

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## WEBOVÁ APLIKACE PRO REALITNÍ KANCELÁŘ

BAKALÁŘSKÁ PRÁCE

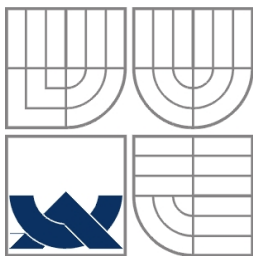
BACHELOR'S THESIS

AUTOR PRÁCE

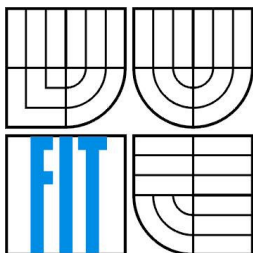
AUTHOR

JAN HÁJEK

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## WEBOVÁ APLIKACE PRO REALITNÍ KANCELÁŘ WEB APPLICATION FOR ESTATE AGENCY

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

JAN HÁJEK

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. LUKÁŠ GRULICH

BRNO 2007

## Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2006/2007

# Zadání bakalářské práce

Řešitel: **Hájek Jan**

Obor: Informační technologie

Téma: **Webová aplikace pro realitní kancelář**

Kategorie: Elektronický obchod

Pokyny:

1. Analyzujte požadavky na webovou aplikaci pro realitní kancelář (vkládání realit, vyhledávání, administrace).
2. Navrhněte strukturu aplikace (zejména datový model, případy použití). Vyberte vhodné programátorské rozhraní.
3. Ve zvoleném prostředí systém implementujte.
4. Testujte, diskutujte použití a možnosti dalšího rozvoje.

Literatura:

- Dle pokynů vyučujícího.

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním paměťovém médiu (disketa, CD-ROM), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Gulich Lukáš, Ing.**, UITS FIT VUT

Datum zadání: 1. listopadu 2006

Datum odevzdání: 15. května 2007

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
Fakulta informačních technologií  
Ústav inteligentních systémů  
Technická zpráva, 3. patro, Žitná 2

---

doc. Dr. Ing. Petr Hanáček  
vedoucí ústavu

**LICENČNÍ SMLOUVA**  
**POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

**1. Pan**

Jméno a příjmení: **Jan Hájek**  
Id studenta: 89662  
Bytem: Oskara Motyky 3/2982, 702 00 Ostrava  
Narozen: 28. 07. 1984, Ostrava  
(dále jen "autor")

a

**2. Vysoké učení technické v Brně**

Fakulta informačních technologií  
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305  
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....  
(dále jen "nabyvatel")

**Článek 1**

**Specifikace školního díla**

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):  
bakalářská práce

Název VŠKP: Webová aplikace pro realitní kancelář

Vedoucí/školitel VŠKP: Grulich Lukáš, Ing.

Ústav: Ústav inteligentních systémů

Datum obhajoby VŠKP: .....

VŠKP odevzdal autor nabyvateli v:

tištěné formě            počet exemplářů: 1

elektronické formě    počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
  - ihned po uzavření této smlouvy
  - 1 rok po uzavření této smlouvy
  - 3 roky po uzavření této smlouvy
  - 5 let po uzavření této smlouvy
  - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**


### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....

Nabyvatel



.....  
Autor

## **Abstrakt**

Tato práce se zabývá tvorbou webové aplikace pro realitní kancelář, ve které budou jak běžní uživatelé, tak administrátoři, schopni jednoduše ovládat celý systém, starající se o prezentaci firmy a aktuální nabídku nemovitostí. Je zde popsán kompletní vývoj, od počátečních specifikací, přes návrhy diagramů až po samotnou implementaci, použitím technologií XHTML, CSS, PHP, MySQL a JavaScriptu.

## **Klíčová slova**

Realitní kancelář, elektronický obchod, informační systém, webová prezentace firmy, XHTML, CSS, PHP, MySQL, JavaScript, databáze, reality, domy, byty, nemovitosti.

## **Abstract**

This thesis is addressed on making a web application for a real-estate agency. In this application both common users and administrators will be able to access and employ the underlying system with ease. The system will be holding up-to-date list of available realties, as well as managing the presentation. The thesis contains description of the complete developing process, including initial specifications, covering diagramme projecting and the final implementation itself, with XHTML, CSS, PHP, MySQL and JavaScript technologies used.

## **Keywords**

Reality estate, electronic shop, information system, company web presentation, XHTML, CSS, PHP, MySQL, JavaScript, database, reality, houses, flats, estates.

## **Citace**

Jan Hájek: Webová aplikace pro realitní kancelář, bakalářská práce, Brno, FIT VUT v Brně, 2007

# Webová aplikace pro realitní kancelář

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lukáše Grulichy. Další informace mi poskytla jednatelka realitní kanceláře AXATEM company s.r.o. Ing. Hana Žáková a její společnice Zuzana Žáková.

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

.....  
Jan Hájek  
30.4.2007

## Poděkování

Děkuji zástupkyním firmy AXATEM company s.r.o. a vedoucímu práce Ing. Lukáši Grulichovi za odbornou pomoc při konzultacích.

© Jan Hájek, 2007.

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

# Obsah

Obsah .....	1
1 Úvod.....	3
2 Analýza požadavků na webovou aplikaci pro realitní kancelář .....	4
2.1 Hlavní účel .....	4
2.2 Uživatelské role.....	4
2.2.1 Neregistrovaný uživatel .....	4
2.2.2 Registrovaný uživatel.....	4
2.2.3 Administrátor .....	5
2.3 Zpracování.....	5
3 Návrh.....	6
3.1 Use-Case diagram .....	6
3.2 ER diagram.....	7
3.3 Normalizace návrhu .....	8
3.3.1 1. normální forma (1NF).....	8
3.3.2 2. normální forma (2NF).....	8
3.3.3 3. normální forma (3NF).....	8
3.4 Popis tabulek .....	8
4 Použité technologie.....	10
4.1 XHTML.....	10
4.2 CSS.....	10
4.3 PHP .....	11
4.4 Javascript.....	12
4.5 MySQL.....	12
5 Implementace.....	13
5.1 Implementace uživatelského rozhraní .....	13
5.2 Celkový vzhled.....	14
5.3 Implementace některých funkcí .....	14
5.3.1 Vyhledávání nemovitostí .....	14
5.3.2 Přidávání nemovitostí .....	14
5.3.3 Přidávání obrázků .....	15
5.3.4 Generování a odeslání nového hesla.....	16
5.4 Přihlašování .....	16
5.5 Připojení k databázi .....	17
5.6 Odesílání e-mailů .....	17



