

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Rezervační hotelový systém

Daniel Poláček

Bakalářská práce

2011



Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

\*Kathy Sierra, Bert Bates: Head First Java, O'Reilly Media 2005, počet stran 720, ISBN-10 1600330002

\*Kline Kevin, Kline Daniel, Hunt Brand: SQL in a Nutshell, O'Reilly Media 2008, počet stran 592, ISBN-10 0596518846

\*Ajit Sagar, Sue Spielman a kol.: Professional Java Server Programming J2Ee 1.4 Edition, Wrox Press 2003, počet stran 850, ISBN-10 1861008139

\*Vivek Chopra, Sing Li, Jeff Genender: Professional Apache Tomcat 6, WROX Press 2006, ISBN-10: 0471753610

\*Sun Microsystems: The Java EE5 Tutorial [online], 2007 [cit. 2009-10-08], dostupný z:

<http://java.sun.com/javase/5/docs/tutorial/doc/docinfo.html>

Vedoucí bakalářské práce:

**Ing. Lukáš Čížek**

Katedra informačních technologií

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

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



prof. Ing. Simona Karamanov, Dr.  
děkan



L.S.



Ing. Lukáš Čížek, 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 11. 05. 2011

Daniel Poláček

## **Poděkování**

Děkuji vedoucímu bakalářské práce panu Ing. Miloslavu Macháčkovi Ph.D. za čas a úsilí, které věnoval mé práci a za jeho odborné rady při vypracování této bakalářské práce.

## **Anotace**

Cílem této práce bylo vytvoření rezervačního hotelového systému pro středně velký penzion na základě informací od ubytovatelů. Po pečlivém prozkoumání situace na trhu jsem začal tvořit vlastní pojetí rezervačních systémů. Při zjišťování aktuálního stavu jsem nejprve zkoumal, jak se vlastně vyvíjely normy, podle kterých se penzion musí řídit. Poté jsem prošel typy ubytování, abych mohl případný návrh systému tvořit univerzálněji a seznámil jsem se s problematikou rezervačních hotelových systémů. Po nashromáždění potřebných údajů jsem vytvořil rozsáhlý návrh databáze a webovou aplikaci.

## **Klíčová slova**

ubytování, penzion, rezervace, stránka, údaje, sezona, klient, databáze

## **Title**

Hotel reservation system.

## **Annotation**

The aim of this work was to create a hotel reservation system for medium-sized pension, based on information from the landlord. After careful examination of the situation of the market, I started to create my own concept of reservation systems. In determining the current market status, I first studied the development of standards which the pension must be followed. Then I went through types of accommodations to be able to design the system more universal and I got into the hotel reservation systems problematic. After the accumulation of necessary data, I created an extensive database design and web application.

## **Keywords**

Accommodation, Guest Houses, reservations, site, data, season, client, database

# Obsah

<b>Seznam zkratk</b> .....	<b>8</b>
<b>Seznam obrázků</b> .....	<b>9</b>
<b>1 Úvodní informace</b> .....	<b>10</b>
<b>2 Teoretická část</b> .....	<b>11</b>
2.1 Vymezení pojmů .....	11
2.2 Historie ubytování u nás a její vývoj .....	11
2.2.1 Standardy a normy .....	11
2.3 Typy ubytování.....	12
2.4 Vývoj rezervací.....	13
2.4.1 Online rezervace a situace na trhu .....	13
2.5 Vývoj aplikačního softwaru .....	14
2.5.1 První kroky .....	14
2.5.2 Výběr vhodných nástrojů .....	15
2.5.3 Tvorba aplikace .....	15
2.5.4 Ladění systému a dokončení.....	15
2.6 Problematika systému.....	16
2.6.1 Data jejich sběr a důvod shromažďování.....	16
2.6.2 Data nutná podle zákona.....	17
2.6.3 Nastavování a správa ubytovacího zařízení.....	18
2.7 Rešerše.....	19
2.7.1 Presentace penzionu a registrace .....	19
2.7.2 Možnosti zákazníka .....	20
2.7.3 Možnosti vedoucího .....	20
2.7.4 Stručný přehled.....	21
2.8 Porovnání se stávajícími systémy.....	21
2.8.1 Subjekt první .....	22
2.8.2 Subjekt druhý.....	23
2.8.3 Subjekt třetí .....	24
<b>3 Praktická část</b> .....	<b>24</b>
3.1 O systému .....	24
3.1.1 Uživatelé systému.....	25

3.2	Návrh databáze .....	26
3.2.1	Registrační část.....	27
3.2.2	Tabulky související s registrací a jejich vazby. ....	27
3.2.3	Hotelová část .....	29
3.2.4	Tabulky související s hotelem a jejich vazby .....	29
3.2.5	Objednávková část.....	31
3.2.6	Vazby mezi jednotlivými díly .....	32
3.3	Vytvoření webové aplikace .....	32
3.3.1	Principy tvorby webové aplikace .....	33
3.3.2	Soubory a složky .....	34
3.3.3	Přehled jednotlivých souborů .....	35
3.4	Vzhled webové aplikace .....	42
3.4.1	Velikost a rozložení stránky .....	42
3.4.2	Barvy .....	43
3.4.3	Písmo .....	43
3.5	Instalace .....	44
3.6	Uživatelská příručka .....	44
3.6.1	Host a zákazník.....	44
3.6.2	Vedoucí.....	45
3.7	Závěr.....	46
	<b>Literatura .....</b>	<b>47</b>
	<b>Příloha A – ER diagram.....</b>	<b>49</b>
	<b>Příloha B – Zpracování dat z formuláře pro úpravu adres.....</b>	<b>50</b>
	<b>Příloha C – Ukázka grafického vzhledu registrace .....</b>	<b>51</b>



## Seznam zkratek

PHP	Personal Home Page
SQL	Structured Query Language
CSS	Cascading Style Sheets
NN	Not Null
PK	Primary Key
FK	Foreign Key
PFK	Primary Foreign Key
HTML	HyperText Markup Language
DML	Data Manipulation Language

## Seznam obrázků

Obrázek 1 - Ukázka z ubytovací knihy .....	16
Obrázek 2 - Ukázka ubytovací karty .....	17
Obrázek 3 - Use Case diagram .....	25
Obrázek 4 - ER diagram registrační část .....	27
Obrázek 5 - ER digram hotelová část .....	29
Obrázek 6 - ER diagram objednávky .....	31
Obrázek 7 - Ukázka komunikace pomocí dotazů .....	33
Obrázek 8 - Ukázka komunikace pomocí procedur .....	34
Obrázek 9 - Adresářový strom .....	35
Obrázek 10 - ukázka vzhledu stránky .....	43
Obrázek 11 - ER diagram .....	49
Obrázek 12 - Layout registrace .....	51

# 1 Úvodní informace

Tato bakalářská práce vznikla jako průřez aktuálního dění na trhu rezervačních systémů pro ubytovací zařízení. O tuto problematiku se zajímám již několik let a právě to mě vedlo k volbě tématu této práce. Snažil jsem se specializovat se na středně velké ubytovací zařízení, jako jsou penziony všeobecně. U těchto objektů je totiž nejpevnější vazba mezi klientem a provozovatelem. U středně velkých penzionů je velmi časté, že se klienti vrací na stejné místo po mnoho let a právě proto je třeba jim vytvářet co nejpříznivější podmínky.

I v dnešní době je rezervačních systémů celkem málo a speciálně u penzionů se na internetovou prezentaci a online rezervaci velmi často zapomíná. Mnoho provozovatelů hřeší na svoji stálou klientelu a domnívají se, že aktuální stav jejich rezervací je dostačující. Opak je bohužel pravdou. Stále více lidí požaduje pohodlné online rezervace a očekávají, že jim tato možnost bude poskytnuta i u ubytování.

Právě proto jsem v této práci shrnul dostupné informace o poskytování ubytovacích služeb a rozebral stávající systémy na trhu s jejich výhodami a nevýhodami. Navrhl vlastní základní systém, který by mohl sloužit pro online rezervace a realizoval jsem jej s pomocí programovacích a databázových nástrojů.

## 2 Teoretická část

### 2.1 Vymezení pojmů

**Ubytovací zařízení:** Za ubytovací zařízení považujeme stavbu nebo část stavby, která je využívána k přechodnému ubytování klientů.

**Středně velký penzion:** Pro potřeby této práce bude pro pojem středně velký penzion myšleno ubytovací zařízení s kapacitou lůžek mezi 15 až 25 s proměnnými službami.

**Rezervace:** Pod pojmem rezervace je myšleno objednání určitého pokoje nebo služby na konkrétní datum, případně čas.

**Katalog:** Katalogem je myšlena webová aplikace shromažďující jednotlivé ubytovací zařízení pod společný rozcestník, kde má každý penzion svoji prezentaci a většinou také možnost rezervace.<sup>1</sup>

### 2.2 Historie ubytování u nás a její vývoj

#### 2.2.1 Standardy a normy

První zmínka o standardizaci ubytovacích zařízení je z roku 1919 kdy byl zákonem číslo 649/1919 Sb. zřízen „Instruktorát pro podniky ku přechovávání cizinců“, který spadl do kompetence ministerstva průmyslu a obchodu. Bylo dohlíženo na splňování provozních podmínek ubytovacího zařízení.

Další krok byl učiněn v roce 1939, kdy došlo k vymezení kategorií a tříd ubytovacích zařízení. Jednalo se o třídy A-D s tím, že první tři třídy byly vymezené pro hotely a čtvrtá pro ubytovací hostince. O třídění rozhodoval živnostenský úřad.

Rok 1948 přinesl velký převrat. Podle zákona 124/1948 Sb. o znárodnění některých hostinských a výčepních podniků a ubytovacích zařízení začaly být veškeré podniky sjednocovány a klasické penziony, jak je známe dnes, prakticky vymizely. Rozdělení do skupin zůstalo zachováno a byly stanoveny pevné ceny podle těchto skupin.

Postupem času vzniklo několik úprav zabývajících se ubytovacími zařízeními, ale podstatný je až datum 1. 4. 1991, kdy byly přidány klasifikační znaky pro kategorizaci ubytování v soukromí a přechází se na nový systém kategorií zahrnující rozdělení různých typů ubytovacího zařízení. Například 5 tříd pro hotel, 3 pro penzion a podobně.

Do roku 1998 bylo zařazování do tříd prováděno na okresních výborech na žádosti provozovatele.

---

<sup>1</sup> Ukázka katalogu: <http://www.hotely.cz/>.

Po datu 1. 7. 1998 se samotné zařazení do tříd stává velmi chaotické. Je sestaveno doporučení pro kategorizaci a podle něj může dokonce sám provozovatel stanovit kategorii svého zařízení. Proběhlo několik pokusů o vytvoření autority pro zařazování do tříd, ale většina bez známek úspěchu. Konečné normy se vytvořily až postupem času pomocí různých svazů a organizací, zabývajících se touto problematikou.

I v dnešní době je v normách pro kategorizaci ubytovacích zařízení velký zmatek. Sice existují normy, podle kterých by se měl provozovatel řídit, ale neexistuje žádný kontrolní orgán, který by dohlížel na kvalitu ubytování. Osobně danou situaci považuji za nešťastnou. Doplácejí na ní nejen samotní klienti, kteří si nikdy nemohou být jisti, kam vlastně jedou, tak samotní ubytovatelé.

Při shromažďování těchto informací jsem vycházel z článku publikovaného na oficiálním webu cestovního ruchu (Hotel Fontána s.r.o, 2009).

## 2.3 Typy ubytování

Ubytování se dělí na různé typy podle velikosti, umístění a poskytovaných služeb (Ministerstvo hospodářství aj., 1994)<sup>2</sup>. Některé z těchto kategorií se vzájemně překrývají nebo jsou jejich hranice prakticky neznatelné. Představme si teď aspoň pár základních typů ubytování:

1. Penzion - Podle dnešních měřítek je penzionem ubytovací zařízení s kapacitou 10 a více lůžek a variabilním stravováním. Většinou se jedná o malé stavby, mnohdy i rodinné domky. Hlavní výhodou tohoto typu ubytování je bezesporu osobní přístup ke klientům, velká flexibilita služeb a průměrně nižší ceny než u velkých hotelů.
2. Hotel – ubytovací objekt s velkou kapacitou lůžek. Řádově se počítá zhruba 35 a více lůžek, tento počet není nikde stanoven. V některých zdrojích je tato hodnota uváděna v pokojích (UNIHOST aj., 2010). Součástí hotelu je zpravidla i restaurace, společenská místnost a zařízení pro trávení volného času.
3. Motel – ubytovací zařízení v blízkosti velkých silnic a dálnic sloužící ke krátkodobému ubytování převážně motoristů.
4. Ubytovna – ubytovací zařízení velmi nízké kvality. Velké společné pokoje mnohdy až o více jak 10 lůžkách se společným sociálním zařízením. Velice často slouží jako nouzové řešení nečekané situace.
5. Privát- menší obdoba penzionu, jinak se ve většině ohledů nijak neliší.

---

<sup>2</sup> Jednotlivé typy ubytovacích zařízení se můžou velmi lišit, protože existuje pouze doporučení pro kategorizace a ne závazná norma.

Samozřejmě existuje bezpočet různých dělení, jako jsou apartmány, kde je více místností se společným sociálním zařízením a kuchyňkou, ale tímto způsobem bývá řešeno i mnoho penzionů. Zde je právě vidět úzká hranice mezi jednotlivými typy ubytování, ale touto problematikou se dále zabývat nebudeme.

## **2.4 Vývoj rezervací**

Možností, jak si rezervovat ubytování v penzionu, je dnes celkem mnoho, ale ne vždy tomu tak bylo. Za počátek by se dala považovat poštovní korespondence. Tento způsob by samozřejmě mohl fungovat i v dnešní době, ale jeho nevýhody jsou zřejmé. Největší slabinou této možnosti je samozřejmě časová prodleva a jistě i to, že známka taky něco stojí. Další možností, která následuje v časovém vývoji, je telefonní rezervace. Podle mého zkoumání aktuální situace na trhu je tato možnost stále nejvyužívanější způsob rezervace. Hlavně u středně velkých penzionů, kde se mnohdy ubytovatel zná s klienty mnoho let, je tento systém velmi populární. Jednak je tato komunikace mnohem osobnější než ostatní možnosti a dalším důvodem popularity telefonních rezervací je určitá nedůvěra mnoha lidí k internetu. Třetí variantou je samotné využití internetu, které v dnešní době plně nahrazuje ostatní zmíněné možnosti. Ať už se jedná o emaily, které jsou levné, rychlé a spolehlivé. Díky tomu mohou nahradit poštovní korespondenci. Případně online komunikátory nahrazující telefonní hovory nebo samotné online rezervační systémy.

### **2.4.1 Online rezervace a situace na trhu**

Při zkoumání aktuální situace jsem byl velice nemile překvapen, jak malá část ubytovacích zařízení používá alespoň nějaký online rezervační systém. Mnohdy jsem narazil i na stránky, které by se daly označit za tragické, a to se jednalo pouze o obyčejné informativní webové stránky. U samotných rezervačních systémů, je trh rozdělen na několik sekcí.

V první sekci jsou „rezervační systémy“ řešené pouze formulářem, který klient vyplní a na základě tohoto formuláře je s klientem dále komunikováno. Drtivá většina těchto systémů je pouhým php formulářem, který se následně odesílá na email ubytovatele. Díky tomu je pak nutná další komunikace, která je ve většině případů nežádoucí. Jako výhoda tohoto systému je pak větší kontrola pro ubytovatele a nižší pořizovací cena.

