; zdroj: vlastní

16. kaple - *Kladrubská* - dnes již neexistuje, ale stála ve Kbelích, v ulici Hůlkova než ustoupila zástavbě. Údajně existuje i několik fotografií již poškozené kaple. V díle J. Dvořáka [1] včetně plánu je kaple ještě uváděna jako stojící.
17. kaple - *Krupecká* - stojí, i když dnes již velmi zchátralá, na parkovišti *Muzea letectví a kosmonautiky*, tedy přes Mladoboleslavskou silnici naproti samotnému muzeu, poněkud schovaná. Je atypicky obrácená čelem k severu.
18. kaple - *Klumenská* - neexistuje, stávala vpravo od Mladoboleslavské ulice na konci areálu *Muzea letectví a kosmonautiky*. Na některých mapách jsou zde zakresleny dvě kaple poblíž sebe. Byla zbourána při budování letiště.
19. kaple - *Kutnohorská* - podle J. Dvořáka stávala v ulici Palackého (pravděpodobně nyní Vrchlabská), v polovině vzdálenosti její křižovatky s ulicí Mladoboleslavská a

náměstím. Podle M. Kurandy [7] je ovšem její umístění sporné, na mapách zde (ani jinde) zakreslená není a pamětníci si žádnou kapli nevybavují.

20. kaple - *Lauretánská* - neexistuje, její poloha je ale přesně známá, dokonce existuje fotografie již poškozené kaple, ze 60. let min. století. Ležela na křižovatce bývalých ulic Nádražní a Alšova.
21. kaple - *Hájecká* - neexistuje, nacházela se na místě dnešního železničního nádraží (zastávky) Kbely, při jehož stavbě byla zbourána.
22. kaple - *Litomyšlská* - neexistuje, stávala pravděpodobně vlevo od Mladoboleslavské ulice, v místě nynějších zahrad.
23. kaple - *Nezamyslická* - existuje, je zachovalá a po opravě. Nachází se vlevo od silnice v poli. Podle J. Dvořáka [1] je (tedy v 70. letech bylo) na této kapli nejvíce původních fresek.
24. kaple - *Nycovská* - existuje, je po opravě a stojí vlevo od Mladoboleslavské silnice před Vinoří.
25. kaple - *Obergürgentálská* - existuje a nachází se na začátku Vinoře vlevo od hlavní silnice v zatáčce naproti Vinořskému zámku. Je ale schovaná v boční uličce (Semtínská) a otočená čelem k plotu rodinného domku. Kaple je v dobrém stavu.
26. kaple - *Pakenská* - také existuje a nachází se přímo v centru Vinoře, u silnice poblíž hřbitova v těsné blízkosti autobusové zastávky. Bohužel to se také podepsalo na její „výzdobě“ sprejery. Je nově omítnutá, ale je ponecháno „okénko“ stropu výklenku, kde je vidět původní invokace „Rosa mystica“ (Růže tajemná). Je pravděpodobně nejznámější kaplí řady (možná právě kvůli své poloze).



Obrázek 2.9: 26. kaple ve Vinoři; zdroj: vlastní

2.4. STRUČNÝ POPIS A POLOHY JEDNOTLIVÝCH KAPLÍ

27. kaple - *Plzeňská* - již neexistuje, nacházela se pravděpodobně vpravo od silnice, v místě zástavby rodinnými domky nad Cukrovarským rybníkem.
28. kaple - *Hradopražská* - zde se již informace rozcházejí. Podle J. Dvořáka [1] tato kaple neexistuje a nacházela se na začátku tehdejšího cukrovaru (který dnes už vystřídal areál *Stro.m Propagace s.r.o.*). To se zdá být evidentní i z mapy II. vojenského mapování. Podle Kurandy [7] a novější literatury je tato kaple ta, která se nachází za zmiňovaným areálem na břehu malého rybníka (který jsem ovšem při mé revizi ani nezaznamenal). Sem byla údajně po vybudování cukrovaru přenesena. Kolem této kaple (pokud to je tedy ona) nyní vede také naučná stezka *Vinoř - Jenštejn*, a tedy jako zastávka stezky je kaple poměrně udržovaná, s informační tabulí a dokonce malým sklíčkem, přes nějž je ve správném úhlu vidět na kapli původní výzdoba. Ještě před několika málo lety (v době vzniku Kurandovy [7] knihy) ale kaple byla poměrně zarostlá. Kvůli přemístění není možné zjistit původní polohu.
29. kaple - *Svatokřížská* - je také sporná. Podle J. Dvořáka [1] se nachází vpravo od silnice za bývalým cukrovarem, ale obrácená čelem k severu. Mohlo by se jednat o kapli která je v novějších zdrojích uváděná jako 28., se kterou se ještě později „hýbalo“. A nebo se také může jednat o úplně jinou, dnes už neexistující kapli, jsou to ovšem jen moje domněnky. Podle M. Kurandy [7] a novějších zdrojů je tato kaple vpravo od silnice, při vjezdu do Podolanky (tedy dnes první „mimopražská“ kaple) na kraji pole, otočená čelem k jihu.
30. kaple - *Svatojakubská* - podle J. Dvořáka [1] existující, při vjezdu do Podolanky, viz 29. kaple podle M. Kurandy. Podle něj tato kaple již neexistuje a Kuranda [7] ani netuší, kde by měla stát. Pravděpodobně v prostoru návsi. Není totiž známo kudy silnice původně stoupala na kopec v Podolance.
31. kaple - *Svatomikulášská* - Podle J. Dvořáka kaple stála vpravo od silnice za mostem přes potok a dnes již neexistuje. Podle Kurandy kaple také neexistuje, ale je sporná. V zahradě za posledním domem v Podolance je kaple stavebně výrazně odlišná (podobná jako např. druhá, nebo ta poblíž 41. kaple) u které není jasné zda do souboru patří (v těchto místech je nepravděpodobná samostatná kaplička bez návaznosti na cestu), zda je nově postavená na místě původní, nebo zda do poutní cesty vůbec nepatří (jak se domnívá J. Dvořák).
32. kaple - *Rokycanská* - tato kaple již celkem bez rozporů existuje a nachází se vlevo od silnice za Podolankou na vrcholu stoupání.
33. kaple - *Římovská* - také existuje, a nachází se vlevo těsně u silnice za odbočkou na Jenštejn.
34. kaple - *Svatohorská* - neexistuje. V minulosti stála vlevo u silnice před Dřevčicemi.
35. kaple - *Sečenská* - existuje a stojí na okraji Dřevčic u silnice. V roce 1973, kdy sepsal J. Dvořák již mnohokrát zmiňovanou publikaci, byly již všechny kapličky od této dál nově omítnuty, takže nešlo rozeznat zbytky maleb. Dnes jsou omítnuty všechny.

36. kaple - *Šternberská* - existuje. Nachází se v Dřevčicích u místní samoobsluhy natočená čelem do uličky (původně staré silnice) směřující ke kostelu a je v dobrém stavu po opravě. Vpravo o několik metrů dříve je navíc kaplička, která do souboru podle J. Dvořáka nepatří. Je také stavebně odlišná a ve výklenku je socha svatého Jana Nepomuckého.
37. kaple - *Strakonická* - neexistuje, stávala za Dřevčicemi vlevo, zhruba v místech kde stávala vodárna, poblíž silnice od Dřevčic k Popovicím. Kaple není zanesena ani v mapách stabilního katastru z roku 1841.
38. kaple - *Strašenská* - stojí, i když už zdálky vypadá že z posledních sil. Nachází se těsně u silnice vlevo naproti cihelny mezi Dřevčicemi a Brandýsem, je podepřená kůly a ovázaná drátem jakoby se každou chvíli měla sesypat na silnici. Tato kaple také není zanesena v mapách stabilního katastru.



Obrázek 2.10: 38. kaple je ve špatném stavu; zdroj: vlastní

39. kaple - *Sušická* - neexistuje, stávala údajně vlevo u můstku přes potok těsně při vjezdu do Vrábí, městské čtvrti Brandýsa nad Labem.
40. kaple - *Tejnecká* - neexistuje, alespoň podle J. Dvořáka a stála vlevo od silnice na začátku Brandýsa nad Labem. Stavebně odlišná kaple (podobná jako ta v Podolance) je několik desítek metrů před 41. kaplí vpravo schovaná v uličce jménem Květnová. Není jasné, zda je tato kaple na místě původní, nebo původní stála zcela jinde.
41. kaple - *Vartenská* - stojí v Brandýsu nad Labem po levé straně (ve směru z Prahy) Pražské ulice, v parčíku. Je opravená a poblíž je také kamenný křížek.

2.4. STRUČNÝ POPIS A POLOHY JEDNOTLIVÝCH KAPLÍ

42. kaple - *Vyšehradská* - poslední dosud stojící kaple se nachází vlevo v Pražské ulici v Brandýse nad Labem, kousek před železničním přejezdem. Je atypická z několika důvodů. Je zhruba o metr vyšší, má zvýšený podstavec, ale jinak je stejného typu. Je již obestavěná zdí a její nika je dodatečně po opravě osazena sádrovou mariánskou plastikou.
43. kaple - *Zbraslavská* - neexistuje, nacházela se v Brandýse nad Labem na náměstí.
44. kaple - *Staroboleslavská* - neexistuje, stála na levém břehu Labe u mostu (toho původního dřevěného, který stál více na jihovýchod proti proudu řeky v ose dnešního můstku vedoucího jen na ostrov). Kapli možná odnesla nějaká povodeň, a protože nový kamenný most přes řeku stál už jinde, nebyla obnovena. Zde byl atypický obraz (rodokmen, viz výše), spolu s obrazem na předchozí kapli již nesouvisel se svatováclavskou legendou.

Kapitola 3

Informační systém z pohledu uživatele

Informační systém je pojem značně obecný a v internetovém světě může vyvolávat pocit čehosi velkého, projektu s velkými objemy dat a mnoha vrstvami uživatelů. Využití pro popis Svaté cesty se může proto zdát poněkud předimenzované, a je jasné, že informační systém v tomto případě budou spíše propracovanější internetové stránky včetně jednoduché databáze, jejichž hlavním cílem bude laickou i odbornou veřejnost informovat a stavu a poloze kaplí poutní cesty a dalších zajímavostech s ní souvisejících. Celý systém tedy rozhodně není hotový během jednoho semestru, ale naopak se bude (alespoň doufám) vyvíjet a růst s postupem času. Bylo proto nutné určit si, co má přednost a co by měl obsahovat již ve svém počátku. Není to ani vyčerpávající množství informací o kaplích, ani „vyplněné“ položky v menu, ale především možnost se do systému zaregistrovat a informace a data vkládat. Ať již jako administrátor, či jako běžný registrovaný uživatel. Na naplnění systému daty a přidání dalších funkcí a zajímavostí bude ještě v budoucnu čas a prostor.

Systém je tedy již funkční po stránce registrace a přidávání dat (a například fotografií), ale mnoho dalších nápadů je stále v záloze a pravděpodobně již od pohledu je jasné že systém ještě „poroste“.

3.1 Popis systému

Jak již bylo řečeno, systém má 3 vrstvy uživatelů - *neregistrovaný uživatel*, *registrovaný uživatel* a *administrátor*. Těch prvních a druhých může být libovolně mnoho, administrátor je jen jeden. Zatímco neregistrovaný uživatel (což je třeba i náhodný návštěvník stránek) má možnost jen informace získávat a fotografie prohlížet, registrovaný již ta samá data (byť ve stále omezené míře) může také editovat, přidávat nové fotografie a v budoucnu například přidávat novinky, či psát o zajímavostech z okolí. Administrátor má již „neomezené“ možnosti, tedy může o kaplích upravovat všechny údaje, fotografie a ostatní uživatele mazat a zároveň on je tím, kdo povolí registraci nového uživatele.

3.1.1 Běžný návštěvník stránek

Největší a hlavní cílová skupina tohoto systému pravděpodobně budou turisté. Jako další návštěvníky systému lze předpokládat milovníky architektury, památkáře, ale také náhodné návštěvníky se zájmem o informace a do určité míry poutníky. Ti všichni budou pravděpodobně stránky navštěvovat jako neregistrovaní uživatelé.

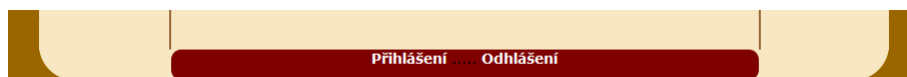


Obrázek 3.1: Úvodní strana

Pokud se na stránky dostane běžný uživatel, tedy ten, který dosud není zaregistrovaný v systému nebo není administrátorem aktivovaný, má možnost stránky a informace „jen“ prohlížet. Tedy zejména prohlížet informace o kaplích, případně jiných objektech a sledovat jejich polohu na mapě. Při načtení hlavní stránky by se měly objevit základní informace s uvítáním, a v budoucnu také přidávané aktuality (např. poutě, koncerty, plánované rekonstrukce ap.) týkající se poutní cesty. Zkrátka typická titulní strana. Po levé straně se nachází hlavní menu, po pravé straně prozatím jen náhodná kaple. Tou je při každém načtení jakékoliv stránky systému náhodně generovaná kaple sloužící jako odkaz na podrobnější informace o ní. Je tvořena fotografií (v případě že kaple neexistuje tak jednotnou černobílou ikonou) a jménem s pořadovým číslem. Pod náhodou kaplí v budoucnu budou důležité odkazy (prioritně formou log), příp. další položky, např. kontakt na administrátora či přímo textový vstup s možností odeslat mu vzkaz.

Záhlaví stránky tvoří jednoduchý nápis „VIA SANCTA/aneb stránky věnované poutní cestě z Prahy do Staré Boleslavi“, fungující zároveň jako odkaz na hlavní stránku projektu. Po stranách jsou dva erby měst které cesta spojuje - levý odkazuje na Prahu, pravý na Brandýs nad Labem - Starou Boleslav. Odkazy se otevřou v nové záložce, což se mi zdá praktičtější, pokud odkazují na stránky externího serveru. V liště pod záhlavím pro neregistrovaného uživatele zatím není nic k vidění, v budoucnu by zde ale mohl být třeba další odkaz na hlavní stránku.

V zápatí, na konci stránky je další lišta, s odkazy na přihlášení a odhlášení ze systému. Protože naprostá většina uživatelů se pravděpodobně registrovat nebude, jsou tyto odkazy „schované“ až dole, aby zbytečně nepřitahovaly pozornost.



Obrázek 3.2: Zápatí stránky

Některé položky menu ještě v době psaní tohoto textu nejsou naplněny obsahem. Nicméně zde je jejich popis:

- **O svaté cestě** - Základní informace
- **Kaple** - Přehledně seřazené kaple poutní cesty
- **Mapa** - Mapy poutní cesty
- **Okolí** - Informace o zajímavostech a památkách kromě kaplí
- **Fotogalerie** - Všechny fotografie
- **Odkazy** - Odkazy na zajímavé a související stránky
- **O projektu** - Důvod a forma vzniku celého projektu



Obrázek 3.3: Podoba menu z pohledu neregistrovaného uživatele

O svaté cestě - Základní informace o poutní cestě, o kaplích této poutní cesty obecně, historie poutní cesty a další zajímavosti.

Kaple - Ikonky odkazující na detaily kaplí, přehledně seřazené a barevně odlišené. Červeně označená okénka znamenají již neexistující kaple a zeleně označená existující (ať už nově opravené či chátrající). Žlutě označené jsou kaple, u nichž není jisté, zda

do poutní cesty vůbec patří, nebo do poutní cesty sice nepatří ale nachází se v její linii. A také kaple, které jsou novější, ale stojí na místě původní starší kaple (viz např. 2. kaple). V tuto chvíli žádná kaple žlutě označená není, ale databáze a systém je na toto připraven. U každé kaple je její jméno a pořadové číslo. Pokud kaple existuje, nebo existuje alespoň její fotografie, je jako ikona použita upravená fotografie. V opačném případě je nahrazena jednotnou černobílou ikonou kaple. Ta, pro zajímavost, vznikla úpravou fotografie 14. kaple v programu *GIMP*.

V horní části stránky je možné zvolit, které kaple chce uživatel zobrazit, zda existující, neexistující, či všechny.

Mapa - V této položce se nachází prozatím jen mapa poutní cesty vytvořená v rozhraní *Google Maps*. Mapa je do určité míry interaktivní, to znamená že všechny stojící kaple jsou v mapě vyznačeny ikonkou na kterou je možné kliknout. Pokud tak uživatel učiní, otevře se informační okénko s fotografií (která je právě nastavena jako titulní) názvem a pořadím kaple, a s počátkem textu ke kapli. Na stránku o konkrétní kapli, kde se nacházejí podrobnější informace je možné přejít kliknutím na odkaz „více“.

V budoucnu v této položce bude vše, co souvisí s mapami poutní cesty. Tedy zejména možnost otevřít mapu Google ve větším okně a podobná mapa vytvořená v *mapy.cz*. Ta by měla být implicitně zobrazená jako „historická“, což znamená jako mapa II. vojenského mapování. Dále by zde mohly být mapy jako soubory PDF či rastrové soubory (např. ke stažení), či jen URL odkazy na jiné zajímavé mapy související s Viou Sanctou.