5.7	Problémy a jejich řešení .....	17
5.7.1	Bezstavovost protokolu HTTP.....	17
5.7.2	Zabezpečení .....	18
6	Možnosti rozšíření .....	19
6.1	Pokročilé vyhledávání .....	19
6.2	Rozšířená editace práv uživatelů.....	19
6.3	Diskuzní fórum.....	19
6.4	Převod vkládaných obrázků .....	19
6.5	HTTPS.....	20
6.6	Kontrolní kód při registraci .....	20
7	Závěr .....	21
	Literatura .....	22
	Seznam příloh .....	23

# 1 Úvod

V dnešní době je pro úspěšný prodej, ať už výrobků svých, nebo zprostředkovávání prodeje jiných, téměř nutností vlastnit minimálně webovou prezentaci firmy, která bude dostupná po celém světě a tedy nepřetržitě k dispozici. Prodej realit je specifický rychle se měnící nabídkou a tudíž i reklama musí být dostatečně flexibilní. Je potřeba i několikrát denně měnit veškeré nabídky, kontaktovat případné zájemce a stále nabízet aktuální údaje o firmě. Proto jsem se rozhodl naimplementovat nástroj, který by umožňoval všechny běžné postupy urychlit a zároveň byl dostupný a spolehlivý. Tato aplikace bude navrhována pro opravdový provoz, ve kterém by se měla ocitnout v druhé polovině roku 2007 a to u společnosti AXATEM company s.r.o.

Vzhledem k použitým technologiím není problém systém aplikovat ve více odvětvích, nejen u prodeje realit. Základem systému je databáze s nabídkou produktů, jenž je potřeba v každém internetovém obchodě.

Následující kapitola je věnována rozboru požadavků na tento systém. Je rozdělena do tří částí a to na hlavní účel aplikace, tedy v čem by měla pomáhat a jaké by měla mít celkové funkce, druhá část se pak zabývá rozбором jednotlivých funkcí uživatelů, vstupujících do toho systému. Tato část se tedy zabývá neregistrovaným a registrovaným běžným uživatelem a administrátorem. V třetí části jsou popsány minimální požadavky na implementační část a to v podobě použitých nástrojů a technologií.

Kapitola třetí se již zabývá samotným návrhem, konkretizováním funkcí a přesnou specifikací možností uživatelů. Správný návrh je základním kamenem k úspěšnému zakončení každého projektu, proto je zde jak popis všech funkcí v Use-Case diagramu, tak rozbor kompletní databáze v ER diagramu. Oba návrhy spadají do jazyka UML, který je taktéž v úvodu kapitoly stručně popsán.

Čtvrtá kapitola je zaměřena na použité technologie, jako jsou XHTML, PHP, JavaScript, CSS, MySQL, na problémy s jejich užitím, jejich historii a předpokládaný rozvoj.

Další kapitola se již věnuje samotné implementaci aplikace, přihlašovací postupům, připojení databáze, zabezpečením proti nežádoucím vnikům, typem uživatelských účtů a jejich rozpoznáním, registračním procesem, přidáním nemovitosti, obrázku a jejich změnami, administrováním uživatelů a automaticky generovanými a rozesílanými hesly.

Nakonec zbývá jen popis možných rozšíření systému a jeho celkového vylepšení, které by ještě připadalo v úvahu, ale nebylo uskutečněno v důsledku omezených časových prostředků určených pro implementaci.

Závěrem shrnu celkový přínos aplikace a splnění požadavků definovaných v zadání.

## **2 Analýza požadavků na webovou aplikaci pro realitní kancelář**

Při návrzích všech systémů je nutné objasnit si celkové požadavky a ty postupně implementovat. V této kapitole shrnu všechny požadavky na tuto webovou aplikaci.

### **2.1 Hlavní účel**

Hlavním účelem této aplikace je zobrazovat aktuální nabídku nemovitostí k prodeji či pronájmu. Důležité je především, aby byl zákazník informován o všech hlavní skutečnostech daného objektu hned v tomto systému, to se týká především lokality, kde se daná nemovitost nachází. Dispozice a celková zastavěná plocha budou na druhém místě v parametrech, podle kterých se bude zákazník řídit, v některých případech společně s cenou, která hraje taky dost velkou roli při tak závažném rozhodování. Mezi dalšími důležitými faktory jsou i vlastnictví, vzhledem k možnosti brát na vybraný byt či dům hypotéční úvěr a typ zdiva, jelikož panelový byt s umakartovým jádrem může celkový pocit nakonec pokazit. Zbývá už uvést jen doplňující informace jako počet pater nemovitosti, popř. podlaží uvedeného bytu, přesnou polohu (ulice, číslo orientační, číslo popisné) a zda je instalován plyn, telefon či internet. Pro možnost dalšího doplnění je také nutné přidělit místo poznámkám.

### **2.2 Uživatelské role**

V tomto systému se nacházejí tři typy uživatelů. Zde popíši jednotlivé role.

#### **2.2.1 Neregistrovaný uživatel**

Má možnost procházet prezentaci firmy a nahlédnout do nabídky nemovitostí. Vyhledávání nemovitostí probíhá podle výše uvedených údajů, jako je druh nabídky, lokace, dispozice, plocha, cena, vlastnictví a typ zdiva. Ke každému výpisu nemovitostí patří i detail jednotlivých objektů společně s fotogalerií. Jedinou možností jak pokračovat hlouběji do systému, a tím využít i další funkce, je registrace.

#### **2.2.2 Registrovaný uživatel**

Má možnost oproti neregistrovanému zadat do systému kontaktní údaje o své osobě, díky kterým může být později na vlastní žádost kontaktován a mohou mu tak být podány podrobnější informace o vybraných nemovitostech. Žádost probíhá pomocí přímo určeného formuláře, který se zobrazí po

kliku na příslušný odkaz. Neměla by zde být potřeba uvádět žádné další údaje, ve formuláři by měl být hlavně prostor pro upřesnění dotazu.

Dále má možnost vybrat si vytipované objekty v systému a uložit si je do oddělené tabulky od ostatních, pro lepší přehled při dlouhodobém hledání. S touto funkcí souvisejí i doplňkové akce, které musí mít uživatel možnost provádět nad svojí tabulkou, jako je další přidávání a mazání.

V případě zapomenutého hesla je po vyplnění registrované e-mailové adresy odesláno elektronickou poštou heslo nové a rovněž změněno v systému.