Druhá sekce je rezervační systém, kde je vytvořena přímo rezervace klientem a to nejčastěji na konkrétní pokoj nebo službu. Tento systém nevyžaduje prakticky žádnou další komunikaci a klient má dobrý přehled o svých rezervacích. V tomto systému ihned vidíte, zda je požadovaný pokoj k dispozici a v případě zájmu si jej můžete objednat. Velkou výhodou je právě vynechání zbytečné komunikace a tím zkvalitnění služeb pro zákazníka. Díky volbě tohoto systému se může zvýšit obsazenost penzionu. Bohužel u těchto systémů je cena mnohdy velmi vysoká a právě proto není tento systém tak rozšířený. Cena se přímo odvíjí od práce, která je na daném systému potřebná. Zatímco první varianta může fungovat pouze na úrovni php u druhé varianty je již potřebná poměrně rozsáhlá databáze. Těchto systémů je velmi málo, osobně jsem narazil jen na pár takovýchto systémů a u všech jsem našel celkem významné nedostatky.

Třetí sekce je hybridem předchozích dvou. Obsahuje jak možnost registrace a objednávání po přihlášení uživatele, tak i možnost rezervace přes formulář. U těchto systémů je formulářová část pouze drobným rozšířením. Osobně považuji tento systém za zbytečný, protože vyplnění údajů do formuláře většinou zabere stejné množství času jako vyplnění registrace.

Katalogy a integrované systémy jsou jistě nedílnou součástí webových prezentací a rezervací. V tomto případě je sjednoceno velké množství ubytovacích zařízení do jednoho systému. Každý takovýto objekt má v katalogu vlastní prezentaci a velice často také přímo možnost rezervaci ubytování. Vesměs se tento systém dá dělit na dvě skupiny. V první skupině jsou katalogy placené, kde ve většině případů není uveden přímý kontakt na ubytovací zařízení, ale kontakt zprostředkovává provozovatel katalogu, samozřejmě za provizi. Druhým systémem funguje trochu jinak. Ubytovací zařízení jsou v katalogu umístěna zdarma a je zveřejněn přímý kontakt na penzion či hotel. Rezervace v obou systémech je často řešena pomocí online systému PREVIO (MITON Previo s.r.o., 2002). Jedná se o krokovou rezervaci, kde je výsledek zaslán do databáze. Provozovatel se pak přihlásí do tohoto systému a v přehledném kalendáři vidí obsazenost svého objektu. Jedná se o zajímavý nápad, ale jeho obrovskou slabinou je cena. Ubytovatel zaplatí poplatek a k tomu měsíční paušál, který se v plné verzi pohybuje nepřiměřeně vysoko.

## 2.5 Vývoj aplikačního softwaru

Vývoj aplikace je velice individuální a závislý na dostupných prostředcích. Dá se vytyčit několik základních kroků, přes které musíme projít.

### 2.5.1 První kroky

Všeobecně základem návrhu každého systému je zjištění, jak vytvářený systém má fungovat. Nejjednodušší cesta je poradit se s někým, kdo se v dané problematice pohybuje. I když jsou k dispozici určité publikace o tom, jak fungují penziony, jsou prakticky nepoužitelné. Trendy se vyvíjejí velmi rychle a nejcennější informace jsou vždy z praxe. Takže musíme najít ubytovatele ochotné sdělit nám svoje pracně nabyté poznatky, což se může ukázat jako velký problém. Někteří provozovatelé nám ochotně poradí s tím, jak penzion funguje a co všechno je třeba k jeho bezproblémovému provozu, ale jen málo z nich má na své ubytovací zařízení nějaký rezervační systém. Když už najdeme vhodný objekt s odpovídající aplikací, narazíme na problém, že provozovatel tohoto objektu nemá ani nejmenší tušení jak jeho rezervační systém vlastně funguje.

Také je nezbytně nutné dozvědět se potřebné informace o normách a povinnostech, které se na takovéto ubytovací zařízení vztahují. Optimálním zdrojem jsou opět provozovatelé, ale pozor. Tyto informace je nutné si ověřit, protože bývají velmi často nepřesné a zkreslené<sup>3</sup>.

---

<sup>3</sup>Je dobré podívat se na dané normy a poplatky přímo do zákonů a podle nich dále postupovat k dalším zdrojům.

## 2.5.2 Výběr vhodných nástrojů

U vytváření hotelového rezervačního systému se tento systém dá rozdělit na dvě části. První část je úložiště dat a druhá informační, která tyto data zobrazuje. U první části musíme dbát na co největší spolehlivost. Zatímco u druhé spíše na vizuální efekt.

**Nástroj pro ukládání dat:** Pro ukládání dat máme více možností, ale osobně považuji jako nejlepší použití databáze. Pro vytváření takového systému je nutné uložení relativně velkého množství informací, které je třeba přehledně skladovat mít k nim rychlý přístup. Osobně bych doporučoval produkty firmy Oracle. Máme k dispozici buď placenou databázi Oracle v aktuální verzi 11g, případně databázi MySQL, která je zdarma. Přes obě tyto aplikace jdou spravovat velice kvalitní databáze.

**Nástroje pro tvorbu stránek:** Samotné stránky můžeme psát buď ručně, nebo využít některý z nástrojů pro tvorbu webových stránek. Osobně se přikláním k ručnímu psaní. Jde sice o zdlouhavější možnost, ale máme mnohem větší přehled o vytvářeném systému. Ať už zvolíme jakýkoliv nástroj, musíme mít na paměti, že vzhled stránky je velice důležitý a má klíčovou roli v úspěchu naší práce.

## 2.5.3 Tvorba aplikace

Předpokládáme, že máme základní informace a zvolili jsme nástroje, které budou pro naši práci potřeba. Teď je čas na první návrh aplikace. Nejprve je nutné navrhnout základ systému, za což by se dal považovat model klient – ubytovací zařízení - rezervace. Tedy základní registrace klienta, základní údaje o penzionu jako jsou pokoje, počet lůžek, sezóny, ceny a nakonec samotná rezervace pokoje. S tímto návrhem je vhodné navštívit ubytovatele a požádat jej o radu. Je jasné, že každý má nastavené jiné požadavky, a proto takovýto systém částečně směřuje podle sebe. To je nezbytně nutná vlastnost pokud máme vytvořit něco nového, ale v tomto případě je také potřeba držet se osvědčeného modelu. Pojetí může být odlišné, ale základní stavební kameny celého systému musí být stejné. Pokud se nám dobře podaří vytvořit takovouto kostru, máme z velké části vyhráno. Je také třeba si rozmyslet, jak bude daná stránka vypadat a co všechno by měla obsahovat. Návrh databáze může být obsáhlejší, ale rozhodně je nežádoucí zapomenout na nějakou službu.

## 2.5.4 Ladění systému a dokončení

Ladění je nejzdlouhavější část celé tvorby. Jsem si naprosto jistý, že na poprvé se nikomu nepovede tak dokonalý návrh systému, aby nebylo nutné něco předělat nebo přidat. V této části je taky nejčastější a velice nutná komunikace s lidmi, kteří se v dané problematice vyznají, protože je zde největší prostor pro vlastní nápady. Díky tomu se může stát, že vytvoříme systém, který by byl zcela bezcenný, pokud nás někdo nebude směřovat správným směrem. Nakonec je dobré systém nějakou dobu testovat, než se přejde na ostrý provoz.



## 2.6 Problematika systému

### 2.6.1 Data jejich sběr a důvod shromažďování

Zjišťování co největšího množství údajů je pro chod penzionu naprosto nezbytné. Shromažďované údaje můžeme rozdělit do tří pomyslných skupin:

1. Údaje nutné pro samotnou administrativu a chod penzionu,
2. kontaktní údaje,
3. rozšiřující údaje.

Pokud hovoříme o první skupině, jedná se především o údaje, jako jsou jméno, příjmení, adresní údaje, které jsou nutné pro vyplnění ubytovací knihy. Tento dokument je ubytovatel povinen vést v písemné podobě a na vyžádání předložit ke kontrole. Tuto kontrolu většinou provádí cizinecká policie a dále tato kniha slouží jako podklad pro placení místních poplatků.

Pořadové číslo	Datum narození	Trvalý pobyt - stát	Datum příchodu	Poznámka: (Číslo pokoje, zp. Úhrady, SPZ vozidla aj.)
		Obec		
Příjmení		Část obce	Datum odchodu	
Jméno	Občanský průkaz - pas	Ulice č.p. (ev.)		

Obrázek 1 - Ukázka z ubytovací knihy

V druhé skupině jsou zařazeny veškeré možné kontakty, ať už se jedná o telefon, který je zpravidla nejpoužívanějším kontaktem u středně velkých penzionů nebo celou škálu v dnešní době rozšířených a používaných komunikačních nástrojů. Obecně platí, že čím více možností jak kontaktovat zákazníka máme, tím lépe. Pomocí emailů můžeme rozesílat informace pro ubytované. Díky sociálním sítím typu facebook můžeme informovat o novinkách, akcích a podobně<sup>4</sup>.

Třetí pomyslnou skupinou jsou údaje rozšiřující. Tato skupina je zaměřená především na zkvalitňování služeb a zlepšení marketingu. Sem můžeme zařadit prakticky jakoukoliv informaci o klientovi. Například se může jednat o informace typu oblíbené jídlo, květiny, domácí mazlíčci, nebo zda je daná osoba kuřák či ne. I když se na první pohled mohou tyto informace zdát nepodstatné, mají na spokojenost zákazníka obrovský vliv. Uvedu malý příklad: Do penzionu přijede rodina s dětmi, o které víme, že jeden člen má rád růže, další preferuje plzeňské pivo a mají psa. V případě, že dáme do pokoje růže,

<sup>4</sup> Sílu těchto sítí mnoho podnikatelů podceňuje, ale v dnešní době jsou pro zveřejňování informací mnohem účinnější než aktuality na stránkách ubytovacího zařízení.

vychladíme několik piv a připravíme ke dveřím psí misky, dosáhneme už v prvních několika minutách mnohem větší spokojenosti u zákazníků.

## 2.6.2 Data nutná podle zákona

Samotná administrativa vedení penzionu vyžaduje sběr určitých údajů o klientech. Tyto údaje se rozdělují do dvou skupin podle klientely (zahraniční, domácí) a i když některé z údajů jsou shodné, je každá skupina ošetřena jinou právní úpravou (Úřad pro ochranu osobních údajů, 2005). Většina ubytovatelů sbírá tato data dvěma způsoby. V prvním případě si od klientů vyžádají pasy, případně občanské průkazy a údaje z nich doplní do ubytovací knihy. Nebo nechají klienty vyplnit připravené ubytovací karty. K mému velkému překvapení jsem zjistil, že velká část ubytovatelů přesně neví, jaké údaje musí pro danou skupinu sbírat. A nyní si projdeme zmiňované skupiny.

<b>Příhlašovací list</b>	<b>Penzion Poláček</b>	
<b>Doba pobytu</b>	<b>Od . . 2011</b>	<b>Do . . 2011</b>
<b>Jméno</b>		
<b>Příjmení</b>		
<b>Datum narození</b>		
<b>Ulice</b>		
<b>Č.p.</b>		
<b>Město</b>		
<b>PSČ</b>		
<b>Číslo obč průkazu</b>		

PENZION POLÁČEK  
Svoboda nad Úpou  
Sluneční stráž 203  
54224

Obrázek 2 - Ukázka ubytovací karty

První skupina je klientela zahraniční, u které se musejí osobní údaje evidovat v ubytovací knize podle zákona o pobytu cizinců (Úřad pro ochranu osobních údajů, 2005). Rozsah evidovaných údajů je vymezen takto:

1. Jméno a příjmení klienta,
2. den, měsíc a rok narození,
3. státní občanství,
4. číslo cestovního dokladu,
5. počátek a konec ubytování.

U skupiny druhé se shromažďování údajů odvíjí od zákona o místních poplatcích<sup>5</sup>, ze kterého vychází povinnost platit rekreační a lázeňské poplatky (pokud je daná obec zavedla). Do této skupiny spadají klienti české národnosti. Rozsah nutných údajů, je u této skupiny vymezena takto:

1. Dobu ubytování,
2. účel pobytu,
3. jméno, příjmení, adresu místa trvalého pobytu nebo místa trvalého bydliště v zahraničí,
4. číslo občanského průkazu nebo cestovního dokladu fyzické osoby, které ubytování poskytl.

U těchto údajů vycházím z dokumentů dostupných na webových stránkách nakladatelství ekonomické a právní literatury (Sagit, 1999).

### **2.6.3 Nastavování a správa ubytovacího zařízení**

Ač se to na první pohled nezdá, tak je správa ubytovacího zařízení velice živá a proměnlivá věc. Nemůžeme proto vytvořit systém, u kterého je většina dat pevně zadaná. Kdyby byla data v systému neměnná, nemohl by provozovatel pružně reagovat na změny. První věc, u které zjistíme velkou nutnost změn, jsou sezóny.

**Sezóny:** Většina penzionů používá rozdělení roku na tři sezóny, ale vyskytují se i penziony, které používají až pět různých sezón. V takovém případě nám nastává problém, protože přidávat pokaždé novou sezónu pomocí přímého vstupu do databáze by bylo zbytečně pracné. Právě proto je nutné vytvořit v aplikaci modul pro přidání sezóny a jejich úpravu podle potřeby. Podobných případů je mnoho, ať už se jedná o ubytovací poplatky, které stanovuje a vybírá město nebo třeba změnu výši zálohy.

**Objednávky:** Při rezervaci pokoje nebo služby je nutné hlídat několik aspektů, které nám celý systém ovlivňují. Je nezbytné počítat s tím, že klient někdy není soudný a tak je třeba jej limitovat. Například omezením doby rezervace. To znamená, že si klient nemůže objednat pokoj na silvestr dva roky dopředu a podobně. U pokojů je také nutnost sledovat počet lůžek a osob, pro které je pokoj určen. Situace, kdy v hlavní sezóně máme dva pětilůžkové pokoje obsazené dvěma lidmi, se také mívá účinkem.

---

<sup>5</sup>Zákon č. 565/1990 Sb., o místních poplatcích, v platném znění.

**Platby:** Bohužel až nepříjemně často se stává, že cena udávaná ubytovatelem nečekaně naroste a klient se pak diví, kde se určité částky vzaly. Je tedy nutností dát klientovi podrobný přehled o cenách a o částce, kterou je nutné uhradit. Tato nepříjemná skutečnost velmi kazí reputaci českých ubytovatelů. Předcházet této skutečnosti jde tím, že veškerá poplatky započítáme již do ceny pobytu a u přehledu plateb je pak pečlivě rozepíšeme. Proto je dobré mít v systému možnost zadání několika různých typů plateb pro rozčlenění a přehlednost v tom, za co vlastně klient platí.

## 2.7 Rešerše

Jak už jsem ve své práci popisoval, má vytváření rezervačního systému pro středně velký penzion svá pevná pravidla. Optimální aplikace by měla mít ošetřeny všechny tyto popsané případy a být co nejpříjemnější jak vzhledem, tak ovládáním pro klienty.

### 2.7.1 Prezentace penzionu a registrace

Moje aplikace je navržena tak, aby samotná databáze byla co nejobsáhlejší a v případě potřeby se daly dodělat jednotlivé moduly. Naproti tomu je postavena webová aplikace, u které je snaha opačná. Šlo mi o to, aby aplikace byla příjemná pro klienta a nabízela vše potřebné, ale nic navíc. Procházel jsem nasbíraná data od různých ubytovatelů a podařilo se mi zúžit informace, které jsou pro potenciálního klienta podstatné. U informací o ubytování jsem došel k následujícímu výsledku:

1. Informace o poloze,
2. ceník a možnost výpočtu ceny pobytu,
3. pokoje a zjištění obsazenosti,
4. fotogalerie pro lepší představu,
5. zajímavosti v okolí a zábava na volný čas,
6. kontakt pro dotazy a případně pro snadnější nalezení cíle.

Na základě tohoto přehledu jsem sestavil hlavní nabídku informací o penzionu, která je uživateli zobrazena hned při navštívení systému. Po průzkumu u několika desítek lidí jsem toto menu začal považovat za úplné. Tato nabídka je k dispozici jak přihlášeným tak nepříhlášeným uživatelům.

