

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

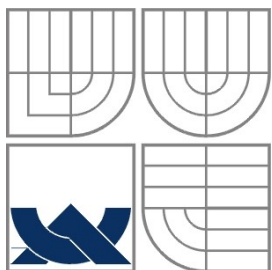
INFORMAČNÍ SYSTÉM PRO CESTOVNÍ KANCELÁŘ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

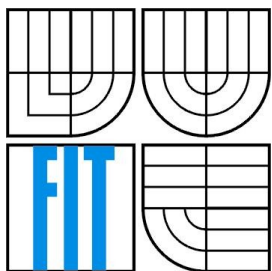
AUTOR PRÁCE
AUTHOR

PAVEL FATRDLA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM PRO CESTOVNÍ KANCELÁŘ

INFORMATION SYSTEM FOR TRAVEL AGENCY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL FATRDLA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PAVEL JURKA

BRNO 2007

Abstrakt

Bakalářská práce se zbývá tvorbou informačního systému s webovým rozhraním pro jihlavskou cestovní kancelář Krasimtour s r.o. Práce začíná analýzou požadavků, na jejichž základě jsou specifikovány jednotlivé případy použití. Následně se věnuje návrhu uživatelského rozhraní a datového modelu systému, reprezentovaného ER diagramem. Rozebírá důležitost dobře propracovaného uživatelského prostředí. Dále pojednává o politice životnosti dat v databázi, uchování a zálohování těchto dat pro potřebné analýzy. Závěrečná část práce popisuje implementaci a testování systému.

Klíčová slova

Informační systém, databáze, entita, atribut, normalizace, PHP, XHTML, UML, MySQL, CSS, JavaScript, DOM, SSL, PDF, ERM, ER diagram, Use-case diagram.

Abstract

This bachelor thesis deals with creating an information system with a web interface for travel agency in Jihlava. The work begins with analyzing requirements. Next, it explores the concept of user interface and datamodel represented by an ER diagram. Finally, the thesis will discuss the implementation and testing of the system.

Keywords

Information system, database, entity, attribute, normalization, PHP, XHTML, UML, MySQL, CSS, JavaScript, DOM, SSL, PDF, ERM, ER diagram, Use-case diagram.

Informační systém pro cestovní kancelář

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Jurky.

Další informace mi poskytli pracovníci cestovní kanceláře Krasimtour s r.o.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Pavel Fatrdla
11.5.2007

Poděkování

Na tomto místě bych rád poděkoval Ing. Pavlu Jurkovi za poskytnutí užitečných rad a odborné pomoci. Dále pak pracovníkům firmy Krasimtour s r.o., se kterými jsem při vytváření bakalářské práce byl v kontaktu.

© Pavel Fatrdla, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Použité technologie.....	4
2.1 UML.....	4
2.2 ERM.....	4
2.3 XHTML.....	4
2.4 CSS.....	5
2.5 PHP.....	5
2.6 MySQL.....	5
2.7 JavaScript.....	6
2.8 DOM.....	6
2.9 SSL.....	6
2.10 JavaScript MD5.....	7
2.11 FPDF.....	7
3 Specifikace a analýza požadavků.....	8
3.1 Uživatelé systému.....	8
3.2 Diagram případů použití.....	8
3.2.1 Popis případů použití.....	8
4 Návrh systému.....	16
4.1 Politika mazání dat z databáze.....	16
4.2 Normalizace.....	16
4.2.1 Normální formy.....	16
4.3 ER diagram.....	17
4.3.1 Popis entit ER diagramu.....	17
4.4 Zabezpečení systému.....	19
4.5 Uživatelské rozhraní.....	20
4.5.1 Rozložení UI.....	20
4.5.2 Vzhled, přehlednost a jednoduchost.....	20
4.5.3 Chybovost uživatele.....	21
4.5.4 Časová odezva při znovunačítání stránek.....	21
5 Implementace systému.....	22
5.1 Životní cyklus webových aplikací.....	22
5.2 Struktura skriptů PHP.....	23
5.3 Použití cizího kódu.....	23
5.4 Popis některých ze skriptů.....	24

6 Testování systému.....	25
6.1 Typy testování.....	25
6.2 Výsledky testování.....	25
7 Závěr.....	27
8 Literatura.....	28
9 Seznam příloh.....	29
Příloha A – Schéma databáze.....	30
Příloha B – Některé z obrazovek systému.....	31

1 Úvod

Současný vývoj informačních systémů je ovlivněn rychlým nástupem internetu. S tím je spojen i rozvoj webových aplikací, jenž skýtají oproti kompilovaným aplikacím řadu bezesporných výhod. Na klientské stanici není potřeba instalovat žádný další software, kromě dnes snad všudypřítomného internetového prohlížeče. Možnost použití libovolného prohlížeče sebou nese zároveň i platformní nezávislost aplikace. Webová aplikace je přístupná odkudkoliv. Mezi největší výhody však patří fakt, že většina logiky aplikace je umístěna na serveru, což umožňuje snadný update. Všichni uživatelé také používají v jeden okamžik stejnou verzi aplikace. Tyto vlastnosti se velice dobře hodí pro kruhový životní cyklus softwaru. Ten dovoluje programátorovi doimplementovat uživateli doslova „za běhu“ další funkce systému.

Druhá kapitola pojednává o použitých technologiích pro analýzu, návrh a vývoj. Vysvětluje důvody, proč jsem si tyto technologie vybral. Velice lehce se zde seznámíte s UML, ERM, XHTML, PHP, MySQL, JavaScriptem a dalšími. Ve třetí a nejrozsáhlejší kapitole se zabývám specifikací a analýzou požadavků. Je zde znázorněn diagram případů použití, následovaný podrobným popisem jejich jednotlivých případů. Po specifikaci požadavků a jejich následné analýze přichází na řadu kapitola čtvrtá. Ta pojednává o samotném návrhu aplikace. Je zde uveden návrh datového modelu systému, vysvětlení pojmů jako normalizace či normální formy. Samotný návrh datového modelu je reprezentován ER diagramem. Poté jsou některé jeho zajímavé či důležité části okomentovány. Kapitola dále obsahuje pojednání o politice mazání dat z databáze, uchování a zálohování těchto dat pro potřebné analýzy. Rozebírá důležitost dobře propracovaného uživatelského rozhraní systému a samotné zabezpečení systému. Kapitola pátá se věnuje vlastnímu procesu implementace systému, využití volně šiřitelného kódu, obsahuje také popis některých zajímavých skriptů systému. V následující kapitole je rozebráno samotné testování systému. Jsou zde vysvětleny různé typy testování a jejich zaměření.

Na příloženém CD naleznete jak zdrojové kódy systému, tak elektronickou verzi tištěného textu.

2 Použité technologie

Informační systém jsem naimplementoval jako systém s webovým rozhraním, což byl také jeden z požadavků zadavatele. Za tímto účelem jsem vybral kombinaci použitých technologií, které tento přístup umožňují. S ohledem na svoje znalosti jsem zvolil možnost použití technologií svobodného softwaru, ve zkratce známého jako „LAMP“ (Linux, Apache, MySQL, PHP). Formát výsledného dokumentu jsem zvolil XHTML, doporučovaný organizací W3C.

2.1 UML

Unified Modeling Language (UML) je unifikovaný modelovací jazyk užívaný nejvíce v softwarovém inženýrství. Využívá grafické syntaxe a sémantiky k vytváření abstraktního modelu systému. UML podporuje objektivě orientovaný přístup k analýze, návrhu či popisu programovaných systémů.

Mezi modely UML např. patří Use-case diagram a diagram tříd. Při výběru databázového systému jsem se rozhodl pro relační a ne objektovou databázi. Z tohoto důvodu místo diagramu tříd používám pro návrh ER diagram, který můžeme chápat jako Class diagram bez metod. K tomu však nutno dodat, že tento diagram již do UML nepatří.

2.2 ERM

Entity-Relationship Modeling (ERM) je modelovací jazyk užívaný k popisu datového modelu relačních databází. K popisu dat systému a vztahů mezi nimi využívá grafické syntaxe a sémantiky.

Mezi modely ERM patří např. ER diagram.

2.3 XHTML

EXtensible HyperText Markup Language (XHTML) je značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW, vyvinutý konsorciem W3C. Je následníkem jazyka HTML, upraveného tak, aby vyhovoval formě XML. Díky tomu je u XHTML značně zpřísněná kontrola syntaxe, než je tomu u HTML. Mezi podstatné rozdíly oproti HTML je, že veškeré prvky jsou párové, a musí být uzavřeny. Hodnoty všech atributů musí být zapsány v uvozovkách. Názvy všech prvků a atributů musí být napsány malými písmeny. V hlavičce každého dokumentu je uvedena DTD (Document Type Definition).