### **2.2.3 Administrátor**

Při použití databází má vždy administrátor nejvíce možností systému, počínaje zadáváním nemovitostí, jejich úpravou a odstraňováním, přes přidávání a odebírání fotek objektů a konče změnou práv, blokováním a mazáním všech zaregistrovaných uživatelů systému.

Změna práv je velice užitečná v případě, že počet uživatelů stoupne natolik, až nebude jeden administrátor schopen zvládat jejich kontrolu, nebo například při výrazném zvyšování počtu nemovitostí v databázi. Pomocí povyšování je také možnost rozdělení funkcí v systému mezi více lidí ve firmě.

V případě nepřiměřeného chování některého ze zaregistrovaných uživatelů a neúměrnému zatěžování emailových schránek pracovníků poskytujících doplňující informace má také administrátor možnost prostého zablokování uživatele, kdy nedojde k vymazání ze systému, jen přijde o možnost využívání pokročilejších funkcí a je jen na administrátorovi, jestli tomuto účet opět odblokuje.

## **2.3 Zpracování**

Každý moderní webový systém by měl splňovat alespoň základní podmínky bezpečnosti a odpovídat standardům psaní HTML. Bezpečnostní minimum se týká především ukládání hesel do tabulek databáze v zašifrovaném tvaru, na to je použita technologie šifrování MD5. Co se týče validity celého systému, měl by odpovídat nejlépe XHTML. Jako požadavek byl tedy zvolen XHTML 1.1, jako poslední známá, funkční a používaná definice pro psaní internetových stránek. Předpokladem pro splnění všech účelů je také použití skriptovacích jazyků, jako jsou například na serverové straně PHP a na uživatelské JavaScript. Vyhovující databáze pro menší systémy tohoto typu je např. MySQL. Použití kaskádových stylů CSS je podmínkou pro splnění validního XHTML.

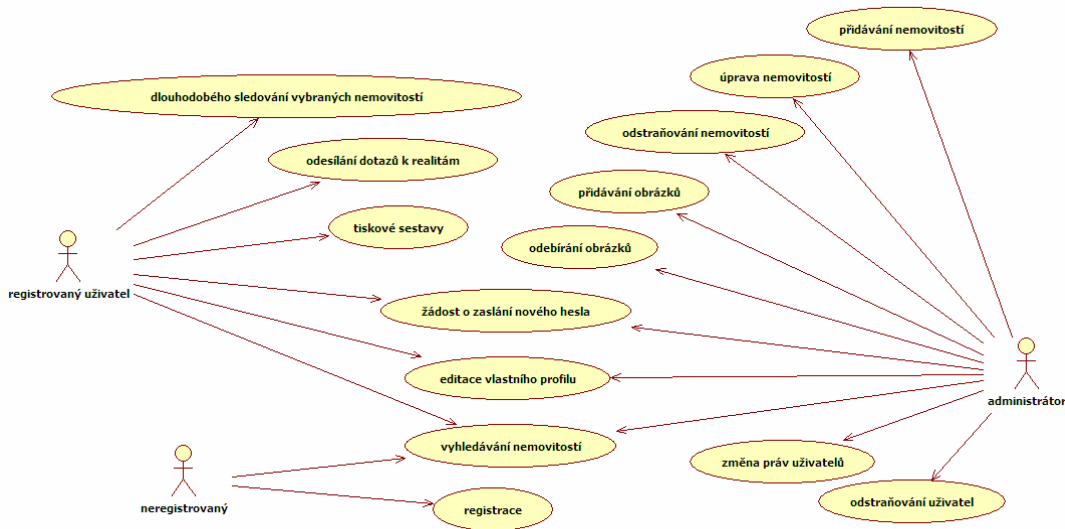
# 3 Návrh

Základem kvalitní aplikace je promyšlený návrh, ve kterém je na první pohled evidentní, čeho všeho a jakým způsobem bude systém schopen. Díky kvalitnímu návrhu je možné aplikaci od počátku implementovat přesně podle určených požadavků a to zabraňuje ztrátě času při přepisu aplikace v případě opomenutí jedné, nebo i více klíčových funkcí.

Všeobecně známým, a také nejpoužívanějším jazykem, pro tvorbu návrhů je jazyk UML (Unified Modeling Language). Ten vznikl v roce 1996 na zakázku firmy Rational Corp. V roce 1997 byl uznán jako průmyslový standard a ostatní analytické jazyky odsunul do pozadí. Skládá se z devíti typů pohledů na aplikaci s tím, že pro každý pohled je vytvořena definice přesně vypadajícího návrhu. Jedná se především o diagramy použitelné při návrhu objektově orientovaných aplikací, jako třeba diagram tříd, objektů, komponent a nasazení. Dále se budeme zabývat jen dvěma variantami a to diagramem případů použití (Use-Case diagram) a stavovým diagramem (ER – Entity relationship diagram).

## 3.1 Use-Case diagram

Tento typ diagramu zachycuje a přehledně zobrazuje, v grafické i textové formě, jednotlivé funkce systému, spolu s aktéry, kteří se na nich mohou podílet. Akce je zobrazena oválem s vepsaným stručným popisem. Aktér (aktor) bývá nejčastěji označován symbolem postavy s popisem jeho funkce v systému (názvem aktora, jako např. administrátor, správce DB, moderátor, aj.). Druhá varianta označení aktéra, používaná spíše pro externí systémy, je obdélník s vepsanou funkcí, nicméně první varianta je daleko přehlednější a pro tento případ názornější.



