

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Rezervační systém pro ubytovací zařízení

Vojtěch Havránek

Bakalářská práce

2010

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vojtěch HAVRÁNEK**
Osobní číslo: **I07613**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Rezervační systém pro ubytovací zařízení**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je vytvořit systém, který bude sloužit provozovatelům ubytovacích zařízení, k evidenci ubytovaných a služeb. Součástí systému bude modul Kalendář, který bude sledovat aktuální stav naplněnosti ubytovacího zařízení, bude upozorňovat na nově příchozí hosty a také bude upozorňovat na termíny úhrad za pobyt jednotlivých hostů. Další součástí bude rezervační rozhraní, kde si budou moct zákazníci rezervovat ubytování a služby.

Program bude vytvořen pomocí jazyka PHP a data budou uložena v databázi MySQL. Pro zákazníky bude viditelný pouze rezervační formulář, který bude nabízet volné termíny.

Ostatní funkce budou v intranetu

Cíl teoretické části: * Porovnat různé rezervační systémy * Shrnout hardwarové a softwarové možnosti oddělení internetu od intranetu * Navrhnout zabezpečení aplikace, aby se zákazník nemohl dostat intranetu Cíl praktické části: * Realizovat navržený systém jako webovou aplikaci pomocí jazyka PHP a databázi MySQL. * Realizovat zabezpečení.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

***KOFLER, Michael, ÖGGL, Bernd. PHP 5 a MySQL 5 : Průvodce webového programátora. BRNO : Computer Press, 2007.**

***GUTMANS, Andi, BAKEN, Stig Saether, RETHANS, Derick. Mistrovství v PHP 5. Brno : Computer Press, 2008.**

Vedoucí bakalářské práce:

RNDr. Josef Rak

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

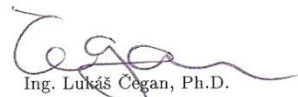
Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2010

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 2. 5. 2010

Vojtěch Havránek

Anotace

Tato bakalářská práce se zabývá problematikou rezervačních systémů a jejich dnes rozšířeným způsobem řešení. Tím jsou online rezervace pomocí webových aplikací. Zaměřuje se tedy na porovnání rezervačních systémů, řešení jejich slabých a silných stránek a také se zabývá zabezpečením aplikace.

Klíčová slova

Rezervace, rezervační systém, ubytovací zařízení

Title

Reservation system for accommodation

Annotation

This bachelor work deals with the reservation systems and their today's often used method of solution, by using online booking web application. Therefore focuses on comparing reservation systems, address their weaknesses and strengths and also deals with security applications.

Keywords

Reservation, reservation system, accommodation

Obsah

Seznam zkratk	8
Seznam obrázků	9
Seznam tabulek	9
1 Úvod	10
2 Ubytovací rezervační systémy	11
2.1 Rezervační systém	11
2.2 Způsob rezervace	11
2.3 Moderní rezervační systémy.....	12
3 Technologie rezervačních systémů	14
3.1 HTML, XHTML a CSS.....	14
3.1.1 HTML.....	14
3.1.2 XHTML.....	15
3.1.3 CSS	16
3.2 PHP.....	17
3.3 Databázové servery.....	18
3.3.1 MySQL.....	18
3.3.2 Oracle.....	19
3.4 Webové servery	19
3.4.1 Apache HTTP Server.....	19
4 Analýza rezervačních systémů	20
4.1 Booking-System BS	20
4.1.1 Struktura a funkce.....	20
4.1.2 Nabízené varianty a služby.....	21
4.1.3 Shrnutí	22
4.2 Czechproduct rezervační systém	22
4.2.1 Struktura a funkce.....	22
4.2.2 Nabízené varianty a služby.....	23
4.2.3 Shrnutí	23
4.3 Previo rezervační systém.....	24
4.3.1 Struktura a funkce.....	24
4.3.2 Nabízené varianty a služby.....	25

4.3.3	Souhrn.....	26
4.4	Celkové shrnutí.....	27
5	Zabezpečení aplikace.....	28
5.1	Zabezpečení PHP.....	28
5.1.1	Session ID (SID).....	28
5.1.2	XSS.....	28
5.2	Zabezpečení databáze.....	29
5.2.1	SQL injection.....	30
5.2.2	Citlivé údaje v databázi.....	31
5.3	Intranet.....	32
5.3.1	VPN.....	33
5.4	HTTPS.....	34
6	Rezervační systém pro ubytovací zařízení.....	36
6.1	Použité technologie.....	36
6.2	HTML stránka.....	36
6.3	PHP aplikace.....	37
6.3.1	Rezervace.....	38
6.3.2	Události a kalendář.....	38
6.4	Databáze MySQL.....	39
6.5	Zabezpečení.....	40
7	Závěr.....	42
	Literatura.....	43
	Příloha A – funkce pro vytvoření události.....	46

Seznam zkratek

HTML	Hypertext Markup Language
XHTML	Extensible Hypertext Markup Language
CSS	Cascading Style Sheets
PHP	Personal Home Page
HTTP	Hypertext Transfer Protocol
VPN	Virtual Private Network
SQL	Structured Query Language
SHA	Secure Hash Algorithm
XSS	Cross-site scripting

Seznam obrázků

Obrázek 1 – Ukázka rezervačního formuláře systému BS	20
Obrázek 2 – Náhled z administrační části systému BS	21
Obrázek 3 – Ukázka rezervačního formuláře systému společnosti Czechproduct	23
Obrázek 4 – Plachta rezervací systému Previo.....	24
Obrázek 5 – Ukázka rezervačního formuláře systému Previo	25
Obrázek 6 – Přehled funkcí variant systému Previo	26
Obrázek 7 – připojení pomocí VPN	34
Obrázek 8 – ER diagram	39

Seznam tabulek

Tabulka 1 – Hodnocení rezervačních systémů	27
--	----

1 Úvod

V dnešní době, všudypřítomných počítačů usnadňujících nám nepředstavitelné množství problémů a stejně tak rozšířenosti světové sítě Internet spojující celý svět, se otevírá mnoho cest k řešení situací, jež nám brání dosažení našeho cíle. S těmito možnostmi se ale kladou větší nároky i na nás samotné. Hlavně na přizpůsobitelnost, flexibilitu a schopnost v co nejkratší době a co nejefektivněji reagovat na změny a nové možnosti, které se nám otevírají každým dnem.

Jednou z těchto možností, při rozšíření Internetu do většiny domácností, bylo pro ubytovací zařízení zrození online rezervačního systému. Horské chaty, penziony, hotely a podobná zařízení mohla zareagovat a využít otevření těchto cest. Dnes, jakožto i webové prezentace takových podniků, je samozřejmostí možnost uživatele si na jejich webových stránkách zařídit rezervaci.

V mé práci vás nejprve seznámím obecně s rezervačními systémy. K čemu jsou, jak fungují a proč jejich vývoj směřoval až k online rezervacím. Dále se podíváme na nejčastější technologie používané právě těmito rezervačními systémy. Shrnu na, co a proč, se které technologie využívají a jednoduše je popíši. Následujícím tématem budou rezervační systémy na českém trhu a jejich analýza. Podívám se na klady a zápory jednotlivých produktů a nebude chybět ani závěrečné ohodnocení.

Důležitým tématem práce je i zabezpečení webových aplikací. Zabezpečení je nezanedbatelná fáze projektů, jako jsou rezervační systémy. Operuje se zde s citlivými údaji jakožto i s peněžními transakcemi. Následky zanedbané bezpečnosti pak mohou být nepředstavitelné. Ochrana přitom není až tak složitá, jak v kapitole naznačím. Po zabezpečení webových aplikací se již budu věnovat jen aplikace své.

2 Ubytovací rezervační systémy

Nejprve bych chtěl vysvětlit, co to vlastně rezervační systém je. Uvést nějaké příklady od těch naprosto nejjednodušších až po složitá a dnes také nejvíce využívaná řešení. Jak se taková rezervace vytvoří. Podíváme se na výhody a nevýhody moderních rezervačních systémů a proč vlastně vznikají.

Rezervační systémy samozřejmě neřeší jen situace spojené s ubytovacími zařízeními, ale rezervace se využívá i u sportovních areálů, restaurací či cestovních kanceláří. Do jisté míry se tyto sféry prolínají, ale v této bakalářské práci se budeme zabývat jen systémy, které jsou zaměřené na ubytovací rezervace.

2.1 Rezervační systém

Pokud si problematiku rezervačních systémů pro ubytovací zařízení hodně zjednodušíme, tak si pod tímto pojmem můžeme představit i primitivní knihu hostů, kde si evidujeme návštěvníky, jejich rezervace a platby, datum příjezdu a odjezdu a další informace námi zhodnocené jako informace důležité k vytvoření nového záznamu ve výše zmíněné knize.

Takový systém dnes ale nikomu stačit nebude. I ten nejmenší podnikatel, který třeba nabízí v letní sezóně dva pokoje na ubytování, si raději vytvoří menší aplikaci v nějakém tabulkovém kalkulátoru, jež mu zjednoduší, zpřehlední a zpříjemní práci s podobnými informacemi, než aby si pracně vytvářel psané záznamy, které jsou navíc náchylnější ke zcizení osobních údajů, či ke zničení celého záznamu.

Pod termínem rezervační systém si dnes tedy představíme sofistikovanější aplikaci, která nám pomáhá se správou hostů, jejich rezervacemi. Upozorňuje nás na nové události, jako je registrace nových hostů, vytvoření nové rezervace či blížící se termín, kdy host musí zaplatit jistou zálohu za rezervované ubytování. Může automaticky generovat e-maily směřované zákazníkům, jež je informují o platbách nebo novinkách v nabízených službách. Vyhledává volné termíny pro jednotlivé pokoje, vypočítává cenu celé objednávky a zabraňuje vícenásobné rezervaci v kolidujících termínech.

2.2 Způsob rezervace

Od kvality rezervačního systému se také odvíjí možnost, jak nám onu rezervaci zpracuje. Ruční zpracování rezervace je samozřejmostí pro každý systém. Pod tímto termínem si představme, že onu rezervaci do systému musíme prostě ručně vložit. Základem je, aby nám aplikace zjistila, zda zákazníkem požadovaný termín je nebo není obsazen a po vložení potřebných informací se rezervace vytvoří a mi ji pak následně můžeme dále spravovat. Funkce ručního zadávání je naprosto nezbytná i v případě, že nám dává rezervační systém možností více.

Pokud ale aplikace nedisponuje jinou variantou, tak uživateli příliš správu nezjednoduší. A nejen uživateli v roli správce chaty.

Host, který by si chtěl rezervovat některý z termínů, musí vynaložit úsilí v podobě příchodu na ubytovnu, zavolání správci chaty, či alespoň napsání e-mailu. Samozřejmě takovéto jednorázové úsilí nemusí přinést své vytoužené ovoce v podobě vyřízení rezervace. Když pomineme, že jsme omezeni ve smyslu, kdy je ubytovací zařízení otevřeno, pro osobní domluvu rezervace, nebo musíme počítat s omezením spojeným s telefonickou rezervací, kde je zase problém s operátorem, jež není na lince čtyřicet hodin sedm dní v týdnu, tak námi požadovaný termín už může být plně zaplněn a stejně se tedy nedostaneme k tíženému výsledku.

Z pohledu správce ubytovacího zařízení je nevýhodou zaměstnanec (nebo práce navíc samotného správce), který musí být u telefonu, na recepci, či zpracovávat e-maily a řešit následné rezervace.