Registrace do systému je uložena na nejběžnějším místě vpravo nahoře, protože zde většina uživatelů danou položku očekává. U samotné registrace jsem zvolil nutnost zadání všech potřebných dat pro vyplnění ubytovací knihy. Pokud se potenciální klient rozhodne pro registraci, očekávám, že našel, co hledal a bude vytvářet rezervaci. Kdybych nevyžadoval všechna potřebná data již zde, musel by klient doplnit chybějící data při samotné objednávce.

### 2.7.2 Možnosti zákazníka

Vzhledem k faktu, že klient do systému zadává svá data, je důležitou částí systému dovolit uživateli úpravu svých údajů. V mé aplikaci jsem vytvořil modul osobní údaje, který je k dispozici po přihlášení uživatele. Jako první část tohoto modulu je přehled zadaných údajů pro kontrolu. Dále pak tři samostatné položky pro úpravu hesla, osobních údajů, kontaktů a adresy. V původním návrhu jsem počítal s možností tyto položky spojit v jednu, ale od návrhu jsem upustil, protože toto spojení se ukázalo jako dost nepřehledné.

Rezervace je asi nejkomplicovanější kapitola. Při tvorbě této části jsem se musel několikrát rozhodovat, co bude vlastně nejlepší. Názory lidí, se kterými jsem se radil, byly dost rozdílné. Systém jsem nakonec postavil na krokové rezervaci, kde v prvním kroku uživatel zadá požadované datum příjezdu a odjezdu. Datum jsem limitoval dvěma způsoby, jednak je možné rezervovat jen dva roky dopředu od aktuálního roku a dále pak je možné rezervovat jen období, na které je vytvořen ceník. V druhém kroku pak vidíme přehled volných a obsazených pokojů a zadáváme počet dospělých, počet dětí a účel pobytu. Zde jsem se rozhodl pro variantu, kde je rozlišeno, zda se jedná o dítě nebo dospělou osobu. Dále se nám uvolní možnost výběru pokoje, kde jsem rezervaci limitoval na pokoje s kapacitou počínající počtem osob plus dvě. To znamená, že pokud bude vytvářena rezervace pro jednu osobu, může si zvolit maximálně třílůžkový pokoj a to jen v případě, že nejsou k dispozici jedno a dvoulůžkové pokoje. Pokoje je možné rezervovat pouze jeden po druhém a ne více pokojů naráz. Tento systém je lepší pro přehlednost. Poslední krok je pak rekapitulace s vypočtenou cenou a potvrzení samotné rezervace.

### 2.7.3 Možnosti vedoucího

Vedoucí, který je hlavní autoritou mé aplikace, má možnost spravovat většinu zadávaných dat do systému. Disponuje seznamem registrovaných uživatelů, u kterých je možnost náhledu na jejich detaily včetně rezervací, které uživatel vytvořil. Může také měnit registrační údaje jednotlivých klientů. Přidávání uživatelů je řešené pomocí registrace, kde vedoucí zadává potřebné údaje. Při změně údajů si vedoucí vybere klienta, u kterého chce měnit registrační informace a s tím dále pracuje.

**Možné akce s uživateli:** Vedoucí může upravovat registrační údaje u jednotlivých uživatelů. Má k dispozici stručný seznam klientů, který může řadit podle potřeb pro snazší vyhledávání. Ze seznamu pak vybírá konkrétního uživatele, se kterým provádí další úkony. Po výběru jsou zobrazena kompletní data uživatele včetně jeho rezervací v systému a částkou, kterou ještě dluží. Klienta lze ze systému odebrat, ale pouze tehdy nemá-li žádné objednávky v systému. V případě, že má vytvořené objednávky, je nutné nejprve odstranit tyto položky a až poté lze odebrat samotného uživatele. Dále je také vedoucímu umožněno měnit veškeré údaje klienta včetně hesla. Pro přidání nového uživatele je použita klasická registrace, kterou vedoucí vyplní a tím vytvoří nového klienta.

**Možné akce s daty o penzionu:** V mém systému lze do penzionu libovolně přidávat pokoje a také je odebírat, ale jen tehdy, není-li na daný pokoj vytvořena objednávka. Vedoucí má i možnost blokovat konkrétní pokoj. Další nastavení, nad kterými má plnou kontrolu se vztahují na sezony a ceník. U sezón je možné přidávat a odebírat jednotlivé položky. Vytvořit je lze jen tehdy, pokud se nepřekrývají zadaná data jedné sezóny s jinou. Odebírání je umožněné kdykoliv, protože u sezóny nejsou žádné specifické požadavky na její uchování. Při vytvoření objednávky se vypočítá cena podle ceníku a částka je uložena zvlášť do jiné tabulky. Vedoucí má možnost kontroly návaznosti jednotlivých sezón a podle toho je dále upravovat. Je nutné, aby na sebe jednotlivé sezóny navazovaly, kvůli výpočtu ceny pobytu.

**Možné akce s rezervacemi:** Vedoucí má k dispozici stručný přehled objednávek, se kterým může dále pracovat. Po výběru konkrétní rezervace se objeví plný náhled na danou položku a možnost jejího odebrání nebo potvrzení. Systém jsem navrhoval tak, aby vedoucí mohl potvrdit přijetí platby a tím uznat rezervaci za platnou. Původně jsem počítal s tím, že by se rezervace sama smazala, pokud by nebyla včas uhrazena záloha. Od tohoto návrhu jsem ustoupil, protože byl zbytečně limitující. Může nastat situace, kdy bude klient dohodnutý s vedoucím, že zálohu nezašle, nebo případ kdy je rezervace vytvořena hned na následující den. Pak by automatické mazání bylo na obtíž.

#### **2.7.4 Stručný přehled**

V mé práci mi šlo o vytvoření ubytovacího systému, kde je základem registrace klienta a až poté je možné zpracovávat samotné objednávky. Snažil jsem se o co největší pohodlí pro klienta. Většina systému na trhu je řešena pouze rezervačním formulářem, kde je pak nutná další komunikace se zákazníkem. Tuto skutečnost jsem ve své aplikaci úplně vynechal. Tím jsem docílil možnosti kompletní online rezervace, u které klient ihned vidí, zda má daný pokoj objednaný a může i sledovat přijetí zálohy a mnoho dalších věcí. Další velkou výhodou tohoto systému je absence vyplňování ubytovací karty. Což je mnohdy nepříjemná nutnost po dlouhé cestě do penzionu. Příjemná je i rezervace konkrétního pokoje. Některé systémy tuto možnost nenabízejí a tak můžeme těžko ovlivnit, zda dostaneme náš oblíbený pokoj. Také jsem se zaměřil na co nejpříjemnější vzhled, dostatečně velké a čitelné písmo a vysokou přehlednost stránky.

Tento systém považuji za velice zdařilý a v mnoha ohledech jedinečný. Při samotné instalaci systému je možné ovlivnit mnoho věcí přesně podle požadavků vedoucího.

## **2.8 Porovnání se stávajícími systémy**

K porovnávání jsem si vybral tři subjekty, které jsem předtím v mezích možností prostudoval. Najít vhodné objekty, kde byla ochota od provozovatele podělit se o svůj systém je velmi málo. Bohužel se ukázalo, že většina provozovatelů nemá tušení, jak jejich rezervace vlastně fungují. U jednoho systému, který zde budu porovnávat, jsem měl možnost nahlédnout i do zdrojových kódů aplikace u druhých dvou mi byl systém pouze vysvětlen.

Rezervační systém, který by byl podobný mé aplikaci, je velice vzácný. Našel jsem jen velmi málo takovýchto aplikací a většina z nich byla použita na luxusní ubytovací zařízení. I přes snahu kontaktovat provozovatele těchto zařízení jsem uspěl pouze v jediném případě. Přesto, že ochota provozovatele byla na nezvyklé úrovni, nezískal jsem potřebné informace pro porovnání. Systém byl tvořen firmou na zakázku a o vnitřním běhu jsem se nedozvěděl žádné použitelné informace. Při porovnávání s mojí aplikací jsem se zaměřil na několik aspektů:

1. Vzhled a přehlednost aplikace,
2. informativní hodnota stránek,
3. tvorba rezervace,
4. přehlednost a kontrola stavu rezervací.

Hodnotím známkami 1 pro úspěch porovnávaného systému, 0 shodnost, -1 pro neúspěch v porovnání s mým systémem.

### **2.8.1 Subjekt první**

Stylový penzion Stará pošta<sup>6</sup>, který se nachází v malém městečku Potštejn. Penzion funguje zhruba 1 rok a od samého začátku má webové stránky. Kapacita penzionu je 17 lůžek. Stránky penzionu jsou na první pohled velice dobře udělané, rozumně zvolené barvy, rozměry stránky a přehlednost jsou silnou zbraní tohoto webu. Informativní hodnota je na vysoké úrovni, texty jsou krátké a poskytují všechny potřebné informace. Osobně bych volil kratší menu, některé položky by se daly spojit. Vzhledem k tomu, že stránky nebyly tvořeny firmou, ale vlastními silami jsou neobyčejně kvalitní.

V samotném webu je klasický rezervační formulář, který zasílá vyplněná data metodou POST do jednoduché rekapitulace objednávky a odesílá je na email. Nad formulářem jsou dobře zpracované informace o rezervačním systému i s dalšími kontakty. Toto ubytovací zařízení je i v několika webových katalozích. Provozovatelka tyto způsoby prezentace nehodnotí nijak kladně a od placených katalogů dokonce ustupuje. Dále je pak tento penzion veden v rezervačním systému PREVIO. Tento systém používají některé katalogy k vytvoření rezervace na ubytovací zařízení. Za nejúspěšnější způsob rezervace na tomto penzionu pak považuje přímo webový formulář a na druhém místě telefonické spojení.

---

<sup>6</sup><http://www.penzionpotstejn.cz/index.php>.

Porovnání s mojí aplikací:

1. Vzhled je v porovnání s mým systémem nápaditější 1,
2. informativní hodnota je opět podobného rozsahu hodnocení 0,
3. pouhý rezervační formulář, nutná další komunikace, velmi rychlé odpovědi 0,
4. úplná absence kontroly a přehledu obsazenosti -1.

Velmi zajímavě zpracovaný web, velká ochota a snaha provozovatelky celkem dobře nahrazující fakt, že je rezervace na stránkách možná pouze rezervačním formulářem.

### 2.8.2 Subjekt druhý

Moderní penzion v Pardubicích<sup>7</sup>. Stránka je na první pohled velmi příjemná a přehledná. Jednoduché menu se základními informacemi je plně dostačující a potencionální zákazník se dozví vše potřebné. Ceník je trochu nepřehledný, ale jsou obsaženy veškeré potřebné informace a při troše hledání je možné dozvědět se vše potřebné. Na první pohled se stránka může jevit jako zbytečně obsáhlá, ale brzy zjistíme, že je obsažena i prezentace restaurace, která je součástí penzionu.

Rezervace je tvořena jednoduchým formulářem, který obsahuje několik netypických voleb, jako jsou počet aut, domácí mazlíčci a podobné. Samotná rezervace je pak tvořena pouze odesláním požadavku na ubytování na email ubytovatele. Díky tomu je nutná další komunikace a není jisté, zda je volná kapacita pro náš požadavek. O průběhu naší objednávky nás tak informuje pouze ubytovatel, což považuji za zbytečně složité a limitující pro obě strany. Přehled volných pokojů zcela chybí, což je obrovská nevýhoda a může vést až ke ztrátě klienta. Stejně jako u prvního subjektu je penzion veden v několika katalozích jak placených, tak neplacených.

Porovnání s mojí aplikací:

1. Vzhled a provedení stránky jsou velmi podobné hodnocení 0,
2. informativní hodnota je opět podobného rozsahu hodnocení 0,
3. obyčejný formulář nutná další komunikace -1,
4. úplná absence kontroly a přehledu-1.

Spíše informativní aplikace, zajímavá s běžnou rezervací. Rozhodně ale patří k těm lepším zástupcům na našem trhu.

---

<sup>7</sup> <http://www.ubytovani-pardubice.eu/>



### 2.8.3 Subjekt třetí

Luxusní hotelový areál v Mladých Bukách<sup>8</sup> nabízející veškeré možné služby. Stránky jsou vytvářené na míru od firmy a jsou velice precizní. Interaktivní menu, spoustu animací a netypických grafik dávají tomuto webu nezvyklé kouzlo. Po grafické stránce jistě špička na českém trhu, bohužel díky pestré grafice a velké obsáhlosti se stává značně nepřehledný. Uzounký panel s novinkami, který je o mnoho delší než zbytek stránky se nešťastně čte. Obsažené informace jsou zajímavé, ale až nepřiměřeně nasměrované pouze na hotel. O okolí a možnostech aktivit mimo hotelový areál je věnovaná samostatná pasáž na jiné části webu. Ceník je obsáhlý, ale mohl by být přehlednější a o pár stupňů větší písmo by taky neškodilo. Byl jsem lehce překvapen, když mi validátor u některých stránek zahlásil výstrahu, u takového projektu bych to opravdu nečekal.

Registrace je na první pohled jasná. Hned při otevření jsem neomylně poznal systém PREVIO. Typická kroková rezervace obsahující 4 kroky je jistě dobrým řešením pro takovýto resort. Vzhled tohoto systému v tomto případě lehce pokulhává, takovýto graficky zdařilý projekt by si jistě zasloužil netypické zpracování rezervačního formuláře.

Porovnání s mojí aplikací:

1. Vzhled a provedení stránky je výjimečné 1,
2. informativní hodnota je vyčerpávající 1,
3. rezervace PERVIO přímo v systému je určitě dostačující 1,
4. klient vidí, zda je volná kapacita při rezervaci, ale neovlivní pokoj 0.

Krásně zpracovaný web v kombinaci s rezervačním systémem nabízí snad vše, co klient potřebuje. Jedna z nejlepších prací, na které jsem v době své průzkumu narazil.

## 3 Praktická část

### 3.1 O systému

Můj systém jsem navrhl tak aby co nejvíce usnadňoval práci jak provozovatelům penzionu, tak samotným klientům. Díky dynamickému návrhu systému se v této základní verzi dá použít pro prakticky jakékoliv ubytovací zařízení bez nutnosti systém upravovat nebo do něj zasahovat.

Jak už jsem podotkl v rešerši systému, má aplikace je postavena na obsáhlejší návrhu databáze. Díky tomu je systém snadno rozšiřitelný o mnoho dalších funkcí přesně podle potřeb daného ubytovacího zařízení. Naproti tomu samotná aplikace obsahuje jen to nejnutnější pro rezervaci ubytovací kapacity v penzionu.