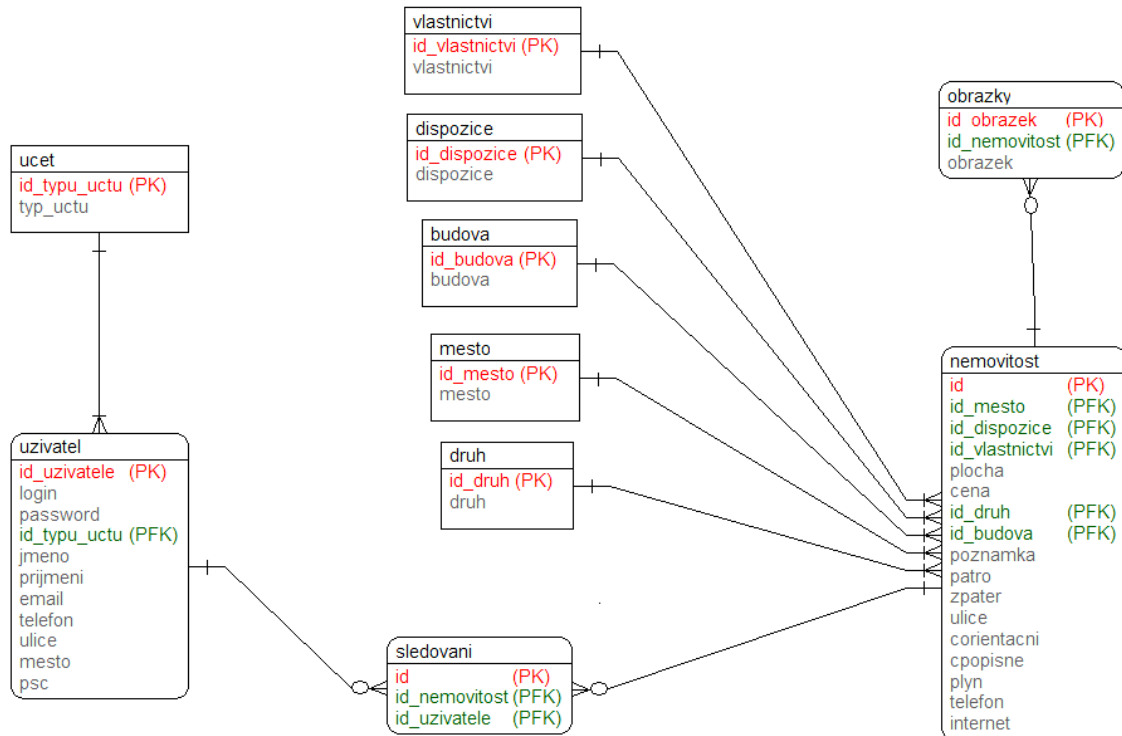
Obrázek 3.1 - Use-Case diagram

## 3.2 ER diagram

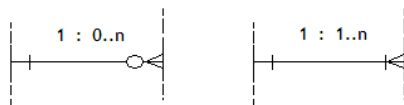
Další velice užitečný nástroj, tentokrát spíše pro návrh databáze, je opět z balíku UML a je jím ER diagram. Toto, opět grafické i textové schéma, zachycuje přehledně strukturu tabulek v databázi, společně se vztahy mezi nimi a s rámcovým obsahem každé z nich. Tabulky jsou značeny obdélníky, v nichž horní část (hlavička) obsahuje název tabulky a je od spodní části, v které se nacházejí jednotlivé názvy sloupců tabulky, oddělena vodorovnou čarou.

Správný návrh databáze samozřejmě musí obsahovat všechny důležité skutečnosti pro její vytvoření, proto se v seznamu sloupců tabulek můžeme setkat se znaky <<PK>> - primární klíč a <<FK>> - cizí klíč.

Uvedený diagram je nakreslen pomocí freeware programu Toad Data Modeler, který značí klíče jinak, než je obvyklé, nicméně na pochopení tato změna nemá vliv („(PK)“ a „(PFK)“).



Použité relace:



Obrázek 3.2 - ER diagram

## 3.3 Normalizace návrhu

Správný návrh databáze je klíčový pro dlouhodobé fungování systému. V případě nalezení chyby v návrhu v běžně užívaném systému se toto může projevit velkou časovou i finanční ztrátou společnosti. Pro základní kontrolu optimálního návrhu byly sepsány normy, které určují, jak by entity, které jsme v tabulkách používali, měly fungovat, co přesně mají znamenat a zda by se nedaly rozložit pomocí dalších relací a tak podpořit snížení velikosti a tedy zrychlení databáze.

Jako základní se považují tři normální formy:

### 3.3.1 1. normální forma (1NF)

První normální forma říká, že obsah buněk tabulek musí být atomický, tzn. dále nedělitelný. V případě této konkrétní databáze můžeme demonstrovat např. na tabulce uživatelů. Každý uživatel má své jméno, které se skládá ze dvou částí. V nesprávném případě by tabulka obsahovala jeden sloupec pro jméno i příjmení. Problém by byl třeba v případě, kdy by například administrátor potřeboval najít uživatele podle křestního jména a nebyl by schopen samostatně jméno jednoduše vytáhnout z databáze a porovnat. Proto musí být tyto dva atributy rozděleny do dvou sloupců.

### 3.3.2 2. normální forma (2NF)

Tabulka vyhovuje 2NF právě tehdy, pokud vyhovuje 1NF a zároveň pokud je každý atribut, který není primárním klíčem na něm přímo závislý. To v důsledku znamená, že převedením tabulky na 2NF (dekompozicí - rozložením) zčásti zabráníme nežádoucí redundanci, jelikož se omezí výčet sloupců, které už by měly být v samostatné tabulce.

### 3.3.3 3. normální forma (3NF)

Třetí normální formě vyhovuje taková tabulka, která splňuje 2NF a zároveň neobsahuje žádný atribut, který není primárním klíčem a byl by tranzitivně závislý na jiném klíči. Řešením je ještě přísnější dekompozice tabulky, než při převádění do 2NF. Po splnění 3NF dosahujeme tabulek bez zbytečných redundancí.

## 3.4 Popis tabulek

Navržená databáze obsahuje deset tabulek. Hlavní jsou tabulka nemovitostí a tabulka uživatelů.

Tabulka nemovitostí obsahuje jako primární klíč identifikační číslo nemovitosti, dále obsahuje pět cizích klíčů, které odkazují na primární klíče tabulek měst, vlastnictví, typu budov, dispozic a druhů nabídek. Ty jsou uvedeny pro snížení redundance v hlavní tabulce a pro snadné vyhledávání podle uvedených parametrů. Dalšími atributy každé nemovitosti z této tabulky jsou její plocha, cena,

adresa, která je po splnění 2NF rozdělena na ulici a orientační a popisné číslo domu, počet pater nemovitosti a číslo patra v případě nabídky bytu. Dále jsou doplněny informace o zavedení plynu, telefonu a internetu. Prostor pro komentář zde poskytuje sloupec poznámek. Povinnými atributy je uvedených pět cizích klíčů, ulice, cena a plocha.

Druhou důležitou tabulkou jsou uživatelé. Zde je primárním klíčem ID uživatele, cizím klíčem je pouze typ účtu, který se opět podle normálních forem nachází v samostatné tabulce. Dále se zde nachází sloupec pro uživatelské jméno, heslo a emailovou adresu. Všechny tyto atributy jsou povinné (not Allow Null). Informace o uživateli doplňují sloupce pro jméno, příjmení, telefon, ulici, město a poštovní směrovací číslo.