Okolí - Protože je velmi pravděpodobné, že samotné kaple roztroušené kolem rušné silnice a za nepřívětivými průmyslovými areály příliš velkým lákadlem pro turisty nebudou, server bude nabízet také informace o jiných zajímavostech poblíž trasy Svaté cesty. Také může být zajímavé spojit například část poutní cesty s návštěvou hradu Jenštejn či některého z významných staroboleslavských kostelů. Právě o historických či přírodních zajímavostech by se zde měl návštěvník stránek dozvědět podrobnosti. Pravda, těch přírodních zde moc není, ale přinejmenším Vinořský park a rybníky stojí za návštěvu, spolu s naučnými stezkami ve Vinoři (Vinořský park - Satalická bažantice a nová NS Vinoř - Jenštejn).

Fotogalerie - Systém již nyní obsahuje velké množství fotografií kapliček (převážně pořízených z vlastního fotoaparátu při revizi cesty), další budou pravděpodobně přibývat. Mimo jejich vyobrazení u samotných kaplí, budou všechny zde, přehledně uspořádané. Budou tu také (a zejména) fotografie, které nesouvisí s žádnou konkrétní kaplí či souvisí s jinými zajímavostmi.

Odkazy - Je mnoho informací, na které je lepší odkázat, než je přímo psát. Budou zde odkazy na různá občanská sdružení a instituce zabývající se poutní cestou (např. *ULTREIA* či *Matice staroboleslavská*), dále na internetové stránky obcí, Národního památkového ústavu a na jiné zajímavé stránky.

O projektu - Obsah této položky bude stručný - budou zde shrnuty důvody a okolnosti vzniku celého projektu a také možnosti jeho vývoje v budoucnu.

Pokud uživatel přejde na detail kaple (ať už odkazem z náhodné kaple, z klepnutí na některou z ikon v položce „Kaple“ menu nebo např. odkazem v mapě), zobrazí se mu vyčerpávající informace (pokud už byly doplněny) o dané kapli. Její název, a pokud existuje tak menším písmem v závorce doplňující název. Např. pro 40. kapli se zob-

razí „Tejnecká kaple“ s podtitulem „(Týnec u Klatov)“. Kliknutím na „předchozí“ nebo „další“ lze přejít na sousední kapli poutní cesty. Pod názvem jsou heslovitě základní informace o kapli - první je její pořadí v poutní cestě. Protože u několika kaplí není ještě jasné přesné pořadí (viz 2.4), je v tomto projektu vycházeno z publikace Miroslava Kurandy[7]. Dále je ke kapli uvedena invocace Panny Marie ve formě oslovení, např. „Panno Ctihodná“ u 15. kaple. Pod ní je jméno a popis donátora, který je u každé kaple jiný. Dále u kaplí, které existují nebo jejichž poloha je přesně známa je uvedeno katastrální území. To funguje zároveň jako hypertextový odkaz na internetové stránky obce (či městské části). Na závěr hlavičky jsou u existujících kaplí uvedeny její zeměpisné souřadnice - tedy zeměpisná šířka a délka v šedesátinném dělení.

<= předchozí
další =>

Obergürgentálská kaple

(Horní Jiřetín u Duchcova)

- 25. v pořadí
- zasvěcená invocací litanie: "**Nádobo Výborné Pobožnosti**"
- donátor kaple: **J. Bedřich, hrabě z Valdštejna**
- kaple leží v katastrálním území **Vinoř**.
- souřadnice kaple: 50° 8' 29" North, 14° 34' 35" East.

Výjev zachycuje sv. Václava jako kajícího v doprovodu bl. Podivena, který kráčí v jeho šlépějích.

Na vyobrazení se spatřuje další část z legendy - sv. Václav, kráčející v zimě nocí sněhem bos a s ním bl. Podiven, zahřívající své nohy v jeho šlépějích.

V horní, mariánské části obrazu, je milostná socha jiřetínské Madony - stojící korunovaná postava s děťátkem v ruce obklopená rozkvetlými keři.

V kartuši na vnější straně kaple nahoře je nakreslen valdštejnský erb.

Kaple je dnes v dobrém stavu.



Obrázek 3.4: Hlavička 25. kaple

Pod hlavičkou každé kaple je její fotografie, která je právě vybrána jako titulní. Tedy samozřejmě jen pokud fotografie existuje. Pokud je k obrázku uložen popis, při najetí kurzorem na fotografii se zobrazí. Zároveň s fotografií začíná hlavní text, popisující především stav kaple, její polohu, původní výzdobu a její význam, souvislosti zasvěcení určité Madoně a případně osud donátora dané kaple. Následuje vodorovnou čarou oddělená modlitba mariánského cyklu (ta původně vepsaná v levé boční stěně niky) a svatováclavského cyklu (pravá boční stěna niky). Ten se většinou vztahuje k výjevu vy-

obrazeném v původní výzdobě kaple. Pokud existují k dané kapli nějaké poznámky, jsou uvedeny pod modlitbami, stejně jako případné odkazy.

Nakonec jsou přehledně v tabulce zobrazeny fotografie ke kapli. Vždy je v levé části její náhled a v pravé podrobnější informace. Fotografie lze klepnutím na náhled otevřít v novém okně a v plné velikosti. V pravé části je u každé kaple tučně její popisek (pokud je zadán), název uloženého souboru, datum pořízení fotografie (pokud je známo), její velikost v kB, rozměry v pixelech, případně autor či zdroj fotografie. Pokud je daná fotografie nastavena jako titulní, je to v tabulce také uvedeno.

3.1.2 Možnosti registrovaného uživatele

Větší část práce ale zatím spočívala v tvorbě prvků typických právě pro registrované uživatele, tedy především nahrávání fotografií, jejich úprava a úprava informací o kapli. K registraci nového uživatele je nutná domluva s administrátorem. A to nejlépe předem, ale teoreticky vzato klidně až po registraci. Pokud se chce uživatel registrovat, musí nejprve klepnout na odkaz „Přihlášení“ dole na liště. Přihlásit se ale zatím nemůže, pokud není registrovaný. Je zde tedy odkaz „Registrace“, na který je nutné kliknout. Zobrazí se jednoduchý registrační formulář spolu s upozorněním o nutnosti kontaktovat administrátora (a s aktuálním e-mailem na něj). Formulář obsahuje 3 položky, všechny povinné: *uživatelské jméno*, *heslo* a *e-mail*. Pokud uživatel nějakou položku nevyplní, formulář není možné odeslat a zobrazí se upozornění. Stejně tak se zobrazí upozornění, pokud uživatel vyplní heslo v němž chybí symbol @. Takový e-mail se považuje za neplatný.



Obrázek 3.5: Registrace nového uživatele

Uživatelské jméno musí být zvoleno takové, které dosud v systému není registrované. V opačném případě se po odeslání formuláře tlačítkem „Registrovat“ vypíše upozornění že není možné uživatele pod tímto jménem registrovat spolu s odkazem zpět k registraci. Pokud je registrace úspěšná, zobrazí se hlášení o úspěchu. V tuto chvíli je sice uživatel registrovaný, zatím je ale pravděpodobně jeho účet neaktivní. Až administrátor uživatele povolí (viz níže), uživatel má teprve možnost se úspěšně přihlásit. Pokud na přihlášení přejde

odkazem z úspěšného hlášení o registraci, bude již přihlašovací formulář předvyplněný. Přihlášení probíhá jednoduše zadáním uživatelského jména a hesla. Pokud je již uživatel přihlášen a klikne na „Přihlášení“, systém ho upozorní dotazem zda se chce přehlásit na jiný účet.

První co se po přihlášení viditelně změní, je podoba menu, v němž jsou nyní navíc 2 položky. Pro přehlednost jsou barevně odlišené, u žlutých položek menu tedy víme, že jsou vyhrazeny jen pro registrované uživatele.



Obrázek 3.6: Menu z pohledu registrovaného uživatele

A co se za nimi skrývá? Pokud registrovaný uživatel klepne na „Editace“, má možnost upravit některé informace o kapli (viz obr. 3.7). Tedy základní popis kaple, modlitby, poznámky, případně upravit, přidat nebo smazat internetový odkaz ke kapli. Nejprve ale musí vybrat v roletovém menu, kterou kapli chce upravit. Formulář odešle klepnutím na tlačítko „Uložit“. Pokud by se uživatel pokusil zadat odkaz ve špatném formátu nebo s protokolem *http://*, systém ho upozorní a formulář neodešle. Stejně tak pokud by se pokusil odeslat údaje aniž by vybral kapli.

Po odeslání formuláře má uživatel možnost upravit/přidat další informace či přejít na detail upravované kaple.

Další položka menu přístupná jen pro registrované uživatele je „Přidat foto“. Uživatel má tedy možnost nahrání fotografie na server, zatím ale jen ke konkrétní kapli. Po klepnutí na tlačítko „Vybrat soubor“ vybere na disku svého počítače danou fotografii, kterou potom uloží kliknutím na „Odeslat“. Systém zatím funguje jen pro fotografie ve formátu JPEG, v budoucnu se počítá také s formátem TIFF. Zobrazí se hlášení o úspěšném nahrání a jméno fotografie, pod nímž je její náhled a velikost v kB. Pokud by náhodou již existovala v systému fotografie se shodným názvem, zobrazí se upozornění, a pro srovnání také náhled již dříve uložené fotografie spolu s velikostmi. Název nové fotografie je v takovém případě doplněn o „-(2)“, aby nedošlo k jejímu přepsání.

V tuto chvíli je fotografie uložena, ale v databázi zatím není a také její náhled ještě

Editace informací

27. - Plzeňská

popis kaple:

Zachycuje sv. Václava, který před odchodem do Staré Boleslavi před svou smrtí přijímá v pozadí hostii. V popředí se loučí se svými druhy.
Sv. Václav, ač se dozvěděl, že má být v Boleslavi při hostině zavražděn, nedal se odradit a do Boleslavi jel, ale nechal se dříve zaopatřit církevními svátostmi umírajících - to je příběh vyobrazení této kaple.

mariánský cyklus:

Věže Davidova a nejsilnější obrano všech věrných křesťanů Panno Maria, před jejímžto svatým obrazem v Plzni, vždycky katolickým městě, horlivá starokatolická pobožnost více než od sta let na každý pátek od roku do roku mši svatou o tvém neposkvrněném početí skrze samé mládenečky zpívati dává; pro tu srdečnou bolest, kterou měla, když jsi svého milého syna viděla hořce plakat na

svatováclavský cyklus:


Svatý Václave, jenž pozván byv od bratra do Boleslavě a předzvěděv svou smrt, opatřil jsi se církevními svátostmi; pak svolav kněží, s pláčem jsi se s nimi rozřehnal; potom již na cestě výstrahu dostav od jednoho křesťana, že tě bratr zabítí miní, odpověděls, že chceš rád umřítí pro Krista Pána. jenž prvé umřel

Obrázek 3.7: Formulář pro editaci informací o kapli

neexistuje. Uživatel má nyní ještě možnost „vycouvat“ klepnutím na odkaz „Nechci fotografii uložit“ v dolní části stránky. V takovém případě se fotografie smaže. Pokud fotografii uložit chce, musí vybrat v roletovém menu které kaple se týká (obr. 3.8). Dále má možnost upravit název souboru, autora, datum pořízení, nebo popis, přičemž vše také může zůstat prázdné. Kolonky „Autor nebo zdroj“, „Datum pořízení“ a „Popisek nebo poznámka k fotografii“ jsou předvyplněny z Exifu fotografie, a tak nemusí být správné. Pokud v Exifu není uveden autor, předvyplní se uživatelské jméno toho, kdo fotografii přidává. Klepnutím na „Uložit“ se formulář odešle. Systém zkontroluje, zda je formulář v pořádku - tedy formát data, zda vůbec takové datum existuje a zda je vybrána kaple. Pokud by uživatel jméno souboru jen smazal, systém na základě náhodného čísla a aktuálního času vygeneruje náhradní a pod tímto soubor uloží. Odesláním formuláře se tedy fotografie uloží do databáze a zároveň se vytvoří náhled pro následné rychlejší načítání. Zobrazí se hlášení o úspěchu (či případné poznámky např. o vygenerovaném náhradním názvu) a přehledně uspořádané informace o fotografii. Také její náhled,

Přidání fotografie

Byl úspěšně načten obrázek s názvem "017-kaple asi 17.JPG":



Velikost: **789 kB**

Zatím ale v databázi není, vyplňte nebo opravte prosím následující údaje o fotografii:

Poznámka: Kromě kaple není žádná položka povinná.
Předem vepsané datum pořízení může být špatné (fotoaparát nemusí správně nebo vůbec ukládat metadata).

Vyberte kapli, k níž se fotografie vztahuje:

Název souboru (není podstatný):

Autor nebo zdroj:

Datum pořízení:

Popisek nebo poznámka k fotografii:

Obrázek 3.8: Formulář pro přidání nové fotografie

na který je možné klepnout a fotografii zobrazit v nové záložce v plné velikosti. Uživatel má nyní 3 další možnosti - přejít na detail kaple, jejíž fotografii nahrával, znovu upravit informace o fotografii, nebo nahrát další.

Pokud se registrovaný uživatel podívá pozorněji na informace o jednotlivých fotografiích ke kapli, najde tam navíc odkaz na jejich úpravu. Pokud na „upravit informace“ klepne, zobrazí se náhled fotografie (opět s možností zobrazit fotografii v nové záložce) a formulář na úpravu informací o souboru (viz obr. 3.9). Velikost a rozměry fotografie jsou dané, ale je možné změnit kapli k níž se vztahuje, jméno souboru, datum pořízení, autora a popisek k fotografii. Navíc je zde možnost zatrhnutím radiobuttonu nastavit danou fotografii jako titulní. Pokud uživatel již nechce informace upravovat, vrátí se zpět ke kapli kliknutím na odkaz „Zpět ke kapli č. ...“. V opačném případě klikne na tlačítko „Uložit“.

velikost fotografie:	2392 kB
rozměry fotografie:	2448 × 3264 pixelů
kaple, k níž se fotografie vztahuje:	15. - Kájovská ▾
jméno souboru:	kap_15_10.JPG
datum pořízení:	2011-02-26
autor nebo zdroj:	Ondřej Pospíšil
poznámka/popisek:	<input type="text"/>
nastavit jako titulní:	<input type="radio"/>
<input type="button" value="Uložit"/>	
Zpět ke kapli č. 15	

Obrázek 3.9: Formulář pro úpravu informací o fotografii

Systém v takovém případě uloží informace do databáze, a pokud byla fotografie nastavena jako titulní, tak se zároveň vytvoří větší náhled (tedy pokud ještě neexistuje) pro rychlejší načítání stránky a tato fotografie se zobrazí v hlavičce detailu dané kaple.

Dalším viditelným rozdílem oproti tomu, když není uživatel přihlášen je horní lišta. Tam se totiž objeví informace o tom, pod jakým jménem je uživatel přihlášen. Pokud se na uživatelské jméno klikne, dostaneme se do správy uživatelského účtu (viz obr. 3.10). To znamená v tomto případě jen úpravu hesla či kontaktního e-mailu. Po rozkliknutí příslušné úpravy je možné položku změnit.

Přihlášen(a) jako **adam**

Správa uživatelského účtu

uživatel:	adam	(nelze změnit)
e-mail:	pospisil.ondrej@seznam.cz	
nový e-mail:	<input type="text" value="pospisil.ondrej@seznam.cz"/>	<input type="button" value="Ulož nový e-mail"/>
		<input type="button" value="Zrušit"/>
původní heslo:	<input type="text"/>	
nové heslo:	<input type="text"/>	
nové heslo - kontrola:	<input type="text"/>	
	<input type="button" value="Ulož nové heslo"/>	<input type="button" value="Zrušit"/>

Obrázek 3.10: Správa účtu. Ve výřezu je též vidět horní lišta

Pro změnu hesla je nutné z bezpečnostních důvodů nejprve zadat staré a to nové pro kontrolu dvakrát. Také lze odeslat v jednu chvíli jen jeden formulář, tzn. že není možné zároveň upravit e-mail i heslo. Před odesláním formuláře systém zkontroluje správnost starého hesla a zda jsou obě nová hesla shodná. V případě e-mailu kontroluje jeho správnost, tím že obsahuje symbol @. Po provedených změnách se tedy formulář odešle klepnutím na „Ulož nový e-mail“/„Ulož nové heslo“.

3.1.3 Možnosti administrátora

Aby se uživatel přihlásil jako administrátor, stačí zadat do přihlašovacího formuláře jako uživatelské jméno „admin“, správné heslo a systém sám pozná, že jde o administrátora. Možnosti administrátora jsou ještě větší než u běžného registrovaného uživatele. Po přihlášení se v menu objeví oproti běžné registraci dvě nové položky: „Vkládání informací (admin)“ - ta nahrazuje položku „Editace“ - a „Správa uživatelů“ (viz obr. 3.11).