Výše zmíněné nedostatky autoři rezervačních systému samozřejmě dávno odhalili, a proto pro náročnější uživatele svých aplikací hledají nová řešení. Existuje software, který je třeba stavěn pro dotykové displeje. Taková aplikace se pak nastaví tak, aby byla schopna přijmout a automaticky zpracovat objednávku přímo od zákazníka. Pro hosta tento proces není nikterak usnadňující. Stále musí dorazit k onomu zařízení a zadat své přání. Správci na druhou stranu ubude plno rezervací, které by musel sám zpracovat. Tento systém ho pak následně upozorní v administrační části o nové rezervaci a tím je ušetřena práce v podobě kontaktu se zákazníkem, řešení termínu a vkládání rezervace.

Nejefektivnějším a proto také nejvyužívanějším řešením rezervací je proto rezervace online za pomoci webové aplikace. Téměř každý podnik dnes nějakou webovou prezentaci má. O to více je důležitější právě u takových zařízení, jako jsou zařízení ubytovací. Pak není problém zimplementovat do stránek i online rezervační systém. Pro správce je zjednodušení obdobné jako u předchozího příkladu s dotykovým displejem. Automatické zarezervování zákazníkem pro něj nepřináší další práci na víc a může tak mimo jiné ušetřit za pracovníka, který rezervace přijímal a tím zaplatit onu implementaci online rezervace. Hlavní přínos je ale pro zákazníka. Ten si v klidu a pohodlí svého domova může rezervaci obstarat. Odpadá tak zbytečné telefonování, mailování či osobní domlouvání ubytovaného termínu. Vzhledem k dnešním možnostem plateb nemusí host ani pro případnou splátku zálohy vstávat od počítače. Celý proces se tím pádem zjednodušuje oběma stranám, a pokud chce majitel jakéhokoliv ubytovacího zařízení v dnešní konkurenci uspět, je pro něj téměř nezbytností zmíněnou službou disponovat.

2.3 Moderní rezervační systémy

V předešlém povídání jsme se propracovali k závěru o online rezervačních systémech a jejich nejvhodnějším použití. Proto bych pod pojmem moderní rezervační systém viděl právě ten, který umožňuje online rezervace pomocí webové aplikace.

Snad veškeré výhody takového systému již byly řečeny. Kdybych je shrnul do jedné věty. Levné usnadnění práce prodejce i kupujícího zajišťující podnikateli konkurenceschopnost. Taková věta jasně podporuje myšlenku na pořízení podobného

rezervačního systému. Skýtá ale toto řešení i nějaké záporné body? Může se nás vůbec nějaký zápor zmocnit tak, že se přikloníme na druhou stranu? Podívejme se tedy na opačnou stránku věci.

Ať už mluvíme o jakékoliv online službě, tak jsou povětšinou skloňovány se slovem bezpečnost. Není se čemu divit. Služba je totiž přístupná odkudkoliv a to je fakt, nad kterým je nutno se pozastavit. Konkrétně u rezervačních systémů musíme brát v potaz, že většina obchodníků po nás bude požadovat nějaké kontaktní údaje, či osobní informace. Pokud tedy jsou požadovány citlivé údaje, mnoho zákazníků může znejistět, jestli s nimi bude nakládáno opravdu tak, jak poskytovatel služby slibuje. I přes jeho nejlepší úmysly je tu stále riziko napadení systému z vnějšku. Systém s citlivými údaji totiž pracuje a může být tak náchylný k propuštění citlivých dat třetí osobě, která se o to pokouší. Ještě větší problém pak může nastat u online plateb.

Poskytovateli služby internetový robot [1], které jsou také často označovány jako „boti“ Internetový robot nebo častěji jenom bot je počítačový program, který pro svého majitele opakovaně vykonává nějakou rutinní činnost na internetu - obvykle sbírá data, odesílá a zpracovává požadavky na služby vzdálených serverů. Taková věc pak může zahltit databázi, vytvořit falešné rezervace atp.

Jak tyto problémy řešit se v této bakalářské práci ještě budu zabývat. Tudíž bych chtěl jen vyzdvihnout, že práce se zabezpečením online aplikací je na dobré cestě a drtivá většina takovýchto služeb je dostatečně ochráněna. Samozřejmě se nacházejí i výjimky, které pak snášení vlnu nejistoty na podezřívavé klienty při používání kvalitních aplikací. Také se říká, že každý obranný systém je prolomitelný, ale já bych nebyl až tolik skeptický.

Sečteno podtrženo. Pokud není podnik natolik malý, že by se mu nevyplatilo investovat do online rezervačního systému, tak bych s jeho implementací do webové prezentace neváhal.

3 Technologie rezervačních systémů

V předešlé kapitole toho bylo obecně o rezervačních systémech napsáno dosytnosti. Přesto jsem se zatím vůbec nezabýval jejich technologickým řešením, jazyky a protokoly, které využívají a ve kterých jsou tvořeny.

3.1 HTML, XHTML a CSS

Vzhledem k tomu, že rezervační systémy jsou povětšinou webové aplikace, tak ke svému využití musejí být implementovány do webových prezentací, které jsou stavěny na v nadpisu zmíněných jazycích, na které se teď podrobněji podíváme.

3.1.1 HTML

Pod zkratkou HTML se schovává název Hypertext Markup Language. Jde o značkovací jazyk využíván pro tvorbu stránek v systému WWW (World Wide Web), jež zprostředkovává publikaci dokumentů na Internetu. Jazyk HTML vychází ze staršího univerzálního značkovacího jazyka se zkratkou SGLM (Standard Generalized Markup Language). Tak jak se vyvíjely webové prohlížeče, tak se s nimi vyvíjel i značkovací jazyk HTML. Současně s ním byl vytvořen i protokol, jež se stará o přenos v počítačové síti HTTP (Hypertext Transfer Protocol). [2]

Vývoj verzí jazyka se datuje k roku 1991 a prošel rozsáhlými změnami, které je zbytečné zde popisovat. Předpokládanou poslední a dnes využívanou verzí je verze HTML 4.01. Ta je v souladu s mezinárodní normou ISO 8879 - Standard Generalized Markup Language (ISO8879).[3]

Jazyk využívá značky zvané tagy, jež jsou v drtivé většině párové (mají počáteční a koncový tag, který se liší od počátečního tím, že před názvem disponuje znakem lomítka) a jsou ohraničeny úhlovými závorkami (<, >). Část dokumentu, jež obsahuje počáteční tag, nějaký obsah a koncový tag se nazývá *element*. Počáteční tag pak může obsahovat i *atribut*, který se skládá z *názvu* a *hodnoty*. Pro ukázkou uvedu příklad:

```
<p align="right"> Ahoj, světe </p>
```

Předepsaná struktura, kterou musí plnit každý dokument v HTML, pokud tedy chce být validní, začíná deklarací DTD (Document Type Definition - je povinná až od verze HTML 4.01), která může vypadat třeba takto:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Struktura pak pokračuje kořenovým elementem *html*, který je určen počátečním tagem `<html>` a zakončen koncovým tagem `</html>`. Tento element reprezentuje celý dokument. Následuje pak hlavička dokumentu, v níž jsou obsažena *metadata*, jež obsahují doplňující informace o dokumentu jako je například jazyk, autor, popis, klíčová slova či použité kódování. Hlavička je uzavřena mezi tady `<head>` a `</head>`. Do posledního dělení

struktury patří element *body* dán párovým tagem `<body>`. Tento element už obsahuje vlastní text dokumentu.

Tagy ještě můžeme rozdělit do tří skupin:

- Strukturální – vytváří formu a strukturu dokumentu (`<p>`, `<h1>`)
- Popisné – popisují povahu obsahu elementu (`<tittle>`)
- Stylistické – určují vzhled elementu při zobrazení

Od stylistických tagů se pomalu ustupuje. Nahrazuje je použití kaskádových stylů, kterými se v této práci ještě budu zabývat.

3.1.2 XHTML

Zkratka XHTML je odvozena z anglického extensible hypertext markup language, což si můžeme přeložit jako „rozšiřitelný hypertextový značkovací jazyk“. Byl navrhnut jako pokračující značkovací jazyk po HTML verzi 4.01. Nakonec ale došlo k vývoji nové verze značkovacího jazyka HTML s označením HTML 5. Současně se ještě pracuje na vývoji jazyka XHTML 2. [4]

Používaných verzí XHTML na rozdíl od svého předchůdce je více. Verze XHTML 1.0 je rozdělena na tři verze (*Strict*, *Transitional* a *Frameset*). Spíše se ale používají jen první dvě jmenované.

XHTML Strict je vytvořen především pro osvobození od formátovacích značek, které rozvrhují stránku. Počítá se v něm s použitím kaskádových stylů a upouští od některých popisných elementů.

Naproti tomu XHTML Transitional využijete, pokud chcete vytvořit webovou stránku pro zastaralé prohlížeče, které ještě nerozumí kaskádovým stylům, či využívají některé zavrhnuté elementy.

Nejvíce využívanou verzí XHTML je verze 1.1, která vynechává téměř všechny prezentační vlastnosti. Je velice podobná s verzí XHTML 1.0 Strict, ale snaží se o větší kompatibilitu s dalšími a budoucími dokumenty z rodiny XHTML.

Nejmarkantnějšími rozdíly mezi HTML a XHTML je zakončení všech tagů a to i nepárových. Nepárové tagy můžeme zakončit následujícími způsoby:

- ` `
- ``
- ``

První dva znázorněné tagy je z různých důvodů nedoporučeno používat. Nejčastěji se tedy setkáme s posledním znázorněním nepárového tagu v XHTML.

Další důležitý rozdíl tkví v rozdílu malých a velkých písmen, jelikož XHTML je jazyk case sensitive. Tagy deklarované a odkazované v DTD, jsou psány malými písmeny, a proto toto pravidlo musíme dodržovat i my. Dále veškeré hodnoty atributů musí být v uvozovkách a dokument také musí začínat XML deklarací. Rozdílů se najde víc, ale pro naši představu jsou výše zmíněné dostačující.

3.1.3 CSS

O kaskádových stylech (převzato z anglického Cascading Style Sheets), jsem se již několikrát zmínil, a tak je na čase si i o nich něco povědět. Stejně jako předešlé jazyky i kaskádové styly jsou hojně využívány ve webových prezentacích ubytovacích zařízení a jejich rezervačních systémech. Definice nám říká, že se jedná o jazyk, kterým vytváříme vzhled aplikací psaných v jazycích HTML, XHTML a v poslední řadě XML. Kaskádové styly byly navrženy, stejně jako ostatní výše zmíněné jazyky, organizací W3C. [5]

Momentálně konečnou a používanou verzí je verze CSS 2.1. Práce ale stále pokračují na verzi CSS3. Jak již zde bylo řečeno, tak HTML obsahuje spousty elementů, jež nepopisují obsah ani strukturu dokumentu, ale určují jeho vzhled. Autoři kaskádových stylů měli cíl, jímž bylo právě oddělení obsahové stránky dokumentu od stránky vizuální.

Kaskádové styly mají také svou strukturu. Ta se skládá z pravidel, které obsahují selektor a blok deklarací. Blok je pak tvořen seznamem deklarací, jež každá z nich disponuje nějakou vlastností. Hodnota vlastnosti následuje za dvojtečkou a deklarace je zakončená středníkem. Uvedu příklad pravidla:

```
body {
  background-color: SteelBlue;
  color: Black;
  margin: 0px auto;
}
```

Aplikace se s kaskádovými styly propojuje třemi způsoby. Prvním je inline zápis přímo do počátečního tagu elementu. Takový styl je pak určen jen pro daný element.

```
<p style="color: red;"> Ahoj, světe. </p>
```

Příliš se ale nevyužívá. Další způsob je definování stylu pomocí elementu *style*. Obsah mezi tagy `<style>` a `</style>`, se pak formátuje podle předepsaných selektorů.