Zbývají tabulky obrázků a tabulka informací o sledování realit uživateli. První zmíněná obsahuje ID obrázku, cizí klíč ID nemovitosti a binární pole s obsaženým obrázkem ve vložené velikosti a formátu JPG.

Tabulka sledování pak obsahuje jen ID jednotlivého záznamu o sledování a dva cizí klíče, ID uživatele a ID nemovitosti, kterou se rozhodl přidat do svého virtuálního seznamu sledovaných realit.



# 4 Použité technologie

## 4.1 XHTML

Prvním značkovacím jazykem bylo HTML verze 2.0. Vydala jej společnost IETF v roce 1995 jako standard pro psaní webových stránek. Tato specifikace neobsahovala mnoho prvků ke tvorbě graficky zdařilých prezentací a tak se dále vyvíjela na verzi 3.2, jež vydalo Webové konsorcium W3C. Bohužel do této verze byly zařazeny i mnohé prvky, které neměly využitelný strukturální význam. Proto W3C vydalo další specifikaci, verzi 4.0, která je hojně používána ještě dnes. Vzhled dokumentu se již převážně přenechává kaskádovým stylům CSS (Cascading Style Sheets), jež se většinou píší do odděleného souboru, proto se také úprava designu stává mnohem jednodušší a rychlejší. Verze 4 byla ještě rozdělena na tři části, kde jedna (HTML 4.0 Transitional) umožňovala stále používání veškerých vzhledových funkcí, jako verze 3.2, ale obsahovala i výhody nového HTML 4.0 Strict. To, jak již název napovídá, je nejpřísněji braná definice uvedené verze. Jako třetí varianta vznikla norma Frameset, která umožňovala použití rámcových prvků.

Tento velký pokrok se neobešel bez chyb, které se časem vyskytly a byla vydána revize 4.01.

Vývoj XHTML začal tím, že byla potřeba definovat HTML jako část XML (eXtensible Markup Language). Tím došlo ke zkombinování některých vlastností XML a HTML a vznikl zatím nejvhodnější nástroj pro tvorbu webu. Dostal název eXtensible HyperText Markup Language a měl by být nejpoužívanějším standardem pro psaní internetových prezentací. Nicméně nová definice s sebou nese i nároky na přeškolení všech zainteresovaných designérů a ne každý byl ochoten již na tuto, relativně novou, specifikaci přejít a stále se často setkáváme se stránkami, které nevyhovují ani HTML 4.0 a obsahují základní chyby při tvorbě stránek.

První specifikací je XHTML 1.0, která je stejná jako HTML 4.01, ale je již rozšířena o pravidla XML. Dále se od vzniku XHTML nepodporuje formátování textu jinak, než použitím CSS a spojením s XML se klade velký důraz na strukturování, z čehož vyplývá striktní uzavírání párových i nepárových tagů, podle jejich typu.

## 4.2 CSS

Vzhledem k mohutnému rozmachu HTML bylo potřeba vymyslet nástroj, kterým by bylo možno jednoduše formátovat velké bloky textu a celé weby jednodušším způsobem, než přepisováním u každého tagu. Proto vznikly kaskádové styly označovány anglickou zkratkou CSS (Cascading Style Sheets), jež umožňují pohodlně definovat formátování tříd, tabulek, odstavců a ostatních bloků, které tak můžeme pohodlně nastavovat a následně také velice rychle měnit pouhou změnou stylpisu (soubor, jež obsahuje definice formátování tagů v HTML, často „styl.css“). Tato technologie tedy

umožňuje například změnu velikosti textu v celém dokumentu, barvy pozadí, fonty a spousty dalších užitečných funkcí.

Příklad:

```
P { font: bold 14px arial; } – všechny odstavce textu budou psány fontem arial, o velikosti 14 pixelů a tučně.
```

## 4.3 PHP

PHP je jeden ze skriptovacích jazyků, které se řadí do skupiny server-side (skript se vykonává na straně serveru a k uživateli jde již vygenerované čisté HTML). Tento způsob programování webů má své plusy i mínusy. Výhodou je, že se k uživateli nikdy nedostane zdrojový kód skriptu, pokud samozřejmě někde nenastane chyba v podobě nedokončených párových označení části HTML, psané pomocí PHP. Další velkou výhodou je, že nemůže nastat nekompatibilita na straně klienta vzhledem k tomu, že se opravdu na jeho počítač dostane jen HTML kód. Bohužel má tato skupina skriptovacích jazyků i nevýhody, jejichž vyřešení by kolidovalo s bezpečností, jak již byla výše popsána. Tou je nemožnost reagovat bezprostředně na klientskou událost, jako bývá často například přejetí myši přes políčko buňky.

Začátek kódu PHP se v HTML většinou značí „<?“ nebo také přehledněji „<?php“, ukončení pak musí být ve formátu „?>“. Celý kód v tomto úseku musí být psán čistě jazykem PHP a jakákoliv chyba je okamžitě serverem hlášena a skript neproveden.

Příklad HTML dokumentu s výpisem v PHP:

```
<HTML>
<HEAD>
<TITLE>Testovací stránka</TITLE>
</HEAD>
<BODY>
<?
echo "Testovací výpis v PHP";
?>
</BODY>
</HTML>
```

## 4.4 Javascript

Druhou kategorií skriptovacích jazyků jsou klient-side. Tato skupina umožňuje bezprostředně reagovat na uživatelskou akci, bohužel má ale klient také přístup ke zdrojovému kódu skriptu, takže prakticky nepoužitelné pro jakékoliv manipulování s citlivými údaji. Dále je zde možnost, že klientský počítač nebude podporovat tento jazyk (to se týká především starých webových prohlížečů, nebo také zakázaného používání tohoto typu skriptu v prohlížeči). Proto je hojně používán hlavně pro efektní změny vzhledu při událostech na myši, dále na příjemnější a rychlejší kontrolu vyplňování formulářů, nebo změny formátování textu, pomocí identifikátorů definovaných pro použití v CSS (id, class).

## 4.5 MySQL

Při programování informačních systémů je nutné zvolit, kam ukládat veškerá data. Ukládání do souborů je nevýhodně vzhledem k pracnosti při vyjímání konkrétních dat a proto přišly na řadu databáze. MySQL je ideální volba pro webové aplikace, jelikož není náročná jako třeba Oracle a zároveň poskytuje téměř všechny funkce, které potřebujeme. Ostatní se pak dají nahradit buď další kombinací MySQL dotazů, nebo pomocí PHP.