---

<sup>8</sup> <http://www.grundresort.cz/>

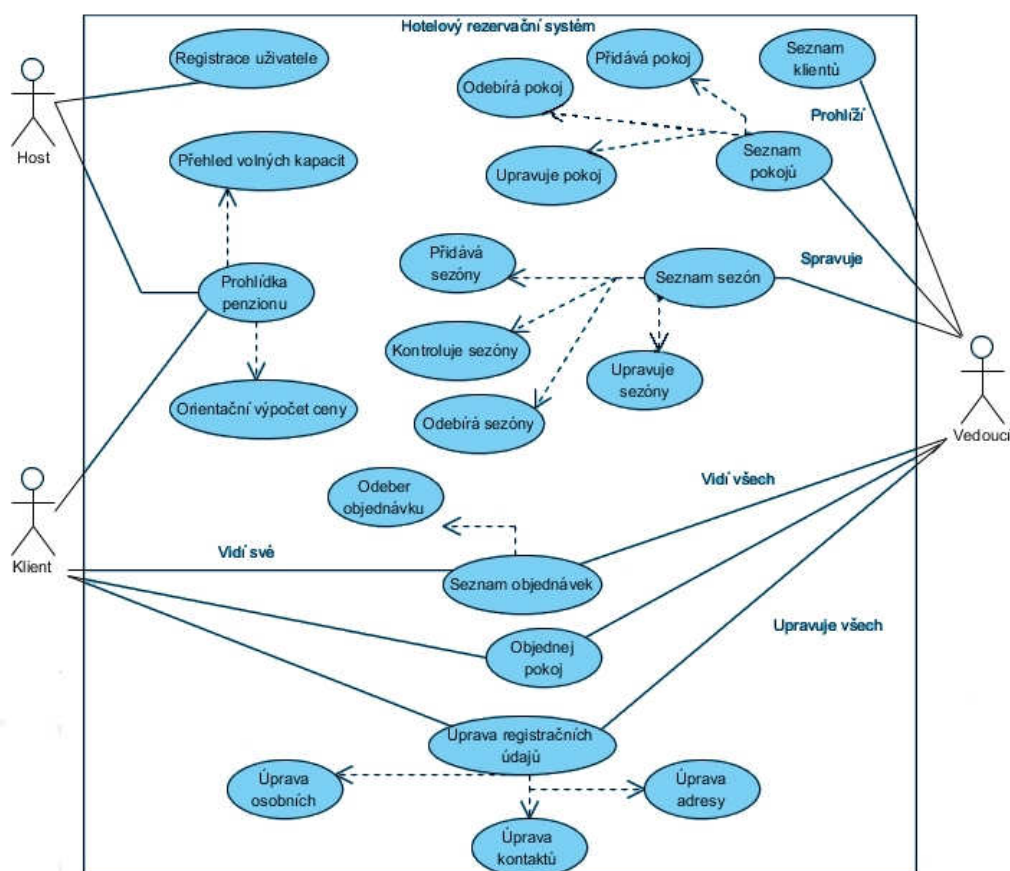
Webovou stránku, která je prezentací ubytovacího zařízení, jsem tvořil tak, aby byla přehledná, jednoduchá a přitom účelná. Zvláštní důraz jsem kladl právě na přehlednost a co největší informativní hodnotu. Samozřejmostí je i dostatečně velké písmo a výběr barev s dobrým kontrastem, protože špatně čitelná stránka klienta spíše odradí.

### 3.1.1 Uživatelé systému

Při návrhu systému jsem počítal se čtyřmi různými skupinami pro obsluhu mé aplikace. Postupně se však ukázalo, že funkce recepčního je pro potřeby středně velkých penzionů zbytečná. Přesto jsem v mé aplikaci nechal náznaky této funkce, kdyby došlo k situaci, že bude potřebná. Další skupinou pak měli být V I P zákazníci, pro které by platily jisté slevy a výhody. Opět zde mám pouze náznak, pro zprovoznění této možnosti by bylo nutné zasáhnout do několika procedur a nastavit velikosti těchto bonusů. Ve finální verzi systému pak počítám s těmito skupinami:

1. Host,
2. zákazník,
3. vedoucí.

Samozřejmě má každá z těchto skupin své specifické možnosti. Pro lepší pochopení vysvětlím celý systém na Use Case diagramu.



Obrázek 3 - Use Case diagram

Host je jakýkoliv návštěvník naší stránky. Ve svém systému jsem stavěl na předpokladu, že návštěvník webové stránky si jde pro nějaké informace. Proto má host možnost kompletní prohlídky prezentace penzionu a s tím související činnosti, jako je přehled volné kapacity nebo orientační výpočet ceny. Aby se host stal zákazníkem, musí provést registraci.

Zákazník (klient) má stejné možnosti náhledu do prezentace ubytovacího zařízení jako host. Jedná se o celkem logický krok. Nemůžeme potenciálního zákazníka odříznout od informací o penzionu a nechat ho se pro zjištění každé drobnosti odhlásit. Oproti hostu má ale širší možnosti. Vzhledem k tomu, že informace zadané při registraci se mohou měnit, případně se může stát, že je uživatel zadal chybně<sup>9</sup>, je nutné klientovi umožnit tato data upravovat. Dále je možné provádět rezervace na konkrétní pokoj a s tím spojené přehledy jednotlivých objednávek.

Vedoucí může po přihlášení měnit a mazat registrační údaje všech uživatelů systému. Pro orientaci v klientech má k dispozici přehledný seznam veškerých uživatelů. Stejným způsobem přistupuje také k objednávkám. Na rozdíl od předchozích dvou skupin nemá přímý přístup do prezentační části. Předpokládám, že provozovatel ví vše o svém penzionu, a proto jsem tento přístup vynechal. Jako jediný pak má přístup ke změnám dat v hotelové části. To znamená, že může přidávat, měnit a odebírat sezóny, ceníky a hotelové pokoje.

### **3.2 Návrh databáze**

Při navrhování databáze jsem vycházel z informací, které jsem nasbíral v době průzkumu aktuální situace na trhu. Při návrhu jsem prošel několika fázemi, než jsem se dostal ke konečné podobě. Vytvořil jsem databázi, která je obsáhlejší, než je vytvořená webová aplikace. Důvod je jednoduchý, chtěl jsem navrhnout takový systém, který by se dal snadno rozšířit a obsahoval vše, co by mohlo být do budoucna potřeba. Myslím, že se mi tento cíl podařilo celkem dobře splnit.

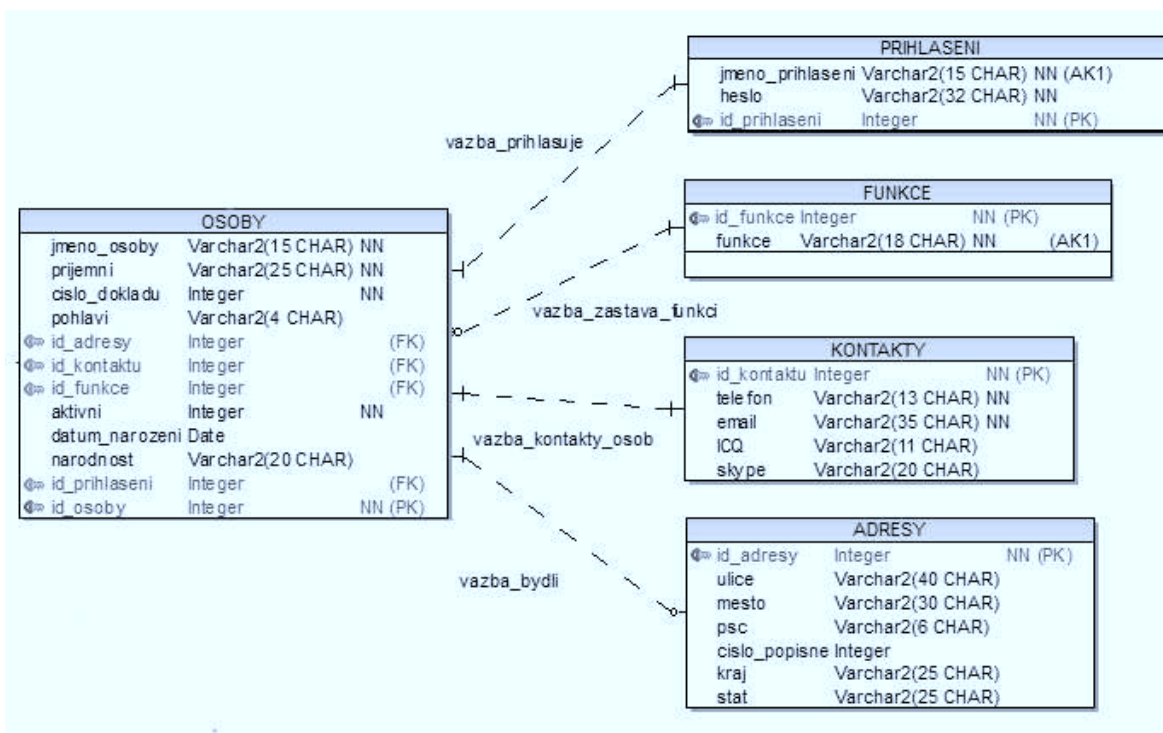
Návrh jsem pro jeho rozsáhlost rozložil na několik částí, které budu postupně procházet. Toto rozdělení vytváří poměrně ucelené bloky. Při tvorbě databáze jsem postupoval právě po těchto blocích, které jsem postupně ladil a až na konec dával do jednoho velkého systému.

---

<sup>9</sup> Vstupy do systému jsou ošetřené, ale nelze zabránit například překlepům.

### 3.2.1 Registrační část

Jak už jsem podotknul v mé práci, sbírání informací o klientech (i o těch potenciálních) je velmi důležité. Musel jsem brát ohled i na jednoduchost a rychlost registrace. V konečné práci jsem se přiklonil k jednodušší registraci a sběr dalších informací nechávám na uvážení ubytovatele mimo systém.



Obrázek 4 - ER diagram registrační část

I když je projekt stavěný pro českou klientelu, do návrhu databáze jsem vložil i údaje potřebné pro případné rozšíření o klientelu zahraniční. V mém řešení systému mám sbíraná data rozčleněná do tří tabulek. Do samotné registrace nám pak vstupuje ještě tabulka čtvrtá.

### 3.2.2 Tabulky související s registrací a jejich vazby.

**Tabulka osoby:** V této tabulce mám uložené klíčové údaje pro celý systém. Stejně tak jako je pro penzion nejdůležitější klient, tak se celá moje aplikace protíná právě v tomto klíčovém bodě. Pro potřeby vedení evidence jak českých, tak zahraničních klientů jsou společné tyto rysy: jméno, příjmení a číslo dokladu. Právě proto jsou tyto údaje vedené jako NN. Číslo dokladu může mít dva různé významy. U české klientely jde o číslo občanské průkazu a u klientely zahraniční jde pak o číslo pasu. Zjišťoval jsem z různých zdrojů formát těchto čísel a ve všech případech se jedná pouze o číslo různé délky, proto může být tato položka jako typ integer. Mohlo by se zdát, že by tento údaj mohl být primárním klíčem, ale tuto variantu jsem nevolil hned ze dvou důvodů. Jednak tuto položku vkládá do systému sám uživatel při registraci, a i když je vkládání striktně omezené, stejně nelze nikdy omezit vše. Druhý problém by se mohl skrývat v duplicitě údajů. I kdybychom připustili, že se v České republice nenajdou dvě shodná čísla

občanského průkazu, může nastat problém u klientely zahraniční. V případě, že by čísla cestovních dokladů nějaké země měla stejný počet číslic, jako u země jiné, vzniká reálná šance duplicity údajů.

Pohlaví je údaj nepovinný a čistě jako možné rozšíření pro marketingové účely. Například pro řazení odkazů na zajímavá místa a aktivity v okolí podle toho, zda je přihlášená žena nebo muž. Položka aktivní je pro případnou deaktivaci uživatelů v systému, která by mohla sloužit místo odstranění uživatelů. Osobně jsem volil odstranění, ale mohly by nastat případy, kdy nebude žádoucí uživatele plně odstranit. Poslední dvě pro nás aktuálně nezajímavé položky jsou datum narození a národnost. Obě dvě jsou nutné pouze pro evidenci zahraničních klientů a v mém systému nepoužívané.

**Tabulka přihlášení:** V této části databáze jsou uchovávány přihlašovací informace. Konkrétně se jedná o přihlašovací jméno a heslo. Délka hesla je omezena na 32 znaků, což je délka textu pro kódování pomocí funkce md5. Všechny údaje v této tabulce musí být zadané.

**Tabulka funkce:** Tabulka obsahující funkce uživatelů v systému. Je pevně zadávaná při instalaci systému a dále neměnná. Funkce jsou zvolené tak, aby vyhovovaly provozovateli penzionu. V našem případě se jedná o vedoucího a zákazníka. Dále je systém navržen tak, aby bylo možné přidat funkci recepce a V I P zákazník.

**Tabulka kontakty:** Kontaktní údaje jsou pro případnou komunikaci se zákazníkem nezbytně nutné. Pro mou aplikaci jsem volil jako povinné dva údaje, a to email a telefon. Další dva údaje jsou volitelné pro větší pohodlí zákazníka. Velikost položek je vybrána tak, aby vyhovovala formátům daných kontaktů. U emailu a skypu není pevně daná délka, proto jsem volil rozměr, který jsem považoval za dostatečně velký. Zatím co u ICQ a telefonu se dala velikost přesně určit, telefonní číslo se skládá z předvolby a devíti čísel. To znamená znak +, tři číselné hodnoty a pak zbylých devět, proto je tato položka omezena na 13 znaků. U ICQ a podobných komunikátorů je tento formát nastaven na tři čísla, pomlčku, další tři čísla opět pomlčka a konečná tři čísla, takže v konečném součtu 11 znaků.

**Tabulka adresy:** U adresy je možné ponechat všechny údaje nulové, samozřejmě až na samotné identifikační číslo, které už z principu jako primární klíč nemůže zůstat prázdné. Vycházím z údajů nutných pro evidenci klientů, kde je adresa nutná pouze u českých klientů. Proto je systém udělán tak, aby zahraniční klient nemusel mít žádnou položku v tabulce adresy, ale mohli jsme mu jí přidat v případě, že se o něm nějaké adresní informace dozvíme.

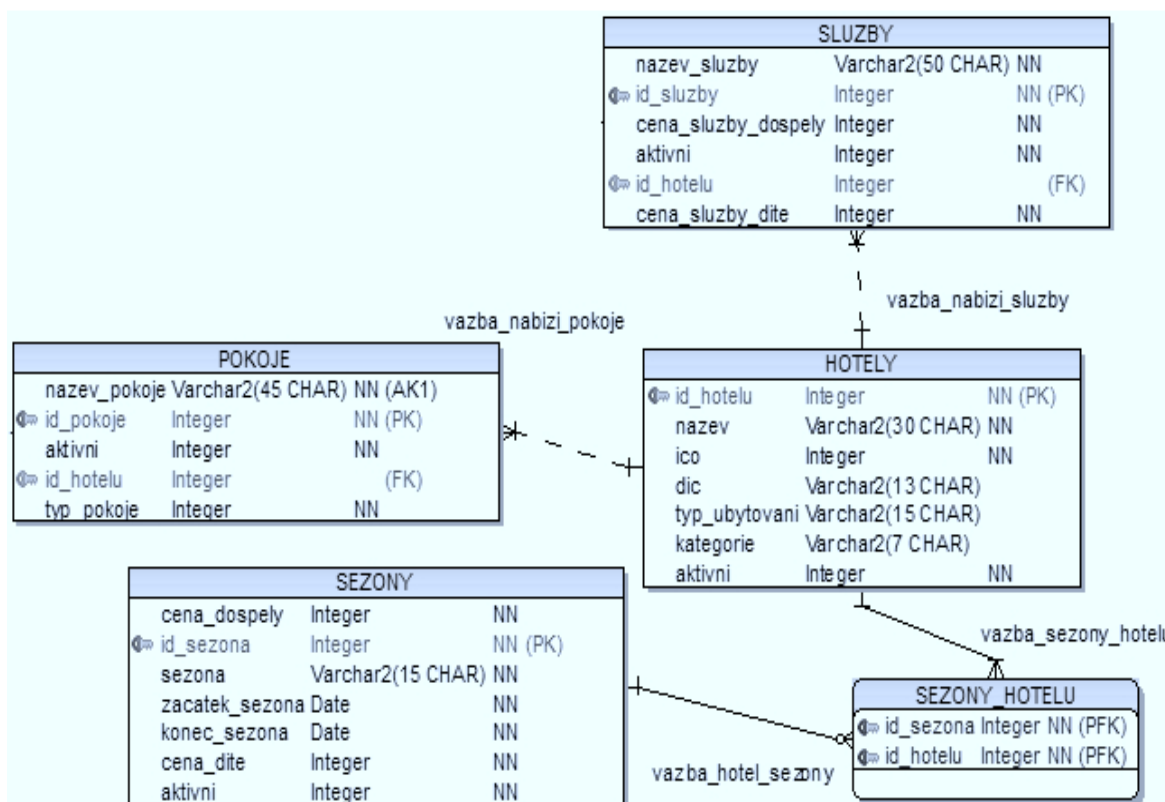
**Vazby mezi tabulkami:** Vazba mezi osobami a přihlášením je 1:1 což znamená, že na jeden záznam v tabulce osoby připadají právě jedny přihlašovací údaje.