Nejhojněji využívaným způsobem je pomocí externího CSS souboru, jež se připojí k dokumentu v hlavičce. Skládá se pak z pravidel, které určují vzhled stránky. Připojení v hlavičce vypadá následovně:

```
<link rel="stylesheet" type="text/css" href="style.css" >
```

Výhodami CSS ve spojení s HTML pak jsou například rozsáhlejší možnosti ve formátování textu a celé webové stránky. Absence složitého vyhledávání stylu nějakého textu v HTML kódu, kvůli jeho změně. Veškeré formátovací položky jsou totiž obsaženy v externím souboru. Možnost dynamické změny elementů pomocí Javascriptu a další výhody plynoucí z jeho použití.

3.2 PHP

PHP, původně převzato z *Personal Home Page*, je rekurzivní zkratka pro termín *Hypertext Preprocessor* (Hypertextový preprocesor). Jedná se o skriptovací programovací jazyk, vytvořený za účelem programování dynamických internetových stránek. V našem případě, kde mluvíme o rezervačních systémech, vytvořených jako webové aplikace, programátoři začleňují PHP jazyk přímo do struktury jazyka HTML a XHTML. Jinak ho ale můžeme využít i v desktopových a konzolových aplikacích. [6]

Uživatel webové aplikace by se k PHP skriptům neměl dostat. Ty jsou totiž prováděny na straně serveru a uživatel už vidí jen jejich výsledek. Při tvorbě se autoři inspirovali jazyky jako je C, Java, Pascal nebo Perl. PHP je velmi univerzální, co se týče operačního systému, ale podporuje také velké množství databázových systémů a internetových protokolů.

K tvorbě webových aplikací se ke kombinaci s PHP používá nejčastěji databáze MySQL. O té si v této práci ještě něco řekneme, stejně tak jako o webovém serveru Apache, bez něhož by nám byly PHP skripty k ničemu. Následuje malá ukázka PHP skriptu, která nám vypíše na obrazovku text „Ahoj, světe“.

```
<?php
    echo "Ahoj, světe!";
?>
```

Tento skript, jak jsem výše zmínil, tedy uživatel neuvidí. Uvidí jen text vypsáný mezi uvozovkami. Přitom je PHP zdrojový kód vkládán přímo do HTML dokumentu. Oba zdrojové kódy tedy najdeme v jednom souboru. PHP kód je ale zřetelně oddělen od HTML a to, jak je patrné s předešlé ukázky, označením začátku PHP kódu pomocí znaků `<?php` a ukončením znaky `?>`. Uvedu ale ještě dvě možnosti. [7]

```
<script language="php">
    echo "Ahoj, světe!";
</script>

<? echo "Ahoj, světe!"; ?>
```

Dlouhý a krátký zápis, který sem uvedl, není k tolik k vidění. Krátký zápis můžete použít jen v kombinaci s parametrem *short_open_tag*. Dlouhý zápis se využívá hlavně pro Frontpage, jež si s jinými způsoby zápisu nerozumí.

Zajímavostí skriptovacího jazyka PHP, na rozdíl od jiných programovacích jazyků (a to i přes to, že z nich vychází), je řešení datových typů. Většina jazyků totiž vyžaduje při definování proměnné i určení jejího datového typu. Ten se u PHP proměnných předem definovat nemusí. To už PHP rozezná sám a podle toho se přizpůsobí.

Další výhodou jazyka PHP jsou pole. Ona výhoda spočívá převážně v jejich flexibilitě, neboť lze na pole ukazovat i jinak než čísla. Taková vlastnost chybí u jiných programovacích jazyků. PHP pole se tedy skládá z klíčů, jež jsou datového typ Integer

nebo String, a k nim přiřazených hodnot libovolného datového typu. Pro lepší pochopení předved jednu ukázkou:

```
$mojePole = array(
    "jmeno" => "Vlastimil",
    "prijmeni" => "Starý"
    "email" => "vstary@email.cz"
);
```

Ukázka nám vytvoří pole pod proměnou *\$mojePole*. Obsahuje tři hodnoty, které jsou schovány pod svými klíči. Pole u PHP pak samozřejmě může využít i dalšího vnořeného pole. Jako chválu PHP bych ještě asi připojil slušné zpracování dokumentace, výtečnou podporu u hostingových služeb a také velmi svobodnou licenci.

Nepozastavil jsem se zatím nad verzemi jazyka. Počátek bychom mohli datovat někdy k půlce roku 1995, kdy byla představena světu verze 1.0. Od té doby ale prošel jazyk rozsáhlým vývojem. Poslední verze, vydaná začátkem března 2010, je 5.3.2.

3.3 Databázové servery

Pro registrované uživatele a jejich rezervace, pokoje a jejich vybavení a další potřebné věci, důležité ke správě chaty, či jiného ubytovacího zařízení, potřebujeme také nějaké úložiště dat. Tímto úložištěm pro nás bude nějaký databázový systém. Základem takového systému jsou samozřejmě tabulky, což je jeden z databázových objektů. Uvedu přehled dalších důležitých objektů:

- trigger – česky spouště, je mechanismus, který spustí nějakou činnost při definované události.
- pohledy – pomocí SQL příkazů vytvořené tabulky, které třeba spojují více tabulek do jedné.
- procedury a funkce – si můžeme představit stejně jako procedury a funkce v jiných programovacích jazycích. Jednoduše provádí určitou sekvenci příkazů nad danými tabulkami. Funkce navíc disponuje nějakou návratovou hodnotou.

Nyní bych se podíval na některé konkrétní databázové systémy. [8]

3.3.1 MySQL

Ač tento databázový systém vytvořila švédská firma MySQL AB, vlastníkem je společnost Sun Microsystems. Komunikace v této multiplatformní databázi probíhá pomocí jazyka SQL. Pokud zůstaneme u webových aplikací, tak se jedná o nejpoužívanější databázi. Tento titul si vysloužila díky jednoduché implementaci do různých operačních systémů, díky svému vysokému výkonu a v neposlední řadě díky tomu, že se jedná o volně šiřitelný software. MySQL bylo vytvořeno především pro co nejlepší rychlost. Nemůžeme se potom divit, že donedávna ještě nepodporovalo výše zmíněné databázové objekty, jako pohledy, trigger nebo uložené procedury. Momentálně jsou ale právě díky programátorům

webových stránek tyto vlastnosti doplňovány. Verze MySQL 5.1 je v dnešní době nejnovější verzí. V přípravě je také verze MySQL 6. [9]

3.3.2 Oracle

Oracle jako moderní multiplatformní databázový systém disponuje velmi pokročilými možnostmi zpracování dat. Samozřejmostí je podpora jazyka SQL, dále jazyka PL/SQL (rozšiřuje vlastnosti SQL pro vytváření uložených procedur, funkcí atp.) nebo i objektových databází. Oracle Database 11g můžeme označit jako verzi aktuální.

Jeho vysoký výkon a pokročilé možnosti si ale berou svou daň. Daň v podobě menší rychlosti a větší sofistikovanosti, která pak brání programátorům webových aplikací v jeho výběru, na místo databázového systému MySQL. [10]

3.4 Webové servery

Různé webové servery disponují různými vlastnostmi. Pár jich mají ale společných. Webový server musí být připojen k počítačové síti a přijímá HTTP¹ požadavky, jež vyřizuje a vrací odpověď počítači, který požadavky vznesl. Odpověď si například můžeme představit jako dokument HTML. Pokud byl požadavek vyřízen v pořádku, tak součástí odpovědi se odešle i stavový kód s hodnotou 200. Jiné hodnoty stavového řádku (3xx, 4xx, 5xx) označují chybová hlášení.

Informace, vracené webovým serverem, jsou buď předpřipravené soubory. V našem případě většinou stránky HTML. Nebo, podle požadavků uživatele, zpracuje data nečastěji z databáze, zpracuje a zformuluje do formátu HTML, aby neměl webový prohlížeč problémy s jeho prezentací.

3.4.1 Apache HTTP Server

Naprostá většina webových stránek využívá webového serveru Apache. Konfigurace a nastavení serveru, není zrovna nejprimitivnější operací. K hlavní konfiguraci využíváme souboru *httpd.exe*. K dalším nastavením máme k dispozici ostatní konfigurační soubory. V dnešní době ale konfiguraci nemusíme téměř vůbec řešit. Pokud si chceme svůj vlastní webový server rozjet na osobním počítači, tak je nám k dispozici sousta utilit, s jejíž pomocí nás otevírání konfiguračního souboru mívá. [12] Nejnovější kompletně dokončenou verzí je Apache HTTP Server 2.2.15. [13]

¹ HTTP (Hypertext Transfer Protocol) – internetový protokol vytvořený pro výměnu hypertextových HTML dokumentů [11]

4 Analýza rezervačních systémů

V mém úvodním tématu, kde jsem se obecně věnoval rezervacím a rezervačním systémům, jsem mluvil o konkurenční schopnosti podniků, které využívají online rezervační systém. Díky tomu nám dnes vyrostla nejedna firma prodávající své rezervační systémy. Na nejrozšířenější z nich bych se teď trochu podrobněji podíval.

Vždy nejprve krátce představím uvedený rezervační systém. Poté se podíváme na jeho strukturu, či nabízené služby a varianty. V závěru pak uvedu nějaké shrnutí, které zkalkuluje plusy a mínusy.

4.1 Booking-System BS

Booking-System BS je jeden z projektů společnosti R-INCON, s.r.o., sídlící v Praze. Byl vytvořen nejen pro rezervační systémy, ale také pro rezervaci vstupenek, půjčoven a třeba i balíčků služeb. Prezentován je na stránkách [HTTP://WWW.B-S.CZ/](http://www.b-s.cz/). [14]

4.1.1 Struktura a funkce

Struktura BS se dá rozdělit do tří sekcí. Jde o aplikaci pro zákazníky, poskytovatele a zprostředkovatele. Jednotlivé sekce teď podrobněji popíši.

Booking system pro zákazníky je aplikace, se kterou pracuje potenciální host na webové stránce konkrétního ubytovacího zařízení. Zřizuje si tedy zde registraci a následnou rezervaci, program mu vypočítává cenu objednávky a umožňuje zaplatit pomocí předem nastavených platebních služeb.

Vyberte období pro rezervaci a měnu

Datum příjezdu: ( kalendář)

Datum odjezdu: ( kalendář) nebo počet nocí:

Zvolte měnu:

Datum zadávejte ve formátu den.mesic.rok

 **dvoulůžkový double - exclusive**

Dispozice:	Vybavení:	Cena:
<ul style="list-style-type: none">• pevných lůžek: 2• přistýlek: 1• postýlek: 1	<ul style="list-style-type: none">• balkon• vířivá vana• minibar• trezor	<ul style="list-style-type: none">• podle období cca 900,00 až 1 100,00 za pokoj

Obrázek 1 – Ukázka rezervačního formuláře systému BS

Rezervace probíhá tak, že zákazník zadá datum příjezdu a odjezdu. Ověří dostupnost lůžek a některé z nich si objedná.

Jedno z rozhraní je rozhraní pro poskytovatele služby. Je tedy určeno pro správce a majitele ubytovacích zařízení. Zde si poskytovatel může nejen vytvořit vlastní prezentaci, ale figuruje zde i nastavení kapacit, provozních dob, cen a slev. Disponuje také rezervačním kalendářem, který funguje jako aktivní modul. S jeho pomocí si podnikatel může i sám vytvářet rezervace přijaté jinak, než z online webové prezentace. Umožňuje také import z jiných (recepčních) systémů.