2.4 CSS

Cascading Style Sheets (CSS) je stylový jazyk používaný pro popis způsobu zobrazení dokumentů, psaných ve značkovacím jazyce, jakými jsou např. XML, HTML a XHTML. Hlavním důvodem vzniku jazyka CSS bylo oddělit popis vzhledu dokumentu od jeho obsahu, a tím možnost provádět změny vzhledu celého dokumentu na jednom místě. K hlavním nevýhodám tohoto jazyka bohužel patří nejednotná interpretace v různých webových prohlížečích.

2.5 PHP

PHP: Hypertext Preprocessor (PHP), původně Personal Home Page, je programovací skriptovací jazyk, jenž byl vyvinut roku 1994 dánským programátorem Rasmusem Lerdorfem. PHP bylo původně určeno pro tvorbu dynamických webových stránek, lze ho ovšem také použít k tvorbě konzolových či samostatných grafických aplikací. Umožňuje jak procedurální, tak od posledních verzí i objektově orientovaný přístup programování.

Skript psaný v PHP je spouštěn na straně serveru. Pokud webový server dostane požadavek na PHP skript, je tento požadavek zpracován PHP procesorem. Ten pak provede všechny příkazy, které jsou ve skriptu uvedené. Volajícímu počítači je pak výsledek, vygenerovaný za běhu skriptu, odeslán způsobem, jakým se odesílají běžné statické stránky.

Jelikož se ke klientovi dostanou už jen statické stránky, odpadají problémy spjaté s případnou nepodporou vývojářského nástroje, neboť pro korektní činnost programu postačí klientovi každý prohlížeč zobrazující HTML.

Skriptovací jazyk PHP je velmi všestranný a jednoduchý jazyk, který může spolupracovat téměř s libovolným webovým či databázovým serverem, tudíž i na libovolné platformě. Data lze ukládat buď do textových souborů nebo do databáze. K tomuto účelu obsahuje funkce, které zprostředkovávají přístup k databázovým serverům jako MySQL, PostgreSQL, MS SQL, Oracle a dalším. Nechybí zde ani podpora komunikace přes ODBC. Pro všechny tyto důvody se tento jazyk těší velké oblibě ze strany programátorů webových aplikací a můžeme ho nalézt na většině webových hostingů.

2.6 MySQL

MySQL je relační databázový server, vyvíjený švédskou firmou MySQL AB, a fungující na mnoha operačních systémech. Firma MySQL AB vydává tento produkt pod dvojí licencí, a to zdarma pro nekomerční účely pod licencí GPL (General Public License), tak pod licencí komerční, placenou.

Jak je již z názvu patrné, komunikace se serverem probíhá pomocí jazyka SQL. Podobně jako je tomu i u ostatních SQL databází, jedná se o dialekt tohoto jazyka s některými rozšířeními.

Již od svého vzniku bylo MySQL zaměřeno spíše na rychlost. Z tohoto důvodu zde chyběla např. podpora poddotazů, uložených procedur, triggerů a pohledů. Většina těchto vlastností byla v posledních letech doplňována, jelikož jejich uživatelům začala scházet.

Mezi hlavní výhody patří jednoduchost ovládání, vysoký výkon, existence aplikačního programového rozhraní (API) pro nepřehledné množství jazyků, a v neposlední řadě také to, že se jedná o volně šiřitelný software.

2.7 JavaScript

JavaScript je multiplatformní programovací jazyk, interpretovaný prohlížečem. Má objektově orientovanou koncepci, podporující objektový model dokumentu (DOM). Jeho autorem je Američan Brendan Eich z tehdejší společnosti Netscape.

Program v JavaScriptu se obvykle spouští až po stažení stránky z Internetu (tedy na straně klienta), narozdíl od ostatních interpretovaných jazyků, jakými jsou například PHP nebo ASP, u kterých dochází ke spouštění již na straně serveru. To umožňuje změnu obsahu stránky či jejího vzhledu, bez nutnosti odeslání a opětovného načtení stránky. Dokáže tak práci s dokumentem urychlit, a v konečném důsledku i snížit vytížení serveru. Typicky se hodí např. při kontrole obsahu formulářů před odesláním. K vytváření a vkládání nových elementů do již načteného dokumentu a pod.

2.8 DOM

Document Object Model (DOM) je objektově orientovaný model dokumentů jakými jsou například XML a HTML. Představuje platformě a jazykově nezávislé aplikační programové rozhraní, které umožňuje přístup k dokumentu jako ke stromu objektů. Použitím tohoto rozhraní je pak programům a skriptům umožněno jednoduše přistupovat k jednotlivým objektům dokumentu. Těm tak dále modifikovat obsah, strukturu či jejich styl.

2.9 SSL

Secure Sockets Layer (SSL), jedná se o protokol/vrstvu vloženou mezi transportní a aplikační vrstvu. Ta poskytuje zabezpečení komunikace pomocí šifrování. Asi nejčastěji se tento protokol používá k bezpečné komunikaci s internetovými servery. Jedná se například o protokol HTTPS, což je zabezpečená verze HTTP.

2.10 JavaScript MD5

Jedná se o volně šiřitelný skript napsaný v JavaScriptu, umožňující používání hašovací funkce MD5 (Message-Digest algorithm 5) na straně klienta. Skript je uvolněn pod BSD licenci.

2.11 FPDF

Free PDF (FPDF) je třída PHP, která dovoluje vytvářet PDF dokumenty. Jedná se o svobodný software, který je volně k dispozici. FPDF je velmi jednoduše ovladatelný, avšak propracovaný nástroj na sazbu PDF pomocí jazyka PHP.

3 Specifikace a analýza požadavků

Jihlavská firma Krasimtour s r.o., která provozuje cestovní kancelář, má požadavek pro svoji společnost, vytvořit informační systém s rozhraním webové aplikace.

System bude umožňovat komplexní správu jednotlivých letovisek, jejich hotelů a pokojů. Bude přehledně zobrazovat obsazenost těchto pokojů, stavy rezervací a informace o platbách. Správu zájezdů, účastníků zájezdů, termínu, slev a služeb. Dále bude uživateli umožňovat tisk sestav, jako např. hromadné letenky a seznamy pro delegáty.

Pracovníci cestovní kanceláře budou se systémem pracovat na 19“ monitorech. Měli již totiž špatné zkušenosti s častým rolováním. Uživatelské rozhraní by tedy mělo být navrženo pro odpovídající rozlišení obrazovky. Nemusí být optimalizováno pro nižší rozlišení. Z toho plyne, že využívat by se měla celá šířka obrazu. To pracovníkům umožní větší přehled při používání systému.

Spolu s nasazením systému by neměly vznikat další příliš vysoké výdaje, jako jsou výdaje na nákup nového hardwaru (HW) či softwaru (SF). Informačního systému by měl být provozován na současném HW vybavení a neměl by být příliš náročný na nákup nového HW.

System má být implementován v jednom z rozšířených jazyků. Data mají být přístupná v otevřené formě, aby mohla být lehce exportována či užívána i jinými aplikacemi.

