

**UNIVERZITA KARLOVA V PRAZE**

**Filozofická fakulta**

**Ústav informačních studií a knihovnictví**

**Studijní program: informační studia a knihovnictví**

**Studijní obor: informační studia**

**BBS. Michal PIVOŇKA, DiS.**

**TVORBA A SPRÁVA KOMERČNÍHO INTERNETOVÉHO PROJEKTU**

**DIPLOMOVÁ PRÁCE**

**Praha 2007**

**Vedoucí diplomové práce:** Doc. RNDr. Jiří Souček, DrSc.

**Oponent diplomové práce:**

**Datum obhajoby:**

**Hodnocení:**

**Prohlášení:**

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje.

V Praze, 8. srpna 2007

.....

Podpis diplomanta

## **Identifikační záznam**

PIVOŇKA, Michal. *Tvorba a správa komerčního internetového projektu [Creation and administration of a commercial Internet project]*. Praha, 2007. 106 s., 5 s. příl. Diplomová práce. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví 2007. Vedoucí diplomové práce Doc. RNDr. Jiří Souček, DrSc.

## **Abstrakt**

Diplomová práce popisuje nejdůležitější metody a postupy používané při analýze, návrhu a implementaci elektronické aplikace. Jsou zde představeny standardizované metody, modely a diagramy určené pro projektování, záznam, vizualizaci a vlastní tvorbu softwarového systému. Popsány jsou i jednotlivé fáze projektu, které následují po jeho implementaci a testování. Jedná se především o správu, podporu a propagaci systému. První kapitola se věnuje významu komplexního návrhu webových aplikací včetně postupu při volbě profesionálního zhotovitele. Druhá kapitola reprezentuje specifikaci a následnou analýzu požadavků na nový systém. Vlastní návrh aplikace popisuje kapitola třetí. Jsou zde zmíněny moderní metody používání speciálních jazyků a technologií (UML, WebML) pro tvorbu modelů a diagramů. Čtvrtá kapitola se zabývá implementací navrženého systému. Popisuje přehled vhodných databází, vývojových prostředí a programovacích jazyků. Následující, pátá, kapitola zmiňuje nejznámější formy testování již hotového systému. Šestá kapitola obsahuje jednotlivé fáze a formy prezentace a správy aplikace. Zabývá se významem doménového jména a hostingu. Základní metody podpory a propagace prezentovaného projektu jsou obsaženy v kapitole poslední, sedmé. Pokud to vyžaduje charakter tématu, jsou jednotlivé kapitoly obohaceny o reálné případové studie, které by měly napomoci lepšímu pochopení dané problematiky. Přílohy práce tvoří definice a struktura smlouvy o dílo a přehledové tabulky nejznámějších českých poskytovatelů webhostingu spolu s přehledem předních českých vyhledávačů a elektronických katalogů.

## **Klíčová slova**

specifikace požadavků, analýza požadavků, návrh webové aplikace, aplikace, prezentace, e-commerce, projektování, databáze, sql, uml, webml, program, podpor, model, diagram, test, doména, hosting, propagace, banner, statistik, seo.

# OBSAH

<b>PŘEDMLUVA</b> .....	<b>7</b>
<b>ÚVOD</b> .....	<b>9</b>
<b>1 VÝZNAM KOMPLEXNÍHO NÁVRHU PROJEKTU A VOLBA ZHOTOVITELE</b> .....	<b>11</b>
1.1 VÝZNAM NÁVRHU .....	11
1.2 VOLBA ZHOTOVITELE .....	12
<b>2 SPECIFIKACE A ANALÝZA POŽADAVKŮ</b> .....	<b>15</b>
2.1 SBĚR POŽADAVKŮ .....	15
2.2 PŘÍPADOVÁ STUDIE SBĚRU POŽADAVKŮ .....	18
2.2.1 <i>Funkční požadavky</i> .....	18
2.2.2 <i>Identifikace uživatelů systému a jejich personalizace</i> .....	19
2.2.3 <i>Požadavky na data</i> .....	20
2.2.4 <i>Ostatní požadavky</i> .....	20
2.3 ANALÝZA POŽADAVKŮ .....	21
2.4 PŘÍPADOVÁ STUDIE ANALÝZY POŽADAVKŮ .....	22
2.4.1 <i>Ukládání vstupních dat</i> .....	22
2.4.2 <i>Layout systému</i> .....	26
2.4.3 <i>Logo a barevné schéma</i> .....	32
<b>3 NÁVRH PROJEKTU</b> .....	<b>35</b>
3.1 MODELOVÁNÍ .....	35
3.2 NÁSTROJE PRO TVORBU NÁVRHU A TVORBY SYSTÉMŮ .....	37
3.3 UML .....	38
3.3.1 <i>Popis jazyka</i> .....	38
3.3.2 <i>Diagramy a modely v UML</i> .....	39
3.3.3 <i>Diagram tříd</i> .....	41
3.3.4 <i>Stavový diagram</i> .....	44
3.3.5 <i>Diagram aktivit</i> .....	46
3.3.6 <i>Use Case diagram</i> .....	49
3.3.7 <i>Business Process Model</i> .....	52
3.3.8 <i>Příklady využití UML</i> .....	56
3.4 WEBML .....	69
3.4.1 <i>Datový model</i> .....	69
3.4.2 <i>Hypertextový model</i> .....	70

<b>4 IMPLEMENTACE PROJEKTU .....</b>	<b>73</b>
4.1 TÝMOVÁ PRÁCE.....	73
4.2 TVORBA DATABÁZE .....	74
4.3 VÝVOJOVÉ PROSTŘEDÍ .....	75
4.4 PROGRAMOVACÍ JAZYK .....	77
4.5 PROJEKTOVÁ DOKUMENTACE.....	78
4.6 PŘÍPADOVÁ STUDIE IMPLEMENTACE .....	79
<b>5 TESTOVÁNÍ A OPTIMALIZACE PROTOTYPU .....</b>	<b>85</b>
5.1 PROGRAMÁTORSKÉ TESTOVÁNÍ FUNKČNOSTI .....	85
5.2 VŠEOBECNÝ ROZBOR .....	86
5.2.1 Analýza nedostatků .....	86
5.2.2 Doporučení úprav.....	86
5.3 AUDIT PŘÍSTUPNOSTI .....	87
5.4 UŽIVATELSKÉ TESTOVÁNÍ POUŽITELNOSTI (user testing).....	87
5.5 HEURISTICKÁ ANALÝZA POUŽITELNOSTI.....	88
5.6 ROZBOR STRÁNEK Z PROHLEDU SEO.....	88
<b>6 PREZENTACE PROJEKTU .....</b>	<b>90</b>
6.1 DOMÉNA.....	90
6.2 HOSTING.....	92
6.2.1 Webhosting.....	92
6.2.2 Serverhosting .....	94
6.3 SPRÁVA.....	95
<b>7 PODPORA A PROPAGACE PROJEKTU .....</b>	<b>96</b>
7.1 REGISTRACE V KATALOZÍCH A VYHLEDÁVÁČÍCH	
7.2 STATISTICKÉ SLUŽBY.....	99
7.3 BANNERY .....	100
7.4 REKLAMNÍ SÍTĚ.....	101
7.5 AFFILIATE PROGRAMY.....	101
<b>8 ZÁVĚR.....</b>	<b>103</b>
<b>SEZNAM POUŽITÉ LITERATURY A ZDROJŮ .....</b>	<b>104</b>
<b>SEZNAM PŘÍLOH .....</b>	<b>110</b>

# Předmluva

## *Motto:*

*„Dvakrát měř a jednou řež!“*

Téma práce (tvorba a správa komerčního internetového projektu) je mi velice blízké. Již několik let se pohybuji ve sféře komerčních internetových projektů, kde se aktivně podílím na jejich návrhu, tvorbě, správě a propagaci. Mé současné pracovní místo (manager elektronického obchodu) spolu s podnikatelskou činností zaměřenou na všechny fáze vývoje webových aplikací mě poznamenalo do té míry, že jsem se stal v tomto oboru profesionálem. V průběhu své praxe jsem ve svém okolí čím dál více narážel na podceňování kvalitní projektové analýzy, návrhu, správy a propagace internetových projektů, která často vedla k nepříjemným, avšak logicky nevyhnutelným důsledkům, což mě nakonec přimělo napsat tuto práci.

Rád bych své doposud nabyté znalosti a zkušenosti ve svém díle nyní zúročil s cílem poskytnout čtenářům srozumitelný popis (a možná i návod) profesionálního postupu při vývoji komerční webové prezentace včetně její propagace.

Práce je rozdělena do sedmi základních kapitol chronologicky reprezentujících jednotlivé fáze vývoje projektu. První kapitola je věnována významu kvalitního návrhu elektronické aplikace včetně výčtu možných hrozeb při jejím nedodržení. Ve druhé kapitole popisuji prvotní vývojovou fázi projektu, kterou je sběr a následná analýza požadavků na budoucí systém. Třetí kapitola reprezentuje vlastní návrh budoucí aplikace a jeho náležitosti. Čtvrtá kapitola čtenáře provede jednotlivými fázemi implementace již navrženého systému. Pátá kapitola je věnována testování a optimalizaci prototypu. V kapitole šesté se zabývám prezentací projektu na internetu a jeho správou. Poslední, sedmou, kapitolu věnuji popisu různých metod podpory a propagace.

K napsání této práce jsem využíval tištěných zdrojů ze své soukromé knihovny i Městské knihovny v Praze. Během přípravy a následného psaní práce jsem leckdy využil teoretických poznatků získaných během studia na Ústavu informačních studií a knihovnictví (ÚISK) FF

UK. Cenné informace a rady mi poskytl můj vedoucí práce Doc. RNDr. Jiří Souček, DrSc., který na výše uvedeném ústavu vyučuje mimo jiné modelování, databázové systémy a značkovací jazyky. Velice cenným zdrojem mi byly odborné elektronické články a ostatní informační zdroje, které jsem vyhledal na internetu. Vybrané části práce jsou obohaceny o případové studie s vlastními konkrétními návrhy na realizaci, což by mělo mít za následek lepší pochopení teoretické části. V neposlední řadě jsem využil svých vlastních znalostí k tomu, abych jednotlivé kapitoly v případě potřeby vhodně informačně doplnil o své poznatky. Veškeré zdroje, ze kterých jsem čerpal, jsou uvedeny v seznamu použité literatury a dalších zdrojů. Citování informačních zdrojů a strukturování bibliografických záznamů jsem provedl v souladu s normami ISO 690 a ISO 690-2. Citované zdroje (nebo jejich úryvky) v práci označuji číslem uvedeným v hranaté závorce (např. [1]). Dané číslo odpovídá číslu záznamu uvedenému v seznamu použité literatury a zdrojů. Čísla, uvedená v horním indexu, odkazují na poznámky pod čarou.

Na závěr cítím potřebu poděkovat své rodině za vytvoření bezvadného pracovního prostředí a za toleranci vůči mým negativním náladám, kterými jsem během psaní této práce občas trpěl.



## Úvod

Význam internetové sítě a její využití v komerční sféře je již ve světě dobře znám. V posledních letech můžeme pozorovat, že i v naší České republice roste využívání internetu a komerčních webových aplikací téměř exponenciálně.

Hlavní příčinou tohoto nárůstu je rozvoj masového budování datových sítí, které umožňují broadbandové (vysokorychlostní) internetové připojení za dostupné ceny, což přimělo doposud váhající jednotlivce si přístup k internetu pořídit. Současný počet tuzemských uživatelů internetu se zvýšil na celkových 4,379 milionu, což je nárůst více jak o pětinu. S internetem tedy již pracuje více než 40 procent obyvatel Česka<sup>1</sup>.

Pro společnosti a podnikatele, kteří své potenciální klienty doposud oslovovali a uspokojovali pouze klasickými metodami (přes kamenné obchody, teleshopping apod.), nyní objevují zcela nový kanál pro akvizici svých potenciálních zákazníků. Důvěra v elektronické nakupování a využívání dalších služeb sítě internet stále stoupá a lze předpokládat, že lidé budou mít čím dál větší tendenci uspokojovat své potřeby (nejen informační) elektronickou cestou.

Všechny výše uvedené faktory by u podnikatelských subjektů měly mít za následek potřebu se na internetu kvalitně a profesionálně prezentovat. Vytvoření webové firemní prezentace, obsahující základní informace o společnosti, jejích zaměření a službách, je dnes standardní záležitostí. S trochou nadsázky můžeme tvrdit, že kdo nemá na webu své stránky, ten jako by nebyl.

Kromě základních webových prezentací, sloužících primárně jako informační zdroj, se na internetu stále více objevují webové aplikace, prostřednictvím nichž můžeme provádět rozmanité úkony, jako je například on-line nakupování, inzerce, diskuse, soutěže atd. Zde se již nejedná o jednoduché několikastránkové prezentace, ale o komplexní internetové projekty. Z hlediska přípravy a návrhu by tyto projekty neměly být v žádném případě podceňovány (což se často děje), jelikož jakékoliv nedostatky či nedorozumění v rámci příprav obvykle předchází následným chybám či nechtěným výsledkům.

---

<sup>1</sup> Informace převzata z výsledků studie agentury Mediaresearch ze dne 24.7.2007

Kromě chybových postupů při tvorbě webových aplikací, se nezdá se setkáváme také se špatnou správou a propagací, která je pro úspěšné fungování projektu stejně důležitá jako jeho vlastní tvorba.

Během jednotlivých fází projektování a správy internetových systémů je tedy zapotřebí kvalitní zpracování podkladů, postupů a vyhodnocení, což by mělo být předmětem této práce.

# 1 Význam komplexního návrhu projektu a volba zhotovitele

Je přirozené, že s rostoucím počtem internetových uživatelů (potenciálních zákazníků), přímo úměrně roste počet internetových aplikací sloužících převážně pro komerční účely. Konkrétně se v poslední době rozrůstá počet elektronických obchodů (e-shopů), které mají za cíl uspokojit zákazníka nejen šířkou a hloubkou nabízeného sortimentu zboží a služeb, ale také se snaží případné elektronické nakupování co nejvíce ulehčit a obohatit o řadu nadstandardních funkcí spojených s nákupem (seznam přání, on-line monitorování objednávky apod.). Jedná se tedy o složitou elektronickou aplikaci s mnoha funkcemi a rozhraními vyžadující kvalitní návrh vypracovaný profesionální řešitelskou firmou.