Vytvoření objednávky

SPECIFIKACE OBJEDNÁVKY

Datum příjezdu	Datum odjezdu	Počet nocí	Měna
<input type="text"/>	<input type="text"/>	<input type="text"/>	koruny

Přiřazený pokoj	Dospělích	Děti 0-3 let	Děti 3-10 let	Stravování	Cena
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	bez stravy	0,00

Přidat další **Vypočítat cenu**

OBJEDNAVATEL

Oslovení:

Jméno:

Příjmení:

RČ/ČP:

Telefon:

Fax:

E-mail:

Obrázek 2 – Náhled z administrační části systému BS

Poslední rozhraní je určeno pro informační centra či cestovní agentury. Tito zprostředkovatelé na základě tohoto rozhraní pak můžou vytvářet nabídky, stejně tak jako provádět rezervace atp. Aplikace disponuje i modulem, jež nabízí nepřetržitý přehled o provizích od poskytovatelů. Umožňuje vytvářet i akční balíčky.

4.1.2 Nabízené varianty a služby

Poskytovatelé ubytování mají na výběr ze dvou nabízených variant, které se liší jejich možnostmi, ale i cenou. [15]

4.1.2.1 ABSOLUTE

První variantou, kterou si představíme, je varianta *Absolute*, která je stavěna hlavně pro náročnější obchodníky. Podle prodejce disponuje prvotřídní aktualizací rezervací. Je vhodná do větších hotelů a zařízení. Uživatelům se zobrazuje aktuální přehled volných lůžek. K dispozici je i využití on-line plateb. Očekává se stálý přístup na internet.

4.1.2.2 STANDART

U levnější varianty je také samozřejmostí on-line rezervace pro zákazníka. Předpokládá se ale, že poskytovatel nevyžaduje nepřetržitý přehled nad internetovými objednávkami. Protože přehled kapacit nemusí být vždy aktuální, je nutno potvrdit rezervaci pomocí SMS či e-mailu.

4.1.3 Shrnutí

Booking system je přehledně vytvořený rezervační systém, který mohou, díky svým variantám, využít jak velké podniky, tak menší penziony a chaty. Přehledné grafické zpracování zajišťuje snadnější práci. Jednoduchost nezpůsobí znechucení potencionálních klientů a spíše potěší. Také svůj systém na webových stránkách poměrně dobře prezentují.

Klady:

- Přehlednost a jednoduchost
- Nabízené možnosti a funkce
- Zahrnutí zprostředkovatele

Zápory:

- Absence zkoušky zdarma

4.2 Czechproduct rezervační systém

Společnost Czechproduct je různorodá firma, jejíž produkty jsou například redakční systémy, webové stránky, internetové obchody. Nabízí také webhosting a, co je pro nás důležité, zřizuje rezervační systém. Informace o něm najdeme na stránkách [HTTP://WWW.CZECHPRODUCT.CZ/44723/REZERVACNI-SYSTEM/](http://www.czechproduct.cz/44723/rezervacni-system/) [16]

4.2.1 Struktura a funkce

Samozřejmostí u každého rezervačního systému je část poskytovatelská a návštěvnická, kterými disponuje i rezervační systém od společnosti Czechproduct. Obě tyto sekce ale neoplývají takovou propracovaností jako předchozí rezervační systém, jež jsem představil. Rezervaci Vám sice od zákazníka bude zaslána. Ošetřeno je, abyste si nemohli zarezervovat termín rok starý, ale nemůžete si přímo na webu ověřit aktuální zaplněnost. Absentuje i základní funkce, jako možnost úpravy objednávky.

Poskytovatelské rozhraní nám tedy zpracovává objednávky zasílané od hostů. Zpracuje je ale takovým způsobem, že vygeneruje automatický e-mail pro klienta a samotné vyřízení objednávky je již na poskytovateli.

Termín pobytu: Od 6. května 2010 ... vyber v kalendáři ...
Do 7. května 2010 ... vyber v kalendáři ...

Jméno: _____
E-mail: _____
Adresa: _____
Telefon: _____

Počet osob: 1 dospělá osoba a Žádné dítě
Z tohoto počtu: nikdo na přistýlce, nikdo v apartmánu

Stravování: bez stravování

Další požadavky: _____

jedná se o závaznou rezervaci
 závaznost rezervace bude potvrzena až při následné komunikaci s objektem
 Uložit údaje ve formuláři pro případné další rezervace.

Odeslat

Obrázek 3 – Ukázka rezervačního formuláře systému společnosti Czechproduct

4.2.2 Nabízené varianty a služby

Další varianty rezervačního systému k dispozici nejsou. Na druhou stranu nám firma Czechproduct nabízí vytvoření celé webové prezentace s již zakomponovaným rezervačním systémem, čehož může spousta menších podniků s radostí využít.

4.2.3 Shrnutí

Rezervační systém nabízený společností Czechproduct patří do skupiny velmi jednoduchých systémů. Směřován je pro menší podniky, kde online rezervace není přímo na denním pořádku a funguje spíše jako zpestření pro potencionální hosty. Firma směřuje svoje služby k začínajícím podnikatelům, kteří potřebují spravit webovou prezentaci obohacenou o jednoduchý rezervační systém.

Klady:

- Možnost vytvoření celé webové prezentace
- Zkouška zdarma

Zápory:

- Možnosti a funkce
- Absence ověření volnosti lůžek zákazníkem
- Chybí automatická rezervace
- Špatná prezentace produktu

4.3 Previo rezervační systém

Společnost MITON Previo s.r.o. je od roku 2006 dceřinou firmou společnosti MITON CZ, s.r.o., která na trhu disponuje spoustou projektů. MITON Previo s.r.o. je ale společnost, jejímž jediným produktem je rezervační systém Previo prezentován na stránkách [HTTP://WWW.PREVIO.CZ/](http://www.previo.cz/). Autoři se honosí informací, že na vývoji aplikace spolupracovalo 11 předních hoteliérů a tím by na trhu neměli mít konkurenci. Podíváme se, co je na tom pravdy. [17]

4.3.1 Struktura a funkce

Jak bývá zvykem, aplikace obsahuje administrátorskou část a část zákaznickou. Poskytovatel má u rezervačního systému Previo opravdu širokou škálu funkcí a možností. V plachtě rezervací je k vidění třicetidenní přehled obsazenosti apartmánů. Přehled je interaktivní, takže při kliknutí na tu kterou rezervaci se otevře dialogové okno s detailem objednávky. Odtud se můžeme dostat i na konkrétního hosta. Touto funkcí disponuje i Booking system, ale nepůsobí tak přehledným dojmem.

The screenshot displays the 'PLACHTA REZERVACÍ' (Reservation Calendar) interface. At the top, there are navigation tabs: REZERVACE, HOSTÉ, POKLADNA, REPORTY, NASTAVENÍ, and NÁPOVĚDA. The main area shows a calendar for August 2007 with a grid of dates. Reservations are represented by colored bars across the calendar cells, with guest names and room numbers listed on the left. For example, reservation 102 [1+1] is for Olga B. on the 1st, and reservation 101 [1+1] is for Karlikovi on the 2nd. The interface also includes a search field for 'Hledání dle hosta', a dropdown for 'Previo hotel', and a clock icon. On the right side, there are sections for 'Příjezdy' (Arrivals) and 'Odjezdy' (Departures) with specific times and names.

Obrázek 4 – Plachta rezervací systému Previo

Navíc nad rozdíl od předešlých rezervačních systémů máme k dispozici funkce další. Cizinecká policie, která zkoumá hosty z jiných zemí, graf vytíženosti hotelu, tisk paragonů, faktur, knihy hostů a dalších úředních dokumentů. Toto a další možnosti rezervačnímu systému Previo nejsou cizí. Správce má také možnost sledovat veškeré činnosti uživatelů v rezervačním systému. Nastavení sezón a sezónních cen je další funkce tohoto systému.

Rezervace

1
Termín

2
Pokoje

3
Zákazník

4
Souhrn

Termín: 6.5.2010 - 7.5.2010 (1 noc) CS ▾

Cena: 6600 Kč ? **Pokoje:** 2x [3+1], 1x [5+1]

Vyplněním tohoto formuláře odešlete poptávku přímo na recepci našeho hotelu.

Přjezd * Odjezd * doplnit čas příjezdu

-

Typ dostupných pokojů	Vyberte počet pokojů	Cena
1. Třílůžkový [3+1]	<input type="text" value="2"/> ▾	3000 Kč
2. Pětilůžkový apartament [5+1]	<input type="text" value="1"/> ▾	2500 Kč

Ceny jsou uvedeny za plně obsazená hlavní lůžka.

Pokračovat

Obrázek 5 – Ukázka rezervačního formuláře systému Previo

Potencionální host musí být s přehledností rezervačního systému také spokojen. Jde o krokovou rezervaci, kde nejprve rezervujete termín a vyberete pokoj. Pak nastavujete jednotlivé parametry v pokoji, což znamená věk osob (od toho se určuje celková cena), stravování (polopenze atd.) atp. Následuje krok zákazník. Zde si vlastně vytváříte účet, na který bude rezervace odkazována. Posledním krokem je krok „souhrn“. Zde si projdete celkovou rekapitulaci. Zkontrolujete zadané informace a vytvoříte rezervaci.

4.3.2 Nabízené varianty a služby

Vzhledem k dosavadním kvalitám nabízené služby není překvapením, že společnost má k dispozici i dvě varianty rezervačního systému. Jak bývá zvykem. Jedna je pro méně náročného uživatele. Druhá naopak pro náročnější klientelu. Jde o verze *Previo LITE* a *Previo PRO*. Pokud bych výhody kvalitnější varianty shrnul, tak jde navíc v podstatě o ucelený hotelový recepční systém. Následná tabulka demonstruje funkce obou variant.

	previo LITE rezervační systém	previo PRO rezervační systém
Správa rezervací a opcí	✓	✓
Databáze hostů a firem	✓	✓
Hotelový účet hosta	✓	✓
Napojení na PrevioPartnery	✓	✓
Evidenční kniha (kniha hostů)	✗	✓
Domovní kniha (cizinecká policie)	✗	✓
Reporty ČSÚ a MěÚ	✗	✓
Účetní doklady (fakturace)	✗	✓
Manažerské reporty a statistiky	✗	✓
Uživatelské účty a práva	✗	✓
Úklid pokojů	✗	✓
Napojení na jiné systémy (restaurační, účetní, kartové, atp.)	✗	✓
Administrace webových stránek	✓	✓

Obrázek 6 – Přehled funkcí variant systému Previo

Společnost Previo nezůstává jen u zřizování rezervačního systému. Svůj rezervační systém doplňuje o další služby. Můžeme si objednat zaškolení obchodním konzultantem, registraci do katalogů a vyhledávačů, popřípadě napojení na zámkový systém SALTO. Pokud nemáme vytvořenu webovou prezentaci, tak (samozřejmě za příplatek) je nám k dispozici spousta šablon. Pro náročnějšího uživatele je možnost vytvořit i celou webovou prezentaci na míru.

4.3.3 Shrnutí

Co říci závěrem? Při analýze tohoto systému jsem byl skoro až nadšen. Autoři velmi dobře prezentují svojí práci. Seznámení s produktem je tak o jednu až dvě třídy lepší, než tomu bylo u předchozích aplikací, kde pro zjištění některých důležitých informací musíte kontaktovat zákaznickou podporu a stejně se nemusíte dobrat k tomu, co jste vlastně chtěli. Na jejich stránkách je k nalezení dokonce demoverze pro vyzkoušení systému. Za své velmi kvalitní služby chtějí autoři i poměrně slušné peníze. Platíme pořizovací cenu plus měsíční poplatek. Menší podnikatelé by asi mohli zauvažovat o levějším řešení. Podle mého by se jim ale vyplatilo zainvestovat do varianty LITE, která není tak drahá a přitom vysoce výkonná a kvalitní.