Obrázek 3.11: Menu z pohledu administrátora

Pokud administrátor klepne na „Vkládání informací (admin)“, zobrazí se obsáhlý formulář, ve kterém je možné upravit veškeré informace o kaplích. Funguje stejně jako formulář pro ostatní registrované uživatele, takže je nejprve potřeba vybrat v roletovém menu kapli pro editaci. Je potom možné editovat její jméno (všechny tři formy), pořadové číslo, její stav (zda existuje, neexistuje nebo je sporná - další roletové menu), zeměpisné souřadnice, zasvěcení, z roletového menu vybrat katastrální území v němž leží či ležela, dále upravit donátora, další texty (popis, obě modlitby a poznámky) a přidat, smazat či upravit odkazy související s kaplí. Odkaz se smaže jednoduše tak, že se vymaže obsah textového vstupu s patřičnou URL adresou. Popis odkazu není povinný, pokud je ale zadán, samotný text funguje jako hypertextový odkaz. Pokud změny uložíme kliknutím na tlačítko „Vložit“, systém před odesláním zkontroluje data - tedy zda je vybraná kaple,

3.1. POPIS SYSTÉMU

zda není zadána jen jedna souřadnice (také kontroluje její formát), a také správnost URL odkazů. Po úspěšném odeslání formuláře je možno opět přejít k dané kapli či upravit další.

The screenshot shows a web form titled "Vkládání informací - administrátor" for editing church data. At the top, a dropdown menu is set to "28. - Hradopražská". The form fields include:

- pořadové číslo kaple: 28
- stav: existující
- celé jméno: Hradopražská
- zkrácené jméno: Hradopražská
- doplňující jméno: (empty)
- Souřadnice: North: 14.5944, East: 50.1518
- zasvěcení: Věži Slonová
- katastrální území: Vinoř - 782378
- donátor: Václav Bílek z Bílenberk
- popis kaple: Zasvěcena Panně Marii Ara Coeli z pražského chrámu sv. Víta. Zachycuje scénu stolování ve Staré Boleslavi, při níž epiklenci Václavovy vraždy třikrát povstanou, ale vždy se srazí a saváňají. V posadí je k odjedu pro sv. Václava přichystán kůň. Václav však zůstává.
- mariánský cyklus: Věže slonová, nejsvětější Panno Maria, jejížto obraz od svatého Lukáš malovaný v hlavním kostele Pražském za drahý poklad kostela toho se chová: skrze tu srdečnou bolest, kterou měla, když jsi viděla svého nejmilejšího syna na kříži umírat, prosím tebe, smiluj
- svatováclavský cyklus: Svatý Václave na něhož, když jsi ve svátek Kosmy a Damiana v Staré Boleslavi u bratra svého stolil, epiklé se hodovníci, třikrát povstali, chtějí tebe samordovati: ale vždycky božskou mocí zdržáni jsou.. ty pak moha na koni, kteréhož

Obrázek 3.12: Formulář pro úpravu informací o kaplích administrátorem - výřez

Druhou položkou menu viditelnou jen pro administrátora je „Správa uživatelů“. Již z názvu je patrné k čemu slouží. Po načtení se zobrazí tabulka s uživateli seřazenými abecedně podle uživatelského jména (viz obr. 3.13).

Ke každému uživateli je v tabulce uveden kromě jména e-mail, a zda je aktivní či nikoliv. Pokud je uživatel aktivní, je v jeho řádce ve sloupci „aktivita“ zaškrtnutý checkbox. Zde má administrátor možnost zakázat či povolit určité uživatele. Pokud se tedy registruje nový uživatel, zde se administrátorovi objeví jako neaktivní a administrátor ho zde podle svého uvážení povolí. Naopak je také možné zakázat takto přístup určitým uživatelům. Po provedené změně je potřeba patřičným tlačítkem na konci tabulky změny uložit. Systém tak uloží do databáze změny.

Je také možné určitého uživatele trvale vymazat z databáze. K tomu slouží tlačítko v pravém sloupci tabulky. Pokud na něj administrátor klikne, zobrazí se nejprve dotaz a poté výzva k zadání hesla (administrátorského). Po jeho správném zadání je uživatel

Správa uživatelů

uživatelské jméno	e-mail	aktivita	
adam	pospisil.ondrej@seznam.cz	<input checked="" type="checkbox"/>	Smazat uživatele
admin	pospisil.ondrej@seznam.cz	aktivní	
jana	pospisil.ondrej@seznam.cz	<input type="checkbox"/>	Smazat uživatele
Uložit změny			

Obrázek 3.13: Správa uživatelů

trvale smazán z databáze.

Další možnost, kterou má administrátor navíc ve srovnání s ostatními uživateli, je mazání fotografie. U podrobných informací o fotografiích v detailu kaple je nyní navíc také červený odkaz na smazání (viz obr. 3.14). Pokud se na něj klikne, zobrazí se nejprve dotaz zda administrátor skutečně chce fotografii smazat, a poté stránka s jejím náhledem. Pod náhledem je název fotografie a výzva k zadání hesla. Pro smazání fotografie je tedy pro jistotu nutné vyplnit správné heslo (viz obr. 3.15). Po klepnutí na „Odeslat“ se ještě zobrazí malý náhled fotografie a informace o úspěšném smazání.

	<p>smazat fotografii</p> <p>název souboru: kap_28_02.JPG datum pořízení: 26. únor 2011 velikost: 3826 kB 2448 x 3264 pixelů autor/zdroj: Ondřej Pospíšil</p> <p>upravit informace</p> <p>smazat fotografii</p>
	<p>název souboru: kap_28_03.JPG datum pořízení: 26. únor 2011 velikost: 3085 kB 2448 x 3264 pixelů autor/zdroj: Ondřej Pospíšil</p> <p>upravit informace</p> <p>smazat fotografii</p>
	<p>název souboru: kap_28_04.JPG</p>

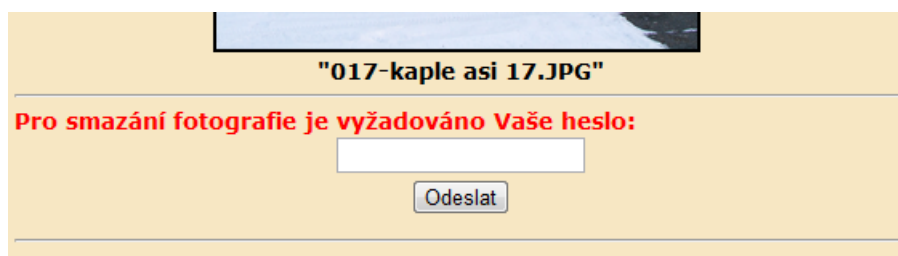
Obrázek 3.14: Detail informací o fotografii z pohledu administrátora

Pokud by se vymazala zrovna fotografie aktuálně nastavená jako titulní, systém automaticky vybere jako titulní jinou, konkrétně první která se vztahuje ke stejné kapli.

Odhlášení administrátora i běžného registrovaného uživatele se provede jednoduše kliknutím na patřičný odkaz na spodním okraji stránky.

3.2 Data

Ač naplnění systému daty nebylo absolutní prioritou bakalářské práce, alespoň pro ukázkou funkčnosti bylo potřeba nějaké informace sehnat. V maximální míře jsem se snažil



Obrázek 3.15: Pro smazání fotografie je nutné zadat administrátorské heslo

využít vlastních zdrojů - tedy množství fotografií pořízených při osobní revizi poutní cesty. Při té jsem si také dělal poznámky a stavu a umístění jednotlivých kaplí. S Petrem Vaverkou jsme s pomocí příruční GPS zaměřili souřadnice všech stojících kaplí, včetně těch které pravděpodobně do Vii nepatří. Tyto souřadnice jsou použity v detailech kaplí, a zároveň na základě těchto souřadnic jsou kaple zobrazeny v mapě. Další fotografie poskytl Národní památkový ústav (NPÚ), stejně jako kopie evidenčních listů památek - tedy jednotlivých kaplí. Informace z těchto listů zatím v systému použity nejsou, ale do budoucna pravděpodobně budou. Další informace, tedy zasvěcení, jejich pořadí a modlitby s texty k některým kaplím jsou získány z publikace [7] s laskavým svolením jejího autora, velkého znalce dějin nejen Prahy 9, Miroslava Kurandy. Donátoři kaplí a základní popisy původní výzdoby jsou také čerpány z diplomové práce Lucie Strolené [6]. Případné odkazy a poznámky, a také třeba katastrální území, v němž kaple leží jsou získány z vlastního pátrání na internetu, především Geoportálu ČÚZK.

Kapitola 4

Informační systém z pohledu vývojáře

4.1 Způsob zpracování

4.1.1 Počítačové jazyky

Protože internetově přístupný informační systém je vlastně fungující webová aplikace, tvůrce si zde nevystačí jen s několika soubory napsanými v HTML. Jednoduché HTML nenabízí prakticky žádné možnosti interaktivních webových stránek. HTML tagy se u každé webové stránky/souboru jednoduše načtou a po načtení klient nemá možnost cokoliv měnit (snad kromě vyplnění formulářů, ty ale zase nemůže odeslat).

V případě informačního systému o poutní cestě *Via Sancta* bylo využito objektově orientovaného skriptovacího programovacího jazyka *PHP*, který je určený především právě ke generování dynamických webových stránek, komunikaci s databázemi, zpracování dotazů klienta, souborů XML, a k mnoha dalším případům. Nakonec se tento programovací jazyk nemusí využívat jen k tvorbě webových stránek, je možné různě zpracovávat lokálně uložené soubory, skriptovat z příkazového řádku, tvořit konzolové a desktopové aplikace atd.

Možností využití PHP je spousta, ale PHP jako způsob tvorby interaktivních webových stránek není zdaleka jediný. Existují například jazyky Pearl (také objektově orientovaný, podobná syntaxe s jazykem PHP) nebo ColdFusion, je možné psát CGI skripty například v jazycích C++, Ruby, Python, Java či využít ASP.NET. PHP je ale nejrozšířenější alternativou pro programování webových stránek na straně serveru [10], nejčastěji využívaný ve spojení s MySQL, Apache a Linux. Taková kombinace se označuje jako „LAMP“, v případě operačního systému Windows „WAMP“.

Aplikace je tedy napsaná v PHP5, konkrétně verzi PHP Version 5.3.4. Psaní probíhalo na notebooku s operačním systémem Windows 7, instalovaným serverem Apache 2.2 a databázovým systémem MySQL 5.1. Přehledněji viz tabulka č. 4.1.

Dále byl využit program a skriptovací programovací jazyk *Matlab*, v němž proběhl jen drobný doplňující přepočítání souřadnic do desetinné soustavy. Pro psaní samotné bakalářské práce byl využit systém \TeX se značkovacím jazykem \LaTeX .

4.1. ZPŮSOB ZPRACOVÁNÍ

název	verze	stručný popis	aplikace v IS
PHP	5.3.4	skriptovací, objektově orientovaný jazyk určený především k tvorbě interaktivních webových stránek (na straně serveru)	základ celé aplikace, všechny stránky projektu jsou psané v PHP
JavaScript		skriptovací, objektově orientovaný jazyk určený především k psaní jednoduchých webových aplikací na straně klienta	zejména vyplňování a kontrola formulářů, Google Maps
SQL		dotazovací jazyk pro komunikaci (tvorba databází, tabulek, plnění daty, jejich úprava, mazání či výpis) s databází	MySQL 5.1 - způsob uložení dat a informací k informačnímu systému - uživatelé, kaple, fotografie, ad..
HTML	4.0	značkovací jazyk určený k publikování na internetu	formát, v němž se hotová webová stránka odesílá klientovi a kterou je prohlížeč schopný zobrazit
CSS		jazyk, popisující zobrazování webových stránek napsaných v HTML (příp. XHTML, XML)	zobrazení/skrytí a vzhled jednotlivých prvků vygenerovaných HTML dokumentů

Tabulka 4.1: Tabulka všech využitých jazyků v průběhu tvorby systému (a bakalářské práce)

4.1.2 Programy, software

Kromě *PHP*, databázového systému *MySQL* a serveru *Apache* bylo pro tvorbu použito několik různých programů. Pro některé (jednodušší a jednorázové) výpočty program *Matlab R2010a*, pro jednorázovou úpravu fotografií pro některé náhledy fotografický software *Zoner Photo Studio 13*. Pro samotné psaní skriptů (PHP, JavaScript) a kódů HTML a CSS byl využit populární textový editor *PSPad verze 4.5.4* vyvíjený českým programátorem *Janem Fialou*, testování a zobrazování vygenerované webové stránky probíhalo zejména ve webovém prohlížeči *Google Chrome ver. 12.0*, který obsahuje také JavaScriptovou konzoli, průzkumník DOM struktury serveru a další zajímavé funkce, s pomocí kterých se snáze nacházejí chyby např. v JavaScriptu. K tvorbě grafiky, podkladových ploch a některých jednoduchých ikon bylo také využito aplikace *GIMP* pro vytváření a úpravu rastrové grafiky. K vytvoření databáze a počáteční správu a vytváření tabulek se použilo pouze příkazové řádky. Později k vytvoření souborů **.sql* pro přesun databáze na server laboratoře fotogrammetrie se ale využil program *MySQL Workbench*, který také umožnil vytvořit *EER diagram*. K samotnému psaní práce v \LaTeX bylo potom využito aplikace *Texmaker*. Vše je přehledně popsáno v tabulce č. 4.2.

software	verze	licence	stručný popis	aplikace v IS
Matlab	R2010a	proprietární software	programové prostředí pro numerické výpočty	některé výpočty se souřadnicemi
Zoner Photo Studio	13	proprietární software	český program, bitmapový editor sloužící zejména k úpravě fotografií	úprava některých fotografií pro potřeby náhledu, zpracování vlastních fotografií vložených do systému
PSPad	4.5.4	freeware	český, univerzální editor pro psaní zdrojových kódů (s podporou jak např. PHP, C++ či Java, tak HTML, TeX a mnoho dalších) pro platformu MS Windows	v editoru psány veškeré zdrojové kódy PHP, JavaScript a HTML
Google Chrome	11.0	Google Chrome Terms of Service	webový prohlížeč s některými funkcemi pro vývojáře	testování a zobrazování veškerých vygenerovaných stránek, zobrazování vygenerovaného zdrojového HTML kódu, případná kontrola JavaScript skriptů
GIMP	2.6.11	GNU GPL	aplikace pro vytváření a úpravu rastrové grafiky	tvorba pozadí, hlavičky, menu, ikon a jiných grafických objektů
MySQL Workbench	5.2.33	GNU GPL	program pro správu databáze MySQL	přesun databáze a tabulek na server, tvorba EER diagramu
Texmaker	2.0	GNU GPL	editor určený k sazbě dokumentů \LaTeX	psaní bakalářské práce

Tabulka 4.2: Tabulka využitých programů v průběhu tvorby systému (a bakalářské práce)

4.2 Architektura

Možností, jak „propojit“ zdrojové soubory, ať už malého, či velkého internetového projektu je mnoho. První možností je napsat prakticky jen jeden skript, který obslouží všechna volání a jednotlivé webové stránky. V tomto souboru, pojmenovaném obvykle *index.php*, se zobrazí vždy určitá část, oddělená většinou podmínkovými řídicími strukturami např. na základě HTTP metod GET a POST. Jediná výhoda tohoto způsobu tvorby webových stránek je ale snad jen ta, že s jedním souborem se snadněji manipuluje. Nevýhodou je zejména u větších projektů značná nepřehlednost. Pokud má skript tisíce nebo desetitisíce řádků, velmi špatně se v něm cokoli hledá. Navíc je URL v prohlížeči buď stále stejná (v případě využívání metody POST a JavaScriptu v maximální míře - může to být i záměr a výhoda), nebo značně nepřehledná s velkým množstvím parametrů (v případě využívání metody GET).

Další možností je vytvořit jeden skript pro každou stránku a každou funkci. Tímto způsobem byl (do určité míry) vytvořen i tento systém - v hlavní složce se nachází skript *index.php*, který zahrnuje skript *homepage.php*. Ostatní jednotlivé stránky již fungují jako

jednotlivé soubory, uložené v hlavní složce. V každém souboru je zahrnut nejdříve soubor s HTML hlavičkou, umístěný ve zvláštní složce *./includes*, dále skript generující hlavičku stránky, boční menu a pravou část stránky (*header.php*). Před zahrnutím těchto skriptů jsou navíc definovány určité proměnné, které se v těchto skriptech využívají pokaždé jiné (konkrétně například obsah tagu `<title>`, či informace, která položka menu je právě aktivní). Tím, že se v jazyce s pomocí funkce `include` nebo `require`, resp. `include_once` nebo `require_once` do kódu jednoduše zahrne další soubor **.php*, jakoby byl od začátku na daném místě kódu, lze předtím dynamicky určit proměnné, potřebné v zahrnovaném skriptu. Rozdíl mezi zmiňovanými funkcemi je následující:

- `include 'soubor.php'`; - zahrne na dané místo soubor *soubor.php*.
- `require 'soubor.php'`; - zahrne na dané místo soubor *soubor.php*, ale pokud je zahrnovaná adresa (zde právě *soubor.php*) neplatná, skript oznámí na rozdíl od předchozí funkce chybu.
- `include_once 'soubor.php'`; - zahrne na dané místo soubor *soubor.php*, pokud ještě nebyl zahrnut.
- `require_once 'soubor.php'`; - zahrne na dané místo soubor *soubor.php*, pokud ještě nebyl zahrnut ale pokud je zahrnovaná adresa (zde *soubor.php*) neplatná, skript oznámí na rozdíl od předchozí funkce chybu.