## 1.1 Význam návrhu

Jak již bylo uvedeno výše, tvorbě složitých systémů by měla předcházet kvalitní analýza, specifikace a návrh. Bohužel stále velký počet zadavatelů elektronických aplikací tuto část výrobního cyklu podceňuje a nevěnuje jí příliš velkou (respektive žádnou) pozornost, což má za následek vznik různých problémů vyplývajících z nedorozumění mezi zadavatelem a zhotovitelem.

Projektování a programování internetových aplikací a informačních systémů se dá s trochou nadsázky přirovnat k jednomu z nejstarších umění - umění budovat, tedy stavět [1]. V minulosti lidé představovali autora i interpreta, architekta i stavitele v jedné osobě. Uměli tedy vytvořit návrh vzhledu budoucí stavby a zároveň ji uměli vytvořit. Od nepaměti se také pro stavbu používaly plány, modely a další nástroje potřebné k úspěšné realizaci.

S rozvojem mechaniky, statiky, nauky o materiálech a dalších technických disciplín se profese stavitelů rozdělila na dvě části. Někteří se začali zabývat pouze navrhováním vzhledu a vnitřního uspořádání staveb a druzí zajišťovali vlastní realizaci staveb právě podle výše uvedených plánů a návrhů.

K obdobné změně, ke které došlo v minulosti u stavitelství, dochází v posledních letech i u profese tvůrců informačních systémů a aplikací. Stále více se vyhraňují dvě vzájemně spolupracující profese: návrháři architektury projektů a vlastní realizátoři těchto návrhů (to

však neznamena, že nebudou nadále existovat společnosti, které poskytují návrh i implementaci projektu v jednom řešení). Je však nutné poznamenat, že tyto dvě profese ve světě tvorby informačních systémů zdaleka nedosáhly takové úrovně, jak tomu bylo a je u stavitelů budov. Velmi často se etapy návrhu a realizace stále ještě vzájemně prolínají a mnohokrát se stává i to, že během výstavby informačního systému/aplikace dochází k zásadním změnám celého projektu. Tyto problémy přímo souvisejí s absencí projektového návrhu a nedostatečnými nebo mylnými informacemi o jeho funkčnosti. Pravidlem potom bývá překročení časových i finančních plánů a není výjimkou, že se započatá stavba informačního systému vůbec nedokončí. Pokud je zhotoviteli aplikace předložen komplexní návrh projektu včetně jednotlivých diagramů a modelů, pak je zcela srozuměn se vzhledem/funkčností aplikace a případné nesouvislosti může vyřešit ještě v teoretické rovině projektování, což je samozřejmě mnohem méně finančně náročné.

Jestliže tedy software neprojektujeme způsobem krátkých implementací a následného intenzivního testování (tzv. extrémní programování), je teoretická příprava více než žádoucí. U tvorby rozsáhlejších informačních systémů a aplikací je absence projektového návrhu nemyslitelná.

## 1.2 Volba zhotovitele

Výběr kvalitního zhotovitele návrhu webové aplikace hraje velice důležitou roli. Můžeme mít zcela jasné představy o cíli a funkčnosti našeho budoucího projektu, nicméně pokud se role realizátora těchto představ zhostí nekvalitní nebo diletantská firma, zhotovení návrhu bývá často vágní a pro programátorskou firmu nepoužitelné.

Při výběru vhodné firmy bychom se měli řídit následujícími pravidly doporučenými Asociací dodavatelů internetových řešení [31]:

- **Dostatečný obrat společnosti:** roční obrat špičkového poskytovatele internetových řešení by měl být alespoň 10 mil. Kč ročně z poskytování webových služeb.

- **Nezávislá klientela:** poskytovatel nesmí pracovat pro klienta, který se podílel na celkovém ročním obrátu poskytovatele více než 50 %. Poskytovatel by neměl být vlastněn ani spoluvlastněn žádným ze svých klientů.
- **Dostatečný počet úspěšně dokončených projektů:** měli bychom se informovat o počtu a stavu dokončených projektů, které mnohdy prozradí vše o stylu a kvalitě práce poskytovatele.
- **Ověřitelné reference:** vyžádejte si konkrétní kontakty na klienty poskytovatele za účelem potvrzení prezentované reference.
- **Struktura klientů:** struktura klientů často napoví o zaměření a specializaci konkrétního poskytovatele.
- **Oborová řešení:** poskytovatel by měl nabízet případové studie, ve kterých dokáže svou znalost daného oboru a prokáže tak přínos referenčního řešení.
- **Právníká osoba a plátce DPH:** i student může dobře navrhnout/zpracovat technické řešení, ale se svým živnostenským listem nemůže nabízet profesionální servis na odpovídající úrovni. Poskytovatel by měl být registrovaným plátcem DPH.
- **Vlastní technické zázemí:** měli bychom se zajímat o HW a SW prostředí, ve kterém jsou vyvíjeny svěřené projekty.
- **Provoz www aplikací na vlastních serverech:** profesionální poskytovatel provozuje vlastní servery, případně využívá jejich pronájem, a to vždy s připojením na páteřní síť u poskytovatele připojení k internetu. Příliš levný webhosting s připojením pod 100 Mb/s není vhodný pro provoz profesionálních aplikací.
- **Dostatečný počet vlastních kmenových zaměstnanců:** pokud má poskytovatel 5 zaměstnanců, stěží kvalitně zvládá více jak jeden projekt. Vyžadujte proto minimálně 15 kvalifikovaných zaměstnanců. Společnost s jedním manažerem, která zprostředkovává práci najatým začátečníkům, nemůže zajistit dostatečně kvalitní výsledek.

- **Servis 24/7:** profesionální servisní podpora je základem úspěšné spolupráce a je nutné ji zajišťovat 24 hodin denně a 7 dní v týdnu i po dobu dovolených či nemocí.
- **Profesionální konzultace a know-how:** problém se může mnohdy vyřešit ještě před dokončením projektu pouze tím, že podobný problém již v minulosti nastal.
- **Spolupráce s odborníky v daném oboru:** projekt vždy řídí odborník na danou oblast problematiky, který je náležitě obeznámen s prostředím klienta.
- **Využívání nových technologií:** profesionální poskytovatel sleduje trendy (někdy je i určuje) a vždy nabízí využívání nových, moderních řešení. Zároveň však pracuje na bezpečných technologiích.

Je žádoucí, aby byla mezi zadavatelem a zhotovitelem sepsána řádná smlouva. Tímto si obě strany přesně stanoví předmět práce, termíny, případné sankce a další náležitosti, čímž se vyvarují možných budoucích nedorozumění a nejasností. Přílohy č. 1 a 2 obsahují definici a strukturu takovéto smlouvy (smlouvy o dílo).

## 2 Specifikace a analýza požadavků

Jednou z klíčových věcí je pochopení potřeb zadavatele a umění přeložit specifikaci jeho potřeb z jazyka uživatelského do jazyka tvůrce a realizátora aplikace. Uživatel a tvůrce mají totiž každý jinou kvalifikaci, jiné zázemí, jiné vzdělání a zkušenosti. A právě na překlenutí této mezery mezi reálným světem uživatelů/zadavatelů a světem počítačů, jsou zaměřeny metody návrhů systémů.

### 2.1 Sběr požadavků

Postup při vývoji webových aplikací je v dnešní době již záležitostí určité konvence v podobě obecného standardu/schématu. Životní cyklus projektu se dělí do následujících fází:

- sběr a specifikace požadavků
- analýza
- návrh
- implementace
- testování
- nasazení a údržba

Aby mohla být započata práce na návrhu a projektování elektronické aplikace, musí být stanoveny přesné a podrobné požadavky na budoucí systém. Kvalitní specifikace požadavků tedy přímo souvisí s úspěchem celého návrhu a implementace webové aplikace.

Fáze sběru a specifikace požadavků je velice často podceňována. Případná modifikace nebo rozšíření zadání může ve finále vést k změně datového modelu a zdrojového kódu, což často stojí nemalé peníze a čas.

Požadavky na budoucí software určuje sám zadavatel, a to buď formou verbální, písemnou nebo kombinací obojího. Písemná specifikace může zároveň sloužit jako podklad pro výběrové řízení na budoucího tvůrce projektu a většinou má velice vysokou úroveň kvality a komplexnosti. Po předložení písemné formy specifikace zhotoviteli zpravidla následuje

ústní pohovor, ve kterém se eliminují případné nejasnosti, popřípadě dochází k modifikaci/rozšíření stávající specifikace o další požadavky.

Výstupy vzniklé z daného rozhovoru jsou pak sepsány do řádné písemné specifikace, která již většinou nepotřebuje žádné úpravy.

Zadavatel by měl mít alespoň základní představu o svém budoucím projektu. Proces stanovení požadavků na nový systém, aniž by měl zadavatel alespoň nějakou představu a požadavky, bývá zpravidla zdlouhavý a neefektivní.

Součástí základních úvah by mělo být:

- určení cíle a účelu budoucí prezentace/aplikace
- určení cílové skupiny pro kterou se bude webová prezentace/aplikace vytvářet
- definice cíle webové prezentace/aplikace z hlediska marketingu
- alespoň částečná definice struktury a obsahu našich budoucích stránek
- provedení grafického designu (bude grafický design na internetu odrážet design například tištěných prezentací?)
- podklady pro tvorbu webové prezentace (grafické prvky, databáze, text)
- určení preferovaného softwaru (při komunikaci s návštěvníkem stránek)
- získání informace o tom, jestli půjde o jednorázové řešení tvorby prezentace/aplikace bez dalších budoucích rozvoju či úprav
- požadavky na registraci internetové domény a webhostingu



- tvorba požadovaného harmonogramu projektu
- určení týmu pracovníků, kteří budou komunikovat při tvorbě webové prezentace s týmem vývojářů
- určení maximální výše rozpočtu, případné naplánování splátkového kalendáře

Veškeré požadavky, potřebné pro návrh budoucího systému, členíme do následujících skupin:

**Funkční požadavky:** jedná se o základní druh požadavku, který tvoří korpus celé specifikace. Obsahuje popis vhodné architektury aplikace, modularity a funkcí, které musí systém realizovat. Pro obrazové vyjádření tohoto druhu specifikace je ideálním nástrojem tzv. *Use Case*<sup>2</sup> diagram (tento typ diagramu je podrobně představen v kapitole 3.3.6).

**Identifikace uživatelů systému:** v této fázi se určují všechny typy uživatelů budoucí aplikace. Opět se zde dá využít diagramu typu *Use Case*.

**Požadavky na personalizaci:** určují, zdali budou všechny služby pro všechny typy uživatelů identické, nebo zdali budou určité funkce a služby dostupné pouze určité skupině uživatelů. Jedná se zde například o přístupová práva do určitých sekcí aplikace.

**Požadavky na data:** specifikují, jaká data bude aplikace uchovávat a následně využívat. Podrobný výčet dat a jejich vztahů se poté stanovuje v rámci datového modelování.

**Ostatní požadavky:** zpravidla se zde určují požadavky na celkovou transparentnost a uživatelskou přívětivost systému.

---

<sup>2</sup> Grafické zobrazení dynamické struktury systému z pohledu uživatele

Dalším požadavkem může být výkonnost a dostupnost aplikace. Výkonnost je nesmírně důležitá, pokud se počítá s přístupem velkého množství uživatelů daného systému. Je nežádoucí, aby se aplikace tzv. zahltla požadavky, což může mít za následek zpomalení procesu vyřizování požadavků nebo dokonce ukončení aplikace. Dostupností je míněna doba, kdy aplikace funguje zcela bez problémů. Stoprocentní dostupnost lze jen velmi těžko zajistit, a tak je standardem dostupnost kolem 99%. Abychom eliminovali riziko ztráty dat (a tím i dostupnosti), je vhodné veškerá data pravidelně zálohovat.

Velice důležitá je bezpečnost systému. Cílem bezpečnostního požadavku je například ochrana osobních dat uživatelů, zabezpečení přístupových hesel nebo šifrování datového přenosu.

Standardním požadavkem by měla být i tzv. rozšiřitelnost neboli možnost rozšiřovat aplikaci o další doplňkové funkce, které budou dříve či později zapotřebí. U elektronických obchodů musíme například počítat s možností přidání dalších kategorií a subkategorií až do n-té úrovně (podle přání zadavatele).

## **2.2 Případová studie sběru požadavků**

Výše uvedené poznatky se nyní pokusím převést na konkrétní projekt. Jako vzorový systém si zvolíme internetovou inzerci zaměřenou na nákup a prodej armádního zboží.

Cílem budoucího systému je vytvořit internetový inzertní systém, který by v reálném čase zobrazoval veškeré vložené (tematicky řazené) nabídkové/poptávkové inzeráty.

Systém by měl být využíván všemi uživateli internetu, kteří chtějí prodat nebo koupit nějaký produkt armádního zaměření.

Hlavní požadavky budou následující:

### **2.2.1 Funkční požadavky**

- zpracovávat, ukládat do databáze, prezentovat a mazat inzeráty
- vyhledávat jednotlivé inzeráty podle klíčových slov
- zakládat diskuse nebo do již existujících diskusí přispívat

- umožnit odesílat odpovědi na soutěže (s využitím elektronického formuláře) a posléze je vyhodnocovat
- umožnit zasílat rychlé reakce/dotazy na jednotlivé inzeráty
- umožnit zasílat dotazy/připomínky správcům serveru prostřednictvím formuláře
- zasílat pravidelné e-maily registrovaným uživatelům
- umožnit uživatelům registraci do systému a následné přihlašování

## **2.2.2 Identifikace uživatelů systému a jejich personalizace**

Jednotlivé role uživatelů včetně jejich práv a omezení jsou vypsány níže.