Klady:

- Nepřeberné množství funkcí

- Přehledný dizajn
- Doplnkové služby
- Skvělá prezentace produktu
- Zkouška zdarma

Zápory:

- Vyšší cena

4.4 Celkové shrnutí

V předchozích kapitolách jsem představil tři rezervační systémy domácího trhu a teď je řada na celkové bilancování. Nejprve uvedu tabulku hodnocení důležitých předmětů jednotlivých rezervačních systémů. Známkovat budu jako ve škole. Stupnicí jedna až pět, kde jednička znamená nejlepší.

Tabulka 1 – Hodnocení rezervačních systémů

	Booking SystemBS	Czechproduct	Previo
Funkce a možnosti	2	5	1
Přehlednost a jednoduchost	2	3	1
Dizajn	1	3	1
Další služby	3	3	1
Prezentace produktu	3	4	1
Cenová dostupnost	2	1	3
CELKOVÉ HONOCENÍ	2,2	3,2	1,3

Z celkového hodnocení ve výše uvedené tabulce si už asi můžeme představit, který z těchto rezervačních systémů bude vítězem. Pár slov ale přeci ještě dodám. Pokud bych jako podnikatel nebyl zrovna ve finanční tísní, neváhal bych a šáhl po levnější verzi rezervačního systému Previo. Na druhou stranu můžu být začínající podnikatel, který je ve fázi testování a zkoušení svého podniku. Pak si asi nemá cenu dělat vrásky s drahými rezervačními systémy a objednat si produkt u Czechproduct. Pokud se mi ale podaří na trh více prorazit, určitě zapřemýšlím o co nejbližší výměně. Zdálo by se, že pak Booking SystemBS zůstane neatraktivní pro jakoukoliv klientelu. Není tomu tak. Jeho sekce pro zprostředkovatele se někomu může zdát jako velmi dobře použitelná výhoda. V kombinaci s levnějším tarifem pak klient raději zainventuje do tohoto rezervačního systému. Já, ač být velký či malý podnikatel, bych určitě šel cestou rezervačního systému Previo.

5 Zabezpečení aplikace

Již v úvodu práce jsem se zmínil o tom, že někteří lidé jsou stále skeptičtí, co se týče bezpečnosti jakýchkoliv internetových služeb. Bojí se, aby nepřišli o své peníze, nebo aby někdo nezneužil jejich citlivých informací. Mají proč?

Zabezpečení webové aplikace, jako je rezervační systém, je nezanedbatelným procesem, kterému musí být věnováno co nejvíc prostoru. Poskytovatel si nemůže dovolit být nevěrohodný. Pokud by se pak setkal s podobným problémem, mohl by ztratit důvěru u klientů. Výsledný ušlý zisk by chtěl vysoudit na prodejci produktu.

5.1 Zabezpečení PHP

Skriptovací jazyk PHP je k útokům na citlivá data velmi náchylný. Popíši zde, na co útočníci své útoky směřují, jaké druhy útoků používají a jak se proti nim můžeme bránit.

5.1.1 Session ID (SID)

Session ID je jedinečný identifikátor, který je přidělen každému návštěvníkovi stránky. Jedinečný proto, že dva návštěvníci nemůžou mít stejné ID. S tímto ID je pak spojena globální proměnná `$_SESSION`. Tento systém je pak jednou z cest, jak kontrolovat pohyb uživatele na webových stránkách. Proto manipulace se session ID bývá základem většiny útoků. Prohlížeč si toto ID musí pamatovat. Dělá to pomocí systému cookies, nebo využívá URL, což je ještě nebezpečnější než předchozí zmiňovaný systém. [18], [19]

Se session ID je spojeno několik útoků. Útočník třeba může získat cizí ID a použít ho pro svoje potřeby. Tento útok se pak nazývá *Session Hijacking*. Zabránit tomu můžeme pomocí protokolu HTTPS². Jako doplňkovou ochranu využijeme kontrolu IP adresy.

Session Fixation je jeden z dalších druhů útoků, kdy nastaví Session ID útočník klientovi a jakmile se přihlásí uživatel, tak tuto hodnotu použije pro sebe. Dochází pak k přesvědčování uživatele třeba klamavými reklamami. Ochranou buď funkce `session_regenerate_id`, které vygeneruje nové session ID a útočník pak původní nebude moci využít. Funkci je nejlepší používat při provádění kritických operací.

5.1.2 XSS

Pojem XSS označuje termín *Cross Site Scripting*, což je druh útoku, jež vkládá nebezpečný skript přímo do obsahu webové prezentace. Nebezpečný kód je psán většinou v klientském skriptovacím jazyce, jako je například Javascript³. Existují tři typy těchto útoků.

² HTTPS – „Je nadstavba síťového protokolu HTTP, která umožňuje zabezpečit spojení mezi webovým prohlížečem a webovým serverem před odposloucháním.“[20]

³ Javascript - je multiplatformní, objektově orientovaný skriptovací jazyk používaný jako interpretovaný programovací jazyk pro webové stránky

První zmíním *útok okamžitý*, který okamžitě vykonává nebezpečný kód. Útočník se pak snaží přesvědčit uživatele, aby klikl na odkaz s nebezpečným kódem. Pokud toto uživatel udělá, může pak skript dostat citlivé údaje, uložené v prohlížeči uživatele.

Dalším útokem je *perzistentní útok*. Zabezpečená aplikace, která nekontroluje vkládaná data běžného uživatele, může mít pak problém v podobě navíc vloženého nebezpečného skriptu útočníkem. Jedná se především o komentáře a diskusní fóra.

Posledním, mnou zmíněným, útokem bude *lokální útok*. Ten je velmi podobný útoku okamžitým s rozdílem, že ke zpracování nebezpečného kódu se využije existujícího klientského skriptu. Lokální skripty mají přístup k souborům na lokálních discích. To pak ohrožuje lokální webové aplikace.

Hlavní zabezpečovací stránkou je nepropouštění jakékoliv skriptovací konstrukce do veřejné části webu, aby je HTML jazyk nezpracovával a tím neměnil třeba vzhled dokumentu pomocí kaskádových stylů. Nejsnazším řešením je pak překódovat HTML ohraničovací značky `<` a `>` na entity. Entita je, pro představu, kódový zápis nějakého znaku. Pro výše zmíněné znaky existuje kódový zápis `<` a `>`. Pomocí těchto entit se nám zobrazí přímo vepsaný zdrojový kód, aniž by se prohlížeč snažil o jeho přeložení.

K takovému ošetření využijeme v PHP funkci `htmlspecialchars()`. Ta převádí právě výše zmíněné speciální znaky a HTML značky na entity. Funkce sice trochu zvýší velikost přenášeného textu, ale přináší nám absolutní bezpečnost.

Problém nastává, pokud potřebujeme do textu vkládat alespoň nějaké HTML značky, jež chceme prohlížečem přeložit (může jít o odkazy či obrázky). V této situaci můžeme využít například systému BBCode⁴. [21] Jedná se o značkovací jazyk, pomocí kterého formátujeme příspěvky například na online diskuzích. Značky uzavíráme do hranatých závorek, které jsou, při odeslání, přetransformovány do HTML kódu. Znázorním příklad takového kódu:

```
[url=http://www.google.cz/]Google[/url]
```

Tento kód se nám přetransformuje do, pro nás známého, HTML kódu:

```
<a href="http://www.google.cz/">Google</a>
```

Jako autor webové aplikace zodpovídáte i za její bezpečnost. Pokud pracujete i s citlivými údaji, tak je bezpečnost aplikace o to více důležitá, a proto se musíte držet alespoň základních bezpečnostních pravidel.

5.2 Zabezpečení databáze

U rezervačních systémů je samozřejmostí i databáze s důležitými daty. Někam prostě musíte ukládat rezervace, informace o klientech a uživateli, systém lůžek a další data,

⁴ BBCode - Bulletin Board Code, což znamená „kód nástěnek“

bez kterých se neobjedete. S tímto ale souvisí i riziko, že se někdo do databáze dostane, upraví její obsah, nebo se jí kompletně zmocní. Jak tomu předcházet?

5.2.1 SQL injection

Pokud potřebujeme z databáze získat nějaká data, tak v PHP aplikaci formulujeme SQL dotazy spojené s vstupními daty a směřované do databáze. Kde se tomu tak stává, je pak cílem případného útočníka, který se následně snaží o zmanipulování vstupních dat do podoby nějakého SQL dotazu. Tím SQL dotaz nabude formy, která z databáze dostane informace citlivé, jež se stávají ulovenou kořistí pro útočníka. Souhrnně tyto útoky nazýváme *SQL injection*. Možnost na výše zmíněný útok dát, jako programátor webové aplikace, převážně dvěma prohrašky. Jeden z nich je zanedbané filtrování escape sekvencí, druhý pak tkví ve špatné práci s datovými typy v databázi. [22], [23]

Nejjednodušší představení problému, kdy vkládáme do SQL dotazu nějaké vstupní parametry, si představíme na následujícím příkladu. Máme k dispozici přihlašovací formulář, který zasílá metodou POST přihlašovací údaje, které se zpracují v souboru *actionLog.php*:

```
<form method="post" action="actionLog.php">
  <input type='text' name='id'>
  <input type='password' name='heslo'>
  <input type='submit' name='prihlaSub' value='Přihlásit'>
</form>
```

Aplikace disponuje souborem *functions.php*, ve kterém máme autentifikační funkci, jež kontroluje přihlášení. V případě úspěchu vrací hodnotu *true*:

```
function autentifikace($id, $heslo)
{
  $sql = "SELECT id_user FROM users WHERE id_user = "
        . $id . " AND password = '" . $heslo . "'";

  $res = mysql_query($sql);
  while($row = mysql_fetch_object($res)) {
    $r = ($row->id_hosta);
  }
  if($r == $id) {
    $bool = true;
  } else {
    $bool = false;
  }
  return $bool;
}
```

Následná ukázka demonstruje, jak bude vypadat SQL dotaz při zadání uživatelského id a hesla:

Id: 23

```
Heslo: mojeHeslo
SELECT id_user FROM users WHERE id_user = 23 AND password = 'mojeHeslo';
```

V tomto případě je vše v naprostém pořádku. Co když ale trochu upravíme přihlašovací údaje:

```
Id: 23
Heslo: 'OR 1=1 --
SELECT id_user FROM users WHERE id_user = 23 AND password = ''OR 1=1 --';
```

Takovýto SQL dotaz momentálně vrací id s hodnotou 23, aniž bychom potřebovali znát heslo. Podmínka, kde kontrolujeme heslo, byla obejita podmínkou, kterou jen ověřujeme, zda se jedna rovná jedné. Výsledná funkce tedy vrací hodnotu true. Při zadání různých jiných SQL dotazů do přihlašovacích údajů, či jinam, poté můžeme přijít i o celé tabulky.

Ochrana před vkládáním útočných SQL dotazů do formulářů je velmi jednoduchá. Využijeme zde PHP funkce *mysql_real_escape_string*, která upravuje řetězec na bezpečný pro použití v SQL dotazu. Výsledná funkce bude vypadat takto:

```
function autentifikace($id, $heslo)
{
    $sql = "SELECT id_user FROM users WHERE id_user "
        . mysql_real_escape_string($id) . " AND password = '"
        . mysql_real_escape_string($heslo) . "'";
    ...
}
```

5.2.2 Citlivé údaje v databázi

Přesto, že máme databázi zabezpečenou od SQL injection, stále bychom neměli zapomínat na citlivé údaje v databázi. Třeba taková data o hesle je jeden z údajů, který by nemusel být ani pro oči administrátora, a tak je vhodné tyto data zabezpečit proti přímému čtení.