Zde byla tedy využita především tato architektura, kdy složka obsahuje více skriptů **.php*, pro každou načítanou stránku jeden. Dále se také využívá psaní prostého HTML kódu - tedy kde je to možné, HTML kód se přímo píše do souboru, ne v php tagu příkazy `echo ''`; či `print ''`;

Dále, častěji využívané funkce, které je zbytečné psát v každém skriptu znovu, jsou také odděleny do zvláštních souborů (uložených ve složce *./includes*) a pokud jsou potřeba, jsou zahrnuty většinou příkazem `require`. Jedná se například od skript zajišťující spojení s databází:

```
<?php
$conn = new mysqli("localhost", "root", "melnik", "viasancta");
if (mysqli_connect_errno())
{
    die("Funkce mysqli_connect selhala: " . mysqli_connect_error());
}
?>
```

Nebo také skript, kontrolující zda má klient právo k načtení dané stránky, například určené jen pro registrovaného uživatele či administrátora. Třídy jsou psány také do zvláštních souborů, a ty jsou ukládány do složky *./classes*. Všechny soubory *JavaScript*, zahrnované v HTML hlavičkách, jsou potom uloženy ve zvláštní složce *./js*, pracovní PHP skripty, které jsou většinou využity jen jednou, například pro hromadné načtení dat do databáze, jsou uloženy ve složce *./pracovni*. Všechny fotografie a rastrové soubory,

které nejsou fotografiemi kaplí (například hlavička stránky, erby, ikona mapy ap.) jsou uloženy ve složce `./foto_ostatni`.

Dále hlavní složka obsahuje čtyři složky s fotografiemi. Adresář `./foto_kaplicky` obsahuje samotné originální fotografie kaplí, adresář `./foto_nahledy_for_kaplicky` náhledy upravené v programu *Zoner Photo Studio* pro ikony v seznamu kaplí a pro zobrazení náhodné kaple. Složka `./foto_kaplicky_nahledy_mensi` obsahuje malé náhledy původních fotografií, které se automaticky vytvoří při načtení nové fotografie a její přidání do databáze. Nakonec složka `./foto_kaplicky_nahledy_vetsi` obsahuje větší náhledy fotografií. Tyto náhledy se vytvoří vždy ve chvíli, kdy je daná fotografie zvolena za titulní k dané kapli (pokud náhled již neexistuje).

4.3 Využití databáze

Použití databázového systému je v systému nezbytné - do tabulek se ukládají jak informace o kaplích, případně jiných objektech, tak například o fotografiích, nebo samotných uživatelích systému. Díky své jednoduchosti, výkonu a především podpory ze strany jazyka PHP padla volba na databázový systém *MySQL*.

4.3.1 MySQL

Databázový systém *MySQL* byl vytvořený švédskou firmou *MySQL AB*, nyní vyvíjený *Oracle Corporation*. Je multiplatformní, lze jej instalovat jak v Linuxu, tak v MS Windows a jiných operačních systémech. Aktuální verze je 5.1.33, připravuje se 6.0. Hlavními přednostmi jsou jednoduchost a rychlost, avšak v posledních letech se díky novým vlastnostem MySQL stává poměrně robustním databázovým serverem. Komunikace probíhá s pomocí známého jazyka SQL.

4.3.2 Schéma databáze

Jednoduchá databáze vytvořená pro účel tohoto informačního systému byla vytvořená z příkazové řádky, jednotlivé tabulky, jako jednorázová akce také. Jejich plnění daty a úprava těchto dat již probíhala z roztředí formulářů a s pomocí jazyka PHP. První prací byla tedy instalace *MySQL Server 5.1* na počítač, hned další potom založení databáze *viasancta*, s pomocí příkazu `CREATE DATABASE viasancta;`

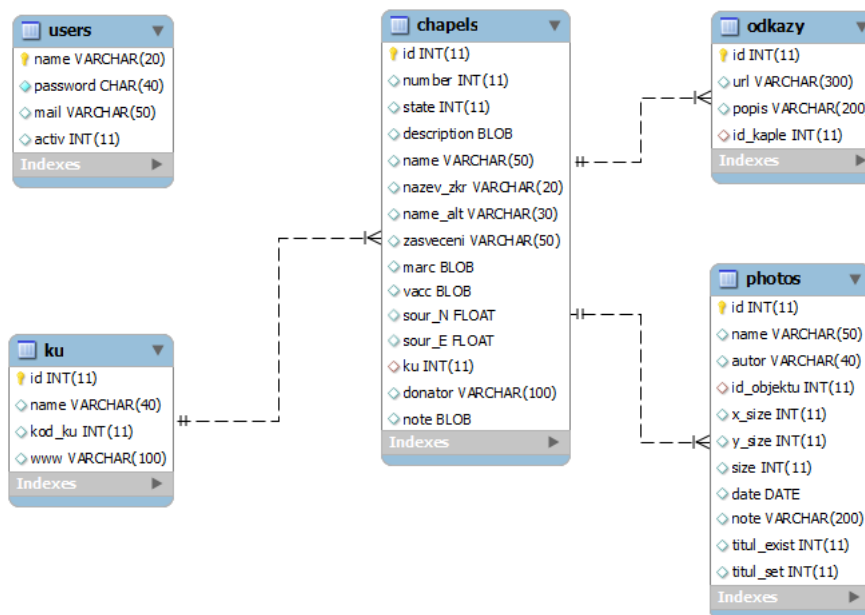
Příkazem `USE viasancta;` se poté nastavila databáze **viasancta** jako aktivní a všechny příkazy probíhaly v jejím kontextu. V průběhu tvorby informačního systému se poté vytvářely nové tabulky příkazem `CREATE TABLE nizev_tabulky (nizev_sloupce1 datovy_typ1, nizev_sloupce2 datovy_typ2, ...);` Pokud bylo zapotřebí vypsát seznam tabulek databáze, použil se příkaz `SHOW TABLES;`

K dodatečné správě databáze, jejího přesunu na server laboratoře fotogrammetrie a také pro vytvoření *EER diagramu* byl později krátce využit program *MySQL Workbench*.

Databáze *viasancta* tedy obsahuje 5 tabulek: `chapels`, `ku`, `odkazy`, `photos` a `users`. Na obr. 4.1 je jejich schéma.

Za hlavní tabulku je možné považovat *chapels*. Tabulka kaplí (a v budoucnu pravděpodobně nejen kaplí) má 15 sloupců.

4.3. VYUŽITÍ DATABÁZE



Obrázek 4.1: EER diagram databáze viasancta

- **id INT(11) PRIMARY KEY NOT NULL** - jednoznačný identifikátor kaple (či jiného objektu), nastavený tedy jako primární klíč, hodnota musí být různá od nuly. Na tento identifikátor se odkazují také sloupce *id_kaple* z tabulky *odkazy* a *id_objektu* z tabulky *photos*.
- **number INT(11)** - pořadí kaple v poutní cestě. V případě objektů jiných než kaple nebo u kaplí nezahrnutých do cesty bude hodnotou *NULL*.
- **state INT(11)** - číselnou hodnotou vyjadřuje, zda kaple existuje (hodnota 1), již neexistuje (hodnota 2), nebo je její zařazení do poutní cesty sporné (3). V budoucnu budou mít jiné objekty nebo kaple nesouvisející s poutní cestou hodnoty vyšší než 3.
- **description BLOB** - obsahuje samotný text ke kapli, její popis - např. historii, popis původní výzdoby nebo umístění kaple. Datový typ *BLOB* umožňuje ukládat řetězce do délky maximálně 65 535 znaků.
- **name VARCHAR(50)** - celý název kaple. Tedy i s jeho vysvětlením, např. *Obergürgentálská (Horní Jiřetín u Duchcova)*.
- **nazev_zkr VARCHAR(20)** - samotný název kaple, bez jeho vysvětlení, např. *Obergürgentálská*.
- **name_alt VARCHAR(30)** - vysvětlení názvu kaple - místa původu madony, např. *Horní Jiřetín u Duchcova*.
- **zasveceni VARCHAR(50)** - mariánské oslovení, u každé kaple jiné, např. *Panno Nejmoudřejší*.

- **marc BLOB** - modlitba z mariánského cyklu. Díky datovému typu *BLOB* řetězec může být delší než 255 znaků.
- **vacc BLOB** - modlitba ze svatováclavského cyklu. Díky datovému typu *BLOB* řetězec může být delší než 255 znaků.
- **sour_N FLOAT** - severní zeměpisná šířka. Ta, pokud je známá (v případě existující kaple), je uložena v desetinném dělení, např. *50.1719* jako datový typ *FLOAT*. V opačném případě má hodnotu *NULL*.
- **sour_E FLOAT** - východní zeměpisná délka. Pokud je známá, je uložena v desetinném dělení, např. *14.6398* jako datový typ *FLOAT*.
- **ku INT(11)** - identifikátor (interní) katastrálního území, v němž kaple leží. Tato hodnota se odkazuje na sloupec *id* tabulky *ku*.
- **donator VARCHAR(100)** - donátor neboli stavebník kaple. Jméno, příp. stručné vysvětlení donátora kaple, např. *K. Leopold, hrabě Carreto Millesimo*.
- **note BLOB** - poznámka ke kapli. Tedy text, přímo nespádající do *description*, ale také související s kaplí, „pod čarou“. Datový typ *BLOB* opět umožňuje délku řetězce delší než 255 znaků.

Tabulka *photos*, která má v tuto chvíli největší počet řádků, obsahuje informace o fotografiích (tedy ne fotografie samotné), především jejich jména - tedy jména souborů, podle kterých je posléze možné fotografie načítat ze složky a zobrazit. Tabulka má 11 sloupců:

- **id INT PRIMARY KEY NOT NULL auto_increment** - id dané fotografie. MySQL přiřadí tuto hodnotu automaticky při vytvoření nového řádku (tedy nahrání nové fotografie na server) a zajistí, aby hodnota byla jedinečná.
- **name VARCHAR(50)** - název souboru (prozatím jen typu JPEG), pod kterým se skrývá daná fotografie. Tento řádek je asi nejdůležitější, protože umožňuje identifikaci správného souboru a jeho zobrazení. Datový typ umožňuje ukládat názvy fotografií s délkou do 50 znaků.
- **autor VARCHAR(40)** - autor nebo zdroj fotografie. Pokud neexistuje nebo není známý, obsahuje hodnotu *NULL*.
- **id_objektu INT(11)** - identifikátor kaple, případně jiného objektu (v budoucnu), ke kterému je fotografie vztažena. Tento sloupec odkazuje na sloupec *id* z tabulky *chapels*.
- **x_size INT(11)** - sem se ukládá šířka fotografie v pixelech. V PHP je potom totiž pravděpodobně jednodušší a rychlejší tento jednou zjištěný údaj načítat spolu s ostatními informacemi z databáze než ho pokaždé znovu počítat.

4.3. VYUŽITÍ DATABÁZE

- **y_size INT(11)** - výška fotografie fotografie v pixelech. Platí to samé jako u sloupce *x_size*.
- **size INT(11)** - velikost fotografie v bajtech. Také se vyplatí údaj jednou spočítat a uložit do databáze než zjišťovat při každém požadavku na informace o fotografii.
- **date DATE** datum pořízení fotografie (pokud existuje nebo je známé). Uložené ve speciálním typu pro ukládání data (kalendářního, bez času) - „rok-měsíc-den“, např. *2011-02-26*.
- **note VARCHAR(200)** - stručný popis nebo poznámka k fotografii, například popis toho, co zachycuje. Datový typ umožňuje délku popisku maximálně 200 znaků.
- **titul_exist INT(11)** - informace o tom, zda již existuje větší verze náhledu fotografie (potřebná jako titulní náhled). Hodnota 1 značí, že existuje, 0 znamená že neexistuje.
- **titul_set INT(11)** - informace o tom, jestli je tato fotografie právě nastavená jako titulní (k detailu kaple). Opět 1 značí že je, 0 znamená že není.

Další tabulka se jmenuje *odkazy*. Obsahuje všechny internetové (URL) odkazy, prozatím k samotným kaplím nebo jiným objektům, uložené přehledně v samostatné tabulce, protože k jedné kapli může být přiřazeno více odkazů. Tabulka má 4 sloupce:

- **id INT(11) PRIMARY KEY NOT NULL auto_increment** - jednoznačný identifikátor odkazu, podle kterého odkaz PHP, příp. JavaScript pozná. Při vytvoření nového odkazu MySQL automaticky nastaví jedinečnou hodnotu (vyšší než předchozí).
- **url VARCHAR(300)** - URI adresa odkazu která je uložena bez schématu (tedy bez „http://“, např. *lfgm.fsv.cvut.cz*).
- **popis VARCHAR(200)** - stručný popis odkazu, nepovinná položka. Pokud je nastaven, použije se jako text, na který bude uplatněn hypertextový odkaz.
- **id_kaple INT(11)** - identifikátor kaple nebo jiného objektu, ke kterému je odkaz vztažen. Odkazuje na sloupec *id* tabulky *chapels*.

Tabulka *chapels* se svým sloupcem *ku* odkazuje na tabulku *ku*. Ta obsahuje všechna katastrální území, ležící v okolí poutní cesty. Kaple (nebo později jiný objekt) může být tedy přiřazena k určitému katastrálnímu území. Tabulka *ku* má 4 sloupce:

- **id INT(11) PRIMARY KEY NOT NULL auto_increment** - jednoznačný identifikátor katastrálního území, jehož jednoznačnost MySQL automaticky zajistí. Na tuto hodnotu se odkazuje atribut *ku* tabulky *chapels*.
- **name VARCHAR(40)** - jméno daného katastrálního území (maximální délka řetězce 40 znaků). Např. *Radonice u Prahy*.

- **kod_ku INT(11)** - šestimístný kód katastrálního území podle ČSÚ.
- **www VARCHAR(100)** - internetová adresa, odkazující na stránky katastrálního území nebo obce (většinou), ve kterém KÚ leží. Opět uložena bez schematu „http://“.

Poslední tabulkou je *users*. Ta s ostatními není nijak propojená, stojí zvlášť, a obsahuje seznam všech registrovaných uživatelů systému. Jednoznačným identifikátorem zde není id, ale přímo uživatelské jméno (PHP aplikace a JavaScript zabrání registrace nového uživatele se jménem, které již existuje). Tabulka má 4 sloupce:

- **name VARCHAR(20) PRIMARY KEY NOT NULL** - uživatelské jméno sloužící zároveň jako primární klíč. Délka je omezená na 20 znaků a stejně jako heslo musí být povinně vyplněno.
- **password CHAR(40) NOT NULL** - heslo, uložené v hashované podobě. Zašifrovaná podoba hesla se do databáze ukládá z důvodu bezpečnosti. Pokud by se případný útočník do databáze s nešifrovanými hesly dostal, snadno by je přečetl a zneužil. Šifrovací funkce *sha1* vrací řetězec o délce přesně 40 znaků, datový typ je tedy *CHAR*, který má přesnou délku. Typ *CHAR(n)* má tu vlastnost, že pokud je mu přiřazen řetězec kratší než *n*, zbylé znaky se zaplní mezerami.
- **mail VARCHAR(50)** - e-mail uživatele. Je důležitý zejména při registraci, kdy má administrátor možnost si ověřit identitu uživatele.
- **activ INT(11)** - indikuje, zda je uživatel aktivní nebo nikoliv. V praxi to znamená, že po registraci uživatele administrátor zprovozní účet změnou této hodnoty (z 0 na 1), příp. později účet zakáže (změnou hodnoty 1 na 0).

4.4 Popis jednotlivých skriptů

V této kapitole projdu jednotlivé skripty a popíšu stručně jejich obsah (funkce, proměnné, hlavní myšlenky ap.).