U vazby mezi osobami a funkcemi je věc trochu jiná. Osoba zastává pouze jednu funkci, ale funkce může připadat více osobám anebo taky žádné. Příkladem jsou funkce

zákazník, která náleží více uživatelům systému. Funkci vedoucí zastává pouze jeden uživatel a funkci recepcce. U kontaktů je situace stejná jako u přihlašovacích údajů. Poslední vazbou je vazba na adresy, kde adresa musí odpovídat jednomu záznamu v tabulce osoby, ale osoba nemusí mít adresy zadanou.

### 3.2.3 Hotelová část

V druhé části databáze se zabývám sběrem dat týkajících se hotelu. Asi nejpodstatnější tabulkou této části je tabulka sezony. V ní jsou uchovávána data jednotlivých sezon a ceník s tím spojený. V původním návrhu jsem přemýšlel nad vazbou tabulky hotel na kontakty a adresy, ale nakonec se tyto vazby ukázaly jako zbytečné. Adresa a kontakt na hotel budou pouze na webových stránkách.



Obrázek 5 - ER digram hotelová část

### 3.2.4 Tabulky související s hotelem a jejich vazby

**Tabulka hotely:** V této tabulce je uloženo několik informací o hotelu. Při návrhu jsem počítal s tím, že by se počet hotelů v systému mohl zvýšit, proto jsem do projektu zahrnul i tuto tabulku. Položka id hotelu, která je primárním klíčem této tabulky, slouží k identifikaci jednotlivých položek. Název by v případě více hotelů sloužil k rozlišení ubytovacích zařízení, délku jsem volil 30 znaků, protože není možné přesně určit, jaká délka bude nutná. IČO a DIČ jsou informace o ubytovacím zařízení, které by mohly být do budoucna použity pro fakturaci ubytování a služeb. Typ ubytování by sloužil v případě, že by se do systému mělo přidat další ubytovací zařízení jiného typu než penzion. Kategorie pak určuje označení, které známe v podobě hvězdiček. Aktivace umožňuje držet hotel mimo systém, než budou všechny potřebné položky nakopírovány do databáze.

**Tabulka sezony:** Zde jsou uloženy informace o jednotlivých sezonách hotelu. V praxi má ubytovací zařízení více sezon s různými cenami. V České republice se nejčastěji užívají tři sezony: letní, zimní a silvestrovský termín. U každé sezony je třeba znát datum začátku sezony a datum jejího konce. Položka sezona, do které se zapisuje název, je volena na délku 15 znaků, což by mělo být více než dostačující. Další nutné položky jsou samozřejmě ceny, které se liší podle jednotlivých sezon. Do návrhu systému jsem zahrnul možnost nastavit dvě různé ceny, a to za dospělého a za dítě. Osobně v práci využívám obě dvě, ale podle mého průzkumu většina středně velkých penzionů poskytuje slevu na ubytování pro dítě jen velmi zřídka. Důvod je jednoduchý, dítě při ubytování má vesměs stejné nároky jako dospělý člověk. Energií je spotřebováno zhruba stejně a úklid po rodině s dětmi bývá mnohem náročnější. U služeb je situace jiná. Tam se slevy většinou poskytují. Takže položka cena dítě by nemusela být v databázi vůbec obsažena, ale při běhu by se absence této položky mohla ukázat jako chyba. Poslední položka je aktivita sezony. Vzhledem k tomu, že ubytovatel má možnost zadat sezony na dlouhou dobu dopředu, lze tímto způsobem omezit, jak dlouho dopředu bude možné rezervovat ubytování. Případně se tato položka dá využít u propadlých sezon. Bud může být sezona odstraněna z databáze úplně, nebo pouze deaktivována z důvodů archivace. V mém návrhu počítám s odstraňováním sezon.

**Tabulka sezony hotelu:** Mezi tabulkami hotely a sezony nám vzniká vazba M:N. Hotel má více sezon a v případě, že budeme mít více hotelů, může vzniknout situace, kdy se sezony u jednotlivých hotelů setkají se stejnými daty začátku a konce.

**Tabulka pokoje:** Každé ubytovací zařízení má logicky více pokojů. Pokoj v systému identifikuji pomocí id pokoje. Tato položka je zároveň primárním klíčem. K určení položky pro uživatele používám název pokoje. Délku jsem volil možná až přehnaně velkou, ale není možné určit, jak budou pokoje v hotelu pojmenované a nechtěl jsem ubytovatele limitovat. Název pokoje je unikátní údaj, ale používat ho jako primární klíč nepovažuji za dobré, protože ho zadává ubytovatel. Každý pokoj spadá pod určitý hotel, proto je zde obsažena položka identifikace hotelu jako cizí klíč. Poslední položka je typ pokoje. V mém systému určuje počet lůžek, které jsou v daném pokoji.

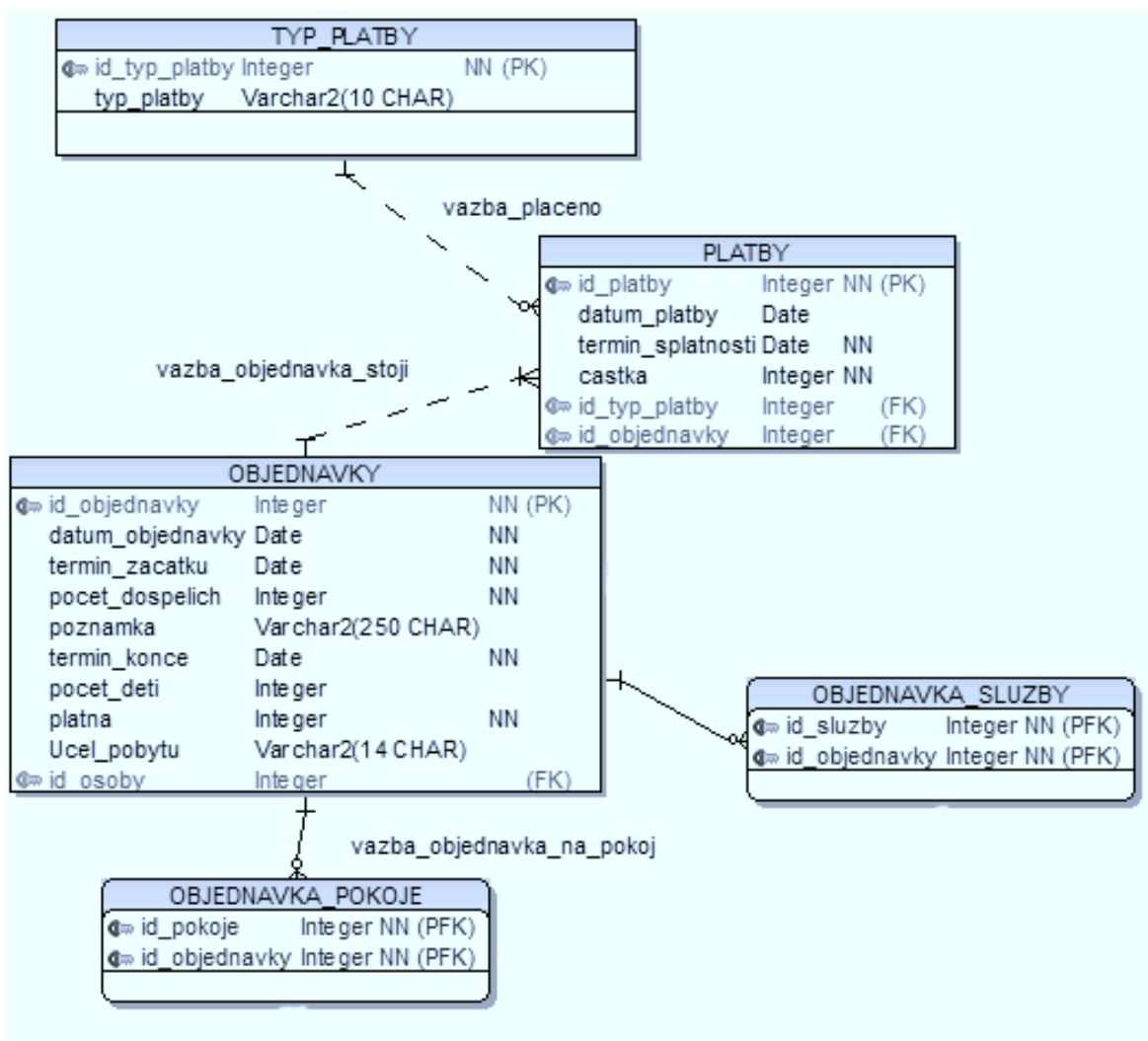
**Tabulka služby:** Poslední tabulka v této části systému. Je podobná jako entita pokoje, ale navíc obsahuje cenu za službu pro dospělého klienta a cenu pro dítě. U této položky se jedná o pouhé rozšíření a není v systému dále používána.

**Vazby v hotelové části:** Vazba mezi hotelem a pokoji udává, že pokoj spadá pouze do jednoho konkrétního hotelu a hotel má více pokojů. Hotel musí mít vždy aspoň jeden pokoj. Stejná situace je u služeb. Vazba na sezony je pak trochu jiná. Jedná se o vazbu M:N kdy hotel má víc sezon a sezona může spadat pod více hotelů. Z důvodů, aby nevznikaly duplicity záznamů v případě, že by se sezony jednotlivých hotelů shodovaly.



### 3.2.5 Objednávková část

Třetí a poslední část je určena pro shromažďování údajů o rezervacích pokojů a služeb. Jedná se o nejkomplicovanější úsek celého návrhu. Všechny rezervace v systému jsou ukládány do tabulky objednávky, která je navázána na tabulku plateb. Na tuto část by bylo možné navázat účetnický systém pro fakturaci.



Obrázek 6 - ER diagram objednávky

**Tabulka objednávky:** Tato tabulka je srdcem celé této části. Objednávky se vztahují na služby a pokoje. K identifikaci v systému používám id objednávky jako primární klíč. Položka datum objednávky nám v systému určuje dobu, kdy byla objednávky vystavena. Termín začátku a konce určují, od kdy do kdy bude daná služba nebo pokoj k dispozici pro klienta. Tyto položky nesmí být nulové. Počet dospělých a dětí určuje pro kolik lidí je objednávka vystavena. Počet dětí nemusí být zadán. Poznámka je v systému pro zadání speciálních požadavku při objednávce, jde tedy o dlouhý textový řetězec, proto takto velký rozměr. Platnost rezervace zadává ubytovatel při přijmutí zálohy. Účel pobytu je pak nutný při objednávce pokoje.



**Tabulka platby:** V této tabulce jsou uložena data o platbách. Při vytvoření objednávky je zde ukládána částka, jako typ integer, kterou má uživatel uhradit. Doba splatnosti je datum, který se vytváří z aktuálního data objednávky, ke které přičítám 7 dní, do kdy musí být záloha zaplacená. Datum platby zadává vedoucí při přijetí zálohy, a tím se stává rezervace platná. Částka je počítána podle počtu osob a dnů pobytu. Mám zde také dva cizí klíče, pro určení o jaký typ platby se jedná a na jakou objednávku je platba vystavena.

**Tabulka typ platby:** Slouží pro rozlišování o jakou platbu se v systému jedná. Zda jde o zálohu, hlavní platby, doplatek a podobně. Tato tabulka je při instalaci systému pevně nastavena a uživatelé systému nemají možnost jí měnit.

**Tabulky objednávka pokoje a objednávka služby:** Vzhledem k tomu, že v obou případech vznikají mezi objednávkou a službou, případně pokojem vazba více na více, je nutné vytvořit tyto dvě entity pro funkčnost databáze. Obě obsahují pouze primární cizí klíče tabulek, které spojují.

**Vazby v rezervační části:** Vazba mezi typem platby a platbami určuje, že platba má vždy jen jeden typ, ale typ může náležet více platbám nebo žádné pokud není v systému vytvořená objednávka. U vazby objednávka stojí je pak situace obdobná s tím, že při vytvoření objednávky musí být v platbách minimálně jedna platba a ta náleží právě jedné objednávce.

### 3.2.6 Vazby mezi jednotlivými díly

Výše popsané části jsem po jejich vytvoření spojil do jednoho celku. Většinu vazeb jsem již popsal, takže nám zbývá několik, které je třeba objasnit. První je vazba mezi osobami a objednávkami. Vzhledem k tomu, že osoba nemusí mít vytvořenou žádnou objednávku, nebo jich může mít libovolný počet, je zde vazba nula až N, ale na druhou stranu rezervace se vytváří pro jednoho konkrétního jedince, takže je zde vazba 1. Mezi registrační a hotelovou část jsem dřív vazby měl, ale považoval jsem je za zcela zbytečné, tak se v mém konečném návrhu systému nevyskytují. Mezi rezervační částí a hotelovou jsou hned dvě vazby. Jedna pro pokoje a druhá pro služby. Vazby jsou zde M:N, takže jsou použity i tabulky pro vytváření objednávek na pokoje a služby, které jsem již popisoval výše.

## 3.3 Vytvoření webové aplikace

K tvorbě samotné stránky jsem využíval jednoduchý program jménem PSPad, ve kterém jsem psal veškeré zdrojové kódy a to od těla samotné stránky, až po PHP příkazy. Pro lepší orientaci bylo nutné vytvořit adresářový strom pro jednotlivé části systému. Tento strom bude podrobně popsán níže. Grafickou podobu stránky jsem dělal pomocí kaskádových stylů.

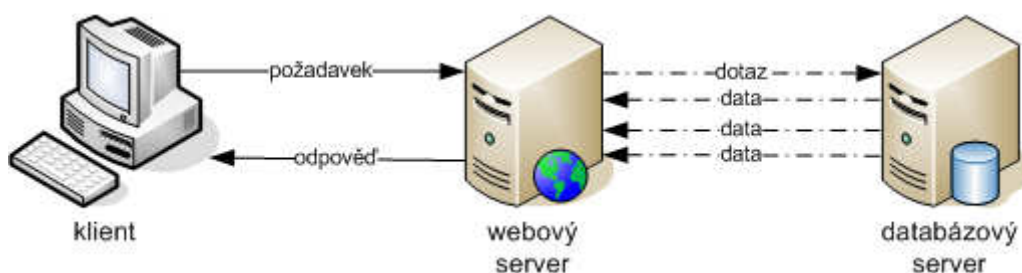
### 3.3.1 Principy tvorby webové aplikace

Při vytváření aplikace jsem se řídil několika pravidly pro větší bezpečnost a stabilitu systému. Nejvíce chyb vzniká při komunikaci webové aplikace s databází. Většinu dat do systému zadává přímo uživatel, díky špatně vloženému řetězci může vzniknout chyba. Bud neúmyslná, nebo v případě snahy o napadení aplikace úmyslná.