V aplikaci tedy můžeme zabezpečit ukládání hesel zakódovanou formou, místo původně formy textové. Databáze tak vidí jen formu zakódovanou. Zde se jedná o tzv. jednocestné zabezpečení, jež je většinou realizováno pomocí *hashovacích algoritmů*. Díky nim ze vstupního řetězce libovolné délky vytvoříme řetězec délky fixní. Výstupu algoritmu se říká *hash*, neboli *otisk*. Hashovací algoritmy by měli podléhat určitým pravidlům. [22]

Odolnost proti kolizi je požadavek, který zaručuje, že dva různé řetězce nebudou mít totožný otisk. Těžko bude takový požadavek platit vždy, protože převádíme na řetězec omezené délky řetězec délky libovolné.

Lavinový efekt nám popisuje vlastnost, kdy malá změna na vstupním údaji vytvoří změnu velkou u otisku.

Zahashované heslo útočník jen velmi těžko odstraní. Pro odhalení kódu mu nezbude nic jiného než generovat všemožná hesla a následně je porovnávat s konkrétním otiskem. Zde se naskytne problém, pokud je heslo vytvořeno kombinací jen několika písmen. Další bezpečnost svého hesla si už určuje uživatel sám. Tzv. síla hesla se posuzuje podle množství a druhu kombinovaných znaků. Samozřejmě platí pravidlo: „čím víc, tím líp“. [24]

Další fakt, zjednodušující útočnickovi práci, je nebezpečí stejného hesla u více uživatelů. Obyčejný hash nám tento problém nevyřeší. Pomůže nám s tím ale jistý způsob nazývaný „*solení*“. Při kódování hesla se na vstupu přidá navíc nějaký řetězec, neboli *salt(sůl)*. Ta je pokaždé jiná a otisk dvou hesel již nemůže být stejný. Neplatí pro ni nějaká striktní pravidla. Sůl bývá veřejná, ale musí se lišit mezi jednotlivými uživateli. Sůl většinou generujeme náhodně. Musí být někde uložena. Tím místem bývá sloupec vedle přihlašovacích údajů. Pokud chceme sůl schovat přímo do výsledného otisku, využijeme PHP funkci *crypt()*, která vytváří sůl v rámci výsledného hashe. Uvedu příklad použití takové funkce:

```
<?php
    $heslo = 'mojeheslo';
    $otisk = crypt($heslo);
?>
```

Generování náhodné soli přináší složitější autentizaci, jelikož sůl se musí posílat uživateli společně s výzvou. Některé servery proto využívají přihlašovacího údaje jako soli. Takový údaj je potom unikátní (nemůže existovat více uživatelů se stejným přihlašovacím jménem), není potřeba vytvářet nový sloupec a je také vždy k dispozici.

Pro kódování hesel využíváme především algoritmů SHA (Secure Hash Algorithm). SHA vytvořila organizace NSA⁵. Je tvořena jako rodina pěti algoritmů (SHA-1, SHA-224, SHA-256, SHA-384 a SHA-512), z níž poslední čtyři uvedené se společně uvádějí jako SHA-2. Nejvíce se ale používá kódování SHA-1, jež vytváří otisk dlouhý 160 bitů (čísla u ostatních algoritmů z rodiny označují délku otisku v bitech). [25]

Pokud tedy využijeme zakódování pomocí algoritmu SHA-1, doplníme ho o výše zmíněnou sůl, je pak pro útočníka téměř nemožné se v rozumném čase dobrat nějakého výsledku a to je také námi zamýšleným záměrem.

5.3 Intranet

Do teď jsem se zabýval spíše obecně zabezpečením PHP aplikací a jejich databází. Vráťím-li se ke konkrétnímu tématu, čímž jsou rezervační systémy, můžeme se podívat na zabezpečení i z trochu jiné stránky.

Jedná se o online rezervace, tudíž je samozřejmostí, aby uživatel měl přístup do systému odkudkoliv. Tedy pomocí celosvětové sítě Internet. Administrátorskou část ale

⁵ NSA – Národní bezpečnostní agentura v USA

můžeme velmi kvalitně ochránit, před napadením zvenčí, zaimplementováním intranetu, což je počítačová síť využívající stejné technologie jako Internet s tím rozdílem, že se jedná o síť soukromou. Je tedy určena pro malé podniky atp. Tím velmi padá riziko, že by se do systému mohl připojit někdo nepovolaný.

Intranet je sice nezávislý na Internetu, ale pro potřeby rezervačních systémů budeme Internet přesto požadovat. K vytvoření intranetové sítě, tedy krom Internetu, bude ještě zapotřebí instalovat na hlavní (třeba i jediný) počítač server. Základem bude server webový. S jeho pomocí využijeme prohlížeč na provoz vnitřních intranetových stránek.

Počítač s webovým serverem již máme. Webovou prezentaci společně s rezervačním systémem jsme již také rozhodili. Pokud máme v podniku více počítačů, na kterých chceme mít funkční systém, je potřeba vytvořit síť, například pomocí switchů⁶.

Vzhledem k tomu, že chceme, aby náš rezervační systém byl online, tak ho také musíme zveřejnit i na Internetu, pro případné hosty. Připojení z Internetu, do naší intranetové sítě je ale trochu náročnější úkol, který většinu potencionálních hostů, od využití online rezervace, odpudí. Nejlepším způsobem by potom bylo vložení systému pro zákazníky na nějaký internetový webový server, společně s webovou prezentací. Součástí by byla i databáze, kam by se ukládali jednotlivé rezervace. Obsahovala by i pokoje a některé nezbytné informace o uživateli.

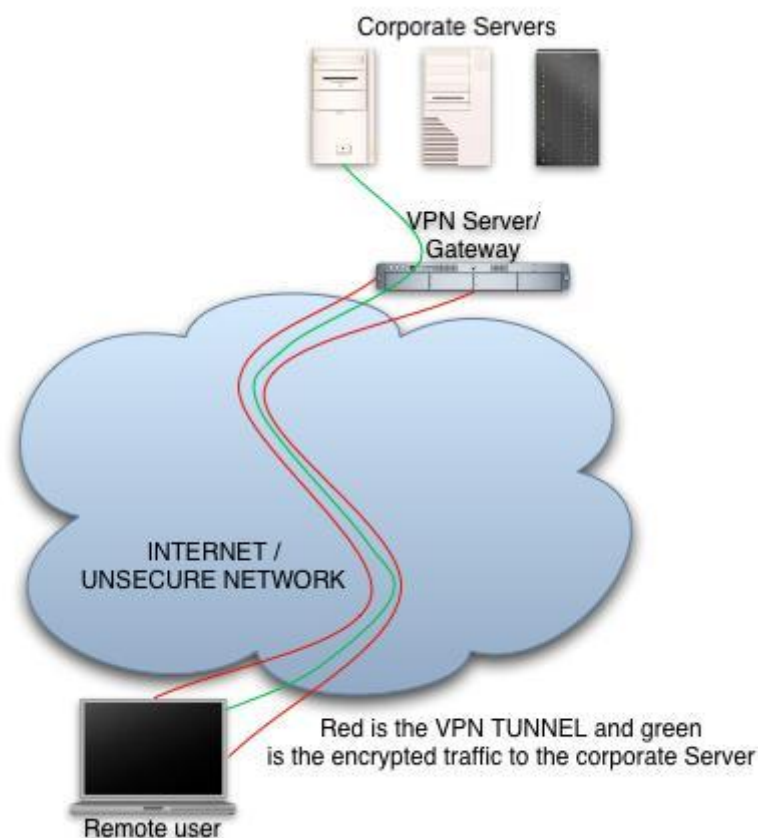
Část administrátorská by již zůstala jen ve vytvořené intranetové síti. Disponovala by funkcí, která by z naší vytvořené internetové databáze následně stahovala potřebné informace do databáze interní. Spousta podnikatelů může vyžadovat připojení do administrátorské části (tedy do intranetu) odkudkoliv. Co potom?

5.3.1 VPN

Pod v nadpisu uvedenou zkratkou VPN se skrývá výraz virtual private network (virtuální privátní síť). VPN je prostředek využívaný pro propojení počítačů ve veřejné (nedůvěryhodné) síti. S jeho pomocí vytváříme spojení, které vypadá jako spojení v interní, a tedy důvěryhodné, síti. Totožnost klientů sítě je ověřována pomocí digitálních certifikátů. Jakákoliv komunikace je šifrovaná a my pak můžeme označit síť za bezpečnou. Obr. 7 demonstruje připojení pomocí VPN.[26]

Takový prostředek využijeme, pokud se chceme dostat z Internetu do naší intranetové sítě. Musíme v podnikové síti zprovoznit VPN server, který následně plní funkci VPN brány, jež zprostředkovává připojení, stará se o šifrování a celkovou bezpečnost komunikace. Jde vlastně o jistou formu síťového tunelování. Tunel nám vytváří právě prostředek VPN.

⁶ Switch – neboli přepínač, je aktivní síťový prvek používaný k propojení segmentů sítě



Obrázek 7 – připojení pomocí VPN

zdroj: http://owebu.blogger.cz/obrazky/owebu.blogger.cz/obrazky/vpn_uvod.jpg

5.4 HTTPS

Pokud někomu přeci jen přijde infrastruktura intranetu s použitím VPN zbytečně moc náročná, je tu další možnost zabezpečení v podobě využití nadstavby síťového protokolu HTTP. HTTPS nám umožňuje zabezpečené a šifrované spojení mezi webovým serverem a prohlížečem.

K šifrování využívá protokoly SSL⁷ a TLS⁸. Princip spočívá v asymetrickém šifrování. Obě strany si tedy před komunikací vygenerují privátní a veřejný klíč. Početím komunikace dojde k výměně klíčů, jež pak každá strana ověří, nejlépe za použití jiného komunikačního kanálu. Využit můžeme i digitálně certifikovaný klíč. Šifrování předejde odposloucháváním komunikace. Bez ověření veřejných klíčů ale existuje riziko útoku „Man in the middle“⁹. [29]

⁷ SSL – „Je protokol, resp. vrstva vložená mezi vrstvu transportní (např. TCP/IP) a aplikační (např. HTTP), která poskytuje zabezpečení komunikace šifrováním a autentizací komunikujících stran.“ [27]

⁸ TLS – je protokol založen na stejném smyslu, jako SSL. Rozdíly mezi nimi jsou nepatrné.

⁹ Man in the middle – je snaha útočníka k odposlouchávání komunikace tím, že se stane prostředníkem. [28]

Protokol HTTPS bych použil hlavně pro část administrativní. Jako zabezpečené spojení je určitě dostačující a přitom jednodušší než zřízení intranetu a následné zprovoznění VPN.

6 Rezervační systém pro ubytovací zařízení

Praktickou částí bakalářské práce bylo vytvořit rezervační systém pro ubytovací zařízení. Aplikace měla být vytvořena pomocí jazyka PHP. Data budou zase uložena na databázi MySQL. Inspirací mi byly přední rezervační systémy, ač jejich kvalitám nemůže má aplikace konkurovat. Jsou to totiž programy vyvíjené již několik let a to zkušeným týmem programátorů. K dispozici mají také diametrálně odlišné prostředky než já. Mou aplikaci v následujícím textu představím.

6.1 Použité technologie

Pro vytvoření výše zmíněného úkolu jsem potřeboval nějakou webovou prezentaci, kde bych mohl vytvoření rezervační systém demonstrovat. Na vytvoření webové stránky jsem využil značkovacího jazyka HTML, konkrétně jeho verze HTML 4.01. Vzhled HTML dokumentu řeším pomocí kaskádových stylů (CSS).