3.1 Uživatelé systému

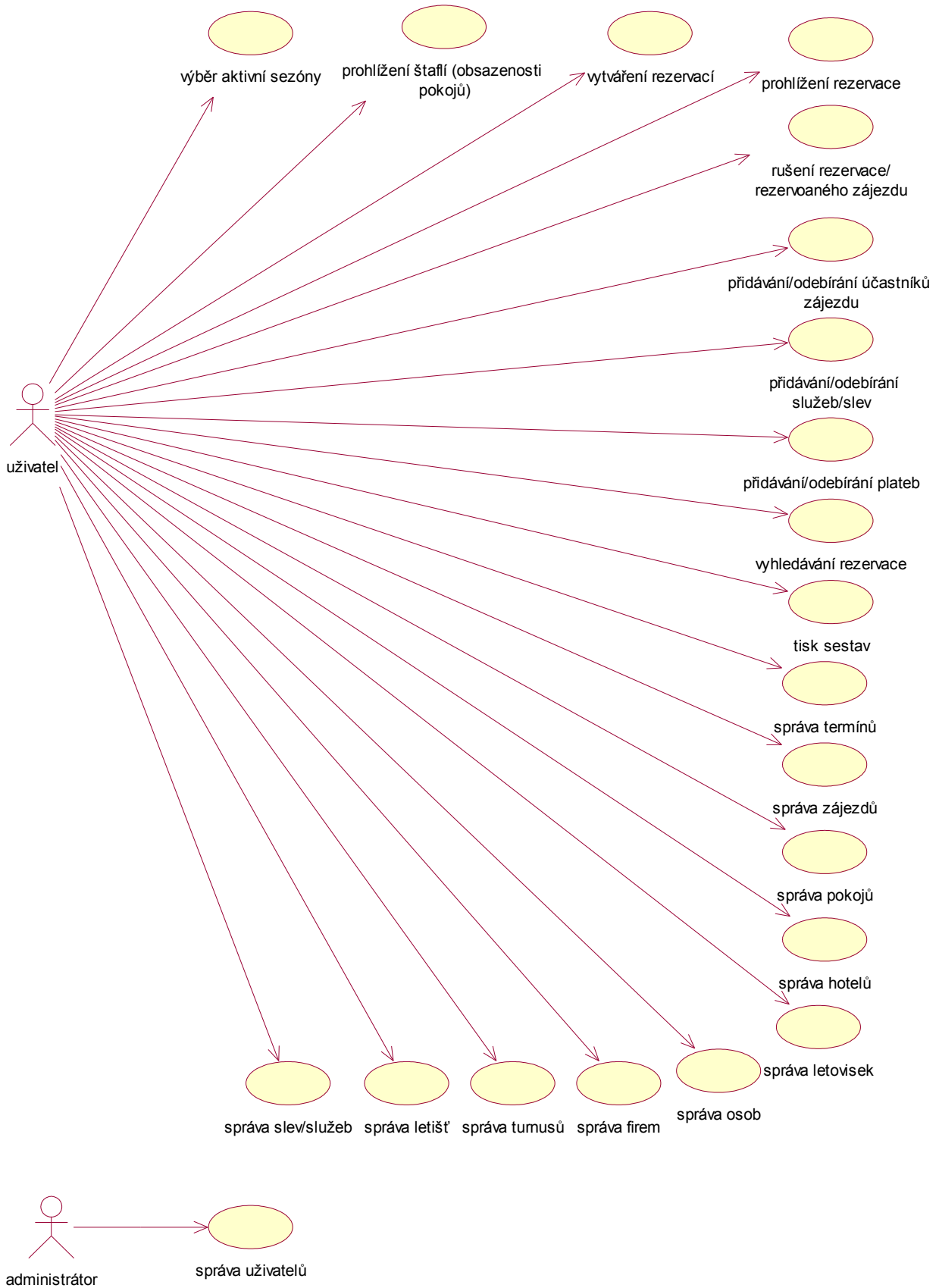
Hlavním uživatelem systému jsou zaměstnanci v pobočce cestovní kanceláře, dále označováni pod pojmem *uživatel*. Uživatel bude s informačním systémem pracovat pomocí svého počítače, a bude mít přístup téměř ke všem funkcím systému s výjimkou správy uživatelských účtů. Tato funkce bude v systému přístupná pouze pro administrátorský účet – *administrátor*.

3.2 Diagram případů použití

Diagram tříd je znázorněn na obrázku 3.1. Z důvodu lepší přehlednosti jsem vazbu <include> v diagramu nepoužil. Každý případ použití je dále podrobněji rozepsán.

3.2.1 Popis případů použití

Pro přesnější specifikaci je u každého, z níže uvedených, případu použití uveden podrobnější popis požadavků na systém.



Obrázek 3.1: Diagram případů použití

Autentizace a autorizace

Uživatel i administrátor se přihlašují přes společnou obrazovku. Uživatel se přihlásí do systému zadáním jména a hesla, kde systém rozpozná, zda-li má do systémů přístup a jaká má práva. Bez přihlášení není možné se systémem nijak pracovat.

Systém bude umožňovat při přihlašování i ostatní komunikaci se serverem možnost použití šifrovaného přenosu.

Při nečinnosti delší jak 30 minut dojde k automatickému odhlášení ze systému, aby bylo zajištěno, že každý bude pracovat pod svým účtem.

Výběr aktivní sezóny

Systém vždy pracuje s jednou aktivní sezonou. Ta je při přihlášení do systému nastavena na aktuální rok. Uživatel může listovat sezonami dopředu či dozadu. Celý systém pak bude pracovat se sezonou, jenž si uživatel vybral. Pokud listuje dozadu, systém pracuje s historií. Listuje-li dopředu, pracuje s budoucností a může např. vytvářet termíny a zájezdy pro příští sezonu.

Informace, se kterou sezonou pracujeme, musí být dobře viditelná ze všech obrazovek. A to z důvodu, aby nedocházelo k omylům způsobených prací ve špatné sezoně. Také listování sezonami musí být uživateli umožněno ze všech obrazovek.

Prohlížení štaflí (obsazenosti pokojů)

Uživatel prochází štaflemlí¹ (viz. obrázek 9.2) a prohlíží si obsazenost pokojů. Může si nechat zobrazit zájezdy od kteréhokoliv měsíce dané sezony, nebo vyfiltrovat zájezdy pomocí upřesnění, které letovisko, hotel nebo konkrétní pokoj si přeje zobrazit. Na výběr má také turnus², který si přeje zobrazit.

U každého zájezdu ve štaflích jsou zobrazeny základní informace o pokoji, cena zájezdu, popř. jméno a příjmení objednavatele či datum vytvoření rezervace. Přímo ze štaflí lze spravovat a vytvářet rezervace.

1 Štaflemlí rozumíme tabulku, která má na vodorovné ose termíny nepřekrývajících se zájezdů a na svislé ose pokoje. Jedno políčko štaflí tedy představuje zájezd v daném pokoji a daném termínu. Zájezd nabývá pro každý termín jednoho ze stavů: volný, rezervován, propadlá rezervace, částečně zaplacený zájezd, zaplacený zájezd, nebo stav, kdy pro daný pokoj není v daném termínu zájezd vytvořen. Různé stavy jsou odlišeny různou barvou políčka.

2 Turnusem rozumíme množinu nepřekrývajících se a obvykle na sebe navazujících termínů zájezdů, tj. termíny zájezdů, jenž můžeme zobrazit na jedné časové ose. Například termíny 15.6. - 21.6. a 21.6. - 27.6. patří do jednoho turnusu. Termíny 6.7. - 12.7. a 10.7. - 17.7. do stejného turnusu nepatří, protože se částečně překrývají.

Vytváření rezervací

Uživatel prochází štaflemy a vidí obsazenost pokojů v průběhu času, a má tak přehled, který pokoj a termín může zákazníkovi nabídnout. Může si vyfiltrovat jen takové zájezdy, o které má zákazník zájem. Např. všechny zájezdy ve vybraném letovisku či hotelu, nebo naopak mít v přehledu všechny zájezdy konající se v požadovaném termínu.

Vytváření rezervace bude možné provést jak na osobu, tak i na firmu. Ke každé z možností bude nutné uvést různé údaje.

K rezervaci na osobu je zapotřebí uvést jméno, příjmení a kontakt na zákazníka. Jedná-li se o zákazníka, který už nějakou rezervaci v minulosti vytvářel nebo se již některého zájezdu účastnil, postačí mu jen nahlášení příjmení. Uživatel si zákazníka nechá vyhledat se všemi ostatními informacemi v seznamu osob. Pokud došlo ke změně některých z údajů, má uživatel možnost je jednoduše v seznamu upravit. Vybraný zájezd pak může zákazníkovi přidat do již existující rezervace nebo mu vytvořit novou.

K rezervaci na firmu je nutné zadat obchodní název firmy, IČO, DIČ a kontakt na osobu, kterou může cestovní kancelář v případě potřeby kontaktovat. Dále již rezervace probíhá stejně jako je tomu u rezervace na osobu. Pokud tedy daná firma už v minulosti nějaký zájezd objednávala, postačí uživateli sdělit obchodní název, a ten si firmu se všemi informacemi nechá vyhledat v seznamu firem. Pokud došlo u některého z údajů ke změně, může uživatel tuto změnu v seznamu jednoduše upravit. Vybraný zájezd pak může firmě přidat k již existující rezervaci, nebo při rezervování zájezdu vytvořit novou.

Při vytváření či editaci rezervace může uživatel lehce přidat slevy, služby a jejich počet k danému zájezdu. Celou dobu při vytváření rezervace je zobrazována celková cena, která je vypočítávána z ceny zájezdu a ceny všech objednaných služeb či slev.