### ***Neregistrovaní uživatelé/inzerenti***

- mají možnost podávat plnohodnotný inzerát
- mají možnost kdykoliv smazat inzerát
- nemají možnost soutěžit o ceny
- nemají možnost zakládat nebo přispívat do diskusí

### ***Registrovaní uživatelé/inzerenti***

- mají možnost podávat plnohodnotný inzerát
- mají možnost svůj vložený inzerát kdykoliv modifikovat a smazat
- po přihlášení se jim při vkládání inzerátu automaticky vyplní kontaktní údaje
- mají možnost soutěžit o ceny
- mají možnost zakládat nebo přispívat do diskusí

### ***Obchodní partner systému***

- má stejná práva jako registrovaný uživatel (viz výše)
- má přístup ke statistikám

### ***Administrátor systému***

- má možnost mazat inzeráty (včetně těch, které sám nepodal)
- má přístup ke kontaktům na jednotlivé inzerenty
- má přístup k vyhodnocení soutěží
- má možnost mazat celá diskusní témata nebo jednotlivé příspěvky

### 2.2.3 Požadavky na data

Jako nástroj pro uchování dat využijeme relační databázi.

Do databáze se budou ukládat všechna data, která uživatel zadá prostřednictvím jednotlivých webových formulářů.

Primárním zdrojem dat bude vyplnění formuláře pro zakládání nových inzerátů. Bude se tedy jednat především o předmět a popis nabízených/poptávaných předmětů včetně inzerentových kontaktních údajů.

Další data budou získávána v rámci registrace uživatelů, za účelem využívání diskusního fóra nebo v rámci soutěží.

Jednou měsíčně se budou veškerá data z databáze exportovat na externí paměťová media (nejlépe CD/DVD).

Inzeráty starší 6 měsíců budou automaticky z databáze smazány.

### 2.2.4 Ostatní požadavky

**Použitelnost:** Celý inzertní systém by měl být založen na jednoduchém způsobu vkládání plnohodnotných inzerátů bez nutnosti se registrovat. Veškeré funkce aplikace by měly být natolik transparentní, aby uživatel neměl žádné problémy se v systému orientovat a pomocí intuitivního rozhraní s ním pracovat.

Prezentace inzerátů by neměla být zavádějící/matoucí. Jednotlivé inzeráty se budou řadit podle jejich charakteru do předem stanovených oblastí (komodit).

Cílem je tedy srozumitelnost a atraktivnost aplikace.

**Výkonnost:** Mělo by se dosáhnout hodnoty 100 požadavků za vteřinu.

**Dostupnost:** Aplikace by měla být zcela funkční na 99%.

**Bezpečnost:** V rámci ochrany citlivých dat (podle zákona o ochraně osobních údajů [45]) budou data přenášena v rámci šifrovaných přenosů. Veškeré přístupy do databáze budou pečlivě ověřovány.

**Rozšiřitelnost:** Aplikace by měla bez problémů akceptovat nastavby a inovace v podobě funkcí i celých přídatných modulů.

## 2.3 Analýza požadavků

Všechny výše uvedené informace se musejí sjednotit do určitých logických výstupů, ze kterých poté budou čerpat vývojáři budoucí elektronické aplikace (většinou se k tomu využívá jazyka UML).

V této fázi bychom měli stanovit, jaké informace budeme chtít prostřednictvím vstupních formulářů ukládat. Jako optimální nástroj pro určení případu užití jednotlivých skupin slouží *Use Case* modelování. Jednotlivé případy užití specifikují účel, vstupní a výstupní hodnoty a samotný průběh užití.

Důležitou součástí analýzy požadavků je vytvoření tzv. datového slovníku. Jedná se o identifikaci jednotlivých objektů a relace mezi nimi. Každý objekt by měl obsahovat informace jako je název, stručný popis, atributy objektu a vztahy k ostatním objektům.

Pokud jsou nám známy všechny uživatelské skupiny (viz kapitola 2.2.2), které budou s aplikací pracovat a známe případy jejich užití včetně datového slovníku, můžeme vytvořit mapu webu. Mapa/mapy webu představují pohled na webovou aplikaci z různých uživatelských pohledů.

V rámci analýzy požadavku na budoucí web by neměl chybět jeho grafický návrh. Určité podklady jako je například logo, barva atd. může dodavatel dodat již hotové v rámci své corporate identity<sup>3</sup>. V každém případě bychom měli od zadavatele získat informace o:

---

<sup>3</sup> Jednotný vizuální styl firmy

- Layoutu (rozvržení jednotlivých prvků) prezentace
- Logu a jeho umístění v prezentaci
- Menu, jeho typu, umístění a případné úrovně
- Barevném schématu
- Jednotlivých typech písma (fonty).

## 2.4 Případová studie analýzy požadavků

### 2.4.1 Ukládání vstupních dat

- **Ukládání dat spojených s vkládáním inzerátu**

Ke vkládání inzertních dat bude docházet prostřednictvím elektronického formuláře o následujících prvcích:

#### *Formulářové prvky „Radio button“*

- definují, zdali se jedná o nabídkový nebo poptávkový inzerát
- možnost zvolení pouze jedné z možných variant „Nabídka“ nebo „Poptávka“

#### *Textové pole „Jméno“*

- obsahuje informaci o uživatelském jménu, které v případě potřeby slouží jako podklad (spolu s níže uvedeným heslem) pro smazání daného inzerátu
- maximální počet znaků je 20
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

#### *Textové pole „Heslo“*

- obsahuje informaci o uživatelském heslu, které v případě potřeby slouží jako podklad (spolu s výše uvedeným jménem) pro smazání daného inzerátu.
- maximální počet znaků je 20
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)
- vyplněná hodnota pole by měla být ve formuláři převedena do hvězdiček

#### *Roletové menu „Kategorie“*

- nabízí možnost si zvolit příslušnou oblast pro inzerci
- pro jeden inzerát se může zvolit pouze jedna kategorie

#### ***Textové pole „Předmět inzerátu“***

- reprezentuje nadpis inzerátu
- maximální počet znaků je 20
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

#### ***Textové pole „Obsah inzerátu“***

- reprezentuje obsah inzerátu
- maximální počet znaků je 2000
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

#### ***Textové pole „E-mail“***

- obsahuje název e-mailového účtu inzerenta
- maximální počet znaků je 100
- textové pole musí obsahovat masku, která bude zajišťovat správnost obsahu i struktury vložené hodnoty
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu) pouze tehdy, je-li hodnota textového pole „Telefon“ prázdná

#### ***Textové pole „Telefon“***

- obsahuje telefonní číslo inzerenta
- maximální počet znaků je 15
- hodnota musí mít pouze numerický charakter
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu) pouze tehdy, je-li hodnota textového pole „E-mail“ prázdná

#### ***Textové pole „Obrázek“***

- obsahuje možnost si zvolit textový odkaz na obrázek, který se poté nahraje (uploaduje) k nám na server

### **• Ukládání dat spojených s registrací uživatele**

Pokud se uživatel rozhodne pro registraci, zobrazí se mu formulář s následujícími prvky:

#### ***Textové pole „Jméno“***

- reprezentuje uživatelské budoucí přihlašovací jméno, které je nutné vyplnit při přihlašování

- maximální počet znaků je 20
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

***Textové pole „Heslo“***

- obsahuje informaci o uživatelském heslu, které je nutné vyplnit při přihlašování
- maximální počet znaků je 20
- minimální počet znaků je 5
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

***Textové pole „Heslo znovu“***

- obsahuje potvrzení uživatelského hesla, které je nutné vyplnit při přihlašování
- pole musí mít stejnou hodnotu jako pole „Heslo“
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

***Textové pole „E-mail“***

- obsahuje název e-mailového účtu inzerenta
- maximální počet znaků je 100
- textové pole musí obsahovat masku, která bude zajišťovat správnost obsahu i struktury vložené hodnoty
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu) pouze tehdy, je-li hodnota textového pole „Telefon“ prázdná

***Textové pole „Telefon“***

- obsahuje telefonní číslo na inzerenta
- maximální počet znaků je 15
- hodnota musí mít pouze numerický charakter
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu) pouze tehdy, je-li hodnota textového pole „E-mail“ prázdná

***Formulářové prvky „Radio button“***

- definují, zdali si registrovaný uživatel přeje zasílat novinky na e-mail či nikoliv
- možnost zvolení pouze jedné z možných variant „ANO“ nebo „NE“



- **Ukládání dat spojených s diskusí**

Formulář pro založení nového tématu obsahuje následující prvky:

***Textové pole „Název tématu“***

- obsahuje název tématu diskuse, který se bude zobrazovat v základním přehledu
- maximální počet znaků je 20
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

***Textové pole „Obsah tématu“***

- reprezentuje obsah tématu diskuse
- maximální počet znaků je 500
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

Formulář pro vložení příspěvku k určitému tématu obsahuje následující prvky:

***Textové pole „Příspěvek“***

- reprezentuje obsah příspěvku do diskuse
- maximální počet znaků je 500
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

- **Ukládání dat spojených se soutěžením**

Formulář pro vložení odpovědi na soutěže bude následující:

***Textové pole „Odpověď“***

- reprezentuje textovou odpověď
- maximální počet znaků je 500
- pole musí být vždy vyplněno (nesmí mít prázdnou hodnotu)

***Formulářové prvky „Radio button“***

- reprezentuje odpověď, kterou si uživatel zvolil
- možnost zvolení pouze jedné z možných variant „Nabídka“ nebo „Poptávka“

### ***Formulářové prvky „Check box“***

- Reprezentuje odpověď, kterou si uživatel zvolil.
- Možnost zvolení jedné i více odpovědí z možných variant.

## **2.4.2 Layout systému**

Grafické rozvržení systému bude uspořádáno pomocí sloupců a panelů (viz obrázek č. 1). Prezentace se bude vertikálně dělit do 3 sloupců o určitém počtu panelů (panely v pravém a levém sloupci budou ve všech úrovních zobrazení identické).

- **Úvodní (indexová) stránka prezentace**

Levý sloupec bude obsahovat následující panely:

### ***Menu***

- bude obsahovat jednotlivé kategorie inzerce
- bude obsahovat odkaz na diskusi
- bude obsahovat odkaz na soutěže
- bude obsahovat odkaz spolupracující servery
- bude obsahovat kontakt

### ***Nový inzerát***

- bude obsahovat odkaz na formulář pro vložení nového inzerátu

### ***Vyhledávání***

- bude obsahovat formulář pro vyhledávání

### ***Reklama***

- bude obsahovat reklamní obrázek (banner)

Pravý sloupec bude obsahovat následující panely:

### ***Novinky***

- bude obsahovat novinkové informace

### ***Vzkazy***

- bude obsahovat formulář pro zasílání vzkazů a připomínek od návštěvníků

### ***Návštěvy***

- bude obsahovat grafický přehled základních statistik návštěvnosti aplikace

### ***Reklama***

- bude obsahovat reklamní obrázek (banner)

Střední sloupec v základním (úvodním) pohledu bude obsahovat následující panely:

### ***Reklama***

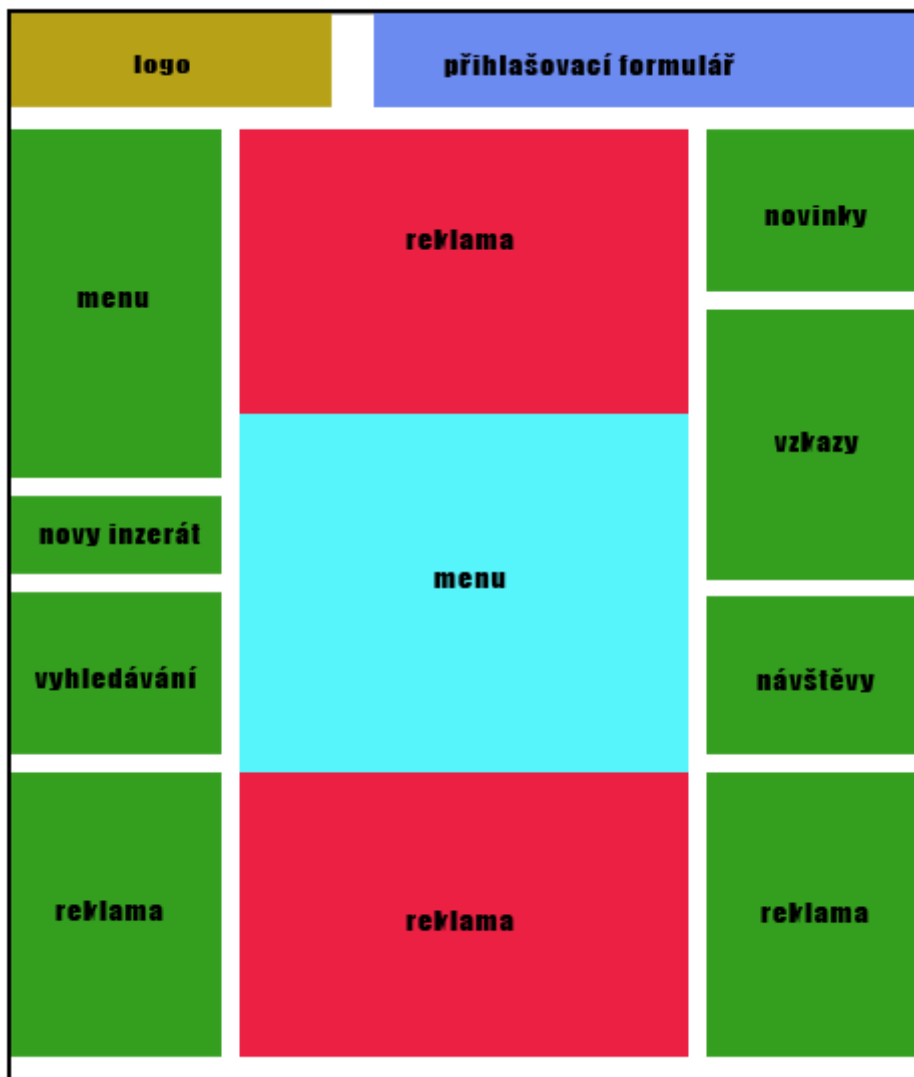
- bude obsahovat reklamní obrázek (banner)

### ***Menu***

- bude obsahovat grafické zobrazení jednotlivých kategorií inzerce

### ***Reklama***

- bude obsahovat reklamní obrázek (banner)



Obr. č. 1: Grafické rozvržení úvodní stránky

- **Úroveň prezentace zobrazující náhled inzerátů**

Grafické rozvržení této části se zobrazí poté, co si uživatel zvolí jednu z kategorií inzerce. Layout je identický s úvodní stránkou kromě střední části. V prostředním panelu se budou zobrazovat odkazy, pomocí kterých uživatel bude moci filtrovat charakter inzerátu (nabídka, poptávka, vše). Pod odkazy se zobrazí 10 nejnovějších inzerátů v rámci zvolené kategorie. Řazení těchto inzerátů bude vždy sestupně (podle data vložení) pod sebou. Na konci prostředního panelu se zobrazí odkazy na další (starší) inzeráty.