Na tvorbu aplikace jsem ani nemohl využít ničeho jiného než skriptovacího programovacího jazyka PHP 5.3.0. Pro přeložení jazyka potřebuji ale nějaký server. Na výběr jsem měl možnost stránky nahrát na nějaký webhosting, ale z důvodu rychlosti a jednoduchosti, jsem si vytvořil server na svém počítači. K tomu jsem použil webový server Apache, verzi 2. 2. 13.

Databází, jež mi byla dána už zadáním, použitou v aplikaci je MySQL. Vzhledem k předchozím zkušenostem bych pravděpodobně při možnosti volby použil databázi Oracle. Databáze MySQL byla pro mě novou záležitostí, se kterou jsem se ještě nesetkal. Ve finále, i přes počáteční obavy z neznáme vody, jsem za tuto volbu rád. Pokud není potřeba prvotřídního databázového systému, tak databáze MySQL je více než dostačující. Práce s ní je mírně jednodušší a není problém si přivyknout. Pro správu mé databáze (konkrétně tedy MySQL 5. 1. 37) bylo využito prostředku s názvem phpmyadmin 3.2.1. Jedná se o přehledné grafické prostředí, vytvořené pro správu databáze.

Webový server Apache s podporou jazyka PHP. Instalaci databáze MySQL společně s aplikací phpmyadmin. To vše jsem nechal na programu s názvem EasyPHP. Konkrétně jde o verzi 5.3.0. Při instalaci tohoto programu se mi nakonfiguruje webový server. Nainstaluje databáze a uživatel má pak o starost méně.

6.2 HTML stránka

V předchozí kapitole jsem vysvětlil, jaké technologie byly v aplikaci použity. Nyní se zaměřím na konkrétní zpracování jednotlivých částí práce.

Jak jsem již výše zmínil. Rezervační systém bylo nutné někde demonstrovat. Pro tuto potřebu jsem si vytvořil HTML stránku s využitím kaskádových stylů. Jedná se o klasický HTML dokument rozdělený pomocí divů¹⁰ na dvě základní skupiny. Jednou je

¹⁰ Div – neutrální tag, který v kombinaci s kaskádovými styly obaluje části stránky a dává jí nějaký vzhled.

třída *logo*, která vytváří hlavičku stránky. Druhou třídou je třída *main*, která obsahuje ještě další dvě důležité podtřídy *leftPanel* a *right*. Třída *leftPanel* definuje levý panel, kde je ovládání webu. Třída *right* pak vytváří hlavní stránku webové prezentace, která funguje jako měnící se displej. Uvedu kódový zápis podtřídy *leftPanel*:

```
.leftPanel{
  height: 1600px;
  width: 200px;
  border-style: solid;
  border-width: 2px;
  background-color: CadetBlue;
  float: left;
}
```

Se znakovou sadou jsem měl z počátku problémy. Inspirován knížkou jsem využil sadu latin2 (iso-8859-2), která mi ale dělala problémy v ukládání a načítání dat z databáze. A to i přes to, že databáze je samozřejmě vytvářena ve stejné znakové sadě. Po změně na sadu UTF-8 již bylo vše v naprostém pořádku.

HTML prezentace, jak už jsem zmínil, figuruje jen jako prvek pro demonstraci vlastního rezervačního systému, takže taková důležitost mu vkládána nebyla. Jinak tomu bylo už jinde.

6.3 PHP aplikace

Hlavní náplní mé práce bylo právě navržení PHP aplikace. Chtěl jsem docílit několika funkcím. Samozřejmostí byla registrace hostů a následné přihlášení. Pokud zůstaneme u potencionálních hostů, tak jim měla být umožněna rezervace jak z přihlášeného účtu, tak bez přihlášení s tím rozdílem, že při rezervaci anonymním uživatelem došlo k následnému doplnění o jeho údaje. Po přihlášení měl mít zákazník k dispozici přehled o svém účtu s případnou změnou údajů, či hesla a přehled o svých rezervacích.

Na administrátorskou část byl kladen důraz větší. Je také samozřejmé, že administrátor disponuje řadou funkcí navíc. Uživatele jsem ještě rozdělil na administrátora, který je jediný. A správce chaty, jehož právě výše zmíněný administrátor může povýšit z obyčejného uživatele. Administrátor by měl mít od správce chaty na víc ještě další funkce. Tím je udělování bezpečných IP adres, pro přihlášení důležitých uživatelů (administrátor / správci). O této části si více povíme v zabezpečení.

Správce chaty tedy může být víc. Mojí vizí bylo, aby se správce ihned po přihlášení dozvěděl o důležitých událostech v systému. Tyto události by pak mohli označit jako přečtené a následně je mazat. Také jsem chtěl dospět možnosti spravování pokojů. Zde správce mohl přidávat, upravovat a mazat pokoje své chaty. Také by se vyskytovala možnost rozšíření vybavení a následné přidání vybavení k určitým pokojům. Chtěl jsem, aby správce měl k dispozici i nějaký přehled hostů s přehledem jejich objednávek, jež by

zařizoval modul kalendář. Mou představou byl pohled na měsíční obsazenost jednotlivých pokojů s následným posouváním. Jak jsem se s důležitými kroky popral, popíši níže.

6.3.1 Rezervace

Registrace a přihlášení jsou problémy minimální. Přejdu tedy k hlavní funkci potencionálního hosta, čímž není nic jiného, než rezervace ubytování.

Hlavní složkou této problematiky, bylo zkoumání volných pokojů. Musel jsem dát dohromady funkci, která mi na základě vložených informací (datum příjezdu a odjezdu), vrátí volné pokoje v daném termínu. Věřím, že možností bylo více, ale výsledná funkce, včetně SQL dotazu vypadá následovně:

```
function vratVolnyPokoj($od, $do)
{
    $sql = "SELECT pokoje.id_pokoje, pokoje.pocet_luzek, pokoje.cena FROM
pokoje
WHERE pokoje.id_pokoje NOT IN
(
    SELECT pokoje.id_pokoje
FROM pokoje, pokoj_objednavky, objednavky
WHERE pokoje.id_pokoje = pokoj_objednavky.id_pokoje
AND pokoj_objednavky.id_objednavky = objednavky.id_objednavky AND
((objednavky.prijezd BETWEEN '".mysql_real_escape_string($od)."' AND
'".mysql_real_escape_string($do)."'
OR (objednavky.odjezd BETWEEN '".mysql_real_escape_string($od)."' AND
'".mysql_real_escape_string($do)."'
OR ('".mysql_real_escape_string($od)."' BETWEEN objednavky.prijezd AND
objednavky.odjezd)
)");
    return $sql;
}
```

Nejdůležitějším aspektem funkce byl samozřejmě SQL dotaz, který je vytvořen takto. Hledá id pokoje, které není ve vnořeném selectu, jež vyhledává pokoje v daném termínu obsazené. Tento vnořený select nejprve zkoumá, jestli je daný pokoj v nějaké objednávce. Následně řeší, jestli datum příjezdu, zaznamenané v rezervaci, se nevyskytuje v intervalu zadaném uživatelem. Poté zkoumá to samé, jen s odjezdem zaznamenaným v rezervaci a v poslední řadě zkoumá, zda zadané datum příjezdu od uživatele se nevyskytuje v intervalu příjezdů a odjezdů uložených v rezervacích. Nepopírám, že řešení se dost možná mohl najít i jednodušší. K funkci řešení tohoto ale výtku nemám, tak jsem jej také použil.

Aplikace následně vypíše volné pokoje na obrazovku, uživatel provede výběr pokojů pro rezervaci a proběhne vlastní rezervace.

6.3.2 Události a kalendář

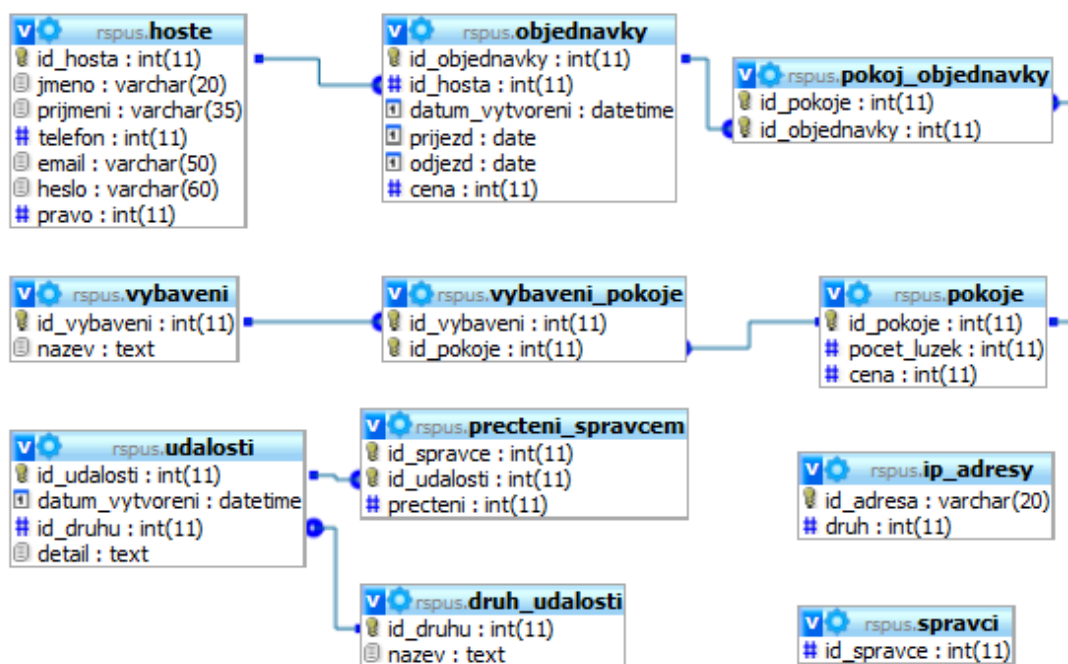
V části administrativní v aplikaci byl nejdůležitější modul kalendář a události. Události jsem řešil tak, že nějaká důležitá činnost klienta či správce bude zaznamenaná do

databáze. Při přihlášení správce nebo administrátora pak prvním, co budou mít na obrazovce, jsou nepřečtené události. Tak by měli mít oba dění v rezervačním systému pod kontrolou. Samozřejmostí také je, že pokud jeden označí událost jako přečtenou a následně jí smaže, tak se mu tak nepodaří, jelikož jí nepřečetl ještě jiný správce. Funkce pro vložení události je vložena v příloze A. Postupuji tak, že nejprve vytvořím přílohu v databázi. Vzhledem k tomu, že id události se mi v databázi vytváří samo, tak následně žádám databázi o vrácení id události nyní vytvořené. To potřebuji pro nastavení přečtení u jednotlivých správců.

Při zobrazení kalendáře se otevře nové okno s aktuálním měsícem. Vypsány zde jsou jednotlivé dny s jednotlivými pokoji a znázorněná je také jeho obsazenost, popřípadě kým je pokoj rezervován.

6.4 Databáze MySQL

S databází nebyla příliš práce. Nejdůležitějším problémem k řešení bylo vymyslet nějakou strukturu celé databáze. Navrhnout jednotlivé tabulky a vymyslet správné relace mezi nimi. Tento problém nejlépe demonstruje pod textem vložený ER diagram (Entity-relationship diagram).