4.4.1 PHP skripty

Základní schéma každé jednotlivé stránky (napsané v PHP) je v projektu následující:

```
<?php
$title = 'HTML titulek';
$menu = 'polozka_menu';
include 'includes/head.php';
include 'includes/header.php';
include 'includes/opravneni_admin.php';
?>
<h2>Nadpis stranky</h2>
//vlastni obsah stranky...
```

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

```
<?php
    $open_page = 'nazev_teto_stranky.php';
    include 'incudes/end.php';
?>
```

Proměnná `$title` obsahuje titulek, který se zobrazí v záložce prohlížeče, tedy obsah HTML tagu `<title></title>`. `$menu` značí položku menu, která je právě aktivní. Tato proměnná nemusí být určena, protože ne na každou stránku se lze dostat z menu. Funkce příkazu `include` je popsána výše, a tak stačí dodat, že zde se načte nejprve soubor s hlavičkou HTML (kde se dynamicky doplní obsah proměnné `$title`), poté záhlaví samotné stránky včetně bočních panelů (a menu), a v posledním zahrnovaném souboru se kontroluje, za je uživatel přihlášený (či dokonce jako admin). Tento soubor se zahrnuje jen do některých skriptů. Následuje Nadpis stránky a její obsah, už přímo jako kód HTML (pokud je to možné). Nakonec se opět s pomocí PHP (v některých případech) určí proměnná `$open_page`, a tato se použije v posledním zahrnutém souboru `end.php`. Ten vypíše zápatí stránky. Následuje popis těchto skriptů, které jsou zahrnovány do více různých souborů.

4.4.1.1 Zahrnované skripty

spojeni.php:

Tento skript zajišťuje spojení s databází, vytvoří nový objekt *mysqli* (parametry konstruktoru jsou hostitel, uživatel, heslo a název databáze). Dále je nutné nastavit databázi v PHP správné kódování. Pokud se nepodaří navázat spojení s databází, skript vypíše chybu.

```
<?php
    $conn = new mysqli("localhost", "user", "heslo", "viasancta");
    $conn->set_charset("utf8");
    if (mysqli_connect_errno())
    {
        die("Funkce mysqli_connect selhala: " . mysqli_connect_error());
    }
?>
```

session.php:

Skript *session.php* (který je zahrnován na počátek každého souboru *header.php*) spravuje uživatelskou relaci. Konfigurační parametr `session.use_cookies` nastaví předávání identifikátoru relace pomocí souborů cookies, druhý parametr `session.use_only_cookies` nastaví, že identifikátor relace poslaný prostřednictvím URL není povolen. Následuje volání funkce `session_start()`, která nastaví modul uživatelské relace a nastaví určitá záhlaví. Nakonec je definována funkce `overeni_uid()`, ověřující zda je deklarováno id relace.

```
<?php
    ini_set('session.use_cookies', 1);
    ini_set('session.use_only_cookies', 1);
    session_start();
    function overeni_uid()
    {
        if (!isset($_SESSION['uid']) || !$_SESSION['uid'])
        {
            header('Location: login.php');
        }
    }
?>
```

opraveni.php:

Skript jednoduše s pomocí identifikátoru relace zkontroluje, zda je uživatel přihlášený. Pokud ne, vypíše varovné hlášení, zahrne ještě soubor *end.php* a běh skriptu ukončí.

```
<?php
    if (!isset($_SESSION['uid']) || ($_SESSION['uid'] != '4'))
    {
        echo '<h3 style="color: red;">Nemáte oprávnění pro tuto stránku!!
            </h3>';
        include 'end.php'; //konec web. stranky
        exit;
    }
?>
```

opraveni_admin.php:

Tento skript funguje podobně jako skript *opraveni.php*, kontroluje ale přímo parametr *name* uložený v relaci, čímž nepovolí načtení stránky pro jiné uživatele než pro jednoho konkrétního (v tomto případě pro administrátora). Tyto dva skripty jsou zahrnovány prakticky jen pro případy, kdyby někdo zadal URL adresu ručně, protože běžnému, nepřihlášenému uživateli se ani odkazy na tyto „VIP“ stránky nezobrazí.

```
<?php
    if (!isset($_SESSION['name']) || ($_SESSION['name'] != 'admin'))
    {
        echo '<h3 style="color: red;">Nemáte oprávnění pro tuto stránku!!
            </h3>';
        echo '<h4>Přihlaste se prosím jako admin!</h4>';
        include 'end.php'; //konec web. stranky
        exit;
    }
?>
```

head.php:

Jediné, co tento skript provede, je že vypíše do výstupu HTML hlavičku se všemi základními informacemi o stránce (použité kódování, zahrnutí souborů JavaScript uložených ve zvláštní složce) včetně titulku stránky, jehož první část se jako řetězec definuje před zahrnutím tohoto skriptu.

head_m.php, head_g.php:

Tyto skripty se od výše popsaného *head.php* liší, ale vycházejí z něj. První má funkci HTML hlavičky pro stránku s mapou GoogleMaps (*mapa.php*) a proto kromě všech informací také definuje adresu `http://maps.google.com/maps/api/js?sensor=true` s funkcemi pro tvorbu této interaktivní mapy. Dále se ve skriptu s pomocí PHP naplní dvourozměrné javascriptové pole `kaple[][]` informacemi o kaplích z databáze. Nejprve se naváže spojení s databází a odešle se dotaz na informace o existujících kaplích:

```
include "spojeni.php";
$mdotaz = "SELECT * FROM chapels WHERE state = 1;";
$mvysledek = $conn->query($mdotaz);
```

Následuje výpis hlavičky HTML, poté v cyklu `while` načtení jednotlivých informací o kapli, včetně souřadnic (ty jsou nejdůležitější). V tom samém cyklu proběhne ke kapli ještě „poddotaz“ na titulní fotografii ke kapli:

```
$poddotaz = "SELECT name FROM photos WHERE id_objektu=$mid
            AND titul_set=1;";
$podvysledek = $conn->query($poddotaz);
while ($podobjekt = $podvysledek->fetch_row())
{
    $titulni_fotka = $podobjekt[0];
}
```

Zde proměnná `$mid` obsahuje id kaple. Metodou `query($poddotaz)` se připravený dotaz vždy spustí a metodou `fetch_row()` se z objektu výsledku dotazu načte jeden řádek.

Poté se řetězec popisu zkrátí na 130 znaků, aby se vešel do `googlemaps infookénka` *InfoWindow* (k tomu slouží funkce `substr()` která má parametry řetězec a index prvního a posledního znaku) a s pomocí regulárního výrazu se odstraní odřádkování. To by totiž způsobilo problémy v javascriptu, kde jsou proměnné na více řádků zakázány. A protože v tomto případě je postup od PHP, přes HTML, JavaScript zpět k HTML (více později), stačí aby v řetězci místo odřádkování byla značka `
`, což v řetězci uloženém v databázi je. Na konci cyklu se vypíše (pořád se vepisuje do tagu `<script type="text/javascript">`) definice jedné kaple, tzn. v každém cyklu se naplní jeden prvek pole dalším polem. Např. pro text, popis aktuální kaple se provede tento příkaz:

```
echo '
kaple[' . $mid . ']["text"] = "' . $description . '";';
```


Odrádkování je zde kvůli přehlednosti vygenerovaného HTML kódu, a v proměnné `$description` je uložen řetězec textu ke kapli.

Skript `head_g` se od základního liší jen v tom, že jej zahrnuje jen hlavní stránka celého projektu, což znamená uvedení navíc klíčových slov, a také metatagu `<meta name="distribution" content="global">`, který znamená právě že toto je hlavní stránka projektu.

header.php

Tento skript vždy vygeneruje HTML kód zobrazující vše, co je u jednotlivých stránek společné, tj. záhlaví stránky, levé menu, horní lištu a pravou nabídku (obsahující zatím jen náhodnou kapli). Kromě toho také naváže spojení s databází.

Nejprve se vytvoří záhlaví (z bitmap uložených ve složce `foto_ostatni`), následuje vykreslení menu. Využívá se přitom několika definovaných CSS-tříd, v závislosti, zda je položka menu aktivní či neaktivní a zda je přístupná jen pro přihlášeného uživatele (či dokonce administrátora) nebo pro všechny. V zahrnujícím skriptu je definována proměnná `$menu` a zde se s pomocí podmínek určí, která položka menu je právě aktivní. Ta se potom vykreslí jinak. Dále se testuje, zda je uložena relace (neboli zda je uživatel příp. administrátor přihlášený) a v závislosti na tom zobrazí nebo nezobrazí tyto „VIP“ položky menu, např.:

```
if (isset ($_SESSION['uid']) && ($_SESSION['name'] == 'admin'))
{
    if ($menu == 'vkladaniadm')
    {
        echo $miadma;
    }
    else
    {
        echo $miadm;
    }
    echo '
<a href="moje_sql.php">Vkládání informací (admin)</a>
    </li>';
    if ($menu == 'sprava')
    {
        echo $miadma;
    }
    else
    {
        echo $miadm;
    }
    echo '
<a href="sprava.php">Správa uživatelů</a>
    </li>';
}
```

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

Proměnné `$miadma` a `$miadm` jsou řetězce - HTML položky seznamu - s různou CSS třídou.

Poté ještě proběhne vygenerování náhodné kaple a zobrazení odkazu na ní. S pomocí funkce `rand()` se vygeneruje náhodné číslo od 1 do 44, poté se odešle dotaz databázi, tentokrát s pomocí vázání proměnných:

```
include 'spojeni.php';
$dotaz = 'SELECT nazev_zkr, number, state FROM chapels
WHERE id = ' . $ch . '>';
$prikaz = $conn->prepare($dotaz);
$prikaz->execute();
$prikaz->bind_result($name, $number, $state);
while ($prikaz->fetch())
{
    $chname = $name;
    $chnumber = $number;
    $chstate = $state;
}
```

Toto je druhá možnost komunikace s databází MySQL. Dál, pokud má kaple status existující, vypíše PHP její ikonu vytvořenou v programu *Zoner Photo Studio* a uloženou ve složce *foto_nahledy_for_kaplicky*, pokud ne, použije univerzální ikonu vytvořenou v GIMPu a uloženou tamtéž. Doplní její číslo a název a vše vygeneruje jako odkaz na detail dané kaple.

Dále se vykreslí horní lišta, a pokud je uživatel přihlášený, vypíše se „Přihlášen(a) jako (jméno_uzivatele)“ s odkazem na úpravu vlastního účtu.

```
if (isset ($_SESSION['uid']) && ($user = $_SESSION['name']))
{
    echo '<div id="loginfo">Přihlášen(a) jako ';
    echo '<a id="loginfoname" href="ucet.php">' . $user . '</a></div>';
}
```

Superglobální proměnná `$_SESSION[]` obsahuje informace o relaci, tedy i jméno uživatele.

end.php

Tento jednoduchý HTML kód se zahrnuje také do každého skriptu a vypíše zápatí stránky včetně odkazů na přihlášení a odhlášení uživatele. V posledním řádku ještě PHP ukončí spojení s MySQL.

4.4.1.2 kaplicky.php

Zdrojové kódy jsou bohatě komentované přímo v souborech, a tak popis zde bude o něco stručnější.

Soubor *kaplicky.php* zobrazuje všechny kaple poutní cesty, resp. ikony s jejich náhledy a odkazy na podrobnější informace. Ve skriptu se na počátku zahrne soubor *kaplicka-Nahled.php*, jenž obsahuje deklaraci stejnojmenné třídy a je umístěn ve složce *classes*. Z HTTP metody *GET* (tedy z URL adresy) se získá informace, které kapličky je potřeba zobrazit. Implicitně je proměnná *show* nastavena na *'vsechny'*. To znamená zobrazení existujících i neexistujících kapliček. S pomocí tohoto parametru se také sestaví dotaz na kapličky. Při načítání výsledku v cyklu metodou *fetch()* se vždy vytvoří nová instance třídy *kaplickaNahled* a na tuto instanci se použije metoda *zobrazNahled()*.

Dále se vypíše odkazy na různá zobrazení kaplí, přičemž se s pomocí přepínače *switch* ten aktuální vypíše jako prostý text.

Třída *kaplickaNahled* má definováno 6 vlastností a 3 metody. Metoda zvaná *__construct* je v PHP vyhrazená pro konstruktor. Definuje vlastnosti a zavolá metodu *sestavNahled()*:

```
function __construct($_jmeno, $_cislo_poradi, $_foto, $_typ,
    $_poradi_nahled, $_identifikator)
{
    $this->jmeno = $_jmeno;
    $this->cislo_poradi = $_cislo_poradi;
    $this->foto = $_foto;
    $this->typ = $_typ;
    $this->poradi_nahled = $_poradi_nahled;
    $this->identifikator = $_identifikator;

    $this->sestavNahled();
}
```

Metoda *sestavNahled()* jednoduše sestaví HTML kód pro vykreslení kaple, přičemž podle pořadí (je konstruktoru předáno jako parametr) spočítá její přesnou pozici v pixelech. Podle stavu kaple také použije jiný identifikátor pro kaskádové styly, stejně tak podkladový obrázek. V případě, že kaple již neexistuje, vybere černobílou ikonu vytvořenou v GIMPu (stejně jako u náhodné kaple, viz výše). Na celek se také vypíše pořadí a jméno kaple. Metoda *zobrazNahled* připravený kód jednoduše vypíše:

```
function zobrazNahled()
{
    print $this->kod;
}
```

4.4.1.3 detail_kaple.php

Tento skript zobrazí podrobné informace o kapli, včetně všech fotografií. Kterou kapli konkrétně, závisí na metaproměnné *id* získané u URI metodou *GET*. Např. URI adresa „*http://localhost/viasancta/detail_kaple.php?id=23*“ zobrazí informace o 23. kapli (Nezamylické).

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

Na počátku skriptu je ještě definována funkce `decToDeg()`, která převádí stupně (zde konkrétně souřadnice) z desetinného na šedesátinné dělení:

```
function DecToDeg($sour_vstupni_double, &$stupne_int, &$minuty_int,
    &$vteriny_int)
{
    $stupne_int = (int)$sour_vstupni_double;
    $mh = 100*((( $sour_vstupni_double-(double)$stupne_int)/10)*6);
    $minuty_int = (int)($mh);
    $v = ((( $mh-(double)$minuty_int)/10)*600);
    $vteriny_int = (int)((( $mh-(double)$minuty_int)/10)*600);
    if ($v - (double)$vteriny_int > 0.5)
    {
        $vteriny_int++;
    }
}
```

Nejprve je ze superglobální proměnné `$_GET` získáno číslo kaple a odeslán dotaz na všechny informace v databázi k této kapli. Pokud je navíc ke kapli zadáno katastrální území, odešle se dotaz na podrobnější informace také na tabulku *ku*. Poté se postupně vypíší základní údaje o kapli, pokud jsou v databázi (jméno, upřesňující jméno, pořadové číslo, zasvěcení, donátor, katastrální území - to dokonce jako odkaz na stránky obce). Souřadnice, pokud jsou známy, se s pomocí výše uvedené funkce přepočítají do šedesátinného dělení a zobrazí.

Poté se odešle dotaz k tabulce *photos* na fotku, která je nastavena jako titulní a tato se zobrazí ze složky *foto_kaplicky_nahledy_vetsi*, kde je již uložena ve správné velikosti a tak načítání netrvá dlouho. Pokud kaple již neexistuje a fotografie také ne, jednoduše se nezobrazí. Dále se vypíší informace o kapli, modlitby mariánského a svatováclavského cyklu a případné poznámky. Poté se odešle další dotaz, tentokrát na tabulku *odkazy*, který vrátí odkazy související s danou kaplí. Pokud má výsledek nejméně jeden řádek, vypíše se.

Poslední databázový dotaz hledá v databázi, v tabulce *photos*, všechny fotografie k dané kapli a informace o nich. Pro správné, české zobrazení data pořízení fotografie je potřeba nastavit národní nastavení zobrazování data a času:

```
setlocale(LC_TIME, "csy");
```

Velikost fotografie je ještě přepočítána na *kB*, pro lepší čitelnost. Datum je poté převedeno z formátu MySQL *date* do přívětivější podoby., např. *2011-02-26* je převedeno na *26. únor 2011*:

```
$datum = strftime('%#d. %B %Y',strtotime($Fdate));
$datum = mb_convert_encoding($datum,"UTF-8","ISO-8859-2");
```

`$Fdate` je původní datum z databáze, funkce `strtotime()` převede datum z řetězce (různé formáty jsou dané, funkce rozumí například těm z normy ISO 8601) na časové

razítka, což je počet vteřin uplynulých od 1.1.1970. Funkce `strftime()` tento údaj převede zpět na řetězec, a to podle uvedeného formátovacího řetězce (den. měsíc rok). Pro správné české zobrazení data je ještě potřeba s pomocí funkce `mb_convert_encoding()` tento řetězec převést na kódování UTF-8.

Následuje zobrazení malého náhledu fotografie (ze složky *foto_kaplicky_nahledy_mensi*), sloužící zároveň jako odkaz na obrázek v plné velikosti a přehledné zobrazení informací o ní (popisek, pokud je vložen, velikost, jméno souboru, rozměry, datum pořízení, autora, zda je fotografie nastavena jako titulní). Pokud je uživatel přihlášen, zobrazí se navíc odkaz na úpravu informací a pokud je přihlášen jako administrátor, tak také odkaz na smazání.