Prohlížení rezervace

Uživatel si prohlíží vybranou rezervaci (viz. obrázek 9.4). Vidí informace o osobě nebo firmě, na kterou je rezervace napsána a datum, kdy byla rezervace vytvořena. Dále je zde uvedeno číslo rezervace, informace ke všem rezervovaným zájezdům, jako jsou termíny odletu a návratu, informace o pokojích, hotelu, letovisku a ceně zájezdu.

U každého zájezdu je také uveden seznam účastníků zájezdu, ve kterém jsou vidět informace o účastníkovi – jméno, příjmení, pohlaví, datum narození, místo odletu a návratu.

Ke každému zájezdu je uveden i seznam objednaných služeb obsahující údaje – popis služby, kolik jednotek služby je objednáno, její jednotkovou cenu, a celkovou cenu služby.

U rezervace musí být uvedeno, kolik činí její celková cena k zaplacení, tzn. součet cen všech zájezdu a jim přibojednaných služeb či slev. Musí zde být uvedena i částka, která byla již zákazníkem uhrazena.

Rušení rezervace/rezervovaného zájezdu

Uživatel si prohlíží vybranou rezervaci, kde vidí přehled všech rezervovaných zájezdů. Má možnost kterýkoliv z rezervovaných zájezdů stornovat. Při stornování všech zájezdů dané rezervace dochází i k samotnému zániku rezervace.

Pokud je rezervace ve štaflích starší 14 dnů a stále nebyla uhrazena záloha, jedná se o propadlou rezervaci a je označena jinou barvou. Uživatel tedy vidí, že může v případě zájmu jiného zákazníka tuto rezervaci zrušit. A zájezd pak nabídnout jinému zákazníkovi.

Přidávání/odebírání účastníka zájezdu

Uživatel si prohlíží rezervaci a vidí, kteří účastníci jsou k zájezdu nahlášeni. Má možnost je odebrat nebo přidávat další. Při přidávání účastníka se uživateli zobrazí informace o zájezdu, ke kterému chceme účastníka přidat (viz. obrázek 9.3). K přidání je zapotřebí zadat jméno, příjmení, pohlaví, datum narození³. Pokud se zákazník dříve již nějakého zájezdu zúčastnil, postačí nahlásit příjmení a uživatel si ostatní informace o zákazníkovi již vyhledá v seznamu osob. Dále je nutno vybrat z možných letišť, odkud účastník poletí, a také letiště, na které se při návratu bude chtít vracet. Po přidání nového účastníka do seznamu se nové okno uzavře.

Přidávání/odebírání služeb a slev

Uživatel si prohlíží rezervaci a vidí, jaké služby jsou k zájezdu objednány s možností odebrat či přidat službu/slevu k danému zájezdu. Při přidávání slev se uživateli zobrazí informace o zájezdu, ke kterému chce služby přiojednávat. Uživatel zadá počet, kolikrát službu objednává.

U přidávání slev k zájezdu musí mít uživatel možnost přidat najednou více druhů slev a jejich počtů.

Přidávání/odebírání plateb

Uživatel si prohlíží obrazovku s informacemi o rezervaci, vidí seznam již vložených plateb a má možnost je odebrat či vkládat další. K platbě bude mít možnost přidat popis.

Tisk sestav

Uživatel si prohlíží obrazovku s přehledem tiskových sestav. Zadá pro který z letů, termínů a letišť si přeje tiskové sestavy vytvořit. Poté si vybere jednu z následujících tiskových sestav:

- a) Tisk seznamu pro delegáta – dokument musí obsahovat hlavičku o jaký let, místo letu, termín a turnus se jedná. Poté seznam všech osob letících v daném termínu z vybraného letiště. V tomto seznamu musí být uvedeno jméno, příjmení, datum narození. Dále musí

³ Údaje - jméno, příjmení, pohlaví, datum narození - jsou vyžadovány u letecké společnosti a při zřizování cestovního pojištění.

obsahovat strukturovaný seznam, ve kterém letovisku, hotelu a pokoji každý z účastníků bude ubytován.

b) Tisk hromadné letenky – dokument musí obsahovat hlavičku, o jaký let, místo letu, termín a turnus se jedná. Jeho součástí je i seznam všech osob letících v daném termínu z vybraného letiště. V tomto seznamu musí být uvedeno jméno, příjmení, datum narození a anglická zkratka o jakou osobu se jedná.

MR – muž starší 12 let

MRS – žena starší 12 let

CHL – dítě ve věku od 2 do 12 let

INF – dítě mladší 2 let

Vyhledávání rezervací

Často nastávají situace, kdy uživatel potřebuje vyhledat rezervaci podle určitých údajů. Např. když na účet cestovní kanceláře dojde platba s číslem rezervace jako variabilním symbolem. Uživatel tedy musí být schopen tuto rezervaci vyhledat, aby mohl provést její úpravy. Nebo v případě, kdy do pobočky přijde zákazník, který si nepamatuje číslo rezervace, a uživatel potřebuje vyhledat všechny rezervace na zadané jméno.

Systém tedy musí uživateli umožňovat vyhledávat rezervace podle příjmení objednavatele nebo čísla rezervace. Uživatel si má možnost nechat vypsat všechny rezervace obsahující zájezdy, které ještě týden před odjezdem nejsou doplacený.

Přímo ze seznamu vyhledaných rezervací je možné se přepnout do správy jedné konkrétní rezervace.

Správa termínů

Uživatel si prohlíží termíny vybraného turnusu tak jak jdou za sebou (viz. obrázek 9.5). Vidí jaké jsou možnosti výběru místa odletu a návratu pro jednotlivé termíny. Termíny může upravovat, odebírat či přidávat nové. Při přidávání nového termínu vybírá v kalendáři datum odletu a datum návratu. Vybere ze seznamu letišť, letiště odletu a letiště návratu. Dále pak turnus, do kterého chce daný termín zařadit. Systém kontroluje zda nezadal termín příletu na dřívější datum, než termín odletu, či zda zadal místa pro přílet a návrat.

Správa zájezdů

Uživatel vidí obrazovku se všemi pokoji nebo může použít některý z filtrů (hotel, pokoj či turnus) k zobrazení jen vybraných pokojů (viz. obrázek 9.6). U každého z pokojů je uveden seznam zájezdů, v nichž bude pokoj nabízen. U každého zájezdu musí být uvedeno – datum odletu, datum návratu,

turnus, cena zájezdu a informace, zda je zájezd již rezervován⁴. Jednotlivé zájezdy pro daný pokoj může upravovat, odebírat či přidávat.

Při přidávání zájezdu uživatel vybere pokoj, ke kterému chce zájezd vytvořit. Vybere jeden z termínů a zadá jeho cenu.

Správa pokojů

Uživatel si prohlíží obrazovku se všemi pokoji. Lze využít filtrů k zobrazení např. jen pokojů určitého letoviska, hotelu nebo konkrétního pokoje. Uživatel je schopen měnit, odebírat nebo přidávat nové pokoje. Při přidávání pokojů zadá označení pokoje, vybere letovisko, hotel. Dále zadá počet lůžek, zda má pokoj vlastní sociální zařízení, balkón. Pro bližší informace lze k danému pokoji přiřadit popis.

Správa hotelů

Uživatel si prohlíží obrazovku se všemi hotely a jejich letovisky, ve kterých se hotel nachází. Lze využít vyhledání hotelu podle jeho názvu. Může zde odstranit, upravit nebo přidat nový hotel. Při přidávání nového hotelu udává název hotelu a letovisko, ve kterém se hotel nachází.

Správa letovisek

Uživatel si prohlíží obrazovku s informacemi o všech letoviscích. Lze využít vyhledání letoviska podle jeho názvu. Uživatel zde může odstranit, upravit nebo přidat nové letovisko. Při přidávání letoviska zadá pouze název letoviska.

Správa osob

Uživatel prochází seznamem osob. Může si nechat vyhledávat osobu podle příjmení, dále ji upravovat, odstranit nebo přidat novou. Při přidávání je nutné uvést údaje – jméno, příjmení, pohlaví a některý kontakt.

Do tohoto seznamu jsou automaticky přidávány i osoby, o kterých uživatel zadává informace např. při rezervaci zájezdu nebo při zadávání účastníků zájezdů.

Správa firem

Uživatel vidí obrazovku se seznamem firem, kde je může upravovat, odebírat či přidávat. Při přidávání firmy je nutné zadat její obchodní název, IČO, DIČ a její kontaktní informace. V seznamu máme možnost vyhledat firmu podle jejího názvu.

Do seznamu jsou automaticky přidávány i firmy, o kterých uživatel zadával informace v jiných částech systému.

⁴ U těch zájezdů, jež jsou rezervovány se uživatel může přepnout do správy dané rezervace.