Obr. č. 2: Grafické rozvržení stránky s náhledy inzerátů

Jednotlivé náhledy inzerátů budou obsahovat následující údaje:

- obrázek k inzerátu
- věc (předmět inzerátu)
- zkrácený obsah inzerátu (250 znaků)
- e-mail inzerenta (pokud byl při vkládání inzerátu uveden)
- telefon inzerenta (pokud byl při vkládání inzerátu uveden)
- datum vložení inzerátu (ve formátu DD:MM:RR HH:MM)
- ikonu pomocí které se vyvolá formulář pro smazání inzerátu
- ikonu pomocí které se vyvolá detailní zobrazení inzerátu

Pro lepší ilustraci rozvržení jednotlivých informací v rámci náhledu inzerátu poslouží níže uvedený obrázek.



Obr. č. 3: Grafické rozvržení jednotlivých informací v náhledu inzerátu

- **Úroveň prezentace zobrazující detail inzerátů**

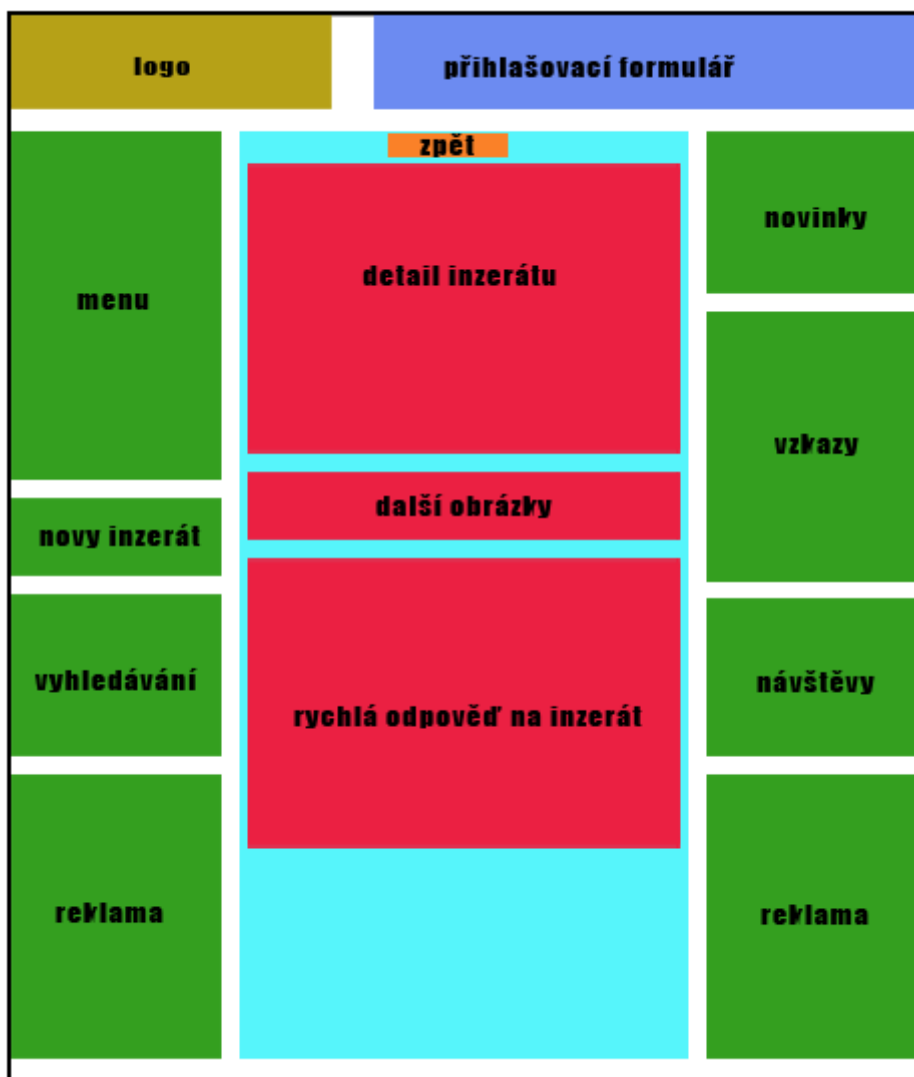
Detail inzerátu se zobrazí, pokud si uživatel zvolí (klikne na příslušnou ikonu nebo na zkrácený text inzerátu) určitý inzerát. Rozmístění grafických panelů v levém i pravém sloupci webové stránky je identické s rozvržením úrovně pro zobrazení náhledů inzerátů.

Ve středním panelu se nahoře zobrazí odkaz, pomocí kterého se uživatel vrátí zpět na úroveň náhledů inzerátů.

Pod odkazem bude uveden panel reprezentující detail inzerátu. Rozvržení jednotlivých informací je stejné jako u náhledu, jen obsah sdělení inzerátu bude nyní úplný.

Další blok zobrazí doplňující fotografie k inzerátu. Obrázky budou řazeny vedle sebe.

Poslední panel bude obsahovat formulář pro okamžitou komunikaci s inzerentem. Následující obrázek by měl pomoci lepě pochopit výše uvedený popis.



Obr. č. 4: Grafické rozvržení stránky s detailem inzerátu

- **Úroveň prezentace zobrazující názvy založených diskusí**

Pokud si uživatel v menu zvolí diskusi, zobrazí se mu přehled všech dosud založených diskusních témat (viz obrázek č. 5). Jednotlivá témata se zobrazí pod sebou. Neaktuální témata budou automaticky mazána na úkor témat aktuálních (nebude existovat žádné stránkování na starší témata). Nad zveřejněnými tématy bude uveden odkaz, prostřednictvím kterého se zobrazí formulář pro založení tématu nového.

Zvolením (kliknutím) určitého tématu se zobrazí jeho detailní pohled včetně příspěvků a formuláře pro vkládání příspěvků (viz obrázek č. 6).

Příspěvkový formulář bude umístěn zcela nahoře. Pod ním se zobrazí přehled vložených příspěvků. Zobrazovat se může maximálně 5 záznamů na jednu stránku. Pokud je počet příspěvků k danému tématu vyšší jak 5, zobrazí se stránkování na příspěvky starší.



Obr. č. 5: Grafické rozvržení stránky s diskusními tématy



Obr. č. 6: Grafické rozvržení stránky s příspěvky

Postup při definici struktury a obsahu ostatních stránek prezentace (stránka o soutěžích, stránka s odkazy na spolupracující servery apod.) bude obdobný, jako u výše uvedených stránek.

### 2.4.3 Logo a barevné schéma

- **Logo aplikace**

Logo inzertního systému se skládá z vlastního názvu budoucí aplikace (v našem případě Armyburza), kde první písmeno „A“ je součástí obrázku trumpetky s vlajkou. Součástí loga je oblouk, který je umístěn nad textem (viz obrázek č. 7).

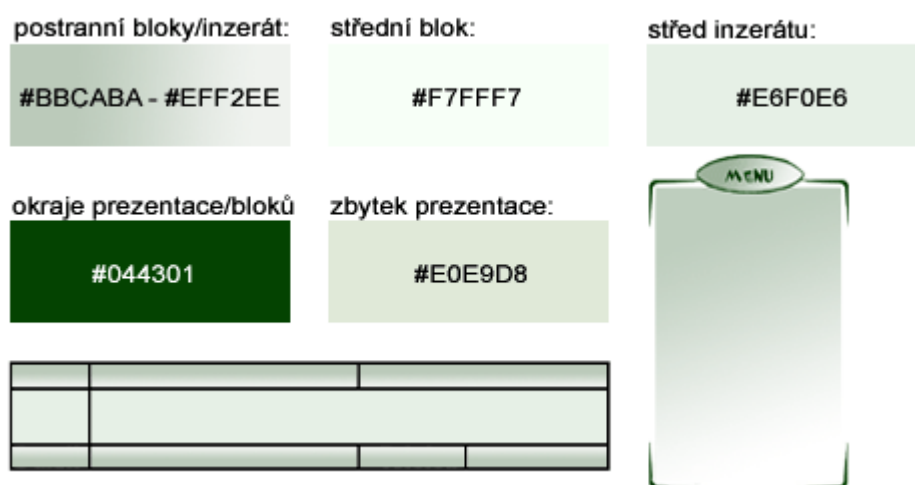




Obr. č. 7: Grafické logo

- **Barevné schéma aplikace**

Všechny barvy budou tvořeny různými odstíny zelené. Jednotlivé bloky v pravém i levém sloupci budou mít barevné pozadí tvořené přechodem dvou odstínů zelené barvy. Stejné přechody se aplikují při zobrazení náhledu a detailu jednotlivých inzerátů. Na spodním obrázku jsou všechny použité barvy znázorněny včetně jejich hexadecimálního ekvivalentu.



Obr. č. 8: Barevné schéma

## Typy písma

Aplikace bude využívat následujících třech typů písma:

**Whimsy TT:** bude aplikován na tvorbu nadpisů jednotlivých panelů

***Arial - Bold Italic:*** je určen pro nadpisy jednotlivých tlačítek

***Verdana:*** tímto fontem bude psán veškerý další text (menu, inzeráty, diskuse, soutěž apod.)

## 3 Návrh projektu

Vyčerpávající představu o složitém informačním systému, který má být realizován nebo který vyžaduje přestavbu, nezískáme většinou jeho slovním (byť dokonalým) vylíčením. Již v historii můžeme nalézt poučení, že pro práce, postavené na abstraktním myšlení, jsou výhodné modelování a simulace. Abychom mohli něco simulovat, musíme mít model.

### 3.1 Modelování

Výraz model známe například z architektury, kde modely projektovaných staveb slouží k názorné prezentaci vytvořeného díla. Jiný význam má model pro malíře či sochaře, kde je jakousi předlohou/reálným vzorem. Za „model“ přípravy určitého jídla bychom pak mohli vydávat též třeba kuchařský předpis. Příslušný výsek mapy se může brát také jako určitý model krajiny.

Z výše uvedeného textu vyplývá fakt, že modelování je jakousi myšlenkovou imitací, abstrakcí, reprodukcí reálně existujícího systému pomocí speciálně konstruovaných modelů. Modelování je tedy jednou z forem poznání a zvláštním prostředkem reprodukce reality. Vztah modelu a objektu je vztahem shody, ne však totožnosti. Model je v určitém vztahu podobný objektu, v jiných vztazích se však od něj nutně liší. A právě pro to, že se model od originálu liší, vzniká možnost vydělit určité podstatné vztahy, snažit se měnit podmínky, hledat určitá řešení, tedy pracovat s modelem tak, jak je s originálem nemožné. Je-li model příliš přesný, ztrácí svůj smysl, je-li příliš nedokonalý, může být zdrojem chyb.

Zkoumání reality je možné myšlenkovými postupy, které lze rozdělit do dvou skupin [1]. Za reprezentanta první skupiny považujeme postupy deduktivní, u kterých učiněný závěr vyplývá z výchozích předpokladů (premis), přičemž odvozování je jisté, nikoliv jen pravděpodobné. Jsou-li tedy pravdivé předpoklady, je pravdivý i sám závěr. Druhou skupinou myšlenkových postupů je indukce. Při indukci se shromažďuje empirický materiál (materiál získaný na základě zkušenosti), na jehož základě se vytvářejí obecné teorie, hypotézy. Při induktivním postupu závěr nemusí nutně vyplývat z výchozích vět či tvrzení, takže výsledek, vzniklý induktivním postupem na pravdivých faktech, může být pravdivý i nepravdivý.

Dedukci je možné provádět přesněji, je-li teorie (v našem pojetí model) vyjádřena pomocí formalizovaného jazyka (například jazyka matematiky). Pak je tyto dedukce možné provádět na základě pravidel daného jazyka. Formalizovaným jazykem nemusí být jen jazyk matematiky. Formalizace v podstatě znamená, že je definováno, jakým způsobem se smějí utvářet symboly a jaká jsou pravidla odvozování. Z tohoto pohledu je formalizovaným jazykem například jazyk UML.

Modelovat nemusíme jen existující procesy a objekty reálného světa. Důležité uplatnění modelů je zkoumání chování objektů, které ještě neexistují, které se zatím pouze projektují. Funkční model informačního systému/aplikace nám například umožní zjistit, zda vytvářený systém bude dělat to, co zadavatel potřebuje a chce. Umožní nám v dialogu s uživatelem zjistit, kde jsme při jeho projektování něco opomněli.

Chceme-li vytvořit funkční model vytvářené aplikace, znamená to v přesném, formalizovaném tvaru popsat svou představu o procesech a struktuře budoucího informačního systému. Předjdeme tím různým nesrovnalostem, které by mohli bez modelování vzniknout. Proto je při návrhu informačních systémů velmi vhodné začít nejprve vytvářet model jako funkční prototyp budoucího systému.

Bez modelu (v papírové či elektronické podobě) můžeme vytvořit nenáročný program pro evidenci titulů domácí filmotéky. U složitějších informačních systémů se však bez modelů jen těžko vyhneme problémům vzniklých na základě špatné/nepřesné specifikace. Existuje řada děsivých případů z praxe, kdy například jistá finanční instituce požadovala tvorbu informačního systému s nejasným zadáním na realizaci pojištění nemovitostí. Podpůrná databáze byla k dispozici plně funkční stejně tak jako návrhy obrazovek (formulářů) určených pro styk s uživateli, informační model však neexistoval. Programátoři poté vytvořili program, který však rozumně nefungoval. V této kritické situaci firma zakoupila CASE nástroj<sup>4</sup>, ale ani ten nepomohl. Teprve pak se nastoupilo na správnou cestu: analýza, návrh, realizace. Celá tato zkušenost stála společnost kolem 2. milionů USD [1].