Tento databázový systém vyvinula švédská firma MySQL AB. Jedná se o multiplatformní databázi, která komunikuje pomocí SQL příkazů, jež obsahují drobné odlišnosti od ostatních SQL databází. Pro svou vhodnost bývá nasazována právě pro webové projekty, společně s užitím technologie PHP a HTML, většinou na serveru Apache, který bývá vydáván často v balíku právě s PHP a MySQL. Díky těmto sadám je rozšířen mezi širokou veřejností.

# 5 Implementace

## 5.1 Implementace uživatelského rozhraní

System je dostupný jak pro přihlášené, tak pro nepřihlášené uživatele. Přihlašovací formulář je umístěn nahoře v hlavním menu, které se pro odlišení od většiny stránek nachází na pravé straně a je vždy zobrazeno. Pod formulářem se také nachází odkaz na stránku s registrací, zde se může návštěvník zapsat a získat tak přístup k dalším funkcím. Registrační formulář je vidět na obrázku 5.4. Obsahuje základní informace o uživateli jako je uživatelské jméno, dvě pole pro heslo, jméno příjmení, emailovou adresu, telefonní číslo, ulici, město a poštovní směrovací číslo. Dále je zde improvizovaný systém proti zneužívání registrace, který je založen na opisování čísel vytisknutých pod formulářem.

Po úspěšném přihlášení formulář zmizí a místo něj se objeví menu uživatele, podle toho, jaká má práva.

Nepřihlášený uživatel má tedy přístupné pouze hlavní menu, ve kterém je možno prohlížet informace o společnosti, služby, nadstandardní služby, seznam realit společně s vyhledáváním, detaily a obrázky, ceny správy nemovitostí a kontaktní údaje.

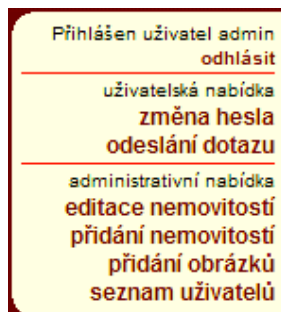
The image shows a search form titled "Vyhledávač" (Searcher). It contains several dropdown menus and input fields. The dropdown menus are labeled "Město:", "Dispozice:", "Druh nabídky:", "Vlastnictví:", and "Zdivo:", all with the value "nezadáno". Below these are two input fields for price: "Cena od: [ ] do: [ ] Kč" and "Plocha od: [ ] do: [ ] m²". At the bottom right, there are two buttons: "Vymazat formulář" (Clear form) and "Vyhledat" (Search).

Obrázek 5.1 - Formulář pro vyhledání realit

Každý uživatel má taktéž možnost nechat si vytvořit tiskovou sestavu s výsledkem vyhledávání, která umožní přehlednější tisk než tisk přímo ze stránky.

Přihlášený uživatel má navíc možnost pohybovat se v dalším menu, které se objeví místo přihlašovacího formuláře. Tam jsou dostupné odkazy pro změnu hesla, formulář pro odesílání dotazů na reality a seznam sledovaných vybraných nemovitostí. Ty se do seznamu přidávají přes stránku detailů dané reality.

Administrátorovi se zpřístupní i druhá část uživatelského menu, kde je možnost odkázat se na formulář pro editaci nemovitostí, přidání nemovitostí, přidání obrázků nebo tabulku, ve které má možnost měnit práva uživatelů.



Obrázek 5.2 - Administrátorské menu

## 5.2 Celkový vzhled

Vzhled jsem volil podle barevné kombinace loga, které je již zavedeno. V hlavičce symbolika realit s logem společnosti uvádí stránku k tématu. Menu, jak jsem již výše uvedl, jsem volil po pravé straně, vzhledem k množství stránek, které jsou navrženy „klasickým stříhem“. V tomhle případě se, myslím, jedná o příjemné narušení stereotypu, které může při celodenních hledáních realit potěšit nejednoho uživatele. Co se vzhledu menu týče, je založeno na informačních rámečcích, ve kterých se nachází aktuální čas, datum, svátek dnes a zítra a samozřejmě počet návštěv. Kontaktní informace jsou pak obsahem další buňky.

## 5.3 Implementace některých funkcí

### 5.3.1 Vyhledávání nemovitostí

Vyhledávání probíhá v uvedeném formuláři (Obrázek 5.1). Je zde vidět pět výběrů typu SELECT, které obsahují slovní část příslušných tabulek databáze. Dále je zde možnost zadání rozmezí plochy a ceny, které jsou pak společně s vybranými možnostmi zařazeny do dotazu a odeslány databázovému serveru k vyhodnocení.

### 5.3.2 Přidávání nemovitostí

Přidávání nemovitostí probíhá ve složitějším formuláři, jelikož tabulka databáze obsahuje 17 sloupců a potřebujeme vyplnit co nejvíce z nich. Prvních pět údajů uvedených v předchozím odstavci je rozšířených o javascriptem ovládané vstupní pole, které slouží pro ruční zadání položky, která se ještě v databázi nenachází. Takový údaj je odesílacím skriptem automaticky přidán do příslušné tabulky

v databázi. Dále se tento údaj v seznamech zobrazuje a není tedy potřeba jej znovu vypisovat. V případě vybrání ze seznamu toto pole mizí. Dále jsou zde zbylé údaje, které by měly v budoucnu klientům napomoci rychlému vyhledání vysněné reality.

**Přidání nemovitosti**

Město: Brno

Dispozice: 4+1

Druh nabídky: vlastní...

Vlastnictví: osobní

Zdivo: vlastní...

Ulice:

PSČ:

Podlaží:

Celkem podlaží:

Č.orientační:

Č.popisné:

Plyn:

Telefon:

Internet:

Poznámka:

Cena: Kč

Plocha: m<sup>2</sup>

Vymazat formulář Odeslat

Obrázek 5.3 - Formulář pro přidání nemovitosti

### 5.3.3 Přidávání obrázků

Přidávání obrázků probíhá s pomocí okna pro výběr souboru na klientském disku. Do formuláře stačí jen přidat číslo nemovitosti, ke které se obrázek má přiřadit a odeslat. Ukládání provádí speciální skript, který načte uvedený soubor jako text, přidá lomítka před funkční HTML znaky a tento řetězec uloží jako binární pole přímo do tabulky databáze.