Správa slev/služeb

Uživatel si prohlíží seznam služeb a slev. Zde si může nechat vyhledat službu podle popisu. Dále má možnost tyto slevy/služby odebírat, upravovat nebo přidávat nové. K přidání nové služby/slevy je zapotřebí uvést její popis a cenu.

Slevy/služby v systému nejsou vázané na žádné kategorie pokojů, ale přiojednávají se jednoduše ke každému zájezdu. Není tedy problém vytvořit novou službu a přiřadit ji pouze jedinému zájezdu.

Správa letišť a turnusů

Uživatel si prohlíží obrazovku s informacemi o všech letištích či turnusech. Může využít vyhledání podle názvu. Uživatel zde může odstranit, upravit nebo přidat nové letiště či turnus. K přidávání zadává pouze název.

Správa uživatelů

Administrátor přidává, odebírá nebo přenastavuje heslo uživatelským účtům.

4 Návrh systému

Po dokončení specifikace požadavků a následné analýzy se projekt přesunul do další fáze a to přesněji k návrhu systému. Na základě požadavků jsem vytvořil datový model systému a návrh uživatelského rozhraní.

4.1 Politika mazání dat z databáze

Ačkoliv je v systému umožněno mazat záznamy, jenž doposud nejsou svázány s ostatními daty, je tato možnost spíše za účelem odstranění chybně vložených záznamů, a ne pro mazání dat z již proběhlých sezón. Většina dat tedy v databázi zůstává. V dnešní době jsou totiž historická data pro firmy velice cenná a byla by škoda se jich zbavovat.

V budoucnu se totiž budou hodit nad těmito daty vytvářet různé pohledy a analýzy. Ty poté firmě mohou napovědět např. jaký má typ zákazníků, jaké služby jsou nejoblíbenější atd. Na jejichž základě budou moci zlepšit obchodní strategii a rozšířit nabídku služeb.

Data jsou navíc uložena v otevřeném formátu, což přináší určitou nezávislost. Není problém pro jinou aplikaci k této databázi přistupovat a pracovat s jejími daty. V případě přechodu na jiný systém lze všechna data přenést do něj.

Aby ovšem data z minulých sezón běžnou práci v systému neznepřehledňovala, pracuje systém pouze s jednou aktivní sezónou. Tudíž vybere-li si uživatel letošní sezónu, zobrazují se mu pouze data aktuální pro tuto sezónu.

Pokud by někdy nastal problém s přeplněním databáze⁵, stačí data z databáze zazálohovat a databázi vyprázdnit.

4.2 Normalizace

Pojem normalizace můžeme chápat jako postup návrhu databáze a vzniku jednotlivých tabulek, které lze snadno a efektivně udržovat. K ohodnocení kvality návrhu tabulky nám slouží různé stupně normálních forem, těch je šest a jsou vzestupně číselně označovány. V čím vyšší formě se tabulka nachází, tím je její návrh kvalitnější.

4.2.1 Normální formy

Jednotlivé normální formy definují podmínky, jenž musí tabulka, s ohledem na datové nebo funkční závislosti mezi jednotlivými atributy, splňovat. Musí být také splněny podmínky všech nižších

⁵ *Myšleno ve smyslu, že by operace nad databází, kvůli velkému počtu záznamů v tabulkách, trvaly příliš dlouhou dobu.*

normálních forem. Např. tedy tabulka ve čtvrté normální formě, splňuje podmínky třetí normální formy a ještě něco navíc.

V praxi většina databází vystačí s normalizací do třetí nebo Boyce-Coddové normální formy, kdy dojde k odstranění většiny anomálií. Proto bych zde chtěl uvést pouze tyto nejčastěji užívané stupně normalizace.

Normalizovat relace do nejvyššího stupně není vždy nutné ani výhodné – kvůli požadované výkonnosti.

1. normální forma (1NF)

Relace je v první normální formě, pokud všechny její jednoduché domény obsahují pouze atomické hodnoty, tj. hodnoty dále již nedělitelné.

2. normální forma (2NF)

Relace se nachází ve druhé normální formě, pokud splňuje podmínky první normální formy a navíc každý atribut, který není primárním klíčem je na primárním klíči plně závislý.

3. normální forma (3NF)

Ve třetí normální formě se nachází taková relace, která splňuje podmínky druhé normální formy a žádný atribut, který není primárním klíčem, není tranzitivně závislý na žádném klíči.

Boyce-Coddova normální forma (BCNF)

Relace je v Boyce-Coddově normální formě, jestliže pro každou netriviální závislost $A \rightarrow B$ platí, že A je nadmnožinou nějakého klíče schématu (superklíč).

4.3 ER diagram

Konečný datový model systému (ER diagram) můžete vidět na obrázku 4.1. Aby byla databáze kvalitní je nutné, aby se vyskytovala alespoň ve třetí normální formě. Právě v tomto stupni normalizace se také návrh databáze systému nachází.

4.3.1 Popis entit ER diagramu

V následující části nebudu popisovat každou entitní množinu a každý její atribut. Chtěl bych se ale zaměřit na ty, u kterých nemusí být na první pohled zřejmé k čemu slouží nebo jsou jinak zajímavé a důležité.

Entita zájezd

Zájezd jako takový vzniká ve chvíli, kdy je mu přiřazen pokoj a termín. V atributu *cena* je uvedena cena za jedno lůžko. Opravdová cena takového zájezdu je pak určena počtem lůžek pokoje krát *cena*.

Entita objednávka

Atribut *datum_rez* určuje datum, kdy byla rezervace provedena. Tato informace je potřebná z důvodu možnosti zjištění platnosti rezervace. Pokud k rezervaci nebyla zaplacená žádná *platba*, je doba její platnosti stanovena na 14 dní.

Celková cena objednávky není nikde uložena, ale je stále vypočítávána z ceny zájezdu a přioobjednaných služeb.

Entita účastník

Entita účastník je mimo jiné popsána atributy *odlet* (a *návrat*), které určují, ze kterého (a na které) letiště chce účastník odlétat (a na které se navrátit).

Entita termín

Entita termín obsahuje atribut *let*. Ten určuje, zda se jedná o odlet na zájezd či návrat.

4.4 Zabezpečení systému

Mluvíme-li o zabezpečení systému, máme na mysli principy, které chrání systém před vniknutím neautorizovaných osob. Ty brání jeho napadení se záměrem poškodit data nebo samotný systém.

Informační systém cestovní kanceláře bude umístěn na webovém serveru připojeném k internetu, aby byl volně přístupný odkudkoliv. To sebou ovšem nese určitá rizika. Stejně totiž jako se může do systému připojit obyčejný uživatel, může se i neoprávněná osoba pokoušet nabourat do systému prakticky odkudkoliv.

Mezi jedno z možných rizik je odposlouchávání paketů u nešifrovaného přenosu. V tomto případě putují přihlašovací údaje po internetu v textové podobě. Útočník si po úspěšném odposlechu tyto informace přečte a nic už mu nebrání přihlásit se pod identitou tohoto uživatele. Heslo uživatele je sice v systému zašifrováno už na straně klienta, avšak tato ochrana ovšem zajistí pouze neprozrazení hesla uživatele. Ovšem s jeho zašifrovanou podobou se již útočník do systému dostane. Proto systém podporuje šifrované spojení se serverem s pomocí tzv. HTTPS (HyperText Transport Protocol Secure). Ten systém proti odposlechu na síti a následnému získání přihlašovacích údajů brání šifrováním veškeré komunikace mezi serverem a klientem.

4.5 Uživatelské rozhraní

Uživatelské rozhraní (UI) je prostředník mezi uživatelem a programem. Umožňuje uživateli s programem komunikovat a zpřístupňuje mu tak naimplementované případy použití.

4.5.1 Rozložení UI

Jeden z požadavků při specifikaci systému bylo vytvořit takové UI, které bude optimalizováno pro 19“ monitory. Tedy pro monitory s rozlišením, s nímž uživatelé v cestovní kanceláři pracují. To samozřejmě umožňuje zobrazit více informací, bez zbytečného rolování. UI je tedy navrženo s ohledem na tuto skutečnost. Např. ve štaflích je díky tomu zobrazen větší počet termínů, než jak by tomu bylo v případě nutné optimalizace pro menší monitory.

4.5.2 Vzhled, přehlednost a jednoduchost

Uživatelské rozhraní je jedna z nejdůležitějších částí programu. Jedná se v podstatě totiž o jedinou část programu, kterou uživatel při používání programu vidí. Sebelepší program s nedomyšleným uživatelským rozhraním, se u odběratele nesetká s kladnou odezvou. Samozřejmě propracované uživatelské rozhraní není jen o vzhledu, avšak též o přehlednosti, jednoduchosti a také intuitivnosti.