Pokud bychom chtěli předcházet rizikům spojeným s tvorbou informačních systémů, měli bychom se držet určitých pravidel:

---

<sup>4</sup> Podpůrný nástroj při vývoji software

- Čím rozlehlejší a komplikovanější je navrhovaný a realizovaný systém, tím více vzniká potřeba plánovat, projektovat, modelovat a simulovat.
- Pokud nebudeme podrobně modelovat, měli bychom alespoň pečlivě a opakovaně probírat s uživateli jejich potřeby. Měli bychom dále podrobně definovat vstupy a výstupy. V neposlední řadě bychom měli vytvářet co nejpodrobnější projektovou dokumentaci.

### 3.2 Nástroje pro podporu návrhu a tvorby systémů

V etapách analýzy, návrhu a částečně implementace hrají nezaměnitelnou roli nástroje typu CASE (Computer Aided System Engineering). Jsou postaveny tak, aby podporovaly týmovou práci při vývoji systému, zajišťují sdílení rozpracovaných fragmentů, správu vývoje, sledují konzistenci modelu systému, automatizují některé procesy, hlídají dodržování metodiky, atd.

Z hlediska celistvosti přístupu k vývoji rozlišujeme tři typy CASE: upper, middle a lower CASE. Upper CASE se využívá pro analýzu systému. Middle CASE zajišťuje etapu návrhu systému a lower CASE etapu implementace. Ne všechny CASE nástroje zahrnují všechny tři typy a ne každý projekt vyžaduje užití všech typů při vývoji. Je-li např. analýza jednoduchého projektu kompletně zpracována v textové podobě a schválena zadavatelem, není nutné používat nákladný upper CASE. Záleží na manažerovi projektu, zda se rozhodne pro použití nástroje, který v sobě zahrnuje všechny tři typy CASE nebo pořídí více nástrojů. Etapy analýzy, návrhu a implementace jsou však úzce svázané. V případě, že se manažer rozhodne pro více nástrojů, je nutné zohlednit, nakolik jsou nástroje integrované, případně jak nákladný je přenos rozpracovaného systému z jednoho nástroje na druhý.

Jedním z kvalitních CASE nástrojů je program *PowerDesigner* od společnosti Sybase [46]. Jedná se o velmi vespělý nástroj pro modelování. Právě prostřednictvím tohoto programu jsou prezentovány všechny níže uvedené modely a diagramy.

Přestože jsou nástroje pro podporu návrhu informačního systému dosti drahou záležitostí, sami o sobě nestačí. Nejsou totiž ničím jiným, než pouze nástroji. Bez příslušných znalostí/zkušeností tedy žádný systém, byť s pomocí nástrojů CASE, nepostavíme.

## 3.3 UML

Abychom mohli všechny výše uvedené informace o projektu efektivně zpracovat do srozumitelné podoby jako pro zadavatele, tak pro řešitele, je vhodné využít jeden ze specializovaných jazyků, který je přímo určen k tvorbě diagramů a modelů týkajících se budoucího projektu. Jeden z nejnámějších a nejpoužívanějších jazyků je UML (Unified Modeling Language).

### 3.3.1 Popis jazyka

UML lze popsat jako standardizovaný jazyk pro záznam, vizualizaci a dokumentaci artefaktů systémů s převážně softwarovou charakteristikou [12].

Obsahuje čtyři základní části:

- notaci UML (syntaxe)
- metamodel UML (sémantika)
- OCL - Object Constraint Language
- specifikaci převodu do výměnných formátů (CORBA IDL, XML DTD)

Při analýze požadavků se často jako nástroj pro komunikaci mezi zadavatelem a řešitelem využívá modelování. Vznikají tak různé modely navrhovaného systému, které mohou být na této úrovni testovány, ověřovány a upravovány. Poté, kdy dojde ke shodě mezi analytikem a zadavatelem, může být model převeden do konkrétního implementačního jazyka.

UML využívá různé typy modelů. Model lze charakterizovat jako souhrn informací, které jsou určitým způsobem strukturovány. Model by měl obsahovat veškeré informace o budoucím systému.

Diagram pak můžeme popsat jako graficky znázorněný pohled na model. Diagram popisuje určitou část modelu pomocí grafických symbolů. Na rozdíl od modelu, jeden diagram málokdy popisuje celý systém.

Analytické modely se snaží hlavně popsat, co by měl daný systém dělat. Návrhové modely pak popisují dekompozici systému na programátorsky zvládnutelné části. Tyto modely se zabývají otázkou jak by to mělo být uděláno. Implementační modely dokumentují implementaci.

Zápisy v UML (pohledy, diagramy) sestavené podle pravidel syntaxe mají pevně stanovený význam. Tímto se zabývá popis sémantiky UML, který je rozčleněn do čtyř vrstev popisující vlastnosti diagramů UML. Na nejnižší úrovni jsou specifikovány vlastnosti primitivních údajů (dat). Jedná se například o typy dat nebo obory hodnoty atributů. O úroveň výše je vyjádřena sémantika uživatelských objektů. Jedná se o tzv. model nebo meta-data. Další úroveň popisuje prvky jednotlivých modelů (metamodel). Popisuje například, co je třída, atribut nebo relace. Poslední vrstva definuje vlastnosti výše uvedených metamodelů (meta-metamodel). Určuje například, jak lze korektně vytvářet nové prvky modelu.

Součástí definice UML je i specifikace výměnných formátů, sloužící primárně pro přenos zápisů v UML mezi různými nástroji. Jako příklad lze uvést formát CORBA IDL (Interface Definition Language) nebo XMI (XML Metadata Interchange).

### 3.3.2 Diagramy a modely v UML

Stěžejním výstupem jazyka UML jsou diagramy, které se dělí do osmi základních typů (podle různých pohledů na systém). Typy diagramu jsou následující:

- **Diagramy tříd a objektů** (class diagrams, object diagrams): popisují statickou strukturu systému, znázorňují datový model systému od konceptuální úrovně po implementaci
- **Modely jednání** (diagramy případů užití – use case diagrams): popisují možné případy použití systému
- **Scénáře činností** (diagramy posloupností – sequence diagrams): popisují scénář průběhu určité činnosti v systému
- **Diagramy spolupráce** (collaboration diagrams): zachycují komunikaci spolupracujících objektů
- **Stavové diagramy** (statechart diagrams): popisují dynamické chování objektu nebo systému, možné stavy a přechody mezi nimi

- **Diagramy aktivit** (activity diagram): popisují průběh aktivit procesu nebo činnosti
- **Diagramy komponent** (component diagrams): popisují rozdělení výsledného systému na funkční celky (komponenty) a definují náplň jednotlivých komponent
- **Diagramy nasazení** (deployment diagrams): popisují umístění funkčních celků (komponent) na výpočetní uzly informačního systému

Statickou strukturu systému vyjadřují diagramy tříd, diagramy spolupráce, diagramy komponent a diagram nasazení. Funkční stránku popisují model jednání, diagramy aktivit, scénáře událostí a diagramy spolupráce. Dynamickou stránku dokumentují stavové diagramy, scénáře událostí, diagramy spolupráce a diagramy aktivit.

Použití diagramů v jednotlivých fázích vývoje je dáno konkrétní metodikou, ale orientačně můžeme říci, že v rámci analýzy se využívají diagramy tříd, model jednání, diagramy aktivit, scénáře činností a stavové diagramy. Pro fázi návrhu jsou typické diagramy tříd, diagramy spolupráce, diagramy aktivit, diagramy komponent a diagramy nasazení. Ve fázi implementace se používají diagramy tříd, diagramy komponent a diagramy nasazení.

UML se snaží být univerzální notací pro všestranné použití od obchodního modelování po detailní návrh systémů pro práci v reálném čase. Notace, která by vyčerpávajícím způsobem pokryla tak široké spektrum potřeb by ale byla velmi komplikovaná. UML proto definuje pouze základní část (tzv. UML core) a umožňuje rozšíření notace podle konkrétních potřeb.

Rozšíření UML lze dosáhnout použitím jiného jazyka pro výrazy. Standardní sémantiku lze doplnit standardními omezeními, která umožňují změnit interpretaci elementů.

Jedná se o:

- **Příznaky** (tags): umožňují přidat k prvku diagramu další informace (atributy)
- **Omezení** (constraints): umožňují specifikovat omezení pro elementy (hodnoty atributů).
- **Stereotypy** (stereotypes): umožňují klasifikovat elementy.

Stereotypy umožňují klasifikovat elementy diagramů a tím vyjádřit další sémantiku. Zapisují se jako klíčová slova. Mohou se také zobrazovat pomocí ikon, což se ale příliš nedoporučuje,



jelikož to porušuje princip jednotnosti z důvodu nestandardnosti těchto ikon. Stereotypy mohou mít následující podobu:

- <<database>> (označení komponenty zabývající se správou dat)
- <<entity>> (označení třídy reprezentující data)

Existují standardní stereotypy , kterými jsou například <<include>> a <<extend>>, jejichž význam je definován sémantikou UML. Příznaky umožňují přidat k prvku diagramu další informace ve formě dvojice *atribut=hodnota*:

*{autor = "Josef Novák", verze=1.2}*

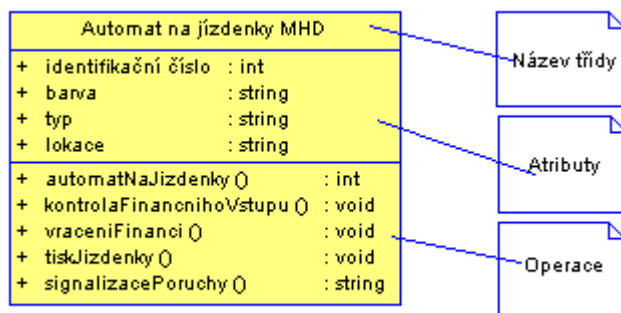
### 3.3.3 Diagram tříd

Tento druh diagramu slouží k zobrazení statické struktury systému prostřednictvím tříd a vztahů mezi nimi.

Ve fázi analýzy a návrhu systému/aplikace je tento typ diagramu velice užitečný. Třída, jako základní element diagramu tříd, sdružuje objekty se společnými vlastnostmi a chováním (sdílejí stejné atributy, operace, vztahy a sémantiku).

**Atributy** jednotlivých tříd jednoznačně charakterizují vlastnost objektu, vypovídají o jeho stavu, případně historii jeho stavů.

**Operace** je funkční složka objektu, která zajišťuje jeho chování. Má formu funkce nebo procedury.



Obr. č. 9: Základní element diagramu tříd

Každý objekt diagramu tříd má svoji dostupnost. Ta určuje, zda a jak mohou ostatní elementy modelu využívat atributy a operace daného objektu. Dostupnost může být následující:

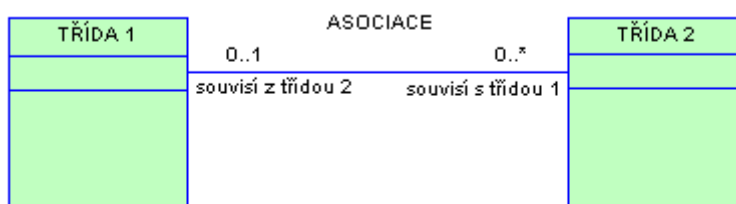
- **- Private** (soukromý): pouze pro danou třídu
- **# Protected** (chráněný): pouze pro danou třídu a její instance (dětské třídy nebo objekty)
- **\* Package**: pouze pro objekty obsažené ve stejném balíčku
- **+ Public** (veřejný): pro všechny objekty

Jednotlivé třídy se dělí do třech základních typů. Prvním typem je model (entitní, konceptuální, aplikační třída), jehož objekty obsahují a zpracovávají data. Druhým typem třídy je View (boundary, třída uživatelského rozhraní). Objekty reprezentované třídou View používají aktoři k interakci se systémem. Posledním druhem třídy je Controller (ovládací třída). Jednotlivé objekty této třídy koordinují chování v systému (definují způsob reakce uživatelského rozhraní na uživatelský vstup).

Velmi důležitou součástí diagramu tříd je vyjádření vztahů mezi jednotlivými třídami. Právě pomocí vztahů máme přehled o jednotlivých provázáních všech objektů v systému.

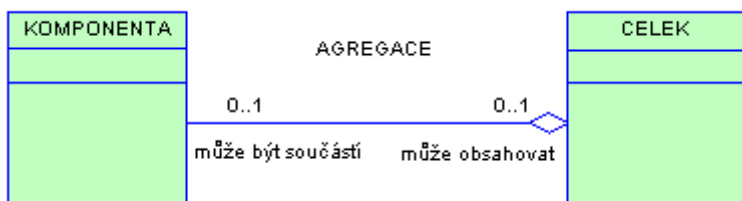
Relace mezi jednotlivými objekty mohou být následující:

**Asociace** – jedná se o sémantický vztah mezi prvky modelu, který specifikuje spojení mezi jejich instancemi. Objekty, které vzniknou z tříd spojených asociací, si budou moci posílat zprávy.



Obr. č. 10: Vztah asociace v rámci diagramu tříd

**Agregace** – je jistá forma asociace, která vyjadřuje vztah celek-část. Element (komponenta) může „přežít“ svůj kontejner (celek), případně se může stát součástí jiného kontejneru (celku).



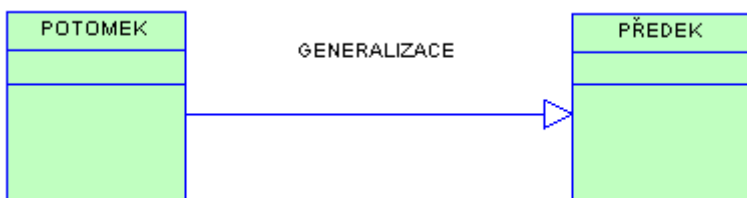
Obr. č. 11: Vztah agregace v rámci diagramu tříd

**Kompozice** – tato relace je silnější formou agregace. Zrušením kontejneru (celku) automaticky zrušíme i obsažený element (komponentu). Daný element může být součástí právě jednoho kontejneru.