Obrázek 8 – ER diagram

Zde máme přehled jednotlivých tabulek, včetně jejich atributů a relací mezi nimi. Když ve zkratce jednotlivé relace popíši. Hosté jsou propojeni s objednávkami (rezervacemi). Následná rezervace pak může obsahovat více pokojů a současně pokoje můžou být rezervovány ve více objednávkách. Liší se akorát termínem. Je proto potřeba zavést mezi tabulku pokoj_objednavky, pro propojení objednávek a pokojů. Jednotlivé

pokoje disponují nějakým vybavením. Druhu vybavení můžou mít pokoje víc a současně vybavení bude obsaženo ve více pokojích. Je zde nutné znovu zavést spojovací tabulku `vybavení_pokoje`. Další tabulky pak souvisejí s událostmi a jejich přečtení správci. Tabulka `správci` je jen vytvořený pohled z tabulky `hosté`. Databáze obsahuje ještě tabulku `ip_adresy`, což je tabulka bezpečných adres pro připojení administrátorů a správců.

6.5 Zabezpečení

Kapitole o zabezpečení webové aplikace jsem se již věnoval. Podíváme se ale, jak jsem vyřešil zabezpečení v aplikaci vytvořené mnou.

Na vytvoření intranetu, následnému zabezpečení s pomocí VPN nemám prostředky. Snažil jsem se ale administrátorské připojení zabezpečit alespoň pomocí kontroly IP adresy. Pokud se tedy připojuje někdo do části pro správce či administrátora, tak je mu kontrolováno z jaké IP adresy se připojuje. V databázi jsou uloženy bezpečné adresy, se kterými se pak porovnává adresa připojovaného jedince. Výsledkem je pak nemožnost se připojit z nebezpečné adresy. Tyto adresy samozřejmě může administrátor spravovat a přidávat další.

Základní zabezpečení aplikace a session ID aplikuji pomocí funkce `session_regenerate_id`. Ochranu proti *Cross Site Scripting*, ve směs nemám kde využít. Vzhledem k tomu, že zákazník nemá možnost vkládání příspěvků a komentářů (ani by neměl v rezervačním systému nějaké využití), tak zde ani ochranu proti XSS v podstatě nevyužívám.

Jinak to je v přístupu do databáze. Vzhledem k tomu, že celá aplikace je na databázi závislá a přístupů do ní je velké množství, největší práci na zabezpečení jsem musel směřovat právě sem. Proti SQL injection bylo nejjednodušší ochranou použití funkce `mysql_real_escape_string`. Tato funkce zabráňuje vkládání nebezpečných výrazů do SQL dotazů. Hesla jsou šifrována pomocí algoritmu SHA-1. Tímto předcházím nebezpečí přímého čtení.

6.6 Srovnání

Těžko říct, jestli mohu srovnávat svou aplikaci s projekty, které prošly několikaletým vývojem, pracuje na nich tým odborníků a mají mnohem lepší zázemí a možnosti. Přesto bych alespoň s jedním z nich obstál.

Z pohledu klienta v rezervačních systémech až takový rozdíl není. Stejně jako u ostatních aplikací je i v té mé možnost výběru příjezdu a odjezdu s následným zarezervováním vybraného pokoje. Stránka administrátora by ale mohla být propracovanější. Modul kalendář mají vyspělé rezervační systémy na vysoké úrovni, které se zatím nemohu moc srovnávat. Přesto se, díky přehledu událostí, hostů a objednávek, či znázornění zaplněnosti lůžek v kalendáři, kvalitami alespoň dotýkám předních rezervačních systémů.

Pokud si ale vezmeme na parádu rezervační systém společnosti Czechproduct, tak má aplikace ho vysoce převyšuje (polehčujícím faktem mu může být, že systém není hlavním produktem společnosti Czechproduct). Absence automatické rezervace ze strany zákazníka je asi největším problémem. Administrátor jí musí zpracovat, sám vložit do databáze a následně o průběhu informovat uživatele. Má aplikace toto všechno obstará samostatně. Včetně informování administrátora. Administrátor, či správce chaty má k dispozici i možnosti na nastavení pokojů a jejich vybavení. Může si pak po nainstalování aplikace vytvořit vlastní chatu a rovnou rozběhnout jeho funkci. Jako jednoduchý nástroj pro nenáročného uživatele by byl mnou vytvořený rezervační systém určitě dostačující.

7 Závěr

Cílem praktické části práce bylo vytvořit rezervační systém pro ubytovací zařízení, který se mi podařilo splnit včetně vloženého zabezpečení aplikace. Budoucnost projektu tkví v rozšíření služeb pro klienty. Rezervační systém by se mohl zakomponovat do šablon a prodávat by se pak již mohla celá webová prezentace. Práce na vlastním systému se zase může směřovat na širší klientelu (zobecnění systému). Ne jen správci ubytoven, ale třeba i kina a divadla by jej mohla využít pro následnou rezervaci vstupenek. Pokud se vrátím k realizaci zabezpečení, tak jeho vývoj nikdy neustane. S příchodem nových nápadů útočníků na prolomení obrany, přijdou i další ochranné záplaty a pomůcky. Těžko říct, jestli budeme moct někdy o nějaké ochraně tvrdit, že je stoprocentní. Vždy se zatím našel někdo, kdo tento argument vyvrátil. Nejzajímavějšími řešeními jsou potom intranetové sítě, upřednostňovány kvůli rychlosti a právě bezpečnému propojení klientů.

Vzhledem k počtu rezervačních systémů na trhu, můžeme předpokládat hojné využití těchto aplikací podnikateli. Služba přitom nepatří mezi nejlevnější. Od požadavků se také velmi liší i její cena. Někteří prodejci si účtují i měsíční paušál včetně počátečních nákladů. Budoucnost tedy tyto rezervační systémy mají určitě. Nabízené služby uspokojí téměř každého zákazníka, a proto další zdokonalení aplikací tkví spíše v nových, levnějších a rychlejších technologiích.

Literatura

- [1] Internetový robot In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 14. 4. 2006, 1. 5. 2010 [cit. 2010-05-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Internetov%C3%BD_robot>.
- [2] HyperText Markup Language In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 16. 7. 2004, 30. 4. 2010 [cit. 2010-05-03]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/HyperText_Markup_Language>.
- [3] RAGGETT, Dave; LE HORS, Arnaud; JACOBS, Ian. *W3.org* [online]. 1999 [cit. 2010-05-03]. HTML 4.01 Specification. Dostupné z WWW: <<http://www.w3.org/TR/html401/>>.
- [4] Xhtml In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3. 1. 2010, 3. 1. 2010 [cit. 2010-05-03]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Xhtml>>.
- [5] Cascading Style Sheets In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2010, 27. 2. 2010 [cit. 2010-05-04]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Cascading_Style_Sheets>.
- [6] PHP In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 6. 2004, 4. 5. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/PHP>>.
- [7] KOFLER, Michael; ÖGGL, Bernd. *PHP 5 a MySQL 5*. Brno : Computer Press, 2007. 607 s.
- [8] Databáze In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 24. 6. 2004, 22. 4. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Datab%C3%A1ze>>.
- [9] MySQL In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 21. 11. 2004, 7. 4. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/MySQL>>.
- [10] Oracle In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 7. 1. 2005, 28. 4. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Oracle>>.
- [11] Hypertext Transfer Protocol In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 15. 6. 2007, 19. 4. 2010 [cit. 2010-05-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Hypertext_Transfer_Protocol>.

- [12] Apache HTTP Server In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19. 11. 2004, 7. 3. 2010 [cit. 2010-05-05]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Apache_HTTP_Server>.
- [13] The Apache Software Foundation. *Apache.org* [online]. 2009 [cit. 2010-05-05]. Apache HTTP server project. Dostupné z WWW: <<http://httpd.apache.org/>>.
- [14] R-INCON Int., s.r.o. *R-incon.cz* [online]. 2000, 2009 [cit. 2010-05-05]. Booking-System BS. Dostupné z WWW: <<http://www.r-incon.cz/cs/projekty/podpora-cestovniho-ruchu/booking-system-bs.html>>.
- [15] Booking SystemBS. *B-s.cz* [online]. 2007 [cit. 2010-05-06]. Booking System. Dostupné z WWW: <<http://www.b-s.cz/>>.
- [16] Czechproduct.cz s.r.o. . *Czechproduct.cz* [online]. 2010 [cit. 2010-05-06]. Rezervační systém. Dostupné z WWW: <<http://www.czechproduct.cz/44723/rezervacni-system/>>.
- [17] MITON Previo s.r.o. . *Previo.cz* [online]. 2002, 2010 [cit. 2010-05-06]. Hotelový rezervační a recepční systém Previo. Dostupné z WWW: <<http://www.previo.cz/>>.
- [18] MALÝ, J. ; KACÁLEK , J. Zabezpečení webových aplikací I. : klientské skriptovací jazyky. *Access server* [online]. 15. 08. 2007, [cit. 2010-05-06]. Dostupný z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2007090001>>.
- [19] VRÁNA, Jakub. *Php.vrana.cz* [online]. 1.7.2005 [cit. 2010-05-06]. Zabezpečení session proměnných. Dostupné z WWW: <<http://php.vrana.cz/zabezpeceni-session-promennych.php>>.
- [20] HTTPS In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 1. 2006, 15. 3. 2010 [cit. 2010-05-06]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/HTTPS>>.
- [21] BBCode In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 19. 5. 2005, 14. 7. 2009 [cit. 2010-05-07]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/BBCode>>.
- [22] MALÝ, J.; KACÁLEK, J. Zabezpečení webových aplikací II. : databáze. *Access.feld.cvut.cz* [online]. 15. 08. 2007, [cit. 2010-05-07]. Dostupný z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2007080002>>.
- [23] SQL injection In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 22. 8. 2007, 19. 4. 2010 [cit. 2010-05-07]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/SQL_injection>.
- [24] TICHÝ, Jan. *Phpguru.cz* [online]. 30. 10. 2007 [cit. 2010-05-07]. Solení hesel aneb Sůl nad zlato. Dostupné z WWW: <<http://www.phpguru.cz/clanky/soleni-hesel>>.

[25] Secure Hash Algorithm In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 3. 1. 2008, 24. 4. 2010 [cit. 2010-05-07]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Secure_Hash_Algorithm>.

[26] Virtual Private Network In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 5. 5. 2006, [cit. 2010-05-08]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Virtual_Private_Network>.

[27] Secure Sockets Layer In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 14. 12. 2004, 26. 6. 2009 [cit. 2010-05-08]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Secure_Sockets_Layer>.

[28] Man in the middle In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2010-05-08]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Man_in_the_middle>.

[29] HTTPS In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, , [cit. 2010-05-08]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/HTTPS>>.

Příloha A – funkce pro vytvoření události

```
function pridejUdalost($druh, $detail)
{
    dbConnect();

    $now = date("Y-m-d H:i:s");

    $sql = "INSERT INTO udalosti (datum_vytvoreni, id_druhu, detail)
VALUES ('".$now."', "
    . mysql_real_escape_string($druh) . ", '"
    . mysql_real_escape_string($detail) . "'" ) ;
    $result = mysql_query($sql);

    $sql = "SELECT id_udalosti FROM udalosti WHERE datum_vytvoreni =
'".$now."' AND
    id_druhu = ".mysql_real_escape_string($druh)." AND detail =
'".mysql_real_escape_string($detail)."'";
    $res = mysql_query($sql);
    while($row = mysql_fetch_object($res)){
        $idUdalosti = ($row->id_udalosti);
    }

    $sql = "SELECT * FROM spravci";
    $res = mysql_query($sql);
    while($row = mysql_fetch_object($res)){
        $id = ($row->id_spravce);
        $sql2 = "INSERT INTO precteni_spravcem (id_spravce, id_udalosti)
VALUES (".$id.", ".$idUdalosti.")" ;
        $result = mysql_query($sql2);
    }
}
```