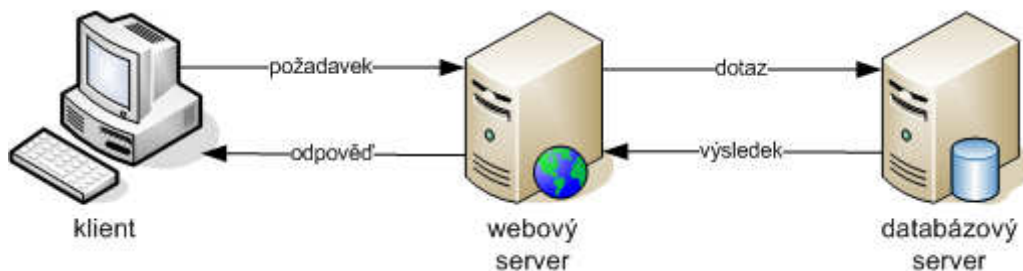
**Snaha o jemné dělení:** U tvorby jsem se snažil rozdělit jednotlivé moduly, procedury a dotazy na co nejjednodušší celky. U každého programování se doporučuje, aby jedna metoda dělala jen jednu věc, a to pořádně. Takové dělení je výhodné pro přehlednost, správu a změny v systému a zajišťuje celkovou flexibilitu aplikace.

**Dotazy na databázi:** Při vznášení sql dotazů je nutné být velmi opatrný, aby nedošlo k chybě nebo dokonce ztrátě dat. Předcházet tomu lze hned několika způsoby (JANOVSKÝ, 2010). V mé práci ošetřuji všechny vstupy pomocí speciálních kontrolních funkcí. Díky tomu jsem schopný omezit formáty zadávaných dat i jejich délky. U zasílaného počtu znaků používám kromě těchto metod také omezení vstupních prvků ve formulářích. Dalším způsobem ošetření vstupu je používání výběrových selektů v html kódu. Tím dáme uživateli jasný výběr dat, která mohou být vložena, a máme jistotu, že nevznikne chyba. Kdyby selhaly všechny předchozí způsoby zabezpečení, jistím svůj systém ještě jednou podstatnou věcí. Veškeré dotazy, od kterých očekávám výstupní data, směřuji na pohledy. V případě, že by se útočníkovi podařilo vpravit nějaký škodlivý kus textu do dotazu a chtěl tím například smazat data v tabulce, tak neuspěje. Pohledy jsou vyvářeny pouze pro prohlížení dat a nejdou přes ně provádět jakékoliv úpravy.

**Vkládání dat do databáze:** Pro uložení uživatelských informací do databáze používám vždy jen procedury. Při použití pouhých DML příkazů by se opět otevíral zbytečný prostor pro útočníky. Po odeslání zkontrolovaných dat do procedury se už samotné příkazy provádějí v jejím těle a ne za pomoci komunikace se serverem. Další nespornou výhodou jsou mnohem nižší náklady na komunikaci mezi webovým a databázovým serverem (LEHOTSKÝ, 2008).



Obrázek 7 - Ukázka komunikace pomocí dotazů



Obrázek 8 - Ukázka komunikace pomocí procedur

**Omezení přístupu uživatelů:** Je jasné, že každý uživatel systému musí mít přístup jen do určité části. Proto jsem ve své práci omezoval vstupy na jednotlivé soubory. K tomu mi sloužila tabulka funkce, kde má každá funkce svou identifikaci. Při vstupu na stránky, které jsou pouze vedoucího, kontroluji, zda je opravdu přihlášen uživatel s odpovídajícím názvem. Pokud ne, dochází k přesměrování na přihlašovací stránku. Vstup do sekce pro zákazníky jsem nijak neomezoval. Pokud bude uživatel přihlášen, může do všech souborů, které jsou pro něj určeny. Když nastane pokus o vstupu u nepřihlášeného uživatele, neprojde podmínka o přihlášení v souboru LogIn.php a dojde přesměrování zpět na úvodní stránku.

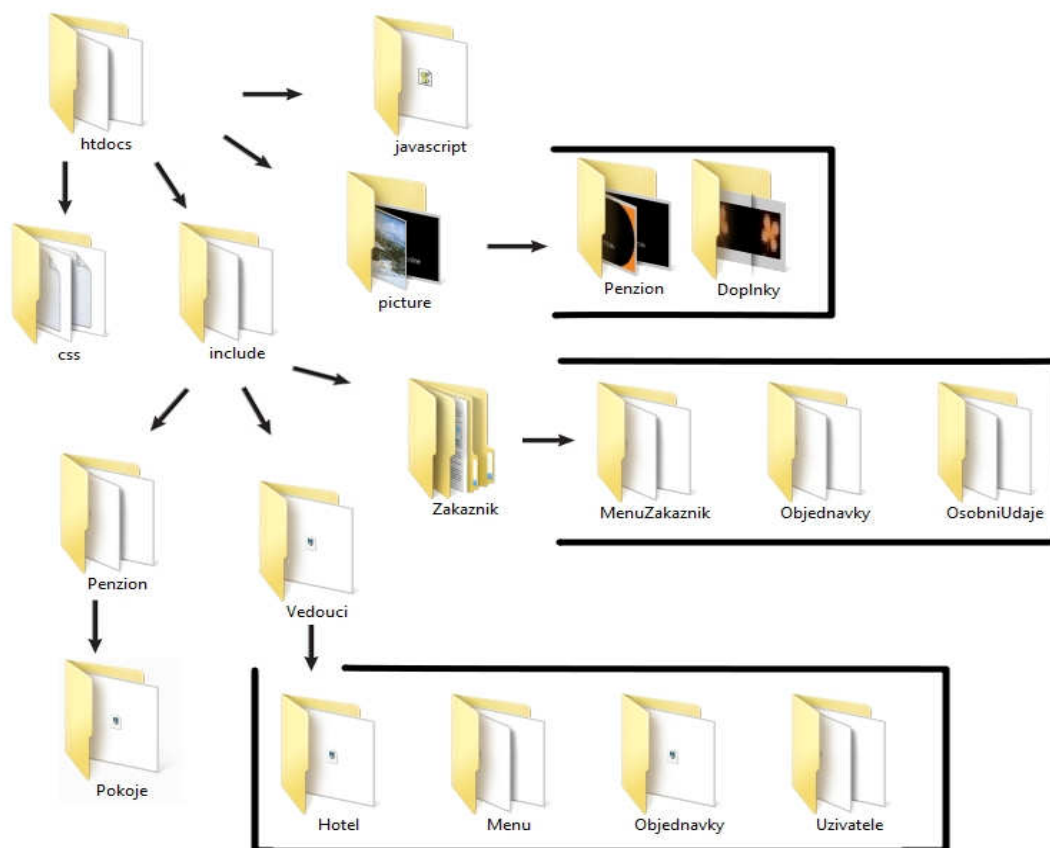
Za zmínku ještě stojí rozhodování a výběry, pro které používám funkci vracející číselnou hodnotu, a tím zase omezují komunikaci mezi webovým a databázovým serverem. Místo několika dotazů v každé podmínce vznesu pouze jeden a jeho vyhodnocení provedu v těle funkce. Ukázky procedur, funkcí a pohledů z mého systému jsou součástí této práce jako přílohy.

### 3.3.2 Soubory a složky

Mnou vytvořený adresářový strom třídí jednotlivé soubory podle místa, do kterého spadají. V první úrovni jsem soubory rozdělil podle jejich typu. Mám zde složky CSS kde ukládám soubory starající se o úpravu vzhledu pomocí kaskádových stylů. Dále zde mám složku picture, která se dále dělí na dvě podskupiny. Zde ukládám veškeré obrázky, které jsou ve webové prezentaci, a také potřebné doplňky, jakou jsou odrážky a podobné. Javaskript je vymezena na samostatné, skripty použité v mém projektu a v include jsou jednotlivé soubory, které vkládám do stránek aplikace.

V druhé úrovni pak člením podle uživatelů, kterým jsou soubory určeny. V položce penzion mám všechny informace týkající se mého penzionu. Ve složce vedoucí pak věci, které jsou potřeba pro administraci stránek a v zákazníkovi položky, které jsou potřeba, aby si zákazník mohl vytvářet a prohlížet své objednávky a podobně.

Třetí úroveň je pak vymezena pro konkrétní obslužné soubory pro jednotlivé uživatele. U vedoucího to je hotel, pro obsluhu hotelu, objednávky pro kontrolu a vytváření objednávek a podobně. U zákazníka je situace stejná.



Obrázek 9 - Adresářový strom

### 3.3.3 Přehled jednotlivých souborů

Když opomenu výše popsané složky s obrázky a CSS styly, je můj systém rozdělen do čtyř skupin. První jsou nezařazené soubory, které slouží buď hostům v systému nebo jsou společné pro všechny skupiny. Další skupiny jsou zařazené do složek podle svého účelu. Jedná se o složky penzion, vedoucí a zákazník. S popisem začnu právě u kořene.

V kořenovém adresáři jsou mimo dalších složek dva hlavní soubory, `LogIn.php` a `index.php`, do obou těchto souborů se dále vkládají moduly podle potřeby uživatele.

**LogIn.php.** Stránka sloužící pro přihlašování uživatelů systému. Při vstupu na stránku webové aplikace je uživatel přesměrován z `index.php` právě na tuto stránku. Tato položka obsahuje vložené menu pro prezentaci penzionu, modul pro přihlašování a odkazy na styly. Při zdání přihlašovacích údajů je zde zkontrolováno, zda existuje daný uživatel a zda jsou jeho přihlašovací údaje správně zadány. Pokud ano, jsou příslušná data nastavena do proměnné `$_SESSION` (MACEK, 2003) a nastane přesměrování na soubor `index.php`.

**Index.php:** Hlavní soubor celé webové části aplikace. Obsahuje vloženou funkci pro vykonávání příkazů směřovaných na databázi. Je zde také odhlašovací modul a includované menu podle funkce uživatele, který je přihlášen. Na začátku tohoto souboru si také hlídám proměnou přihlášení, kterou mám uloženou v session. V případě, že by tato proměnná nebyla nastavena, dojde k automatickému přesměrování na LogIn.php. Stejná situace pak nastává i po stisku tlačítka pro odhlášení.

Nyní přejdu do adresáře include, kde je několik souborů zajišťujících registraci, kontrolu dat, přihlášení a podobné činnosti sloužící převážně pro hosta systému, nebo jsou pro uživatele systému společné.

**Funkce.php:** V tomto souboru mám zapsané dvě funkce pro spojení mezi webovou částí a databází. Jako první popíši funkci „dotaz“, která slouží pro vznesení dotazu na databázi a očekává množinu dat jako výsledek.

```
function dotaz($sql) {  
    $c = oci_connect('Pjupju', 'poldis', '///localhost/XE');  
    $s = oci_parse($c, $sql);  
    oci_execute($s);  
    oci_fetch_all($s, $res);  
    oci_close($c);  
    return $res;  
}
```

Tato funkce přijímá jako parametr nějaký sql dotaz. Například select na tabulku nebo na pohled. První řádek složí k vytvoření spojení na databázi, kde je prvním parametrem uživatelské jméno, druhým heslo pro připojení k účtu Oracle a třetím je určení databáze, ke které se chceme připojit. Oci\_connect nám pak vrací identifikátor spojení, které je potřeba pro další operace (The PHP Group, 2011).

U příkazu oci\_parse připravujeme sql příkaz na provedení. Parametry této funkce jsou identifikátor připojení k databázi, vytvořený pomocí oci\_connect a sql příkaz, který jsme zaslali jako parametr. Funkce nám vrací prohlášení o možnosti provedení dotazu na databázi (The PHP Group, 2011).

Dalším krokem je funkce oci\_execute (The PHP Group, 2011), která provádí výpis příkazu, který předtím připravila oci\_parse. Tato metoda vrací booleanovskou hodnotu o úspěchu. S tímto příkazem je nutné spojit funkci oci\_fetch\_all (The PHP Group, 2011), která vkládá data výsledku do dvourozměrného pole. Prvním parametrem této metody je připravený příkaz, který byl vykonán předchozí funkcí a druhým pak proměnná pro uložení výsledku.

V posledních dvou krocích je pak zavřené spojení pomocí metody oci\_close s parametrem, ve kterém je vytvořené spojení na databázi a navrácení výsledné hodnoty.

Druhá funkce v tomto souboru slouží pro volání procedur, které vkládají a manipulují s daty v databázi. Tato metoda se od předchozí popisované liší jen nepatrně.

```
function dotaz_vloz($sql)
{
$c = oci_connect ('Pjupju','poldis','//localhost/XE');
$s = oci_parse($c, $sql);
oci_execute($s);
oci_close($c);
}
```

Jak je vidět v ukázce, je zde vynechána část pro vkládání příkazu do pole. Protože od této funkce neočekáváme žádnou návratovou hodnotu, je zde pouze otevření spojení, připravení příkazu, jeho provedení a zavření spojení.

**Kontrolni.class.php:** Pro kontrolu většiny zadávaných dat od uživatele používám několik kontrolních tříd, pomocí kterých limituju délky a formáty těchto informací. Tento soubor je určen speciálně pro registrační data. Každá z kontrolních tříd obsahuje dvě pole, jedno pro data, která projdou kontrolou a druhé pro případ vzniknutí chyby. Dále pak je obsaženo několik funkcí, kterým zasílám data jako parametry. Tyto metody jsou si v jádře velmi podobné, proto představím pouze jednu z nich.

```
function RegJmenoPrijemni($poslanaData,$odkud){
    $poslanaData = trim($poslanaData);
    if($poslanaData==' ' or $poslanaData=='Nevyplněné!')
    {
        $this->chyby[$odkud]='Nevyplněné!';
    }
    else if ctype_alpha($poslanaData)==false or
    $poslanaData=='Pouze písmena!')
    {
        $this->chyby[$odkud]='Pouze písmena!';
    }else
    {
        $this->data[$odkud]=$poslanaData;
    }
}
```

Jak je vidět výše, funkce RegJmenoPrijemni kontroluje zadávané jméno a příjmení. Jako parametry posílám dva údaje. První jsou samotná zasílaná data a druhý je název, odkud se tyto informace posílají. Nejprve jsou z údajů pomocí trim (The PHP Group, 2011) ořezány bílé znaky. Pak kontroluji, zda zasílaný parametr není prázdný nebo zda není vyplněný chybovou hláškou. Pokud je splněna první podmínka kontroluji, zaslano proměnou pomocí metody ctype\_alpha, což mi zaručí, že zadané jméno nebo příjmení obsahuje pouze znaky. Když je podmínka vyhodnocena kladně, ukládám do chybového pole hlášku o chybě. V opačném případě jsou pak data uložena do pole pro zaslání na databázi.

V tomto souboru také kontroluji heslo na shodnost, přihlašovací jméno na délku, číselné hodnoty na délku, numerické znaky a vyplnění. Případně specifické položky jako je email nebo telefonní číslo na správně zadaný formát. U emailu je to například jmeno@domena.cz a u telefonního čísla pak na formát +420123456789.

**Registrace.php:** Zde sbírám jednotlivé údaje o uživateli a ty dále zpracovávám a ukládám do databáze. Mám zde includovaný soubor kontrolní třídy pro jednotlivá data a několik pomocných proměnných pro ověřování správnosti samotných dat. Dále pak dvě pole pro možnosti výběru a pole pro vkládání zkontrolovaných dat do procedury. Samotná pole pro zadávání dat jsou v jednom velkém formuláři, který je odeslán pomocí metody POST. Při odeslání jsou data kontrolována několika způsoby, které popíší u jednotlivých položek.

Jako první uživatel zadává přihlašovací jméno, u kterého je nejprve kontrolována délka, kterou jsem omezil na 5 a více znaků, vyplnění daného pole (tento údaj je povinný) a nakonec je vznesen požadavek na databázi v podobě funkce ověřující přihlášení. S touto položkou jsou velmi úzce spojeny další dvě položky, a to jsou heslo a ověření hesla. Zde je nejprve kontrolována délka hesla, která je nastavena na 6 a více znaků z důvodu bezpečnosti a ověření zda jsou hesla shodná. Když tato data projdou první kontrolou, jdou spolu s přihlašovacím jménem do již zmiňované funkce. Ta nám vrací číselné hodnoty podle toho, zda se v systému vyskytuje zadané jméno a heslo. Pokud, ano je vrácena informativní zpráva o chybě.