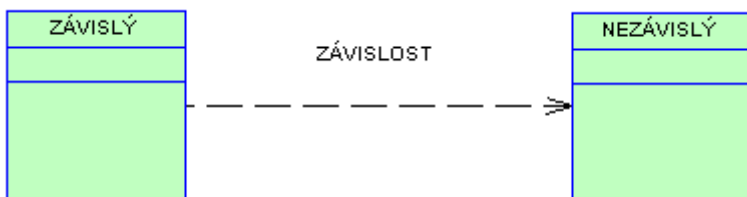
Obr. č. 12: Vztah kompozice v rámci diagramu tříd

**Generalizace** (dědičnost) – popisuje hierarchický vztah tříd, v němž třída „potomek“ dědí atributy a operace svého „předka“. Potomek může mít kromě zděděných charakteristik ještě další vlastní specifické vlastnosti. Zděděné vlastnosti mohou být v potomkovi modifikovány.



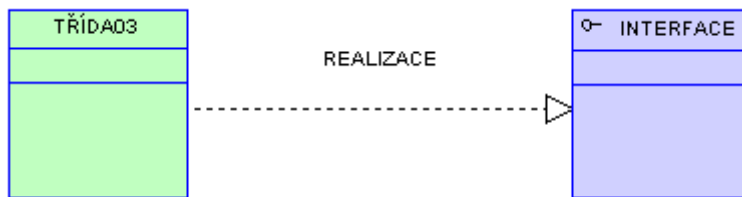
Obr. č. 13: Vztah generalizace v rámci diagramu tříd

**Závislost** – je vztah mezi dvěma třídami, v němž změna jednoho (nezávislého) elementu přímo ovlivní druhý (závislý) element.



Obr. č. 14: Vztah závislosti v rámci diagramu tříd

**Realizace** – jedná se o souhrn všech veřejně přístupných metod dané třídy. Umožňuje vícenásobné využití operací, aniž bychom museli zavádět dědičnost mezi třídami.



Obr. č. 15: Vztah realizace v rámci diagramu tříd

V rámci jednotlivých relací můžeme vyjádřit i tzv. kardinalitu vztahů. Možné stavy jsou následující:

- 0..\* - 0 až více
- 0..1 – 0 nebo 1
- 1..\* - 1 až více
- 1..1 – právě 1
- 1..3 – 1 až 3
- 5,10,20 – 5, 10 nebo 20
- 5..\* - 5 až více

### 3.3.4 Stavový diagram

Tento druh diagramu popisuje stavy jednotlivých objektů a povolených přechodů mezi těmito stavy. Jinými slovy stavový diagram reprezentuje životní cyklus objektu.

Využití diagramu je vhodné zejména při popisu dynamiky objektu (pokud má rozpoznatelné stavy), popisu metody (známe-li algoritmus), popisu protokolu (včetně protokolu o styku uživatele se systémem) a popis systémů pracujících v reálném čase.

Výraz stav bychom mohli popsat jako hodnotu nějakých atributů (tzv. stavové atributy) nebo podmínku/situaci během existence objektu, při jejímž trvání objekt splňuje nějaké podmínky, provádí nějaké aktivity, nebo čeká na nějakou událost.

Základní konstrukty stavového diagramu jsou následující:

### Začátek (start)



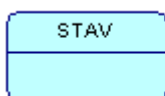
Obr. č. 16: Grafické znázornění začátku v rámci stavového diagramu

### Konec (end)



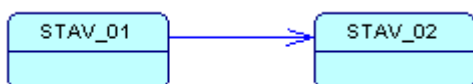
Obr. č. 17: Grafické znázornění konce v rámci stavového diagramu

### Stav (state)



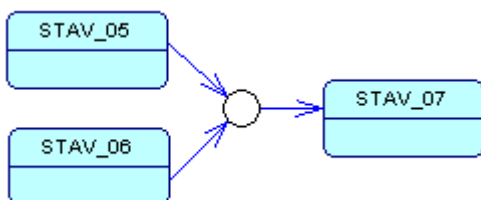
Obr. č. 18: Grafické znázornění základního elementu v rámci stavového diagramu

### Přechod (transition)



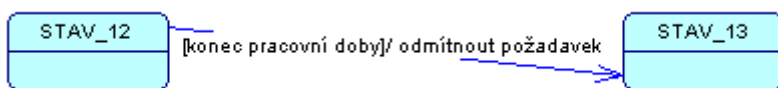
Obr. č. 19: Grafické znázornění přechodu v rámci stavového diagramu

### Větvení a spojování (junction)



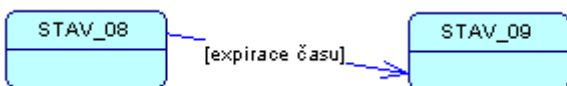
Obr. č. 20: Grafické znázornění větvení a spojování v rámci stavového diagramu

**Událost** (event) – to, co může způsobit změnu stavu. Píše se před lomítkem (event /).



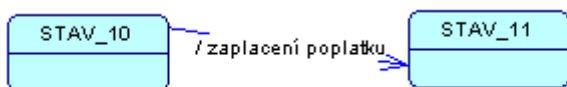
Obr. č. 21: Grafické znázornění události v rámci stavového diagramu

**Podmínka** (condition) – podmínka, která musí být splněna, aby došlo k přechodu ze stavu do stavu. Píše se do hranatých závorek.



Obr. č. 22: Grafické znázornění podmínky v rámci stavového diagramu

**Akce** (action) – to, co se děje při přechodu ze stavu do stavu. Píše se za lomítkem (/ action).



Obr. č. 23: Grafické znázornění akce v rámci stavového diagramu

### 3.3.5 Diagram aktivit

Diagram aktivit má za úkol reprezentovat strukturu (dynamiku) počítačových a organizačních procesů v systému. Je zaměřen převážně na jeho vnitřní chování. Zobrazuje řídicí toky (přechody) mezi akcemi (aktivitami) v systému od počátečního bodu po jeden nebo více koncových bodů. Důraz je kladen na pořadí výše uvedených aktivit.

Diagram aktivit je využíván hlavně při modelování průběhu jednotlivých Use Case a operací v třídách, při modelování podnikových procesů (business process modeling) nebo workflow.

Základní konstrukty diagramu aktivit jsou následující:

### Začátek (start)



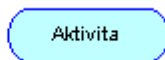
Obr. č. 24: Grafické znázornění začátku v rámci diagramu aktivit

### Konec (finish)



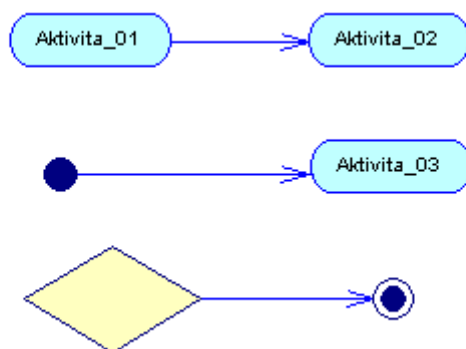
Obr. č. 25: Grafické znázornění konce v rámci diagramu aktivit

### Aktivita (činnost, activity. ActionState – akční stav)



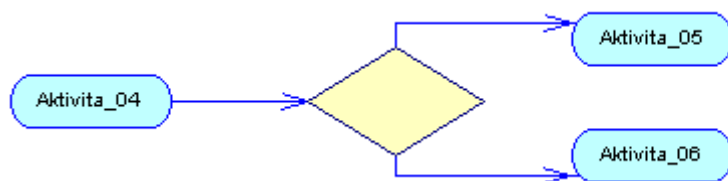
Obr. č. 26: Grafické znázornění aktivity v rámci diagramu aktivit

### Přechod (transition)



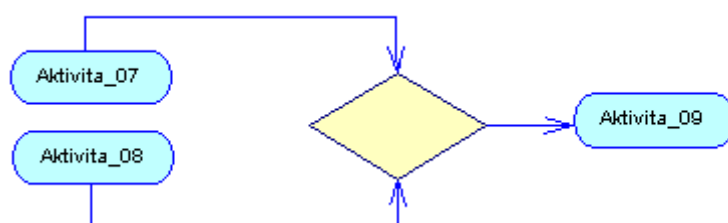
Obr. č. 27: Grafické znázornění přechodu v rámci diagramu aktivit

### Rozhodování (decision) – nepovinný symbol



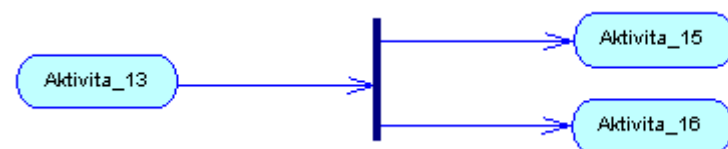
Obr. č. 28: Grafické znázornění rozhodování v rámci diagramu aktivit

### Spojování (merge) – nepovinný symbol



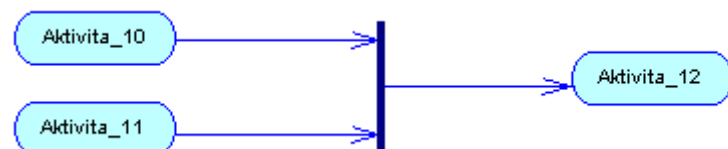
Obr. č. 29: Grafické znázornění spojování v rámci diagramu aktivit

### Větvení (fork)



Obr. č. 30: Grafické znázornění větvení v rámci diagramu aktivit

### Spojování (join, synchronizace)

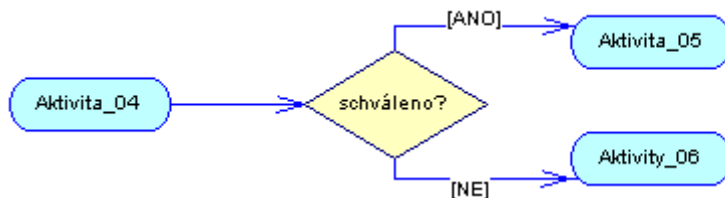


Obr. č. 31: Grafické znázornění spojování v rámci diagramu aktivit



## Podmínka

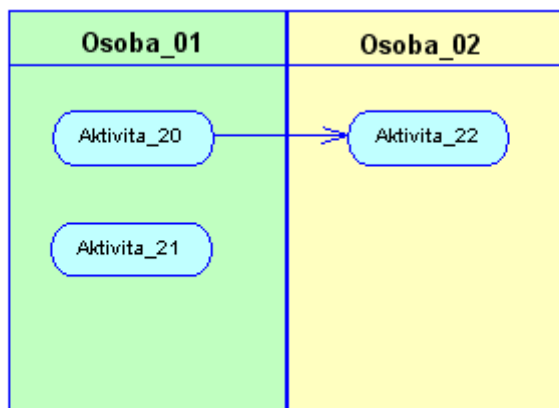
Využívá se při rozhodovacím procesu, kde reprezentuje možné alternativy rozhodnutí. Většinou se jedná o hodnoty „ano“ a „ne“, které se vepisují do hranatých závorek v rámci přechodu.



Obr. č. 32: Grafické znázornění podmínky v rámci diagramu aktivit

## Swimlane (dráha, oblast zodpovědnosti)

Jedná se o části diagramu (pole), za které odpovídá jeden objekt (organizační jednotka).



Obr. č. 33: Grafické znázornění dráhy v rámci diagramu aktivit

### 3.3.6 Use Case diagram

Zobrazuje dynamickou strukturu systému z pohledu uživatele. Je primárně určen k definici chování systému, aniž by odhaloval jeho vnitřní strukturu. Na Use Case diagram můžeme pohlížet také jako na soubor scénářů pro používání systému, kde každý scénář obsahuje sekvenci (posloupnost) událostí, které v jeho rámci probíhají (včetně případných variant) a popis interakce (komunikace) mezi uživatelem (aktorem) a systémem.

Use Case model je grafickým zobrazením části dokumentu specifikace požadavků. Požadavky na budoucí systém/aplikaci lze rozdělit na nefunkční a funkční. Mezi nefunkční požadavky patří například cena, počet uživatelů nebo bezpečnost. Funkční požadavky jsou pak ty, které jsou zobrazitelné prostřednictvím Use Case.

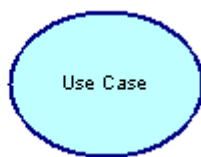
Model Use Case se zabývá následujícími otázkami:

- jaké jsou úlohy aplikace?
- jaké hlavní úlohy a funkce mají být prováděny aktorem?
- jaké informace si přeje aktor získávat, vytvářet nebo měnit?
- bude okolí potřebovat informace o změnách v systému?
- přeje si aktor být informován o neočekávaných změnách?
- budou se z okolí přidávat, ubírat nebo měnit informace v systému (popřípadě jaké)?
- jaké změny v okolí systému povedou k přísunu informací do systému?
- bude aktor informovat systém o změnách v externím prostředí?
- jaké činnosti jsou třeba k administraci systému?

Use Case diagram můžeme využít (spolu s ostatními diagramy) při vlastní specifikaci požadavků na systém, při komunikaci se zadavatelem nebo jako podklad pro řízení projektu.