Zobrazování obrázků pak řeší jiný skript, který načte binární řetězec. Tento řetězec pak pomocí funkce `imagecreatefromstring` převede na obrázek. Z tohoto obrázku pak získá výšku a šířku, podle zadaných parametrů obrázek zmenší a vytvoří hlavičku HTML souboru, který se bude chovat jako obrázek typu JPEG daných rozměrů. Tímto skriptem je pak možné obrázek užít kdekoliv v aplikaci. Díky možnosti zmenšování pomocí výpisového skriptu se dá jednoduše snížit velikost dané stránky a

tím celé načítání zrychlit. To je možné jen proto, že se jedná o skriptovací jazyk typu server-side a proto nedochází ke zmenšování až u klienta, nýbrž přímo na serveru.

### 5.3.4 Generování a odeslání nového hesla

V případě ztráty hesla uživatele je zde naimplementována možnost nechat si vygenerovat nové heslo. To se generuje skriptem, který vybírá náhodně z tří množin znaků (velká a malá písmena a čísla). Podmínkou je ještě ošetřeno, aby se v generovaném řetězci nevyskytly znaky stejné množiny těsně za sebou. Skript je připraven generovat libovolně dlouhá hesla, podle nastavení parametru funkce. Zvolil jsem délku 8 znaků, kterou považuji za dostatečně bezpečnou proti hackingu stylem brutal-force (hádáním řetězce po znacích). Po této proceduře je heslo odesláno společně s uživatelským jménem na příslušný email, dále je zašifrováno pomocí MD5 a zapsáno do databáze místo hesla zapomenutého.

**Registrace**

Přihlašovací jméno:

Heslo:

Kontrola hesla:

Jméno:

Příjmení:

Email:

Telefon:

Ulice:

Město:

PSČ:

Zde prosím opište kód:

**Zaregistrovat**

povinné údaje jsou zvýrazněny tmavší barvou rámečku  
emailová adresa bude použita pro zaslání Vámi vyžádaných detailů nemovitostí,  
popř. pro zaslání zapomenutého hesla

**Vstup do systému**

Jméno:

Heslo:

**Vstup registrace**

**Aktuality**  
zbrusu nový design -  
databáze realit -  
uživatelské účty -

**Menu**  
o společnosti  
služby  
nadstandard  
reality  
ceny  
kontakty

**Info**  
Dnes je čtvrtek,  
10. května 2007.  
**13:05:12**  
svátek má Blažena,  
zítra Světava  
počet návštěv: 129

Obrázek 5.4 - Ukázka designu aplikace a registrační formulář

## 5.4 Přihlašování

Do systému se uživatel přihlašuje vloženým formulářem v menu. Přihlašovací údaje jsou odeslány pomocí metody POST na tutéž stránku, která při každém načítání kontroluje, zda byla zavolána pomocí formuláře a v případě že ano, hledá v tabulce uživatelů v databázi adekvátní uživatelské

jméno, heslo a typ účtu, tedy jakými právy daný uživatel disponuje. Všechny tyto informace jsou dále uloženy do superglobální proměnné `$_SESSION`. Heslo je v ní uloženo v zašifrované podobě pomocí MD5. Tato proměnná je pak využita pro řešení problému bezstavovosti protokolu HTTP, jelikož nejsme schopni zjistit, co se v posledním kroku dělo a to konkrétně jestli je uživatel přihlášen a jak. Uvedená proměnná je uložena v cookies, zašifrována a její délku platnosti určuje nastavení webového serveru. K tomu slouží položka `session.cookie_lifetime` v configuračním souboru serveru „`php.ini`“. Uvedená doba platnosti je zadaná v sekundách.

## 5.5 Připojení k databázi

Připojení probíhá pomocí souboru s přihlašovacími údaji. Název databáze je „`axatem`“. Soubor se jmenuje „`kllice.php`“ a je ve složce společně s ostatními skripty aplikace.

## 5.6 Odesílání e-mailů

Pro odesílání e-mailů aplikace používá open-source PHP třídu PHPMailer, která je dostupná na internetu na adrese <http://phpmailer.sourceforge.net/>. Tato třída je zde důležitá především pro správné kódování textu a pro zalamování řádků samotného těla e-mailu. Je obsažena v souboru `class.phpmailer.php`, který je pak vložen do kódu při odeslání nově vygenerovaného hesla a při dotazu k nemovitosti. Podle této třídy je vytvořen objekt, ve kterém jsou následně nastaveny proměnné a provedena funkce k odeslání. Adresu informátora, kam bude systém odesílat dotazy uživatelů, je možno nastavit v již výše zmíněném souboru „`kllice.php`“.

## 5.7 Problémy a jejich řešení

### 5.7.1 Bezstavovost protokolu HTTP

Hlavním problémem při implementaci je bezstavovost. To znamená, že server neví, která stránka byla v předchozím kroku zobrazena a tudíž by třeba po odeslání dotazu na databázi mohlo nastat, že by se nám zobrazila stránka stejná a nebyla by možnost se z této situace dostat. Tento problém se většinou řeší pomocí změny adresy stránky způsobem, kdy se pomocí metody GET předávají požadované vstupní informace pro načtení. Adresový řádek proto může v některých případech dosahovat poměrně dlouhých řetězců.

Příklad: `http://www.axatem.cz/ibp/index.php?akc=login&id=43#reality_result`



Takto složitá adresa značí, že jsme se hodláme pohybovat v uživatelském menu (tedy po přihlášení) a že budeme zobrazovat detail nemovitosti s identifikačním číslem 43.

Velkým pomocníkem při řešení bezstavovosti jsou SESSIONs. Pro jejich správnou funkci a bezpečnost je nutno v php.ini nastavit session.use\_only\_cookies na „1“. V případě nastavené na nulu se bude řetězec identifikující používanou SESSION odesílat také pomocí metody GET a to může v některých případech vyvolat problémy z validitou. Někdy server přidává do formulářů na nevhodná místa skryté vstupní pole, které mu tento parametr odesílá. V tomto případě bylo porušeno pravidlo pro validní XHTML 1.1 a tudíž bylo nutno aktivovat toto omezení, což v konečném důsledku navíc znamená i vyšší bezpečnost, jelikož není viditelný identifikační řetězec použité SESSIONy.