V tomto adresáři jsou ještě obsaženy soubory pro přihlášení a odhlášení, které nejsou nijak složité ani zajímavé. Dále si ukážeme složku penzionu. V tomto adresáři jsou uloženy prezentační materiály k ubytovacímu zařízení. Většina z nich jsou jen obyčejné informativní stránky bez vstupu do databáze až na dvě výjimky.

**Ceník.php:** V souboru je obsažena tabulka, kterou získávám pomocí sql dotazu na pohled p\_sezony\_na\_hotelu, který mi vrací informace o začátku a konci sezóny, názvu a ceně za dospělého a dítě. Pod touto tabulkou je vložen formulář pro výpočet ceny pobytu. Zde je možné zadat datum příjezdu a odjezdu, počet dospělých a dětí a účel pobytu. Z těchto informací je pomocí funkce pro výpočet ceny vrácena celková cena pobytu, ze které dále počítám zálohu. Nyní si ukážeme pohled p\_sezony\_hotelu:

```
CREATE VIEW p_sezony_na_hotelu AS
  SELECT sezony.id_sezona, sezona , zacatek_sezona, konec_sezona,
  sezony.aktivni, cena_dospely,cena_dite , hotely.id_hotelu, nazev
  FROM sezony, hotely, sezony_hotelu
  WHERE sezony.id_sezona=sezony_hotelu.id_sezona AND
  hotely.id_hotelu=sezony_hotelu.id_hotelu;
```

**PokojePenzion.php:** Jak již napovídá název, jsou pod touto položkou informace o ubytovací kapacitě a s tím spojené informace o obsazenosti. Pomocí dotazu, na pohled vytvořený nad tabulkou pokoje, získávám informace o názvu, kapacitě a popisu pokoje. Při načtení je položka o obsazenosti prázdná. Po zadání data počátku a konce pobytu je nám zobrazeno, zda je daný pokoj v tomto termínu k dispozici.

Nyní se podíváme na složku, která je k dispozici zákazníkovi. V ní je obsaženo více adresářů pro lepší orientaci. Začneme u položky OsobniUdaje.

Zde jsem sjednotil veškeré informace o registračních údajích klienta s možností jejich změny. Jsou obsaženy dva typy souborů. První je přehled registračních údajů, ve kterém jsou přehledně vypsána veškerá registrační data uživatele, druhý typ souboru jsou moduly na úpravu těchto dat. První případ není zajímavý, proto ho vynechám a z druhé skupiny popíši pouze jeden případ z důvodů jejich velké podobnosti.

**UpravaOsobni.php** Jedná se o formulář, do kterého je překopírovaná část registrace. V tomto případě se jedná o položku osobní. Odeslaná data jsou pak kontrolována pomocí kontrolní třídy, překopírovaná do připraveného pole a v případě nenalezení chyb jsou zaslaná do databáze pomocí procedury na úpravu osobních údajů.

Pro uskladnění souborů pro vytváření a obsluhu rezervací pro zákazníka používám složku Objednavky. Jako první projdeme přehled a odebrání objednávek. Poté projdeme jednotlivé kroky rezervace.

**PrehledObjednavek.php:** Při otevření stránky přehledu objednávek vidíme tabulky s jednotlivými objednávkami a pod tím celkový součet cen za všechny objednané pokoje. O vypsání tabulky se stará dotaz na pohled p\_prehled\_objednavek, ke kterému je přidána podmínka pro vypsání objednávek právě přihlášeného uživatele. To zaručuje klauzule WHERE a samotný identifikátor uživatele získávám ze SESSION. O odebrání objednávek pomocí tlačítka umístěného v tabulce se pak stará procedura odstraň rezervaci. Nejprve je vznesen dotaz, zda má být rezervace odstraněna a po potvrzení se provádí již zmiňovaná procedura, která postupně odstraňuje data z tabulek objednávky, objednávky pokoje a platby. Pro zjištění celkové dlužné částky se stará dotaz na pohled, který je následně vypsán.

**VytvořeníObjednavky.php:** základní kámen celého rezervačního formuláře. Sjednocuje veškeré kontroly a do svého těla includeje jednotlivé kroky rezervace. Každý z kroků je samostatný soubor. Toto rozdělení jsem volil pro lepší orientaci a úpravu. V prvním kroku je vložen fieldset s formulářem a několika selecty na volbu data a příjezdu. Když uživatel vyplní a odešle tato pole, přichází kontrola těchto dat. O to se stará funkce\_sezony\_objednavky, která vrací hodnoty 0 až 4. Nula pro úspěch a jedna až čtyři pro situace, které mohou vzniknout. Jsou to: zadání propadlého data, zadání data mimo sezóny, zadání data mimo ceník a větší datum odjezdu než příjezdu. Pokud vše projde v pořádku, ukládám data do proměnných a přistupuji k dalšímu kroku.

Následující modul obsahuje volbu počtu osob. Je možné zvolit počet dospělých a dětí ze selectu a účel pobytu. V horní části modulu je includovaná tabulka s přehledem volných a obsazených pokojů v požadovaném datu z prvního kroku. Po zvolení osob a účelu jsou odeslaná data vložena do proměnných a přechází se k dalšímu kroku. V tomto kroku se provádí výběr pokojů. U tohoto výběru je opět v horní části vidět, jaké pokoje jsou volné a v dolní pak možnost výběru pokoje. Výběr požadovaného pokoje je prováděn



pomocí selectu. Nejprve je funkcí zjištěno, jaké pokoje jsou volné. Do funkce je zaslán počet osob a datum příjezdu a odjezdu. Návrátová hodnota je pak nulová v případě nenalezení vhodného pokoje, nebo typ pokoje, který je volný. To znamená, že v případě zadání jedné osoby se projdou všechny jednolůžkové pokoje. Když je nalezen nějaký volný, vrátím hodnotu 1, když ne prochází se dvoulůžkové a tak dále. Procházeny jsou pokoje maximálně o dvě lůžka větší, než je požadováno. Následně je dalším dotazem zjištěn seznam pokojů, které odpovídají danému typu a do selectu jsou umožněny na výběr pouze ty pokoje, které jsou momentálně volné.

Po potvrzení výběru pokoje se dostáváme do závěrečné rekapitulace. Jednoduchý výpis proměnných, které uživatel zadával v průběhu rezervace. Po potvrzení jsou veškeré tyto informace uloženy pomocí procedury na vytváření objednávek do databáze.

Ve složce pro zákazníky jsou pak ještě jednotlivá menu, ale tyto soubory jsou velice primitivní a nemá smysl se jimi zabírat. Proto nyní přejdu do poslední složky a tou je vedoucí. Velká část souborů u vedoucího funguje obdobně jako u uživatelů s tím, že vedoucí zadá, pro jakého uživatele má být daná položka vytvořena nebo upravena. Začnu u složky hotelu, kde jsou uloženy moduly pro úpravu pokojů a sezon.

**SeznamPokoju.php:** Soubor je rozcestníkem celé zprávy pokojů v ubytovacím zařízení. Základem je tabulka s přehledem všech pokojů v systému. Po výběru pokoje je možné pokoj upravovat a odebírat. Pro tyto funkce jsou připravené dva speciální moduly s formulářem, které přijímají data. Metodou POST je pak odesílají do souboru SeznamPokoju.php, kde jsou zkontrolovány pomocí kontrolní třídy a vloženy do procedury, která se pak starají o vložení dat do databáze.

**SeznamSezon.php:** Je soubor, který je funkcemi a vzhledem velice podobný předcházejícímu. V horní části je ve formuláři pole pro výběr sezon s tlačítky jaká operace se má vykonat a pod tímto formulářem je tabulka s přehledem zadaných sezon v systému. Po výběru z tabulky je id požadované sezony vloženo do výběru a můžeme pokračovat buď úpravou dané sezony, nebo jejím odebráním. Při odebrání je uživatel vyzván k potvrzení a poté je požadovaná položka odstraněna pomocí příslušné procedury. U úpravy pokoje je situace stejná akorát po výběru položky a stisku tlačítka pro úpravu je nad formulář z výběrem includovaný modul se souboru. Před načtením modulu je proveden select pro načtení informací o sezoně a data jsou vložena do formuláře pro úpravu. Po potvrzení operace jsou data zkontrolována pomocí kontrolní třídy, datum začátku a konce je převedeno do datového formátu funkcí date a je volána funkce pro překrytí těchto dvou termínů s jinou sezonou. V případě, že tato funkce vrátí nulu, je volána procedura pro vložení dat do databáze. Poslední možnost je přidání nové sezony. Postup je stejný jako u úpravy, ale není vznesen pohled na dotaz pro zjištění informací o sezoně a po kontrole dat a překrytí je volána procedura pro vložení nové položky.

To je k hotelové části vše a nyní přejdeme do části, kde probíhá zpráva uživatelů. Začnu u souboru pro přehled jednotlivých uživatelů.

**Seznam\_uzivatele.php:** Při vstupu je vznesen dotaz na pohled nad seznamem uživatelů. Z tohoto pohledu se do tabulky vypíše registrovaní uživatelé systému. Seznam je omezen na tyto položky: funkce, jméno, číslo občanského průkazu pro identifikaci a dva kontakty. Při výběru konkrétního klienta je includován soubor `PrehledOsobnichUdaju.php`, kde je vznesen dotaz na stejný pohled jako u přehledu s upřesněním na jednoho konkrétního uživatele. Data získaná z tohoto pohledu jsou pak přehledně vypsána do tabulky. V dolní části souboru je pak vznesen další sql dotaz na pohled nad tabulkou objednávek pomocí kterého jsou vypsány všechny objednávky klienta. Dále se nám nabízí dvě možnosti a to úprava dat uživatele a jeho odebrání. Odstranění uživatele probíhá podobně jaké u odstranění pokojů, nebo sezon. Úprava je funkcemi opět dost podobná. Je vznesen dotaz na příslušný pohled, z toho jsou vypsána data vložena do formuláře a ten je poté kontrolován kontrolní třídou a uložen pomocí procedury. Jediný rozdíl od úpravy osobních údajů samotným klientem je ve spojení všech položek do jednoho souboru s názvem `uprava_udaju.php`. Zde jsou položky rozděleny do fieldsetů a do jednotlivých formulářů. Podle toho jaký formulář je potvrzen se volá patřičná procedura.

Poslední část, kterou si projdeme je vytváření rezervací z pozice vedoucího. Samotná rezervace probíhá stejně jako je tomu u výše popsaného klienta, ale vedoucí mám možnost vytváření rezervací jak na sebe tak na jakéhokoliv existujícího uživatele systému.

**PrehledObjednavek.php:** Při otevření je vznesen dotaz na pohled vypisující přehled objednávek. Po stisku tlačítka detail objednávky, je tento dotaz vznesen znovu s upřesněním konkrétní objednávky. Tentokrát je objednávka vypsána do tabulky i s jednotlivými detaily a tlačítkem pro odebrání. Po zvolení odebrání objednávky je uživatel pomocí jednoduchého formuláře vyzván, zda si je jistý, že se jedná opravdu o objednávku, kterou chce smazat.

**VytvoreniObjednavkyVedouci.php:** Jedná se o stejný modul jako u zákazníka, ale je zde includovaný soubor pro výběr klienta za kterého má být objednávka vytvořena. Vedoucí vybere ze seznamu uživatele a dále s ním pracuje jako by byl za něj přihlášen. Prochází postupně všechny kroky rezervace a při potvrzení je volána procedura na vytvoření objednávky, které data zpracuje a uloží do databáze

V této části nejsou popsány všechny soubory. Většina z nich funguje na podobném principu a není tedy třeba je dále rozepisovat.

## 3.4 Vzhled webové aplikace

Vzhled webové aplikace je velmi důležitou součástí mého projektu. Každý člověk dá na první dojem. Říká se, že u webové prezentace máme necelých 5 sekund, za které můžeme klienta buď získat, nebo ztratit. Právě z těchto důvodů je vzhled webové stránky a s tím i spojená prezentace ubytovacího objektu klíčová pro jeho úspěch.

### 3.4.1 Velikost a rozložení stránky

Volba vhodné velikosti stránky je podstatná pro celkový vzhled a čitelnost stránek. V případě, že stránky uděláme moc široké, budou textové odstavce špatně čitelné. V opačném případě se pak odstavce jeví jako mnohem delší než ve skutečnosti jsou a to mnoho klientů odradí v jejich přečtení.

Při volbě vhodných rozměrů jsem vycházel z předpokladu, že drtivá většina monitorů v dnešní době pracuje na rozměrech 1024x768 (POLESNÝ, 2008). Dále jsem do návrhu zahrnul i fakt, že se čím dál tím více rozšiřují širokouhlé monitory. Proto jsem zvolil šířku hlavního okna aplikace na 960 obrazových bodů a samotný text jsem ještě o další 90 obrazových bodů zúžil. Tím jsem se dostal na 870 bodů, což je podle mě velmi dobře čitelné.

Do hlavičky stránky jsem vložil na levou stranu logo a na pravou stranu přihlašovací pole s registrací. Toto umístění považuji za jedno z nejčastějších. Menu jsem volil řádkové, umístěné pod hlavičkou a široké přes celé tělo stránky. Myslím, že takto vytvořené menu dodává stránce lepší celkový vzhled a je přehlednější. Po přihlášení uživatele je menu rozšířené o další úroveň.

Tlačítka jsem vybral ve stejné barvě jako pozadí a k jejich označení jsem použil jemný dvoubarevný rámeček. To dodává aplikaci zajímavý nádech a přitom jsem zachoval přehlednost pro klienta.



Obrázek 10 - ukázka vzhledu stránky

### 3.4.2 Barvy

Při tvorbě aplikace jsem hledal barvy, které budou výrazné, se silným kontrastem a netypické. Jasná volba byla černá a bílá. Mnoho grafiků tyto barvy podceňuje, ale jejich síla na klienta je velká. Kdybych zůstal jen u těchto dvou barev tak by stránka působila dost ponuře, proto jsem potřeboval najít hřejivou pastelovou barvu. Tato tmavší oranžová se mi jevila jako optimální volba. Upoutá pozornost, ale neprovokuje a má velmi dobrý kontrast jak s černou tak s bílou.

### 3.4.3 Písmo

Je několik kroků, které je třeba dodržet, aby stránka byla přehledná a příjemná na pohled. Základem mé aplikace je dostatečně velké písmo typu Arial Black, které považuji za dobře čitelné a dostatečně výrazné. Barvu písma jsem volil podle pozice a funkce ve dvou barvách a to bílou a černou. Mnohdy se písmo může zdát až zbytečně velké, ale k jeho použití mám jednoduchý důvod. Prezentace penzionu musí být schopná zaujmout všechny věkové generace. Mladší lidé si této skutečnosti prakticky nepovšimnou, ale pro starší je velké písmo mnohdy nespornou výhodou.

## 3.5 Instalace

Zprovoznění systému je celkem snadné a měl by ho zvládnout i průměrný uživatel se základní znalostí databází a webových stránek. Potřebujeme k tomu Oracle databázi, webový server a pokud možno nějaký sql developer. Kostrou databáze jsou tabulky, jejich vytvoření je uloženo v souboru `TabulkyVytvoření.sql`. Stačí zkopírovat obsah tohoto souboru do sql developeru a tím vytvoříme základ naší aplikace. Podobným způsobem je třeba nakopírovat do systému texty ze souboru `SQL příkazy`, kde jsou uloženy obslužné funkce, procedury, pohledy a defaultní data jako jsou pozice v systému, přihlašovací údaje pro vedoucího a podobné. Všechna data mají poznámky k čemu slouží, takže pokud si chceme změnit například přihlašovací údaje vedoucího, stačí si vyhledat příslušné položky. Defaultně je vedoucí nastaven na přihlašovací jméno `Pjupju` a heslo `poldis`. Pokud budeme měnit heslo, je nutné jej v textovém souboru zadávat v kódování `md5`. Převodů na toto kódování je na internetu nespočet. U změny ostatních údajů je nutné vyhledat uživatele se jménem `Daniel Poláček`, ale tyto údaje se již dají změnit přímo v systému.