Z těchto důvodů je určitě důležité, aby se na návrhu uživatelského rozhraní podíleli i samotní budoucí uživatelé.

V návrhu uživatelského rozhraní pro cestovní kancelář se tedy promítly námitky od možných budoucích uživatelů, se kterými jsem problematiku konzultoval. Snažil jsem se, aby UI bylo co nejvíce přehledné a intuitivní. Z tohoto důvodu jsem se držel následujících zásad:

a) Přehledná navigace

Hlavní navigace webové aplikace se provádí skrze jednotné menu, vyskytující se na každé obrazovce. V menu je rovněž vyznačeno, na které obrazovce systému se zrovna nacházíme. Uživatel má tedy pořád přehled, kde se nachází.

b) Lehké přechody mezi stránkami

Na jednu a tutéž obrazovku se můžeme dostat z více míst. Např. na prohlížení rezervace se dostaneme z vyhledávání rezervací, tak ze štaflí.

c) Zobrazování pouze potřebných informací

Na obrazovce se vyskytují pouze informace, které může uživatel potřebovat. Např. při vyhledávání rezervací se v seznamu ke každé rezervaci vypisují základní informace rezervací a ne jednotlivé platby, které zde uživatel znát nepotřebuje.

4.5.3 Chybovost uživatele

Při navrhování uživatelského rozhraní je třeba si být vědom toho, že každý uživatel, který bude se systémem pracovat, pracuje s určitou chybovostí. Je tedy důležité, aby akce se zásadními dopady do databáze nemohly být provedeny na základě jednoho kliknutí či překliknutí.

V případě, kdy uživatel zadá, že chce smazat určitý záznam, systém vypíše v potvrzovacím okně informace o záznamu, který jsme chtěli smazat. Pokud to potvrdíme, potvrzovací okno je uzavřeno a záznam je smazán. Pokud nepotvrdíme, potvrzovací okno se uzavře a nic se nestane. Tímto způsobem je tedy zabráněno nechtěným ztrátám dat.

4.5.4 Časová odezva při znovunačítání stránek

Jelikož je informační systém cestovní kanceláře navrhnut jako webová aplikace, dochází, oproti kompilovaným aplikacím, k delší časové odezvě. Ta je způsobena odesláním informací serveru, zpracování skriptu procesorem PHP na serveru a znovunačítáním a zobrazováním stránek. Při implementaci jsem se tedy snažil této nepříjemné vlastnosti vyvarovat.

K tomuto účelu jsem co nejvíce využíval JavaScriptu. Např. ke kontrole odesílaných dat formulářů. Při výskytu chyby je informace o chybě ohlášena uživateli ještě před posláním na server. Uživatel jí poté může opravit a na server už dojdou korektní data. Dalším případem použití JavaScriptu bylo také dynamické generování formulářů pomocí DOM přístupu. Ten zde lehce umožňuje do formuláře přidávat nové prvky, jakými jsou např. další vstupní pole formuláře.

Díky těmto opatřením se částečně vyhneme znovunačítání stránek a tím si práci se systémem zpříjemníme.

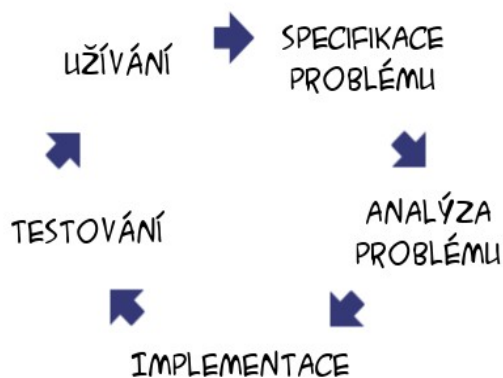
5 Implementace systému

Po návrhu datového modelu a návrhu uživatelského rozhraní, v podobě načrtnutých jednotlivých obrazovek, jsem začal s implementací systému.

Při implementaci jsem používal nástroje PSPad. Pro ladění pak prohlížeč Mozilla FireFox s rozšířeními HTML Validator a FireBug, jenž obstarává komplexní validaci XHTML, CSS a JavaScriptu.

5.1 Životní cyklus webových aplikací

Webové aplikace oproti kompilovaným aplikacím skýtají nespornou řadu výhod. Mezi nimiž bych se zde chtěl zaměřit na jednu z nich. U webových informačních systémů je většina aplikační logiky umístěna na straně serveru, ať již webovém či databázovém. Tím odpadá mnoho problémů při vytváření nových verzí programů. Jako je tomu u kompilovaných aplikací, kdy všem uživatelům systému musíte nainstalovat novou verzi programu. Naproti tomu u webových aplikací se nová verze jednoduše nahraje na server a od tohoto okamžiku ji používají všichni uživatelé. Díky této vlastnosti se u webových aplikací snadno využívá kruhového životního cyklu softwaru, který je graficky znázorněn na obrázku 5.1.



Obrázek 5.1.: Kruhový životní cyklus softwaru

Jak je z obrázku patrné, systém může být nasazen např. jen se základními funkcemi a další rozšiřující funkce se mohou doimplementovat postupně, již v době, kdy uživatel systém používá. I díky této vlastnosti se v současnosti webové informační systémy staly velice populární.

5.2 Struktura skriptů PHP

V této části bych chtěl popsat a částečně tak přiblížit strukturu PHP skriptu za použití ostatních technologií. Většinu obrazovek reprezentují jednotlivé skripty (soubor.php). Tedy např. obrazovka prohlížení objednávek je implementovaná ve skriptu *objednavka.php*. Akce z této obrazovky zpracovává skript *sObjednavka.php*. Analogicky tak *pohoj.php* a *sPokoj.php*, *osoba.php* a *sOsoba.php* atd.

Jelikož ve webové aplikaci nepoužívám rámců (které jsou v této verzi XHTML již zakázané). Používám na vkládání XHTML hlavičky a hlavního menu jiného způsobu.

Ve skriptu *top.php* vytvoříme hlavičku XHTML spolu s hlavním menu, tento skript pak includujeme do skriptů, kde chceme hlavičku a menu umístit. Obdobně pak každý skript ukončujeme includem skriptu *bottom.php*, obsahujícího ukončovací tagy XHTML dokumentu.

Include hlavičky

Začátek XHTML dokumentu

Takto pak vypadá hlavička vložená do každého XHTML dokumentu.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="cs" lang="cs"><head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="content-language" content="cs" />
<title>CK KrasimTour</title>
```

Vložení JavaScriptu

Funkce JavaScriptu, jsou z důvodu větší přehlednosti uloženy v jiném souboru.

```
<script type="text/javascript" src='./_js/default.js'></script>
```

Vložení CSS stylů

Stejně tak uložené skripty jsou odděleny od obsahu XHTML dokumentu