4.4.1.4 `reg.php`

Skript *reg.php* zajišťuje registraci uživatele do systému. Nejprve se kontroluje, zda již nebyl odeslán registrační formulář, poté se odešle databázový dotaz na kontaktní mail administrátora. Ten se vypíše jako kontakt pro schválení registrace. Formulář obsahuje tři povinná pole, kontrolovaná navíc javascriptově: uživatelské jméno, heslo a e-mail. Po odeslání formuláře se z proměnné `$_POST` načtou zadané řetězce, dalším databázovým dotazem se zkontroluje, zda už takové uživatelské jméno neexistuje a pokud ne, heslo se zašifruje funkcí `sha1()` a vše se odešle do databáze do tabulky *users*. Hodnota *activ* je odeslána 0, aby se registrovaný zatím nemohl přihlásit. V případě úspěchu se vypíše zpráva informující o registraci a o schválení administrátorem.

4.4.1.5 `login.php`

Pokud je uživatel již registrovaný, má možnost se přihlásit. K tomu slouží skript *login.php*. S pomocí supreglobální proměnné `$_SESSION` se nejprve zjistí, zda uživatel není přihlášen. Pokud ano, vypíše se upozornění. V opačném případě se testuje, zda již není vyplněn přihlašovací formulář. Pokud není, následuje HTML kód s tímto formulářem. Ten obsahuje dva textové vstupy a odesílací tlačítko. Dále také odkaz na registraci uživatele. Pokud je formulář odeslán, z proměnné `$_POST` je získáno jméno a heslo. S pomocí databázového dotazu se zjistí, zda uživatel existuje, zadal správné heslo a zda je již povolen administrátorem (hodnota *activ* je rovna 1). Pokud ano, vypíše se hlášení o úspěchu a do pole `$_SESSION` se uloží parametry *uid* a *name*.

V případě neúspěchu je vypsáno příslušné chybové hlášení.

4.4.1.6 `logout.php`

Skript na odhlášení se spustí v případě kliknutí na „Odhlášení“ a nezahrnuje žádné hlavičkové soubory. Nejprve je potřeba znovu inicializovat relaci s pomocí funkce `session_start()`, poté proměnné `$_SESSION` přiřadit prázdné pole a nakonec relaci zrušit s pomocí příkazu `session_destroy()`. Nakonec proběhne přesměrování buď na hlavní stránku, a nebo (pokud byla ze zahrnujícího skriptu zadána) na stránku otevřenou předtím. Proto není potřeba zde zahrnovat záhlaví, zápatí ap.

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

4.4.1.7 insert.php

Pokud se uživatel přihlásí (ne jako administrátor), má možnost spustit tento skript kliknutím na položku v menu. Po nezbytném zahrnutí souborů včetně *opraveni.php* se provede dotaz na určité informace o kaplích. U víceřádkových položek (popis, cykly a poznámka) se s pomocí regulárních výrazů (a funkce `preg_replace()`) nahradí odřádkování a uvozovky, a sestaví se dva řetězce. První vytvoří soubor HTML tagů `option`, což jsou prvky roletového menu `select`, druhý postupně vytváří a plní několikarozměrné javascriptové pole `info_vse`. Ke každé kapli se také provede dotaz na všechny odkazy, které se zapíší do javascriptového pole včetně jejich počtu. Definice pole se potom vypíše do HTML:

```
echo '<script type="text/javascript">
var info_vse = new Array(44);
' . $pole . ' </script>';
```

Poté se vypíše, pokud ještě nebyl odeslán, celý formulář. Jeho prvky jsou výběrové pole *select*, do něhož se s pomocí PHP vepíšu volby *option* obsahující jednotlivé kaple (jméno a pořadové číslo), dále velká textová vstupní pole (*textarea*) pro popis kaple, mariánský a svatováclavský cyklus a pole pro poznámku. Dále dva textové vstupy pro přidání nového odkazu URL a jeho popisu. Toto pole je ale implicitně (s pomocí stylu CSS) skryté, zobrazí se až po kliknutí na tlačítko „Přidat odkaz“ (toto a další souvislosti jsou ošetřeny s pomocí JavaScriptu). Na konci formuláře je ještě hlavní odesílací tlačítko (*submit*).

Pokud byl formulář odeslán (a zkontrolován s pomocí JavaScriptu), s pomocí databázového dotazu se data, která byla vyplněna uloží do databáze (do tabulky *chapels*, příp. *odkazy*). Pokud formulář (resp. kaple - při výběru kaple se totiž díky javascriptové funkci `vyplnMalaEditace()`) zobrazilo tolik vyplněných textových polí, kolik ke kapli patří odkazů) obsahoval více odkazů, ty se odeslaly s pomocí pole. Toto pole se nyní postupně načte a odkazy uloží. Aby bylo jasné, které odkazy je potřeba upravit (SQL příkaz `UPDATE`), odešle se ještě jeden databázový dotaz na *id* odkazů dané kaple. Pokud se ve formuláři nějaký smazal, nyní se celý řádek v databázi smaže (SQL příkaz `DELETE`). Poté se ještě s pomocí SQL příkazu `INSERT` doplní případný nově přidáný odkaz.

4.4.1.8 foto_insert.php

Skript slouží k vložení nové fotografie. To je ale zatím podmíněno příslušností fotografie k určité kapli. S pomocí podmínek `if` se skript načítá ve třech různých verzích.

- obrázek ještě nebyl vložen ani uložen do databáze
- obrázek již byl vložen, ale ještě nebyl uložen do databáze
- obrázek již byl vložen i uložen do databáze

Pokud **obrázek ještě nebyl vložen**, vypíše se jednoduchý formulář pro nahrání souboru. Ten obsahuje 3 položky: *file* - pro nahrání souboru, *hidden* - skryté pole pro maximální velikost souboru (nemusí fungovat) a *submit*, tedy odesílací tlačítko.

Pokud **obrázek již byl vložen, ale zatím není uložen v databázi**, načte se ze superglobální proměnné `$_FILES['obrazek']`. Pole `$_FILES[]` obsahuje i další informace o načteném souboru, např.:

- `$_FILES['soubor']` - handler daného souboru - označení `soubor` je dáno parametrem `name` odeslaným z formuláře
- `$_FILES['soubor']['name']` - původní název souboru
- `$_FILES['soubor']['error']` - případné chybové konstanty
- `$_FILES['soubor']['size']` - velikost souboru v bytech
- `$_FILES['soubor']['type']` - MIME typ načteného souboru
- `$_FILES['soubor']['tmp_name']` - dočasné jméno souboru

Pokud existuje proměnná `$_FILES['soubor']['error']`, jsou s pomocí přepínače `switch` ošetřeny všechny možné chyby a do proměnné `$chybova_zprava` je vždy uložen text k dané chybě. Dále se také zkontroluje MIME typ souboru (povoleny jsou zatím jen typy `image/jpeg` a `image/pjpeg`):

```
$povolene_mime = array('image/jpeg','image/pjpeg');
if (!in_array($_obrazek['type'], $povolene_mime))
{
    $chybova_zprava = 'Povoleny jsou pouze formáty JPG a JPEG, <a
        href="foto_insert.php">opakujte načtení</a>.';
    break;
}
```

Pokud je dosud vše v pořádku, prohledá se funkcemi `opendir()` a `readdir()` složka *foto_kaplicky*, zda již neexistuje soubor se stejným názvem:

```
$slozka_obrazky = opendir("./foto_kaplicky");
while($soubor = readdir($slozka_obrazky))
{
    if ($soubor == $nazev_foto)
    {...
```

Pokud je takový soubor nalezen, k názvu nového souboru se s pomocí regulárních výrazů přidá „-(2)“:

```
...
$rozfragovano = preg_split('@\.',
    $soubor,
    -1,
    PREG_SPLIT_NO_EMPTY |
    PREG_SPLIT_OFFSET_CAPTURE);
```

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

```
$pozice_koncovky = $rozfragovano[count($rozfragovano)-1][1];
$bez_koncovky = substr($soubor, 0, $pozice_koncovky-1);
$koncovka = substr($soubor, $pozice_koncovky);
$nazev_foto = $bez_koncovky . "_{(2)}." . $koncovka;
```

Zde se na chvíli zastavím. Zajímavá je tu funkce `preg_split()`, která řetězec (v tomto případě `$soubor`) rozdělí na podřetězce a uloží do pole. Prvním parametrem je vzor který slouží jako rozdělovač řetězce. Vzor

```
'@\.@'
```

znamená jednoduše tečku (zavináče jsou zde v úloze oddělovače, zpětné lomítko ruší speciální význam tečky, která jinak odpovídá přesně jednomu znaku). Druhým parametrem je řetězec, který chceme rozdělit. Třetí parametr je počet dílů na kolik řetězec chceme rozdělit. Jasně by to mělo být v následujícím příkladě:

```
$text='Kobyła má malý bok.';
$text=preg_split('@[\W]+@', $text, 2)
print_r($text);
```

vypíše

```
Array
(
 [0] => Kobyła
 [1] => má malý bok.
)
```

Pokud se tento parametr zadá `-1`, ignoruje se, a je možné předat funkci další parametr(y), a to svislou čarou (rourou) oddělené příznaky. Zde použité:

- `PREG_SPLIT_NO_EMPTY` - zabrání vkládání prázdných prvků do výsledného pole [11].
- `PREG_SPLIT_OFFSET_CAPTURE` - funkce vrátí dvourozměrné pole obsahující text a index prvního znaku nalezeného vzoru [11].

Dále se tedy určí pozice koncovky souboru, s pomocí funkce `substr()` oddělí a nakonec místo tečky vloží řetězec „{(2)}.“. Tím se zajistí především to, aby se soubor se shodným názvem nepřepsal.

Také se vytvoří verze názvu pro uložení souboru do složky. Pokud by se používaly jen názvy bez diakritiky a mezer, byla by situace jednodušší. To ale ve skutečnosti nemusí být dodrženo, a tak se pro potřeby manipulace s fotografiemi (vč. jejich ukládání) vytvoří název také v kódování ISO 8859-2. K tomu slouží jednak funkce `iconv()` z rozšíření `iconv` a jednak funkce `mb_convert_encoding()` z rozšíření `mbstring`, která je použita právě zde.

Dále je potřeba soubor (nyní již máme zajištěné že jím je obrázek ve formátu JPEG) přemístit z dočasného umístění (a jména) na název nový včetně složky. Jako nový název se použije právě ten původní (či upravený), před který se vepíše jméno adresáře *foto_kaplicky*.

K přesunu nového souboru slouží funkce `move_uploaded_file`:


```
if(!@move_uploaded_file(  
    $obrazek['tmp_name'],  
    $cilovy_adresar . $nazev_foto_utf  
)){...}
```

Symbol @ před voláním funkce je tzv. *operátor ticha*, který zabrání zobrazení případných chybových zpráv. Rozměry obrázku v pixelech se získají takto:

```
$velikosti = getImageSize($cilovy_adresar . $nazev_foto_utf);  
$sirka = $velikosti[0];  
$vyska = $velikosti[1];
```

Pokud je vše v pořádku, zobrazí se nyní náhled nahraného obrázku a jeho velikost v kB. Pokud byl nalezen druhý se shodným jménem, zobrazí se oba, spolu s názvy a velikostmi. Dál se vypíše formulář, umožňující upravit informace o fotografii. Do formulářového prvku `select` se opět v cyklu vepíše všechny kaple (k tomu slouží databázový dotaz). Pozastavím se ale u načítání dat z Exifu.

Exif je metodou používanou v digitálních fotoaparátech, programech a dalších zařízeních k ukládání metadat k obrázku. V okamžiku pořízení snímku se ukládá například clona, expozice, typ fotoaparátu, zda byl použit blesk, nebo dokonce náhled fotografie a desítky dalších informací včetně přesného času pořízení snímku (to ale předpokládá že je čas správně nastavený ve fotoaparátu). V různých programech (například právě v *Zoner Photo Studio* nebo v *Adobe Photoshop*) je pak možné k fotografii s pomocí *Exifu* doplnit třeba autora fotografie nebo různé popisky. Podrobná specifikace Exifu je k nalezení na internetové adrese <http://exif.org/specifications.html>. V PHP se touto metodou zabývá rozšíření `php_exif.dll` které je potřeba spolu s rozšířením `php_mbstring.dll` povolit v konfiguračním souboru `php.ini`. V našem skriptu jsme využili hodnoty `'Copyright'` (autor fotografie), `'ImageDescription'` a `'DateTimeOriginal'` - datum pořízení fotografie. Aby to nebylo tak jednoduché, hodnoty bylo nutné překódotovat a datum převést do upřednostňovaného formátu *ISO 8601*:

```
ini_set('exif.encode_unicode', 'UTF-8');  
$exif = exif_read_data("foto_kaplicky/" . $nazev_foto);  
$exif['Copyright']=mb_convert_encoding($exif['Copyright'], "UTF-8",  
    "ISO-8859-2");  
$exif['ImageDescription']=mb_convert_encoding($exif['ImageDescription'],  
    "UTF-8","ISO-8859-2");  
if (isset($exif['DateTimeOriginal']))  
{  
    $datum_porizeni = $exif['DateTimeOriginal'];  
    $datum_porizeni_cele = date("Y-m-d", strtotime($datum_porizeni));  
    $datum_porizeni_rok = date("Y", strtotime($datum_porizeni));  
}
```

Poté se konečně vypíše samotný formulář, kde se v kolonce popisku předvyplní právě hodnota `ImageDescription`, datum pořízení `DateTimeOriginal` a autor `Copyright`. Pokud poslední hodnota neexistuje, předvyplní se jméno uživatele který fotografii vkládá.

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

Spolu s těmito prvky formuláře se odesílají také skrytá pole (`hidden`) s velikostí, rozměry a původním názvem fotografie.

Když **byl obrázek vložen, a zároveň byl odeslán i formulář** (a řádně zkontrolován javascriptovými funkcemi), načtou se nejprve data z formuláře. Poté se velikost pře počítá na MB a zformuluje se databázový dotaz na vložení fotografie do databáze. Samotný soubor se přejmenuje pomocí funkce `rename(puvodni_nazev, novy_nazev)`. Ve skriptu se počítá i s variantou, že název souboru není zadán. V takovém případě se vygeneruje náhradní s pomocí aktuálního času a náhodného čísla:

```
$time = time();
$rand = rand(0,1000);
$novy_nazev = 'kaple_c_' . $id_kaple . '_photo_' . $time . '_' . $rand .
'.JPG';
@rename('foto_kaplicky/' . $puv_nazev_utf, 'foto_kaplicky/' . $novy_nazev);
$chyb_hlaska .= "Soubor nemůže být uložen bez jména, je proto uložen
                pod názvem <span class='zvyrazneno'>'$novy_nazev'
                </span>!  

```

Dále PHP samo vytvoří náhled fotografie vysoký 100 pixelů a ten uloží do složky *foto_kaplicky_nahledy_mensi*. Při načtení webové stránky s detailem kaple se tak zobrazí právě náhledy a ne celé fotografie, což by mohlo při pomalém připojení trvat třeba několik minut (pro desítky fotografií). Manipulace s obrázky je jedna z mnoha schopností grafické knihovny *GD - php_gd2.dll*. Vytvoření kopie a zmenšení fotografie - funkce jsou popsány v komentářích:

```
// vytvoreni nahledu
$nahled_vyska = 100; //jednotna vyska nahledu
$nahled_sirka = $sirka/($vyska/$nahled_vyska); //vypocet sirky nahledu
// vytvoreni obrazku nahledu
$out = ImageCreateTrueColor($nahled_sirka, $nahled_vyska);
// vytvoreni puvodniho obrazku
$puv_obrazek = ImageCreateFromJpeg('foto_kaplicky/' . $name_utf);
// zmenzeni puvodniho obrazku do noveho nahledu
ImageCopyResampled($out, $puv_obrazek, 0, 0, 0, 0, $nahled_sirka,
    $nahled_vyska, $sirka, $vyska);
// ulozeni noveho nahledu do patrcne slozky
ImageJpeg($out, 'foto_kaplicky_nahledy_mensi/' . $name_utf, 90);
// zruseni pracovnich obrazku
ImageDestroy($out);
ImageDestroy($puv_obrazek);
```

Místo funkce `ImageCopyResampled()` lze použít také podobnou funkci `ImageCopyResized()`, která je dokonce rychlejší. Algoritmus pro přepočítání pixelů je ale díky tomu jednodušší a výsledný obrázek znatelně horší. Rozdíl je možné porovnat na obr. 4.2.

Do databáze je také uloženo datum v MySQL formátu `date` a hodnoty sloupců `titul_exist` a `titul_set` jsou na počátku nastaveny z logických důvodů na 0. Poté



Obrázek 4.2: Porovnání kvality zmenšení a přerastrování obrázku s pomocí funkce `ImageCopyResampled()` (vlevo) a `ImageCopyResized()` (vpravo)

se databázový dotaz (`INSERT`) odešle, fotografie spolu se základními informacemi zobrazí a nakonec také vypíšou odkazy na nahrání další fotografie, na detail ke kapli a na úpravu informací o této fotografii - to obstarává následující skript.