Když máme zprovozněnou databázovou část, je třeba nahrát webové stránky na připravený server. Stačí zkopírovat všechny položky z `htdoc` do kořenového adresáře na server. Zde si musíme dát pozor na soubor `funkce.php`, jak jsem již popsal výše, jde o přihlašovací funkce. Pro spojení databází na web je nutné změnit parametry ve funkci `oci_connect` na údaje z databáze Oracle. I přes jednoduchost zprovoznění systému je lepší, aby tento systém spouštěla osoba, která jej dobře zná. Aplikace je stavěna na konkrétní penzion a při spouštění pro jiné ubytovací zařízení je nutné přizpůsobit část dat pro potřeby nového penzionu. Jsou to například počáteční ceníky, data o penzionu a podobné.

## 3.6 Uživatelská příručka

Systém jsem navrhoval, tak aby byl lehký na ovládání. Snažil jsem se o co nejintuitivnější ovládání, proto nemá smysl nějak víc popisovat všechny kroky.

### 3.6.1 Host a zákazník

Začnu u registrace nových zákazníků. Všechny potřebné informace jsou popsány přímo v aplikaci. Přihlašovací jméno musí obsahovat minimálně 5 znaky, heslo pak minimálně 6 znaků a doporučuji použít znaky i čísla pro větší bezpečnost. Jméno a příjmení je třeba vyplnit. Tyto položky přijímají pouze písmena, jiné znaky nejsou přípustné. Číslo občanského průkazu je pak nutné zadat jako 6 nebo 9 za sebou jdoucích čísel. Poštovní směrovací číslo přijímá hodnotu ve formátu tři čísel mezery a dvou čísel. Číslo popisné je pak pouze numerická hodnota. U kontaktů je popis pro všechny kolonky v aplikaci. Úprava zadávaných registračních dat po přihlášení je na stejném principu, pouze rozčleněna do jednotlivých položek. U rezervací nás pak vedou jednotlivé kroky a systém je natolik omezen, že nepřipustí špatnou volbu. Odebrání rezervace se provádí v položce přehledu objednávek, kde je u každé rezervace tlačítko pro odebrání.

### 3.6.2 Vedoucí

Pro obsluhu systému je nutná základní znalost fungování ubytovacího zařízení. S popisem začnu u hotelové části, kde vedoucí připravuje položky o ubytovacím zařízení. Při přidávání pokojů je nutné zadat název pokojů, který musí být jedinečný, počet lůžek a zda má být pokoj ihned aktivní. Zadávaná data jsou ošetřena, takže se nebojte, že něco pokazíte. V případě chyby se vám systém sám ozve, co je špatně. Odebírat lze pouze pokoje, na které nejsou vytvořeny objednávky a stejně tak je to s úpravou. Takže pokud je nezbytně nutné nějaký rezervovaný pokoj odstranit, musíme nejprve odebrat všechny rezervace, které jsou na něj vytvořeny.

V případě práce se sezonami je situace podobná. Samotné ovládání funguje na stejném principu, kde si vyberete položku z tabulky podle id a s tímto prvkem dále pracujete. Pro přidání sezony není nutné nic vybírat. Pro výpočet ceny je nutné, aby se jednotlivé položky nepřekrývaly. Systém je pro tento případ omezen. Zadáním sezony a s tím spojeného ceníku umožňujete klientům vytvořit objednávky. Pro omezení provozu například v době dovolené stačí nevytvářet na dané období položku v tabulce sezon a tím omezíte možnost rezervací. Mějme na paměti, že jednotlivé sezony by na sebe měly navazovat. Systém hlídá návaznost, ale v případě, že nevytváříte ceník z důvodů popsaných výše, bude hlášena kolize v sezonách. Toto řešení je pouze orientační a nijak neovlivňuje systém. Je pouze na vás, jak si dané sezony rozvrhnete. Odebírat jednotlivé položky jde vždy, ale mějme na paměti, že při odebrání sezony zůstávají v systému již vytvořené objednávky na dané období. Stejně tak při úpravě, kdy změním datum začátku nebo konce, případně cenu, jsou již vytvořené rezervace počítány podle starých ceníků a dat. Data zadaná v této tabulce se zobrazují i hostům penzionu, proto si musíme sami rozhodnout, zda ponecháme staré sezóny v ceníku nebo je smažeme.

U správy uživatelů máme k dispozici jejich podrobný přehled a objednávky, které do systému vložili. Pro odebrání je nejprve nutné odebrat všechny vytvořené objednávky a až poté lze odebrat samotného uživatele. Toto nastavení je sice omezující, ale nutné, aby nedošlo k nechtěné ztrátě dat. Změnit registrační údaje uživatele je možné kdykoliv. Změna se nám ihned promítne do rezervací daného uživatele.

U přehledů a zpráv objednávek je ovládání velmi podobné předchozím částem. Vedoucí má k dispozici seznam objednávek, se kterými může pracovat. Objednávky lze odebírat, přidávat a potvrzovat. Odebrat objednávky lze kdykoliv, objednávání není ničím vázané, jak tomu bylo například u pokojů, proto zde dbejte zvýšené opatrnosti. Vytváření objednávek je stejné jako v situaci kdy rezervaci tvoří zákazník. Vedoucí vytváří objednávku na sebe a do poznámky si zapisuje základní údaje o klientovi. Případně je možné vytvářet rezervace na již existujícího klienta. Potvrzení objednávek slouží pro evidenci přijatých záloh. Pomocí jednoduchého výběru potvrdíte objednávky a tím dáte klientovi jasně vědět, že jeho platba zálohy, případně celého pobytu byla přijata.

### 3.7 Závěr

Má aplikace je jistě zajímavým řešením rezervačního hotelového systému. S postupem času, od návrhu této aplikaci po její zprovoznění jsem zjistil, že některé z položek návrhu nebudou pravděpodobně využity, ale jinak má můj systém vše potřebné pro plnou funkčnost. Tato aplikace se dá v mnoha ohledech rozšířit, ať už by šlo o fakturovací systém nad platbami, modul pro tvorbu ubytovací knihy nad objednávkami, nebo třeba rozšíření pro více ubytovacích zařízení a tím vytvoření katalogu. Pro středně velký penzion je tato aplikace plně dostačující a v porovnání s aktuální situací na trhu by se řadila mezi luxusní rezervační systémy. Co se dalšího vývoje této aplikace týká, projekt se bude dále rozvíjet. Již nyní o tento systém projevilo zájem několik ubytovatelů. Dojde k několika úpravám. Za hlavní by se dal považovat přechod na jinou databázi a s tím nutné změny v sql příkazech. Dále je nutné vytvořit modul pro zálohování dat v případě pádu databáze. V průběhu podzimu by pak mohly proběhnout ostré testy systému. Při vývoji tohoto projektu jsem se dozvěděl velké množství nových informací, které mě donutily částečně přehodnotit můj pohled na rezervace. Tato práce byl pro mě velmi zajímavá a v budoucnu jí budu dále rozvíjet a zlepšovat.

## Literatura

**BORONCZYK, Timothy aj. 2009.** *PHP 6, MySQL, Apache: Vytváříme webové aplikace.* Brno : Computer press, 2009. 978-80-251-2767-4.

**GROFF, James R. a WEINBERG, Paul N. 2005.** *SQL kompletní průvodce.* Brno : Computer press, 2005. 80-251-0369-2.

**Hotel Fontána 2009.** Standardizace ubytovacích zařízení. *cestovni-ruch.* [Online] 7. 2 2009. [Citace: 02. 05 2011.] <http://www.cestovni-ruch.cz/kategorizace/klashotel.php>.

**HUSEBY, Sverre. 2006.** *Zranitelný kód.* Brno: Computer press, 2006. 80-251-1180-6.

**JANOVSKÝ, Dušan. 2010.** Jak psát web. *Formuláře.* [Online] 05. 02 2010. [Citace: 5. 5 2011.] <http://www.jakpsatweb.cz/html/formulare.html>.

**LACKO, Luboslav. 2007.** *Oracle. Správa, programování a použití databázových systémů.* Brno : Computer press, 2007. 978-80-251-2767-4.

**LEHOTSKÝ, Zdeněk. 2008.** Výhody databázových procedur. *Programujte.* [Online] 25. 03 2008. [Citace: 05. 05 2011.] <http://programujte.com/?akce=clanek&cl=2008032301-vyhody-databazovych-procedur>.

**MACEK, Petr. 2003.** Session v PHP. *Builder.* [Online] 19. 03 2003. [Citace: 05. 05 2011.] <http://www.builder.cz/art/homepage/phpsession.html>.

**Česko. Ministerstvo hospodářstv. 1994.** Doporučení upravující základní ukazatele pro kategorizaci hostinských a ubytovacích zařízení. *cestovni-ruch.* [Online] 07. 12 1994. [Citace: 02. 05 2011.] <http://www.cestovni-ruch.cz/kategorizace/doporuceni.htm>.

**MITON Previo s.r.o. 2002.** previo. *Hotelový rezervační a recepční systém Previo.* [Online] 2002. [Citace: 02. 05 2011.] <http://www.previo.cz/>.

**POLESNÝ, David. 2008.** Nejrozšířenější rozlišení monitoru je... *Živě.* [Online] 8. 6 2008. [Citace: 5. 5 2011.] <http://www.zive.cz/clanky/nejrozsirenejsi-rozliseni-monitoru-je/sc-3-a-142335/default.aspx>.

**Sagit. 1999.** Sběrka zákonů. *Sagit.* [Online] 23. 12 1999. [Citace: 05. 05 2011.] <http://www.sagit.cz/pages/sbirkatxt.asp?zdroj=sb99326&cd=76&typ=r>.

**STEPHENS, Ryan a PLEW, Roland. 2004.** *Naučte se SQL za 21 dní.* Brno : Computer Press, 2004. 80-7226-870-8.

**The PHP Group. 2011.** oci\_connect. *PHP.* [Online] 06. 05 2011. [Citace: 07. 05 2011.] <http://php.net/manual/en/function.oci-connect.php>.

**The PHP Group. 2011.** oci\_parse. *PHP.* [Online] 06. 05 2011. [Citace: 07. 05 2011.] <http://php.net/manual/en/function.oci-parse.php>.



**The PHP Group. 2011.** oci-execute. *PHP*. [Online] 06. 05 2011. [Citace: 07. 05 2011.] <http://php.net/manual/en/function.oci-execute.php>.

**The PHP Group. 2011.** oci-fetch-all. *PHP*. [Online] 06. 05 2011. [Citace: 07. 05 2011.] <http://php.net/manual/en/function.oci-fetch-all.php>.

**The PHP Group. 2011.** trim. *PHP*. [Online] 06. 05 2011. [Citace: 07. 05 2011.] <http://php.net/manual/en/function.trim.php>.

**UNIHOST aj. 2010.** Oficiální jednotná klasifikace. *unihostostrava*. [Online] 2010. [Citace: 05. 05 2011.] [http://cms.netnews.cz/files/attachments/671805/9-metodika\\_oficialni\\_jednotne\\_klasifikace.pdf](http://cms.netnews.cz/files/attachments/671805/9-metodika_oficialni_jednotne_klasifikace.pdf).

**Úřad pro ochranu osobních údajů. 2005.** Úřad pro ochranu osobních údajů k problémům z praxe. *uouu*. [Online] 11 2005. [Citace: 05. 05 2011.] <http://www.uouu.cz/uouu.aspx?menu=14&loc=346>.



## Příloha B – Zpracování dat z formuláře pro úpravu adres

```
<?php
include 'include/kontrolni.class.php'
$poleKraju=array("Jihočeský", "Jihomoravský", "Karlovarský", "Králověh
radecký", "Liberecký", "Moravskoslezský", "Olomoucký", "Pardubický", "Pl
zeňský", "Praha", "Středočeský", "Ústecký", "Vysočina", "Zlínský");
$poleOdeslani= array
$kontrola = new tridaKontrolyDat


$sql="SELECT * FROM p_prehled_udaju WHERE
id_osoby='".$_SESSION['ID_CISLO']."'";
$vysledek = dotaz($sql);
if (isset($_POST['registrace']))
{
    $kontrola->Vyplnene($_POST['regUlice'], 'RegUlice');
    $kontrola->Vyplnene($_POST['regMesto'], 'RegMesto');
    $kontrola->RegCisloPopisne($_POST['cisloPopisne']);
    $kontrola->RegPSC($_POST['regPCS']);
    $kontrola->Vyplnene($_POST['regStat'], 'RegStat');

    $pocetChyb=count($kontrola->chyby);
    if($pocetChyb==0)
    {
        $poleOdeslani['adresa_id']=$vysledek['ID_ADRESY'][0];
        $poleOdeslani['ulice']= $kontrola->data['RegUlice'];
        $poleOdeslani['mesto']= $kontrola->data['RegMesto'];
        $poleOdeslani['cislo_popisne']= $kontrola-
>data['RegCisloPopisne'];
        $poleOdeslani['PSC']= $kontrola->data['RegPSC'];
        $poleOdeslani['kraj']= $_POST['kraj'];
        $poleOdeslani['stat']= $kontrola->data['RegStat'];

        $sql="BEGIN
procedura_uprava_adresy('".$poleOdeslani['adresa_id']."', '".$poleOdeslani
['ulice']."', '".$poleOdeslani['mesto']."', '".$poleOdeslani['PSC']."', '".$$
poleOdeslani['cislo_popisne']."', '".$poleOdeslani['kraj']."', '".$poleOdes
lani['stat']."'"); END;";
dotaz_vloz($sql);

echo "Vaše údaje byly v pořádku změněny. Děkujeme za váš čas.";
} else{
    echo"Omlouváme se, ale nastala chyba. Zkontrolujte prosím své
údaje.";
}
```

## Příloha C – Ukázka grafického vzhledu registrace



Hotely a penziony  
Ubytování snadno a rychle online

Přihlášení  
Jméno:   
Heslo:

O nás   Ceník   Pokoje a obsazenost   Galerie   Zajímavosti   Kontakt

### Registrace nových uživatelů.

Proč se registrovat? Registrací u nás získáte možnost online rezervace v našem hoteli z pohodlí vašeho domova. Při příjezdu odpadá nutnost ubytovacích formalit a váš rezervovaný pokoj bude v připraven podle vašich představ.

Vyplňte prosím pečlivě vaše registrační údaje budou dále používány pro vaše objednávky. Položky s hvězdičkou jsou povinné u kontaktních údajů doporučujeme vyplnit telefon neí však podmínkou.

**Přihlašovací údaje**

Heslo musí obsahovat minimálně 6 znaků. Doporučujeme použít znaky i čísla pro větší bezpečnost.  
**POZOR REGISTRACE ROZLIŠUJE MALÉ A VELKÉ ZNAKY!**

\*Přihlašovací jméno:  \*Heslo:  \*Heslo znovu:

**Osobní údaje**

Číslo občanského průkazu je 6 nebo 9 místné číslo nejčastěji pod fotkou. Toto číslo potřebujeme pro vedení evidenční knihy hostů děkujeme za pochopení.

\*Jméno:  \*Příjmení:  \*Číslo OP:  Pohlaví: Muž ▾

**Datum narození**

Den: 1 ▾ Měsíc: 1 ▾ Rok: 1941 ▾

**Adresa**

Poštovní směrovací číslo vyplňte prosím ve formátu 000 00.

\*Ulice:  \*Číslo popisné:

\*Město:  \*PSC:  \*Kraj: Jihočeský ▾ \*Stát: Česká Republika

**Kontakty**

Telefon je nutné vyplnit ve formátu: +420789456123, email: jmeno@domena.cz a ICQ: 123-456-789.

\*Telefon:  \*Email:  ICQ:  Skype:

Obrázek 12 - Layout registrace