```
<style type="text/css">@import './_css/default.css';</style>
```

Tělo dokumentu

V dokumentu se pak mezi uvozovacími tagy `<? a ?>` vyskytuje kód PHP.

5.3 Použití cizího kódu

Při implementaci šifrování hesla na straně klienta jsem využil volně šiřitelného skriptu *md5.js*. Tento skript je napsán v jazyce JavaScript a je uvolněn pod BSD licencí. Ta umožňuje jeho volné šíření,

přičemž vyžaduje pouze uvedení autora, informace o licenci, spolu s upozorněním na zřeknutí se odpovědnosti za dílo.

5.4 Popis některých ze skriptů

Zde bych se rád zmínil o některých ze skriptů systému. Chtěl bych se zaměřit na ty z nich, které jsou něčím důležité či jinak zajímavé.

***top.php, bottom.php* – Generování XHTML hlavičky a konce XHTML**

Skripty jsou volány na začátku (*top.php*) a konci (*bottom.php*) téměř z každého skriptu. Při první dochází ke generování hlavičky XHTML a hlavního menu. Při druhé pak k vygenerování konce XHTML dokumentu.

***conf.php* – Údaje potřebné pro připojení k MySQL databázi**

Nachází se zde údaje potřebné k připojení k databázi na MySQL severu. Přesněji se jedná o ip adresu a port serveru, přihlašovací jméno, heslo a název vybrané databáze.

***conn.php* – Připojení k MySQL databázi**

Tento skript je volán v každém skriptu, který pracuje s databází. Údaje nutné k přihlášení získává ze *conf.php*. Při připojování zde dochází ke kontrole zda se podařilo spojit se serverem. Dále zda je na serveru požadována databáze a jest-li k ní máme přístup. Pokud se spojení nezdaří, je vypsáno chybové hlášení a provádění skriptu je zastaveno.

***create_tables.php* – Vytvoření tabulek databáze**

Tento skript je třeba spustit při instalaci systému. Při spuštění se připojí k vybranému serveru a databázi⁶, kde vytvoří zde potřebné tabulky.

***fill_tables.php* – Naplnění tabulek ukázkovými daty**

Skript je užíván při testování nebo předvádění systému zákazníkům. Po vytvoření struktury tabulek je databáze naplněna ukázkovými daty na nichž, je systém prezentován.

⁶ Jedná se o server a databázi jenž je uvedena v *conf.php*.

6 Testování systému

Jednou z posledních fází životního cyklu softwaru před nasazením do provozu je jeho samotné testování. V této fázi se snažíme pomocí testování přijít na nedostatky našeho systému. Jelikož je toho u softwaru k testování velice mnoho oblastí, dá se rozdělit na několikero typů podle, toho co testujeme. Níže jsou uvedeny alespoň některé z nich.

6.1 Typy testování

Testování funkčnosti

U tohoto druhu testování se snažíme odhalit rozdíly mezi funkčností aktuálního systému proti funkčnosti uvedené ve specifikaci. Testování probíhá zkoušením různých scénářů použití systémů a pozorováním, zda nedošlo k nekorektní reakci systému.

Testování použitelnosti

U tohoto typu testování poznáváme, jakým problémům čelí uživatel při používání našeho systému. Zjistíme, které prvky na jednotlivých stránkách nejsou pro uživatele srozumitelné. Zda se uživatel umí na našich stránkách orientovat.

Testování probíhá tak, že budoucí skupině uživatelů je zadána množina úkonů, které budou v systému provádět. Na základě chování uživatelů při používání systému jsou identifikovány potenciální problémy, bránící uživatelům dosahování jejich cílů či snižování efektivity.

Jako metriku můžeme použít např. procento času, kdy se uživatel drží optimální cesty k dosažení cíle nebo počet případů, kdy se musel vrátit zpět.

Testování bezpečnosti

Při testování bezpečnosti se snažíme např. odhalit možnosti nebezpečí vniknutí neoprávněné osoby či úniku informací. Na základě získaných slabín tyto bezpečnostní mezery odstranit.

6.2 Výsledky testování

Testování funkčnosti systému jsem prováděl jak v průběhu implementace, tak i po jejím dokončení. V průběhu testování jsem našel mnohé chyby, jež jsem následně odstranil. Dále jsem testoval bezpečnost systému pokusy o neautorizovaný přístup či odposlech hesla. Na jejichž základě jsem doimplementoval možnost práce se šifrovaným přenosem. Pokud by se ovšem jednalo o reálné nasazení systému, bylo by zapotřebí provést s pomocí budoucích uživatelů také testování použitelnosti a na jeho základě provést úpravy systému.

Testování klienta probíhalo konkrétně na platformách:

- a) MS Windows XP SP2
 - Mozilla FireFox 2.0
 - Internet Explorer 6.0 SP2
- b) GNU/Linux Kubuntu 6.10
 - Mozilla FireFox 2.0

a na serveru:

system:	Debian GNU/Linux 4.0 Etch
webový server:	Apache 2.2.3 s mod_ssl
databázový systém:	MySQL 5.0.32
skriptovací jazyk:	PHP 5.2.0

7 Závěr

Bakalářská práce se zabývá tvorbou informačního systému cestovní kanceláře s webovým rozhraním. Toto téma jsem si vybral, abych prohloubil své znalosti a dovednosti v oblasti informačních systémů.