## 5.7.2 Zabezpečení

Ukládání hesel do databáze je docela nebezpečná záležitost. Proto se používají nejrůznější šifrovací metody, abychom případnému útočníkovi zabránili zneužití nelegálně získaných hesel. To lze zabezpečit celkem jednoduchým použitím jednosměrného šifrovacího algoritmu MD5. Ten je schopen z libovolně dlouhého (nepočítám přetečení proměnných v prostředí, kde je MD5 implementováno a používáno) řetězce vytvořit 30ti bajtový hashcode, který není lehké dešifrovat zpět. Podle mých experimentů, by způsobem brutal-force, tzn. hádáním řetězců a převáděním na MD5 hashcode, trvalo na běžném počítači (Intel Pentium 4 3.0 GHz, 1GB RAM DDR2) uhodnutí osmimístného hesla, ve kterém by byly použity velká a malá písmena společně s číslicemi, zhruba sedm let. Dojít ke správnému výsledku logickou cestou je téměř nemožné, jelikož se při šifrování provádí spousta bitových přesunů a aritmetických operací, které zpětně není možné rozpoznat. Soudím tedy, že pro náš systém není potřeba použití dalších šifrování pro úschovu hesel.

Další možností neoprávněného přístupu do systému by bylo odposlechnutí hesla na počítačové síti. Řešením by bylo použití zabezpečeného přenosu protokolem HTTPS, což by se v případě výskytu citlivějších informací určitě vyplatilo. Dále v kapitole možných rozšíření.

## **6 Možnosti rozšíření**

### **6.1 Pokročilé vyhledávání**

Prvním možným vylepšením by mohlo být pokročilé vyhledávání podle všech parametrů nebo i pomocí fulltextu. Nabízí se zde ale otázka, zda je vůbec potřeba takové vyhledávání. Většina klientů, ne-li všichni, vyhledávají nemovitost podle lokality, tedy města, podle dispozic, ceny a plochy. Mohlo by se také stát, že by se složitějším vyhledáváním stránka zneřehlednila. Je to tedy možnost do budoucna, která se nemusí ukázat tou nejlepší volbou.

### **6.2 Rozšířená editace práv uživatelů**

Možné rozšíření bych viděl i v možnosti administrátora přidělovat detailně různá práva uživatelům. Ani zde není jisté, zda by se tato funkce byla v tomto stavu aplikace opravdu užitečná, jelikož administrátor rozdávat práva může, tudíž má dostatečné možnosti ke spolupráci s jiným zaměstnancem při správě systému. V případě většího rozvoje firmy, návštěvnost třeba přes tisíc lidí denně a několik set nemovitostí měnících se denně, by bylo rozšířené přidělování práv určitě velice užitečnou funkcí.

### **6.3 Diskuzní fórum**

V případě zájmu uživatelů by bylo možné přiimplementovat ještě diskuzní fórum, kde by mohli zástupci firmy odpovídat na časté a veřejné dotazy zákazníků. Diskuzní fórum je bohužel kapitolou samou pro sebe a jeho implementace je dosti časově náročná. Tato varianta by připadala v úvahu při viditelném zájmu zákazníků a obnášela by další složitý vývoj.

### **6.4 Převod vkládaných obrázků**

V konečné fázi vývoje je aplikace schopná nahrávat označené obrázky z klientského počítače přímo do pole v tabulce databáze. Tisk na obrazovku pak probíhá použitým skriptem, který předpokládá, že použitý obrázek je typu JPEG. Možné rozšíření by tedy mohlo být provedeno použitím dalších funkcí PHP, které by byly schopny překonvertovat vkládaný obrázek z různých formátů. Systém je ale navržen pouze pro jeden typ, který je nejběžnějším výstupním formátem z každého digitálního fotoaparátu.

## 6.5 HTTPS

Jak už jsem naznačil v kapitole implementace v části zabezpečení, hrozí zde riziko odchyty hesel po nezabezpečené komunikaci přes počítačovou síť. Řešením by bylo použití zabezpečené šifrované komunikace pomocí protokolu HTTPS. To by ale vyžadovalo další technické prostředky a nejspíš by bylo toto zabezpečení pro systém řešeného typu zbytečné.

## 6.6 Kontrolní kód při registraci

Kontrolní kód k opsání při registraci, který alespoň na základní úrovni stěžuje jednoduchému skriptu neustálýma registracema zahltit systém, je proveden pouze jako řetězec. Tento kód by měl být pro správnou funkci převeden v PHP na obrázek a proložen jinými znaky, které by útočnickovým skriptům zabránily jeho jednoduchému přečtení.

## 7 Závěr

Shrnu-li celou tvorbu, myslím, že byly dodrženy postupy vedoucí ke správně navrženému systému i implementaci. Aplikace se začala rýsovat už při kladení podmínek, přes návrh databáze a návrh diagramu případů použití, přes implementaci, až po testování. Podmínky definované v zadání byly taktéž splněny.

Vzhledem k tomu, že byl systém od začátku navrhován pro konkrétní firmu, která vstupuje na trh s realitami v Moravskoslezském kraji, bylo opravdu dbáno na pečlivé promyšlení všech funkcí, které by měl užitečný systém plnit a tím ulehčovat práci jak administrátorům, tak klientům. Doufám, že například systém sledování realit bude dobře využitelný a díky němu jeden klik zastoupí několikaminutové opětovné hledání těch samých nemovitostí dokola. Stejně tak formulář pro odesílání dotazů, jež nahrazuje e-mailového klienta, kterého v tomto případě nemusíme vůbec spouštět a opět ušetříme čas.

Během implementace jsem měl možnost přiučit se a procvičit si technologie potřebné k tvorbě funkčního webu jako jsou XHTML + CSS, skriptovací jazyky PHP a JavaScript a návrh a užití databáze MySQL. Celková několikaměsíční práce pro mě byla nesmírným přínosem a těším se, až uvidím tento systém zaveden v praxi.

S uvedením do provozu se nese otázka možného rozšiřování do budoucna. V předešlé kapitole jsem se snažil navrhnout pár možných vylepšení, bohužel až praxe ukáže, kde má tato aplikace slabé stránky a pevně věřím, že jich mnoho nebude.

Celá aplikace je validní podle specifikace XHTML 1.1 a splňuje požadavky zadání.

Aplikace je umístěna na internetové adrese <http://www.axatem.cz/ibp>.

# Literatura

- [1] Gutmans, A., Bakken, S. S., Rethans D. *Mistrovství v PHP 5*. Praha, Computer Press, 2005, ISBN 80-251-0799-X
- [2] Welling L., Thomson L. *PHP a MySQL - Rozvoj webových aplikací*. Praha, SoftPress, 2005, ISBN 80-86497-83-6
- [3] Zendulka J., Rudolfová I. *Databázové systémy - studijní opora*, Brno, 2006
- [4] články na serveru <http://www.interval.cz>
- [5] články na serveru <http://www.builder.cz>

# Seznam příloh

Příloha 1. CD s aplikací a skriptem pro vytvoření tabulek v databázi.