4.4.1.9 `foto_zmena.php`

Tento skript zajišťuje úpravu informací již vložené fotografie. Protože velká část řešení a funkcí je podobná ostatním skriptům popsáným výše, zmíním se jen o tom nejpodstatnějším. Tento skript spustíme jedinečně proklikem z odkazu u některé fotografie z detailu kaple. Právo na úpravu fotografie, a tedy správné spuštění skriptu má jen přihlášený uživatel, což je opět ošetřeno zahrnutím na počátek souboru `opraveni.php`. Id fotografie je skriptu předáno s pomocí metody `GET`. V databázovém dotazu na počátku skriptu se hodnoty sloupce `titul_exist` a `titul_set` převedou z jedniček a nul na logický datový typ `boolean`. Další databázový dotaz směřuje k tabulce kaplí (pro formulářový prvek `select`). Obrázek kaple se zde zobrazí zmenšený, ale ten původní. Je totiž jeden, a tak načítání netrvá tak dlouho jako kdyby fotografií bylo 20 jako v detailu kaple. Pod tento větší náhled fotografie se vypíše formulář s informacemi o fotografii včetně přepínače (`radiobuttonu`) s pomocí kterého je možné nastavit tento obrázek jako titulní k dané kapli. Jako skryté formulářové prvky se ještě odešle původní název, šířka, výška, id fota a informace o tom, zda již existuje fotografie ve formě většího náhledu (ve složce `foto.kaplicky_nahledy_vetsi`). Zpracování formuláře probíhá podobně jako toho s přidáním nové fotografie. V několika detailech se však liší. Přejmenovat se zde musí také malý náhled, případně velký, pokud již existuje. Pokud uživatel ve formuláři kliknul na možnost „nastavit fotografii jako titulní“, s pomocí podmínky se nejprve zjistí, zda již existuje větší náhled. Pokud ne, vytvoří se s pomocí `GD` (viz výše) knihovny stejně jako ve vkládání fota, jen s většími rozměry a v databázi se do sloupce `titul_exist` zadá 1. Zároveň se zadá hodnota 1 ve sloupci `titul_set` a u všech ostatních obrázků k dané kapli 0.

4.4.1.10 `mazani_fotografie.php`

Tento skript lze také spustit jen „prokliknutím“ ze stránky detailu kaple, ale na rozdíl od výše popsaného má právo jej plnohodnotně spustit jen administrátor, což zajistí zahrnutí skriptu `opraveni_admin.php`. Identifikátor fotografie je opět skriptu předán

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

prostřednictvím metody `GET`, a je ihned použit k vyhledání informací o obrázku z databáze. Aby zde nedělalo „neplechu“ tlačítko „zpět“ prohlížeče, je zajištěno, že pokud dotaz nevrátí žádný výsledek (což po odeslání formuláře a smazání fotografie nevrátí), skript přesměruje řízení na soubor `kaplicky.php`. Ke smazání fotografie je nutné zadat heslo administrátora. Formulářový prvek `password` se tedy vypíše pod náhled fotografie. Po odeslání (opět javascriptem zkontrolovaného) formuláře se zašifrované heslo porovná s tím vyhledaným v databázi, a pokud souhlasí, smaže se fotografie i její náhled. Dále je prohledána funkcí `readdir()` složka s většími náhledy, a pokud je soubor nalezen, smaže se také a zároveň se nastaví hodnota `1` ve sloupci `titul_exist` pro první fotografii v databázi, která přísluší stejné kapli. Nakonec se také odstraní fotografie z databáze. V případě jakéhokoliv neúspěchu je vypsáno chybové hlášení, ve skriptu se tak značně větví různé podmínky.

4.4.1.11 smazfoto.php

Tento krátký skript slouží jen ke smazání fotografie, v případě že si uživatel před odesláním formuláře rozmyslí vložení nové (viz skript `foto.insert.php`). I zde je nutné zahrnout `opraveni.php`. Jméno fotografie je předáno přes URL metodou `GET`, není jej proto před použitím funkce `unlink()` nutné převádět do kódování *ISO 8859-2*.

4.4.1.12 ucet.php

PHP skript `ucet.php` slouží k jednoduché správě účtu, konkrétně jen změnu e-mailu a hesla. Spustit jej lze klepnutím na uživatelské jméno na horní liště. Částečné skrývání a zobrazování části formuláře je zajištěno Javascriptem. Formuláře jsou vlastně na stránce dva samostatné - jeden na změnu e-mailu (jeden textový vstup pro nový e-mail a odesílací tlačítko) a druhý pro změnu hesla (tři vstupy typu `password` - staré heslo a dvakrát nové heslo a odesílací tlačítko `button`). Na počátku souboru je podmínkami `if` zajištěno zpracování příslušného formuláře, zda byl odeslán. Změna e-mailu je jednoduchou záležitostí a jedná se jen o SQL dotaz `UPDATE`. Se změnou hesla je to složitější. Přesto že to provede před odesláním formuláře JavaScript, jsou porovnána obě nová hesla a také zkontrolováno to staré. JavaScript totiž není v žádném případě vhodný ke kontrole citlivých údajů, lze jej například v prohlížeči snadno zakázat. A tak javascriptová kontrola slouží spíše k tomu, aby se na server zbytečně neodesílala data a skutečná kontrola hesla probíhá až na serveru s pomocí PHP.

V tomto případě je ale bezpečnost řešena ještě více do hloubky. JavaScript těsně před odesláním formuláře hesla smaže a odešle na server místo nich šifrovanou podobu. Není tak možné ani odposlechnout heslo nějakým způsobem z metody `POST`.

4.4.1.13 moje_sql.php

Tento skript zajišťuje prakticky to samé jako `insert.php`, tedy vkládání informací ke kaplím, ale jako přístupný jen pro administrátora je možné u kaplí měnit téměř vše. Pořadové číslo kaple, stav, celé, zkrácené i doplňující jméno kaple, její souřadnice, zasvěcení, je možno z výběrového pole `select` vybrat katastrální území v němž leží,

donátora, popis, obě modlitby, poznámku a odkazy. Jinak je vše řešené stejným způsobem jako právě u *insert.php*, nemá tak smysl skript podrobněji popisovat.

4.4.1.14 sprava.php

Sprava.php je opět soubor přístupný jen administrátorovi a slouží k aktivaci/deaktivaci a mazání jednotlivých uživatelů. Ti se při načtení stránky zobrazí v tabulce. Aktivace či deaktivace uživatelů je řešena s pomocí zatrhávacích políček (checkboxů) a jednoho odesílacího tlačítka. Po odeslání tímto tlačítkem se tak podle uživatele SQL příkazem UPDATE zapíše ke každému uživateli do sloupce `activ` 1 nebo 0. Jména `name` prvků formuláře jsou dynamicky pojmenovaná podle jmen uživatelů. Mazání uživatelů je ale řešeno jednotlivými odesílacími tlačítky, a tak se musí na začátku skriptu projít z databáze jména všech uživatelů a porovnat s odeslaným formulářem:

```
while ($meziradek = $mezivysledek->fetch_row())
{
    $jmeno = $meziradek[0];
    $post = 'smazat_'. $jmeno;
    if (isset ($_POST[$post]))
    {
        if ($_POST[$post] == 'Smazat uživatele')
        {
            $mazaci_dotaz = "DELETE FROM users WHERE name=\"$jmeno\"";
            if ($conn->query($mazaci_dotaz))
            {
                echo "Uživatel <span class=\"$zvyrazneno\">$jmeno</span> byl
                    úspěšně smazán<br /><br />";
            }
        }
    }
}
```

V případě že je nalezen takový uživatel, smaže se SQL příkazem DELETE.

Při výpisu tabulky je u administrátora místo zatrhávacího políčka (checkboxu) text „aktivní“, aby u něj nebylo možné změnit hodnotu.

4.4.2 JavaScript

JavaScript je v projektu využit ve dvou hlavních případech. Implementace *Google Maps* a kontrola formulářů. To první je popsáno podrobněji níže, kontrolu formulářů se pokusím shrnout zde.

JavaScript je interpretovaný, objektový jazyk, určený k tvorbě internetových aplikací. Skript se na rozdíl od PHP posílá přímo v HTML kódu a jeho zpracování proběhne až v prohlížeči (v klientovi). Na tom je tedy JS závislý, ale dnes již funguje téměř na všech prohlížečích.

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

Jak již bylo řečeno, není vhodné JS využívat jako hlavní kontrolu hesel a jiných důležitých údajů, lze jej totiž snadno editovat nebo zakázat. Syntaxe tohoto jazyka je podobná některým dalším příbuzným (Java, C, C++, Python, PHP..), kromě několika důležitějších výjimek. Za prvé je to jazyk *case sensitive*, což znamená že není možné zaměňovat malá a velká písmena, za druhé příkazy není nutné ukončovat středníky, musí ale být jednořádkové. Také není nutné deklarovat typ proměnné, ani klíčovým slovem **var**. Téměř všechny javascriptové funkce v tomto projektu kromě těch souvisejících s mapou jsou umístěny v externím souboru *insert.js*, na který je v HTML hlavičce odkaz:

```
<script type="text/javascript" src="js/insert.js"></script>
```

JavaScriptu je využito například ve formulářích pro změnu informací o kaplích - soubory *insert.php* resp. *moje_sql.php*, kde jsou výběrem příslušné kaple v nabídce **select** spuštěny funkce **vyplnMalaEditace**, resp. **vypln**. Tyto funkce nejprve smažou obsah celého formuláře, poté jej předvyplní v závislosti na vybrané kapli. Také se zároveň vytvoří předvyplněné textové vstupy s odkazy, podle toho kolik jich ke kapli existuje. A jakým způsobem se taková funkce spouští? V příslušném HTML tagu je jako parametr příslušná uživatelská událost, parametrem je tato funkce. Např. pro výběr volby **selectu**:

```
<select name="kaple" onchange="vypln(this,info_vse)" >
```

Uživatelská událost je v tomto případě **onchange** (změna hodnoty políčka - výběr v **selectu**). Pokud funkce rozhoduje o vykonání původního účelu kliknutí (např. hypertextový odkaz nebo odeslání formuláře) - tedy vrací logickou hodnotu, musí i v hodnotě být před voláním funkce **return**. Příklad pochází z odeslání formuláře:

```
<input type='submit' name='potvrzeni' value='Vložit'  
  onclick="return validateForm()"/>
```

Zde se formulář odešle jen pokud funkce **validateForm()** vrátí hodnotu **true**. V tomto případě je uživatelská událost **onclick** (kliknutí na prvek). Uživatelských událostí JavaScriptu je mnoho, např.:

- **onLoad** - při úplném načtení stránky
- **onClick** - při kliknutí na prvek
- **ondblclick** - při dvojkliku na prvek
- **onmouseover** - při najetí myši na prvek
- **onmousedown** - při stisknutí tlačítka nad prvkem
- **onmouseup** - při uvolnění tlačítka nad prvkem
- **onfocus** - aktivace políčka
- **onchange** - při změně hodnoty políčka

- `onSelect` - při vybrání textu myší
- ...a mnoho dalších

Při nahrávání fotografie na server se JavaScript využívá také. Kliknutím na tlačítko odesílající formulář se spustí funkce `insert_photo()`. Ta kontroluje zda byla vybrána kape, zda nebylo jméno fotografie změněno na takové které již existuje, a zda je datum ve správném formátu. Dokonce je volána funkce `kontrolaData()`, která kontroluje zda vůbec datum existuje. Např. 29.2.2011 je neplatné datum a formulář tak není možné odeslat. V každém z případů funkce vrací hodnotu `false` a vypíše upozornění:

```
...
if (kontrolaData(datum) == true)
{
    if(nazev == stnazev)
    {
        var vzkaz_stnazev = "Právě tento název fotografie již existuje,
            změňte prosím její název!";
        alert(vzkaz_stnazev);
        return false;
    }
    else
    {
        return true;
    }
}
else
{
    var vzkaz = 'Neplatné datum.\n Je sice ve správném formátu, ale takové
        datum neexistuje!';
    alert(vzkaz);
    return false;
}
...
```

Podobná funkce (`change_info()`) je volána při změně údajů o fotografii. Kontrola formátu data je zde řešena s pomocí regulárních výrazů, jejichž syntaxe je té z PHP podobná:

```
...
var datum = document.formular.datum.value;
var vzor1 = /^19\d{2}-[01]{1}\d{1}-[0-3]{1}\d{1}$/;
var vzor2 = /^2\d{3}-[01]{1}\d{1}-[0-3]{1}\d{1}$/;
var result1 = vzor1.exec(datum);
var result2 = vzor2.exec(datum);
if (datum != null && datum != '')
```

4.4. POPIS JEDNOTLIVÝCH SKRIPTŮ

```
{
  if(result1 != null || result2 != null)
  {
    ...
  }
}
```

Funkce `reg_validate()` kontroluje, zda je vyplněné uživatelské jméno a heslo a volá funkci na kontrolu e-mailu, `kontolaEmailu()`. Ta kontroluje jednak jestli je e-mail zadán, jednak s pomocí JS funkce `indexOf()` zda obsahuje alespoň symbol '@':

```
if (mail != "")
{
  if (mail.indexOf("@") != -1)
  {
    return true;
  }
  else
  {
    var hlaska = "Špatný formát e-mailu!"
    alert(hlaska);
    return false;
  }
  ...
}
```

Pokud totiž tato funkce vrací hodnotu -1, nebyl znak nalezen. Javascriptovými funkcemi je také řešeno zobrazování či skrývání částí dokumentu, resp. formuláře, jako je tomu např. v úpravě hesla a e-mailu uživatele. Zde JavaScript přistupuje přímo ke stylům CSS. Jeden řádek tabulky je při prvním načtení stránky s pomocí CSS skrytý:

```
<tr id="novyMail" style="display:none;">
```

Pokud se klikne na tlačítko „Změnit e-mail“, spustí se JS funkce `otevriForm()`:

```
<input type="submit" name="zmenit_mail" value="Změnit e-mail"
  onclick="return otevriForm(novyMail, this)" />
```

Tato funkce zajistí změnu CSS stylu tak, aby se řádek tabulky (i s textovým vstupem a odesílacím tlačítkem) zobrazil:

```
function otevriForm(novy, input)
{
  if (novy.style.display == 'none')
  {
    novy.style.display='';
    input.style.display='none';
  }
  return false;
}
```


Funkce musí vracet logickou hodnotu `false`, jinak by se formulář odeslal. Stejným nebo podobným způsobem fungují další funkce, např. `zrus()`, `kontrolaZmenaEailu()`, `otevriForm()`, `otevriNovyOdkaz()` a další. Zastavím se ještě u funkce `kontrolaZmenaHesla()`. Ta nejprve zkontroluje, zda jsou hesla vyplněna a zda je to nové vyplněno dvakrát stejně. Funkce nekontroluje samotná hesla, ale jejich šifrované podoby. Formulář totiž, jak už bylo řečeno výše, na server odesílá již hesla v hashované podobě. Samotný JavaScript ale nenabízí přímo šifrovací funkci pro algoritmus `sha1`, a tak jsem si tuto funkci vypůjčil ze stránek <http://pajhome.org.uk/encrypt/md5>, kde je ke stažení pod BSD licenci. Jedná se celkem o 22 funkcí (z nichž se nakonec volá jen `hex_sha1()`) a tak je vše uloženo v samostatném souboru `hex_sha1.js` a v HTML hlavičce je uvedena cesta k souboru. Ve funkci `kontrolaZmenaHesla()` je využit také potvrzující dialog `confirm`:

```
zmenit = confirm("Opravdu si přejete změnit heslo?");
if (zmenit)
{
    ...
}
```

V Javascriptu jsou 3 základní dialogy:

- **Alert** - upozorňovací dialogové okno
- **Prompt** - dialogové okno vyžadující vstup uživatele, návratová hodnota je vepsaný řetězec
- **Confirm** - potvrzující dialog, vrací logickou hodnotu

Funkce `validateForm()` se spustí při desílání formuláře a kontroluje zda byla vybrána kaple, zda nebyla vyplněna jen jedna souřadnice a také formát URL odkazu, který musí obsahovat nejméně dvě tečky a musí být zadán bez protokolu (`http://`). V průběhu funkce se tak volají ještě dvě další - `validateSour()` a `urlValidate()`.

Správu a kontrolu formulářů zajišťuje celkem 21 javascriptových funkcí uložených v souboru `insert.js`.