Práci na systému jsem začal zjišťováním požadavků kladených na systém. Během několika osobních konzultací s uživateli jsem zjišťoval potřebné informace. Za pomoci těchto informací jsem vytvořil diagram případů použití s podrobnějším vysvětlením všech jeho jednotlivých případů. Při návrhu systému jsem pak na základě specifikovaných požadavků vytvořil datový model databáze vyjádřený ER diagramem. Navrhl jsem obrazovky systému a nechal si je od dodavatele schválit. Implementaci systému jsem pak prováděl v programu PSPad. I v samotném průběhu implementace jsem prováděl drobné úpravy návrhu systému. V případech různých nejasností jsem kontaktoval pracovníky cestovní kanceláře. Testování systému jsem prováděl jak v průběhu jeho implementace, tak i po jeho dokončení.

Výsledný systém splňuje všechny požadavky, jež byly na systém kladeny a mohl by být nasazen do reálného provozu. Systém by mohl být v budoucnu rozšířen i o generování webové prezentace cestovní kanceláře s možností objednávání zájezdů přes internet.

8 Literatura

- [1] KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací*. Václav Urban. 1. vyd. U Průhonu 22, Praha 7 : Grada Publishing, spol. s r. o., 1998. 492 s. ISBN 80-7169-373-1.
- [2] ZENDULKA, Jaroslav, RUDOLFOVÁ, Ivana. *Informace k předmětu Databázové systémy pro akademický rok : Studijní opora IDS* [online]. [cit. 2007-04-10].
Dostupný z WWW: <https://www.fit.vutbr.cz/study/courses/IDS/private/IDS_predn.pdf>.
- [3] HRUŠKA, Tomáš. *Informační systémy (IIS) 2006 : Programování klienta (JavaScript)* [online]. 2007 , 18.2.2007 [cit. 2007-04-13].
Dostupný z WWW:<<https://www.fit.vutbr.cz/study/courses/WAP/private/opory/>>.
- [4] *Wikipedia* [online]. 2001- [cit. 2007-04-15].
Dostupný z WWW: <<http://en.wikipedia.org/wiki/>>.
- [5] SCHMULER, Joseph. *Myslíme v jazyku UML : knihovna programátora*. Petr Somogyi. 1. vyd. U Průhonu 22, Praha 7 : Grada Publishing, spol. s r. o., 2001. 360 s. ISBN 80-247-0029-8.

9 Seznam příloh

Příloha A – Schéma databáze

Příloha B – Některé z obrazovek systému

Příloha C – CD/DVD ...

Příloha B – Některé z obrazovek systému

Pokoj	02.06.2007 09.06.2007	09.06.2007 16.06.2007	16.06.2007 23.06.2007	23.06.2007 30.06.2007	30.06.2007 07.07.2007	07.07.2007 14.07.2007	14.07.2007 21.07.2007
Čís 51, Černomorec Penzion Stamo 2 x	Jan Mokry Celkem: 26500/66122 Kč	Rezervovat Cena: 7003 Kč	Rezervovat Cena: 7005 Kč	Lenka Neumanová Celkem: 14014/14014 Kč	Lenka Neumanová 18.04.2007 Celkem: 14018 Kč	Rezervovat Cena: 7011 Kč	Rezervovat Cena: 7013 Kč
Čís 2, Černomorec Penzion Stamo 2 x	Rezervovat Cena: 7002 Kč	Jiri Stoch 07.04.2007 Celkem: 12512 Kč	Rezervovat Cena: 7010 Kč	Rezervovat Cena: 7014 Kč	Jan Mokry Celkem: 26500/66122 Kč	Rezervovat Cena: 7022 Kč	Rezervovat Cena: 7026 Kč
Čís 3, Černomorec Penzion Stamo 2 x	Rezervovat Cena: 7003 Kč	Rezervovat Cena: 7009 Kč	Rezervovat Cena: 7015 Kč	Rezervovat Cena: 7021 Kč	Rezervovat Cena: 7027 Kč	Rezervovat Cena: 7033 Kč	Rezervovat Cena: 7039 Kč
Čís 4, Černomorec Penzion Stamo 2 x	Jakub Svoboda 14.04.2007 Celkem: 15008 Kč	Rezervovat Cena: 7012 Kč	Rezervovat Cena: 7020 Kč	Rezervovat Cena: 7028 Kč	Rezervovat Cena: 7036 Kč	Rezervovat Cena: 7044 Kč	Rezervovat Cena: 7052 Kč
Čís 5, Černomorec Penzion Stamo 3 x	Jiri Stoch 26.04.2007 Celkem: 21015 Kč	Rezervovat Cena: 7015 Kč	Jan Mokry 22.04.2007 Celkem: 21075 Kč	Rezervovat Cena: 7035 Kč	Rezervovat Cena: 7045 Kč	Rezervovat Cena: 7055 Kč	Rezervovat Cena: 7065 Kč
Čís 6, Černomorec Penzion Stamo 3 x	Rezervovat Cena: 7006 Kč	Rezervovat Cena: 7018 Kč	Rezervovat Cena: 7030 Kč	Rezervovat Cena: 7042 Kč	Rezervovat Cena: 7054 Kč	Rezervovat Cena: 7066 Kč	Rezervovat Cena: 7078 Kč
Čís 7, Černomorec Penzion Stamo 3 x	Rezervovat Cena: 7007 Kč	Rezervovat Cena: 7021 Kč	Rezervovat Cena: 7035 Kč	Rezervovat Cena: 7049 Kč	Rezervovat Cena: 7063 Kč	Rezervovat Cena: 7077 Kč	Rezervovat Cena: 7091 Kč
Čís 8, Sozopol Hotel Laleya*** 2 x	Viktor Novotny 19.04.2007 Celkem: 14016 Kč	Rezervovat Cena: 7024 Kč	Rezervovat Cena: 7040 Kč	Rezervovat Cena: 7056 Kč	Rezervovat Cena: 7072 Kč	Rezervovat Cena: 7088 Kč	Rezervovat Cena: 7104 Kč
Čís 9, Sozopol Hotel Laleya*** 2 x	Rezervovat Cena: 7009 Kč	Rezervovat Cena: 7027 Kč	Rezervovat Cena: 7045 Kč	Rezervovat Cena: 7063 Kč	Rezervovat Cena: 7081 Kč	Rezervovat Cena: 7099 Kč	Rezervovat Cena: 7117 Kč
Čís 10, Sozopol Hotel Laleya*** 2 x	Rezervovat Cena: 7010 Kč	Rezervovat Cena: 7030 Kč	Rezervovat Cena: 7050 Kč	Rezervovat Cena: 7070 Kč	Rezervovat Cena: 7090 Kč	Rezervovat Cena: 7110 Kč	Rezervovat Cena: 7130 Kč
Čís 11, Sozopol	Rezervovat	Rezervovat	Rezervovat	Rezervovat	Rezervovat	Rezervovat	Rezervovat

Obrázek 9.2: Obrazovka štaflí

https://83.240.17.172 - CK KrasimTour - Mozilla Firefox

Přidat účastníka zájezdu

Termín	Letovisko	Hotel	Pokoj	Cena
30.06.2007 - 07.07.2007	Černomorec	Penzion Stamo	Číslo pokoje 2 2 x	2 x 7018 Kč

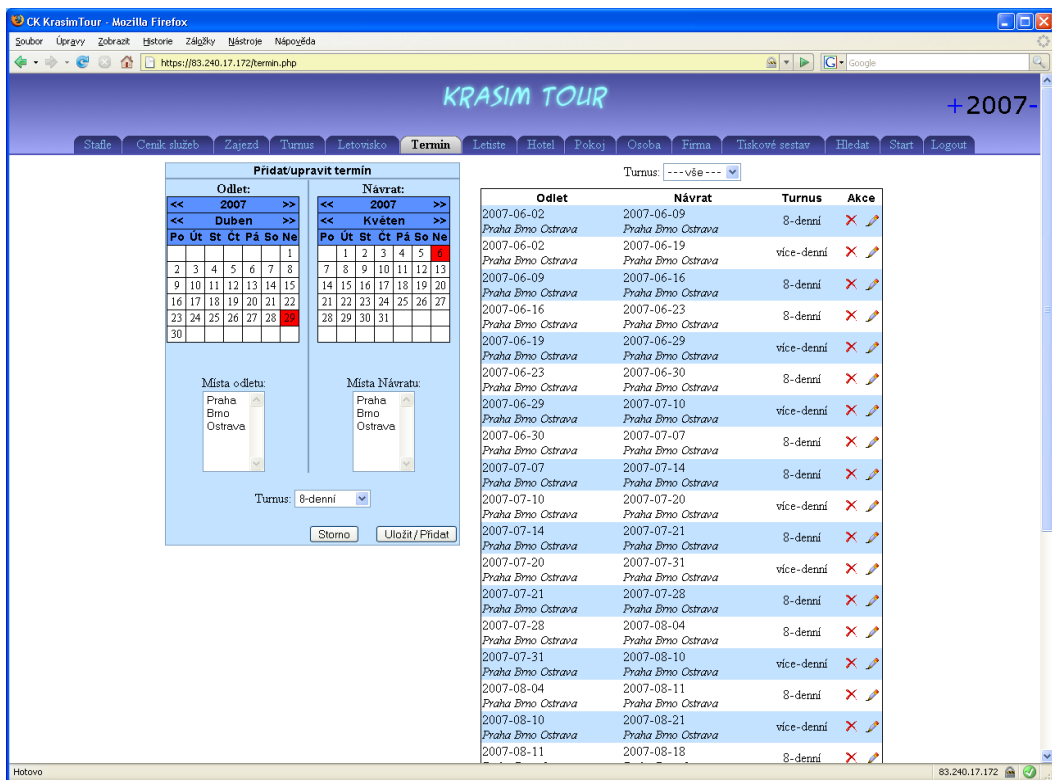
Jméno:
Příjmení:
Pohlaví: Muž Žena
Datum nar.:
(např. 21.07.1950)
Adresa:
(max 60 znaků)
Tel.:
Email:
Místo odletu:
Místo návratu:

Jedná se již o existujícího zákazníka:
Hledat osobu (příjmení):

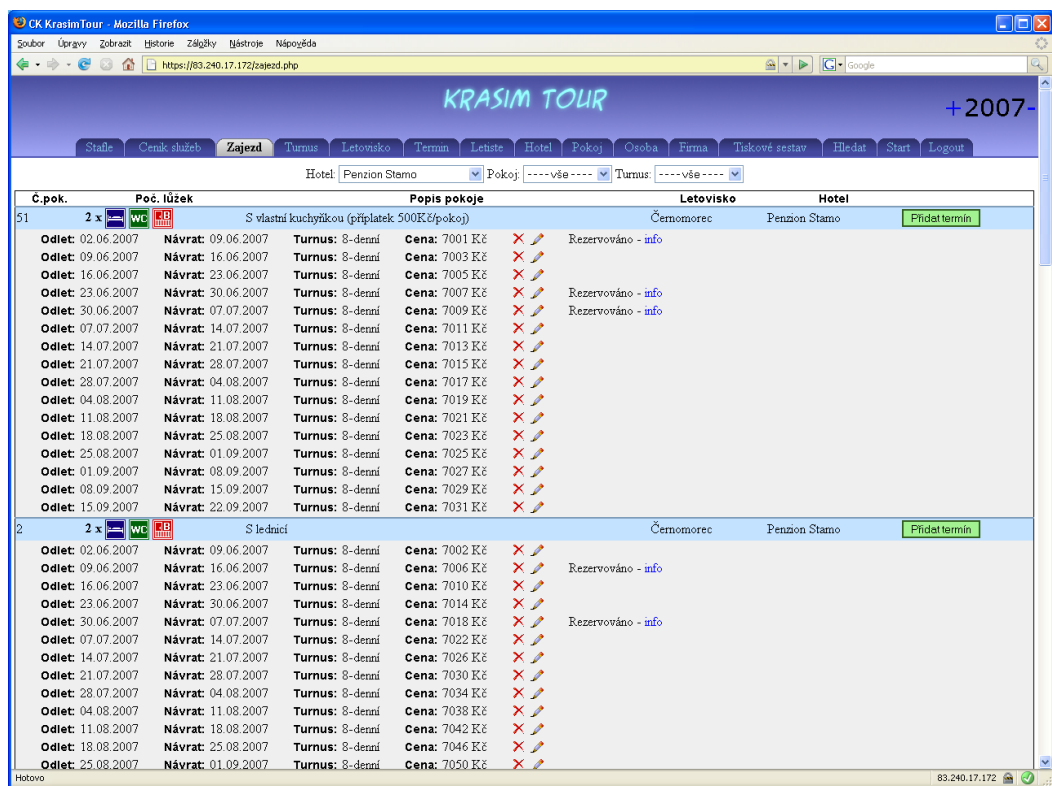
- Dvořák Petr
- Mokrá Markéta
- Mokry Jan
- Neumanová Lenka
- Novotná Jarmila
- Novotny Viktor
- Pruška Pavel
- Sedláková Kristýna
- Stoch Jiří
- Svoboda Jakub

Hotovo 83.240.17.172

Obrázek 9.3: Obrazovka přidávání účastníka zájezdu



Obrázek 9.5: Obrazovka pro správu terminů



Obrázek 9.6: Obrazovka pro správu zájezdů