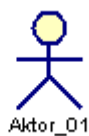
Use Case využívá následujících konstruktů:

**Use Case** (případ užití, typová činnost, užitná činnost)



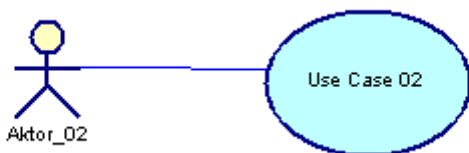
Obr. č. 34: Grafické znázornění případu užití v rámci Use Case diagramu

**Aktor** (aktér, participant, účastník)



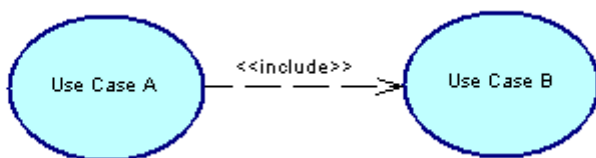
Obr. č. 35: Grafické znázornění aktora v rámci Use Case diagramu

**Vazba mezi aktorem a případem užití** (asociace)



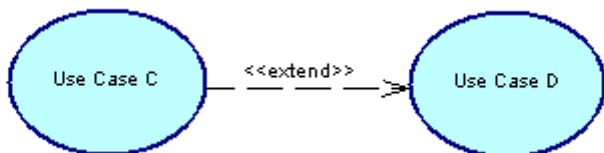
Obr. č. 36: Grafické znázornění vazby aktor - Use Case v rámci Use Case diagramu

**Vazba mezi případy užití – asociace** (include, Use Case „A“ zahrnuje Use Case „B“)



Obr. č. 37: Grafické znázornění asociační vazby v rámci Use Case diagramu

**Vazba mezi případy užití – závislost** (extend, Use Case „C“ rozšiřuje Use Case „D“)



Obr. č. 38: Grafické znázornění závislostní vazby v rámci Use Case diagramu

Součástí diagramu může být i ohraničení samotného systému, které se zpravidla znázorňuje prostřednictvím čtverce nebo obdélníků s barevnou výplní.

Use Case diagram můžeme doplnit charakteristikami určitých případů využití. Jedná se o tzv. *Action steps*, kde například případ užití v podobě ukládání souboru můžeme provést přes horizontální menu aplikace, nebo můžeme využít tzv. kombinaci *hot keys* v podobě kombinace kláves *CTRL + S*. Pokud je dokument, se kterým pracujeme, již pojmenovaný, tento se uloží bez jakýchkoliv dalších kroků. Pokud však dokument ještě jméno nemá, zobrazí aplikace dialogové okno s výchozím jménem souboru, které můžeme změnit. Poté se soubor uloží.

### 3.3.7 Business Process Model

Jedná se o konceptuální model, který odráží vše, co se v podniku děje. Tento model zobrazuje procesy a jejich interakci v systému od počátečního bodu po jeden nebo více bodů koncových. Velký důraz se klade na zobrazení procesů.

Business Process Model můžeme využít jako podklad pro návrh informačního systému podniku, podklad pro reengineering procesů podniku, podklad pro řízení procesů (workflow management) nebo jako podklad pro dokumentaci procesů v podniku (např. pro získání certifikátu jakosti podle ISO 9000).

Business Process Model využívá následujících konstruktů:

**Začátek** (start)



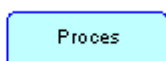
Obr. č. 39: Grafické znázornění začátku v rámci Business Process modelu

## Konec (end)



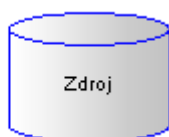
Obr. č. 40: Grafické znázornění konce v rámci Business Process modelu

## Proces (process)



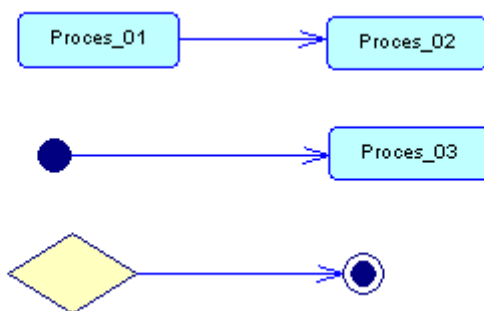
Obr. č. 41: Grafické znázornění procesu v rámci Business Process modelu

## Zdroj (resource)



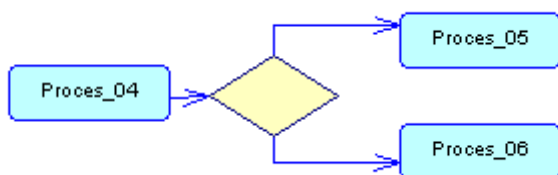
Obr. č. 42: Grafické znázornění zdroje v rámci Business Process modelu

## Tok (flow)



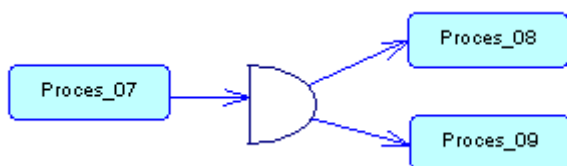
Obr. č. 43: Grafické znázornění toku v rámci Business Process modelu

## Rozhodování (decision)



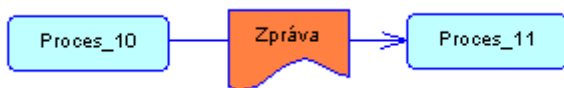
Obr. č. 44: Grafické znázornění rozhodování v rámci Business Process modelu

## Synchronizace (synchronization)



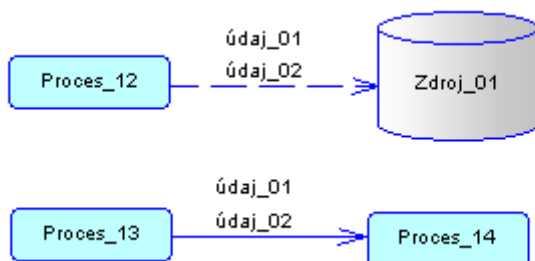
Obr. č. 45: Grafické znázornění synchronizace v rámci Business Process modelu

## Formát zprávy (message format) - přiřazuje se k tokům.



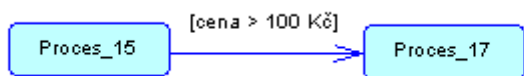
Obr. č. 46: Grafické znázornění formátu zprávy v rámci Business Process modelu

## Data (data) – přiřazují se k tokům a procesům.



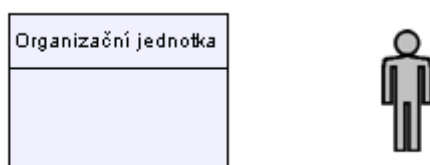
Obr. č. 47: Grafické znázornění dat v rámci Business Process modelu

**Podmínka** (condition) – podmínka, která musí být splněna, aby došlo k přechodu z jednoho procesu do druhého.



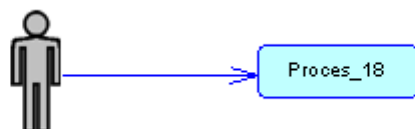
Obr. č. 48: Grafické znázornění podmínky v rámci Business Process modelu

**Organizační jednotka** (swimlane) – entita zodpovědná za proces.



Obr. č. 49: Grafické znázornění organizační jednotky v rámci Business Process modelu

**Přiřazení role** (role association) – spojení ikony organizační jednotky a procesu.



Obr. č. 50: Grafické znázornění přiřazení pole v rámci Business Process modelu

### 3.3.8 Příklady využití UML

V této kapitole jsou výše uvedené teoretické informace převedeny do praxe v podobě případové studie modelování v rámci návrhu nám již známého inzertního systému.

#### **Datový model - konceptuální**

Datové modelování slouží k obrazovému vyjádření budoucí datové struktury systému. Tento druh modelování obsahuje dva typy modelů, které se nazývají *konceptuální* a *fyzický*. Konceptuální datový model je oproti fyzickému obecnější a slouží hlavně pro celkový obraz databázové struktury, který je nezávislý na konkrétním databázovém systému.

Fyzický model již zobrazuje veškeré tabulky včetně datových typů, tabulkových relací a různých dalších specifikací.

V našem příkladě sestrojíme tabulky, které budou uchovávat data o jednotlivých inzerátech, inzerentech, partnerech, diskusích, soutěžích a podobně (viz obrázek 52).

Základní tabulka bude *T\_INZERCE*, která bude obsahovat veškeré informace o podaném inzerátu a jeho tvůrci. Tato tabulka bude dále čerpat informace z dalších tabulek/číselníků. Číselník je druh dvouslupcové tabulky, která obsahuje pouze unikátní data, jež sestávají z pravidla z jednoznačného identifikátoru (zkráceně ID) a textového řetězce. Hlavním cílem používání číselníků je zabránění duplicitě či multiplicitě dat v tabulkách. V naší případové studii využíváme dva číselníky *C\_KATEGORIE* a *C\_PODKATEGORIE*. Oba číselníky odkazují do již zmiňované tabulky *T\_INZERCE*, která obsahuje pouze identifikátory těchto číselníků. Identifikátory se mohou v tabulce libovolně opakovat. Opakují se však pouze čísla a ne celé textové vyjádření číselníků, což by způsobovalo výše uvedenou nežádoucí multiplicitu.

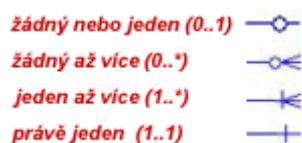
Obdobně jako u tabulky *T\_INZERCE* dochází ke sdílení dat (relacím) i u ostatních tabulek.

Jednotlivé relace mezi tabulkami jsou na obrázku zobrazeny prostřednictvím čar, které mají vždy na obou koncích grafické vyjádření vzájemných vztahů. Vztahy mezi tabulkami určují četnost výskytu dat obsažené v jedné tabulce (resp. v jednom záznamu tabulky) vůči tabulce



druhé. Například tabulka *C\_KATEGORIE* obsahuje záznamy (řádky), které se v tabulce *T\_INZERCE* nemusejí objevit vůbec, ale také se mohou objevit i několikrát. Jinými slovy to znamená, že do dané kategorie nemusí spadat buď žádný inzerát vůbec, nebo do ní může spadat jeden, dva, tři i více inzerátů. Naopak jednotlivým záznamům (inzerátům) z tabulky *T\_INZERCE* smí být přiřazena pouze jedna kategorie, což znamená, že inzerát nemůže být podán do více kategorií.

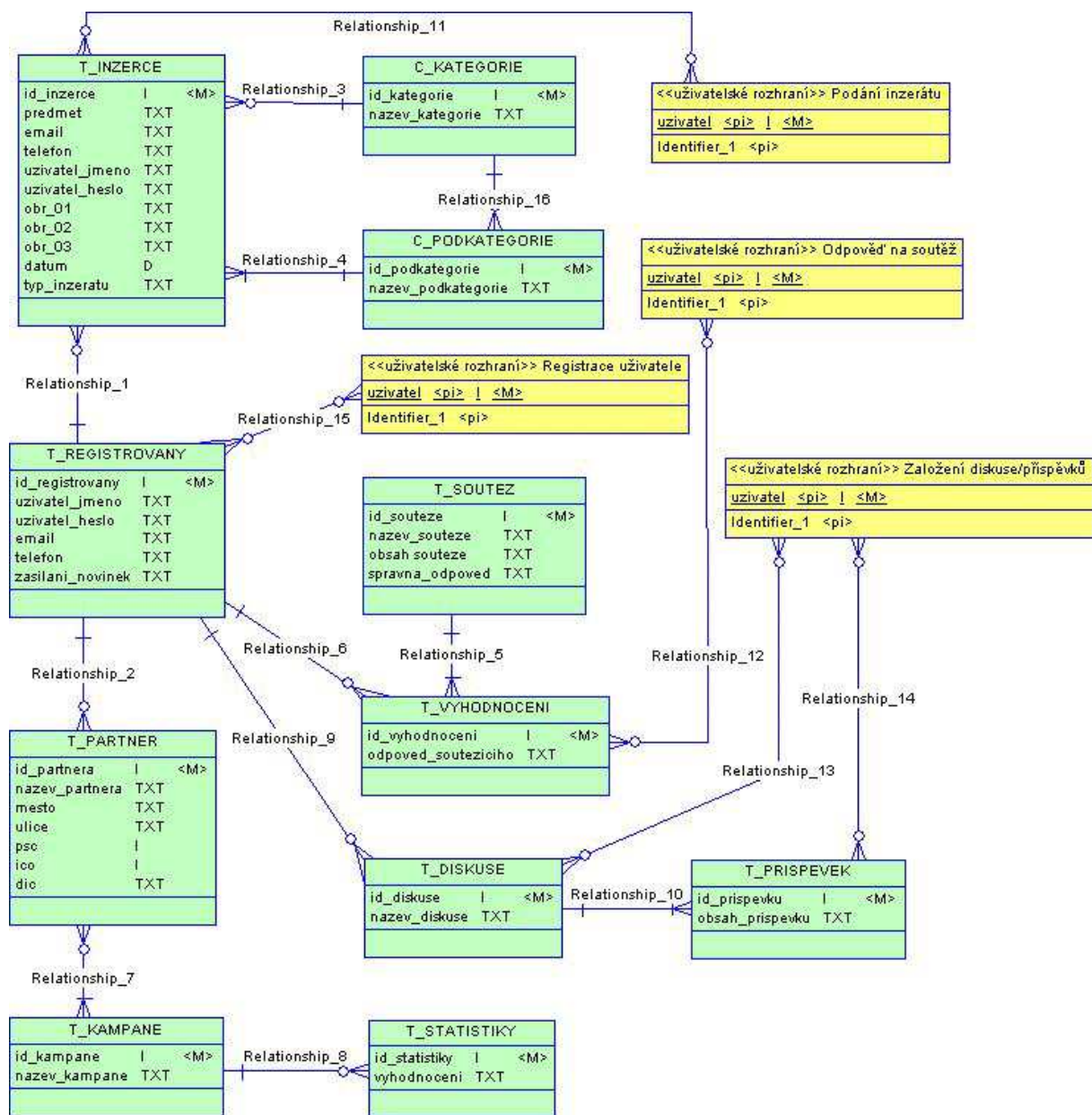
Následující obrázek by měl pomoci se v jednotlivých relacích lépe orientovat.



Obr. č. 51: Přehled možných vztahů mezi objekty

První vztah určuje, že se záznam z jedné tabulky může v tabulce jiné vyskytovat jednou nebo vůbec. Druhý vztah, který jsme si již popsali výše, znamená, že záznam z jedné tabulky se nemusí v tabulce druhé objevit vůbec nebo také i několikrát. Třetí vztah určuje četnost výskytu jedenkrát až vícekrát. Poslední relace definuje možnost výskytu právě jedenkrát.

Na obrázku č. 52 můžeme vidět i tabulky, které jsou podbarveny žlutě. V tomto případě se nejedná o tabulku, ale o uživatelské rozhraní, pomocí kterého se data do jednotlivých tabulek vkládají.