4.4.3 Ostatní kódy

Zbývá už jen soubor s CSS styly. Zde jsem se snažil definovat kaskádové styly alespoň ke všem opakujícím se prvkům, například položky menu nebo náhledy fotografií. Je zde také řešené vyplnění plochy záhlaví, zápatí a těla stránky, jako atribut `background-image`. PNG soubory jsou uloženy ve složce `foto_ostatni`. První ukázka pochází z levé části záhlaví, tj. té s erbem hl. města Prahy:

```
div#header-left
{
    position: absolute;
    left: 0px;
    top: 0px;
    width: 165px;
```

4.5. VYUŽITÍ GOOGLE MAPS

```
height: 165px;
background-image: url('foto_ostatni/header_via_1.png')
}
```

Jednotlivé atributy jsou myslím dostatečně jasné, jen je možná dobré uvést, že `div#header-left` znamená, že tento styl bude použit u prvku `<div id="header-left"`, tedy u oddílu označeným parametrem `id`. Druhá ukázka z levého menu naopak využívá CSS třídu:

```
li.menuitem, li.menuitem_vip, li.menuitem-actual, li.menuitem-actual_vip,
li.menuitem_adm, li.menuitem-actual_adm
{
list-style: none;
border-bottom: 0.1em solid brown;
}
```

```
li.menuitem_adm a
{
display: block;
background-color: #FF6600;
color: brown;
text-decoration: none;
font-weight: bold;
padding: 0.5em 0.7em 0.5em 0.7em;
}
```

První soubor stylů je použit na všechny položky seznamu `` daných tříd (tedy všechny položky menu), druhý jen na odkazy `<a>` uvnitř prvků seznamu `` třídy `menuitem_adm`, tedy položky menu přístupné jen pro administrátora. Třída se v HTML kódu zapisuje parametrem `class` a je jí možné použít u více prvků (na rozdíl od `id`):

```
<li class="menuitem_adm">
```

Navíc v průběhu vzniku informačního systému vzniklo i několik „pracovních“ PHP skriptů, např. pro doplnění informací k fotografiím. Velké množství fotografií pocházejících z mého fotoaparátu bylo totiž přidáno dříve než vznikly skripty pro jejich nahrání, a tak bylo nejjednodušší v cyklu např. zjistit jejich rozměry a doplnit je do databáze.

4.5 Využití Google Maps

Google nabízí zdarma (pro nekomerční použití) zajímavou službu, rozhraní pro tvorbu mapových aplikací. Poslední verze má označení *Google Maps JavaScript API V3* a nabízí nepřeborné množství funkcí pro vytvoření vlastní mapy s daty z Google Maps. V projektu *Via Sancta* je mapa Google využita (zatím) v samostatné nabídce menu jako čtvercová mapa s ikonkami kapliček. Celá jednoduchá aplikace je tedy napsána v JavaScriptu v samostatném souboru *map.js*.

V HTML hlavičce stránky s mapou tedy musí být tento soubor zahrnut, a před tím také adresa rozhraní:

```
<script type="text/javascript"
      src="http://maps.google.com/maps/api/js?sensor=true">
</script>
```

Způsob inicializace javascriptového pole `kaple` je popsán výše v 4.4.1.1 na straně 56. Důležité tedy je, že obsahuje ke každé stojící kapli její id, pořadové číslo, zkrácený název, popis (zkrácený na 130 znaků) a souřadnice. Vše jsou řetězce, až právě na souřadnice, které jsou v poli uloženy jako objekt `google.maps.LatLng`. Při načtení stránky (tedy událost `onload` prvku `<body>`) je spuštěna funkce `inicialzace(kaple)`. Pole je tedy funkci předáno v parametru. Samotná stránka potom obsahuje pouze oddíl `<div>` s určitým id:

```
<div id="map_via" ></div>
```

Na počátku souboru `map.js` je deklarována proměnná `map`, aby bylo možné k ní přistupovat z více funkcí. Poté, již ve funkci `inicializace` je vytvořen objekt mapy, souřadnice středu jsou nastaveny na 28. kapli a přiblížení je nastaveno na hodnotu 12:

```
var souradniceStredu = kaple[28] ["souradnice"];
var nastaveniMapy = {
    zoom: 12,
    center: souradniceStredu,
    mapTypeId: google.maps.MapTypeId.ROADMAP
};
map = new google.maps.Map(document.getElementById("map_via"),
    nastaveniMapy);
```

Poté jsou v cyklu vytvořeny funkcí `placeMaker()` jednotlivé ikony kaplí a rozmístěny na správné souřadnice. Funkci je jako argument předávána vždy jedna konkrétní kaple. Tedy pole s podrobnějšími informacemi. Funkce `placeMaker()` zároveň vytvoří „infookénko“ `google.maps.InfoWindow`, které se otevře při kliknutí na ikonu. K tomu je potřeba vytvořit událost která nejprve zavře případné předchozí (otevřené) infookno a poté otevře nové:

```
google.maps.event.addListener(marker, 'click',
function(event) {
    if(info_open)
        info_open.close();
    info.open(map, marker);
    info_open = info;
});
```

4.5. VYUŽITÍ GOOGLE MAPS

Do infookna je možné vepsat klasický HTML kód včetně formátování a CSS stylů. V tomto případě se vypíše pořadí a jméno kapličky, zobrazí se její fotka a část (oněch 30 znaků) popisu s odkazem na podrobnější informace o kapli. Jako řetězec HTML obsah okénka vypadá následovně:

```
var infookenko = '<div class="infookenko"><h3>'+identifikator+'. - '+
    kratNazev+' kaple</h3>'+
    ''+
    '<p class="text_infookenko">'+text+'...<br />
    <a href="detail_kaple.php?id='+identifikator+
    '">více</a></p></div>';
var info = new google.maps.InfoWindow({content: infookenko,
    maxWidth: 200});
```

V ukázce je také vidět samotné vytvoření objektu `InfoWindow`. Do budoucnosti se nabízí také možnost zakreslení linie (polygonu) původní nebo nynější trasy poutní cesty. Stejně tak možnost zobrazení velké mapy v novém okně nebo zpětný odkaz z detailu kaple na tuto mapu. Samozřejmě přiblíženou na danou kapličku.

Kapitola 5

Závěr

Cíl této bakalářské práce se nemusí zdát úplně jednoznačný. Samozřejmě, již v zadání se mluví o vytvoření internetového informačního systému o Svaté cestě a o jeho naplnění daty. Také o hledání zdrojů dat a vůbec nápadů a přístupů k tvorbě. A to se také podařilo. Nicméně mechanické plnění systému daty není zrovna tvůrčí práce. A tak bylo nutné zvolit kompromis, a vytvořit zejména rozhraní, kterým bude možné data vkládat. Poté již práce může pokračovat třeba paralelně ve více větvích - bude možné na jedné straně zároveň data vkládat, ať už mnou nebo, a to bych považoval za smysluplnější, registrovanými uživateli. Na druhé straně zároveň bude možné pokračovat s vývojem samotného systému, jednotlivých funkcí a třeba na základě připomínek a nápadů lidí se systémem nebo Svatou cestou spojených přidávat nová vylepšení. Tato cesta by mohla být (alespoň po určitou dobu) na mně.

Systém tedy je funkční, je internetově přístupný a obsahuje určité množství dat. Nabízí se jej dále vyvíjet a pokračovat v práci.

Osobně jsem velmi rád, že bakalářská práce (tedy samotný projekt, nemyslím teď tento text) s odevzdáním a případným obhájením neztratí svůj význam, že se nejedná o „bakalářskou práci pro bakalářskou práci“. To, že systém bude, alespoň doufám, reálně využíván a veřejně internetově přístupný (což už je), je pro mě výzvou a zároveň motivací. Samozřejmě si nedělám iluze o kdovíjaké návštěvnosti.

V počátku tvorby této bakalářské práce bylo největší prioritou získávání nápadů a podnětů od patričních lidí s poutní cestou spojených. Ať už z občanského sdružení *UL-TREIA*, z *Národního památkového ústavu* či z *Ústavu pro ekopolitiku*. Návrh velkého a propracovaného informačního systému by pravděpodobně mohlo být téma na samostatnou bakalářskou práci. Zde jsem ale energii spíše než do UML diagramů nebo celkové architektury vložil do získávání nápadů, co by v systému nemělo chybět. Od Prof. PhDr. Ing. Jana Royta například zazněl nápad s uveřejňováním aktuálních událostí a akcí spojených s Viou Sanctou. Tedy zrovna tato funkce dosud funkční není, nicméně je na seznamu nápadů a v budoucnu doufám dojde k její realizaci. Stejně tak nápad paní Michaely Valentové z Ústavu pro ekopolitiku, o.p.s. že by bylo zajímavé a především užitečné v mapě poutní cesty vyznačit jak starou, původní, historickou cestu, tak možné alternativy pro pěší či cyklisty v současnosti, nebyl zapomenut, ale v budoucnosti snad bude naplněn.

Také doufám, že tato bakalářská práce a práce Petra Vaverky také související s Viou

Sanctou jsou spíše začátkem sledu událostí. Především mám na mysli případné opravy některých kaplí, zušlechtění jejich okolí a v ideálním případě jejich spojení pěší, cyklistickou či kombinovanou trasou. Tato samotná cesta už nemusí souviset jen s poutní cestou, ale zkratka s formou dopravy z místa na místo, do školy či do práce. V současné době se na podobné úrovni řeší poutní cesta od Lorety do Hájku, tedy kudy přesně vést stezku pro pěší a cyklisty zejména v místech s možnou kolizí s motorovou dopravou. Měl jsem možnost se zúčastnit schůzky s představiteli Prahy 6, Hostivice a Národního památkového ústavu na Útvaru rozvoje hlavního města Prahy o osudu této poutní cesty, a je vidět že možnosti i snaha jsou. Informační systém by měl tu snahu ještě umocnit tím, že informace budou přehledné a snáze dosažitelné. Pokud se o Svaté cestě ví, je více uložena v povědomí lidí, „věci se snáz hýbou“. Na území Prahy je navíc zapotřebí rychlé jednání, protože výstavba postupuje o dost rychleji než například v okolí Dřevčic.

Co se týká technického řešení, psaní webové aplikace, na PHP padla volba i díky tomu, že je v současné době relativně populární a, upřímně řečeno, zájem o PHP programátory je velký. Škola nás dosud seznámila především s jazykem C++ jako se základem objektově orientovaného programování, a mě zkrátka napadla myšlenka, proč se nenaučit pracovat s PHP. Jako na zkoušku jsem v PHP zhruba v lednu letošního roku vytvořil jednoduché, ale dynamicky generované stránky k závodu v orientačním běhu, s formulářem a několika funkcemi využívajícími framework *Nette*. To byla jakási generální zkouška před začátkem psaní bakalářské práce. Je pravda, že v tomto systému PHP tolik nevyužívá například vlastní objekty a třídy, ale základní konstrukce, tedy zpracování formulářů, práce s obrázky, s relacemi, cookies atd. jsou zde v plné míře využity.

Možnosti do budoucna jsou velké. Systém má jistě mnoho menších chyb, o kterých třeba zatím nevím, také mnoho nápadů a funkcí stále čeká na realizaci.

Jistě bude mít pro systém větší význam interaktivní mapa ve chvíli, kdy v ní bude linií (nebo více liniemi) také vyznačená samotná trasa poutní cesty. Nebo dokonce pokud bude možné registrovanými uživateli cestu upravovat, měnit lomové body a průběh linie.

Další významné vylepšení, které řadím mezi prioritní, je možnost vkládání různých souborů jiných než fotografií (ve formátu JPEG a TIFF) ať už jako mapy, k jednotlivým kaplím, nebo jen obecně, a textového popisu k těmto souborům, co obsahují. Zejména dokumenty ve formátech PDF, PNG, *.txt, RTF a *.doc jsou poměrně důležité. Také proto, že by bylo vhodné vložit výsledek bakalářské práce Petra Vaverky na tyto internetové stránky. Jsem také celkem nadšeným papírovým modelářem, a tak mě již od počátku lákalo vytvořit papírový model jedné z kaplí. Pokud k tomu někdy dojde, mohl by třeba právě tento model být ke stažení ve formátu PDF. Dále různé dokumenty, náčrtky, plánky atd., to vše by mělo být možné registrovanými uživateli na server nahrát a být k dispozici ke stažení. Zprovoznění vkládání fotografií ve formátu TIFF je pravděpodobně jen otázkou připsání několika řádků.

Další velkou oblastí, čekající na implementaci jsou zajímavosti a zejména památky z okolí, které by měly být uloženy v databázi podobným způsobem jako kaple, dokonce možná ve stejné tabulce. Klepnutí na „Okolí“ v menu by potom mělo být způsobem, jak tyto data zobrazit.

Co také považuji za poměrně důležité, je způsob zálohování textů a vůbec informací ke kaplím. Představme si situaci, že uživatel třeba nevědomky smaže obsah textového

pole s popisem kaple a změny uloží. V takovém případě by skutečně data byla smazána z databáze. Řešením by mohlo být vytvořit například druhou, paralelní tabulku, v níž by se ukládala vždy předchozí verze informací o dané kapli. V takovém případě se ale předpokládá, že si uživatel včas uvědomí chybu kterou udělal, nebo chybu objeví jiný uživatel před další editací této kaple. Mohl by potom jednoduše vrátit krok zpět, nebo, v jednodušším případě, kontaktovat administrátora.

V nejbližší době doufám zprovozním také možnost dostat se z detailu kaple odkazem na mapu s kaplemi, přiblíženou na tu konkrétní. Opačný postup již funguje, ale toto by mohla být velmi užitečná funkce. Co se týká map, také by bylo vhodné mít možnost mapu otevřít ve zvláštním okně, kde již stupeň přiblížení nebude omezen velikostí výřezu mapy. Také se nabízí vytvoření podobné mapy s daty ze serveru *mapy.cz*, kde je jako jedna z vrstev mapa II. vojenského mapování.

Také by bylo zajímavé zjistit pokud možno co nejpřesnější polohu již neexistujících kaplí a na základě polohy je přidat do katastrálních území. Také možnost vložit odkaz k mapám by nebyla marná. V současné době se ikona s fotografií kaple zobrazí jen pokud kaple skutečně stojí. Protože u několika kapliček existují (staré) fotografie, ale dnes již by jsme kaple hledali v terénu marně, dává smysl toto „omezení“ zrušit. Další, již zmiňovanou funkcí která by systém mohla v budoucnu obohatit je přidávání událostí a akcí souvisejících se Svatou cestou. Ať už zvlášť či do novinek.

Na další nápady a vylepšení samozřejmě čekám, také ukáže čas co má a co nemá význam. Jak je tedy vidět, práce není u konce. Každopádně systém byl vytvořen, jeho hlavní funkce fungují a data také obsahuje.

Tento text byl napsán v systému L^AT_EX, který je volně k dispozici ke stažení, včetně rozšiřujících balíčků.

Literatura

- [1] DVORÁK Jiří: *Historická geografie 11, Příspěvky k řešení problematice středověkého osídlení a cest*, Praha 1973
- [2] RYNEŠ V.: *Palladium země české*, Praha 1948
- [3] MIŠOVIČ J.: *Víra v dějinách země koruny české*, Praha 2001
- [4] KALISTA Z.: *České baroko. Studie, texty, poznámky*, Praha 1941
- [5] STIBOR Jiří: *Putování po Svaté cestě (Via Sancta) od Špitálské brány po Starou Boleslav která se uskutečnilo v létě roku 2009 Společenstvím Molechet* [online], 2010. Dostupné z <http://molechet.palcat.cz/images/stories/putovani/po-svate-ceste/svata_cesta.pdf>.
- [6] STROLENÁ Lucie: *Svatá cesta (Via sancta) z Pražské katedrály do Staré Boleslavi*, Diplomová práce, Praha 2009
- [7] KURANDA Miroslav: *Z Prahy do Staré Boleslavi svatou cestou podél čtyřiačtyřiceti kapliček a svatováclavskou cestou ze Staré Boleslavi do Prahy*, 1. vyd. Brandýs nad Labem - Stará Boleslav: s.n., 2009. 121 s. ISBN 978-80-254-3648-6.
- [8] TANNER Jan: *Svatá cesta z Prahy do Staré Boleslavě k nejdůstojnější rodičce Boží panně Marii, čtyřiceti a čtyřmi krásnými stavuňky, podle počtu litanie lauretánské titulům jejími obrazy v Čechách slavnějšími ozdobená, ale předtím dávno svatého milého Václava, dědice českého na smrt odjezdem, a potom těla přenešením posvěcená, nyní také malováním životem a zázraků jeho okrášlená*, Praha: Tiskárna jezuitská, 1679
- [9] VEVERKA Bohuslav, ZIMOVÁ Růžena: *Topografická a tematická kartografie, České vysoké učení technické v Praze*, 2008, 198 s. ISBN 978-80-01-04157-4
- [10] *Usage of server-side programming languages for websites* [online]. W3Techs: Aktualizováno 23.5.2011 [cit. 2011-05-23]. Dostupné z <http://http://w3techs.com/technologies/overview/programming_language/all>.
- [11] GUTMANS Andi, BAKKEN Stig Saether, RETHANS Derick: *Mistrovství v PHP 5*, 2. vydání Computer Press, a.s.: 2008, 655 str. ISBN 978-80-251-1519-0

Příloha A

Obsah přiloženého CD

K této práci je přiloženo CD, na kterém jsou uloženy zdrojové kódy a tato bakalářská práce. Má následující strukturu:

- bakalarska_prace
 - bp.pdf - text bakalářské práce ve formátu PDF
- zdrojove_soubory - kořenový adresář projektu, obsahující *index.php*
- sql - složka s **.sql* soubory pro vytvoření a naplnění databáze