Obr. č. 52: Konceptuální datový model

## Datový model - fyzický

Fyzický model datové struktury aplikace je základním detailním modelem, znázorňujícím uložení dat do jednotlivých tabulek a vyjádření vztahů mezi nimi. Každý zelený obdélník (viz obrázek č. 53) reprezentuje jednotlivou tabulku v budoucí databázi. V těchto tabulkách jsou pak vyobrazeny názvy položek (sloupců), do kterých se budou ukládat příslušná data.

Vedle názvu sloupců je uveden i jejich datový typ. Datový typ stanovuje, v jakém formátu budou data ukládána.

V rámci celosvětové konvence existuje několik standardních datových typů:

- **Integer** (ukládá celočíselné numerické hodnoty)
- **Float** (ukládá numerické hodnoty s plovoucí desetinnou čárkou)
- **String** (ukládá různé textové řetězce)
- **Char/Varchar** (ukládá jednotlivé znaky)
- **Text** (ukládá objemné textové řetězce)
- **Date** (ukládá datum)
- **Time** (ukládá čas)
- **Datetime** (ukládá datum i čas)
- **Boolean** (ukládá logické hodnoty pravda/nepravda)
- **Blob** (ukládá data binárního charakteru jako jsou obrázky, zvukové záznamy apod.)

V našem příkladě budeme ukládat převážně text (textový řetězec), který se v modelování označuje výrazem *Varchar*. Číslo v závorkách určuje maximální možný počet uložených znaků. Ve dvou případech použijeme pro text místo datového typu *Varchar* datový typ *Char*. Bude se jednat o položky v tabulkách, kam se budou ukládat pouze jednoznakové zkratky.

Rozdíl mezi oběma výše uvedenými datovými typy je v tom, že *Char* uchovává řetězec v pevně stanové délce, kterou si v závorce určíme. Pokud tedy stanovíme pro určitý sloupec max. počet 50 znaků a poté uložíme řetězec o 20 znacích, zbytek (30 znaků) se musí do databáze také uložit v podobě neviditelných mezer. *Varchar* stanovuje variabilní délku, takže se ukládá pouze reálný počet znaků, přestože máme v závorkách uvedeno číslo vyšší. Ukládání variabilní délky znaků zajišťuje přidaný ukončovací znak. Nevýhoda *Varchar* nastane v situaci, kdy ukládáme stejný počet znaků (např. 5). Jelikož datový typ *Varchar* vždy používá ukončovací znak, bude celkový objem řetězce 6 bytů (5 bytů řetězce + 1 bajt ukončovacího znaku). *Char* uloží pouze vlastní text, proto bude finální objem přesně 5 bytů.

Pokud ukládáme text, jehož délka přesáhne 255 znaků, použijme místo výše uvedených datových typů výraz *Text*.

Dalším datovým typem, který využijeme, je *Integer* (zkráceně *Int*). Pomocí tohoto typu se do databáze ukládají číselné hodnoty.

Posledním typem, který v případové studii využijeme, je *Datetime*. Jak už z názvu vyplývá, do sloupce s tímto typem budeme ukládat hodnotu datumu a času (ve formátu RRRR-MM-DD HH:MM:SS<sup>5</sup>).

Každá tabulka má tzv. *primární klíč*, což je jednoznačný identifikátor pro daný záznam (pole v tabulce). Pole označené primárním klíčem musí být vždy unikátní, nesmí tedy nastat případ duplicity či dokonce multiplicity. Většinou se *primární klíč* označuje zkratkou <PK> (primary key).

V rámci datového modelování používáme i tzv. *cizí klíč*. Ten se označuje zkratkou <FK> (foreign key) a stanovuje, že hodnoty sloupce označeného tímto klíčem se vyskytují ještě

---

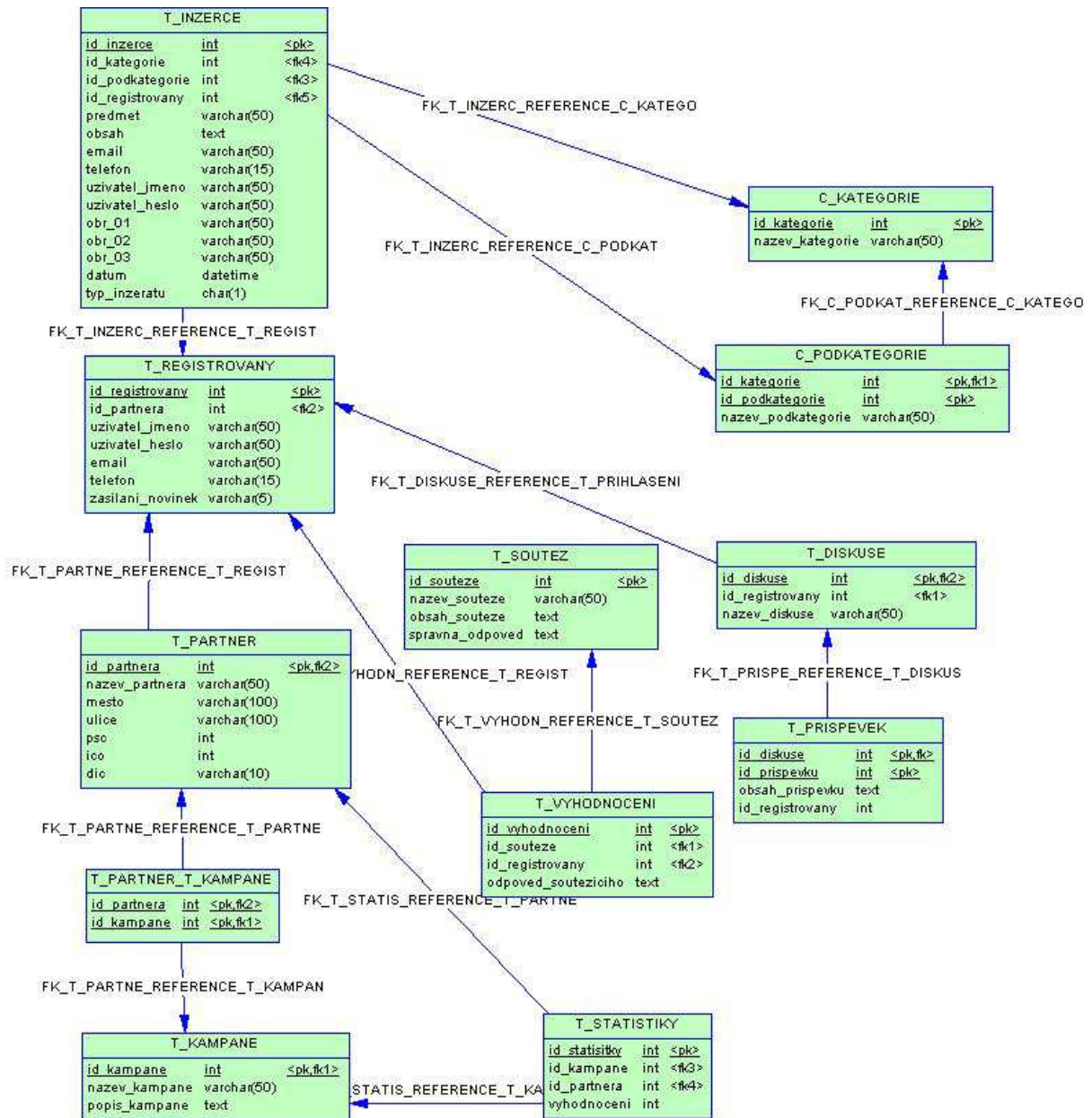
<sup>5</sup> RRRR: rok, MM: měsíc, DD: den, HH: hodina, MM: minuta, SS: vteřina

v jiné tabulce. Obě tabulky jsou právě díky cizímu klíči spolu provázány a mohou tak spolu komunikovat. V našem případě máme *cizím klíčem* označen například sloupec *id\_kategorie* v tabulce *T\_INZERCE*. Znamená to, že pomocí numerické hodnoty v tomto sloupci budeme moci zjistit hodnotu textovou (sloupec *nazev\_kategorie*), která je obsažena v tabulce *C\_KATEGORIE*, a to právě díky sloupci *id\_kategorie*, který je primárním klíčem této tabulky.

Posledním klíčem, se kterým se můžeme setkat je tzv. složený, což je klíč skládající se z několika sloupců, jejichž kombinace hodnot je jedinečná a lze podle ní daný záznam jednoznačně identifikovat. Na našem modelu se setkáme se složeným klíčem např. v tabulce *C\_PODKATEGORIE*, kde máme sloupce *id\_kategorie* a *id\_podkategorie*. Hodnoty v jednotlivých sloupcích, označující kategorii a podkategorii, se mohou libovolně opakovat. Nesmí však dojít k případu, kdy se bude v tabulce opakovat kombinace hodnot, čímž se naruší unikátnost záznamu.

Na obrázku je také zobrazena tzv. pomocná tabulka (*T\_PARTNER\_T\_KAMPANE*). Jedná se o tabulku, která neslouží k projekci datových záznamů, ale k propojení vícečetného vztahu mezi dvěma dalšími tabulkami (v našem případě *T\_PARTNER* a *T\_KAMPANE*). Jelikož partner může mít zakoupeno více reklamních kampaní a zároveň jedna kampaň může být využívána více partnery, vzniká nám vztah, kde bychom dosáhli v jednotlivých tabulkách nežádoucí multiplicity. S využitím pomocné tabulky, obsahující pouze identifikátory obou tabulek s libovolnou četností výskytu, budou vždy záznamy v tabulce partnerů i kampaní vždy unikátní.

Jednotlivé relace mezi tabulkami jsou znázorněny šipkami.



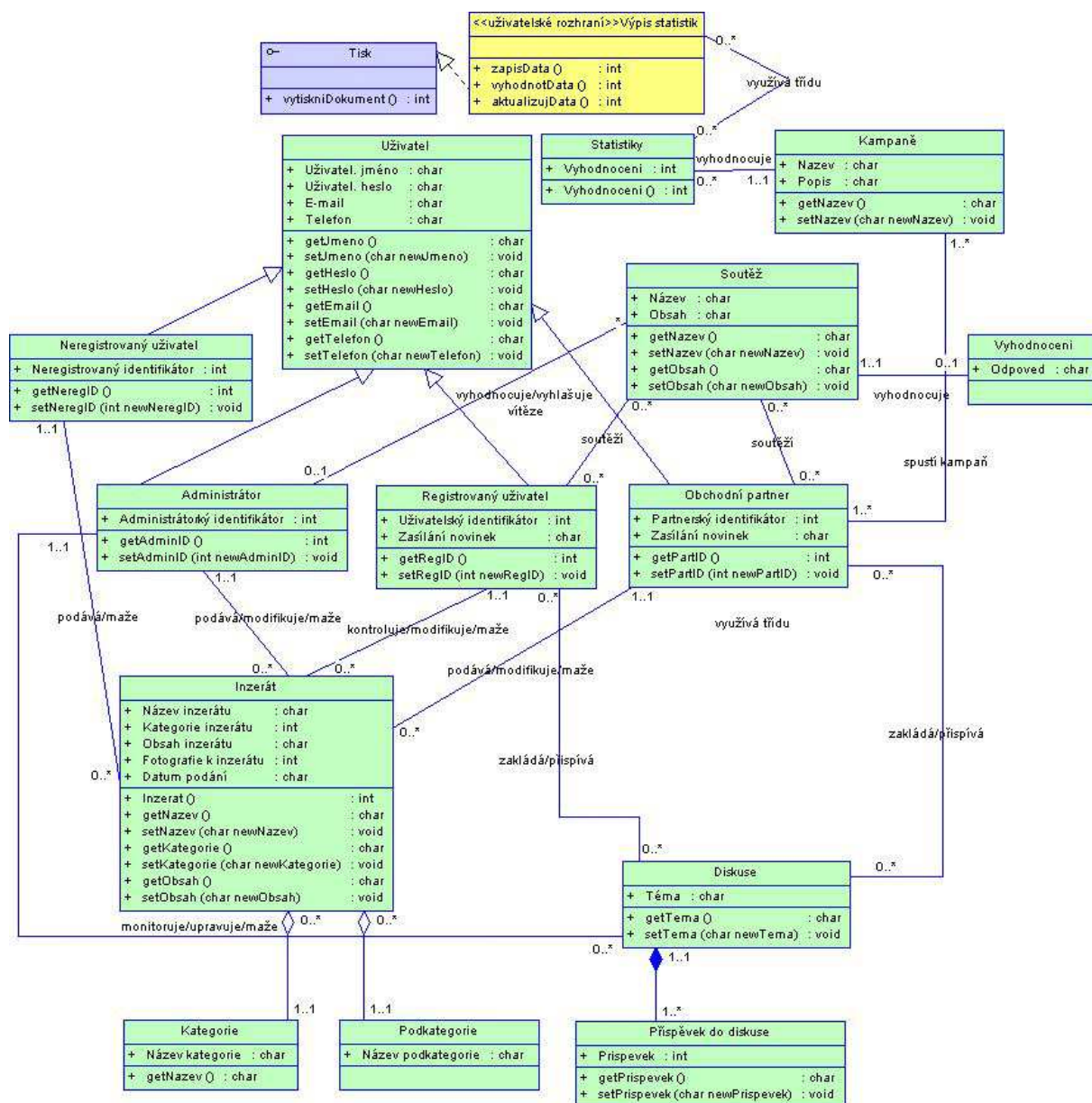
Obr. č. 53: Fyzický datový model

## Diagram tříd

Tento diagram nám slouží jako konceptuální model pro zobrazení statické struktury se všemi entitami (např. *uživatel*), vztahy (např. *soutěží*) a atributy (např. *e-mail*) systému. Diagram se využívá ve fázi analýzy, návrhu i implementace a jeho cílem je graficky vyjádřit sdružení objektů se společnými vlastnostmi a chováním.

Veškeré vztahy včetně jejich kardinality (násobnosti) byly vysvětleny v teoretickém popisu tohoto diagramu (viz kapitola 3.3.3).

Žlutě vybarvená tabulka znázorňuje uživatelské rozhraní pro tiskovou formu výpisu statistik včetně tisku samotného (fialová tabulka) .



Obr. č. 54: Diagram tříd



## Use Case diagram

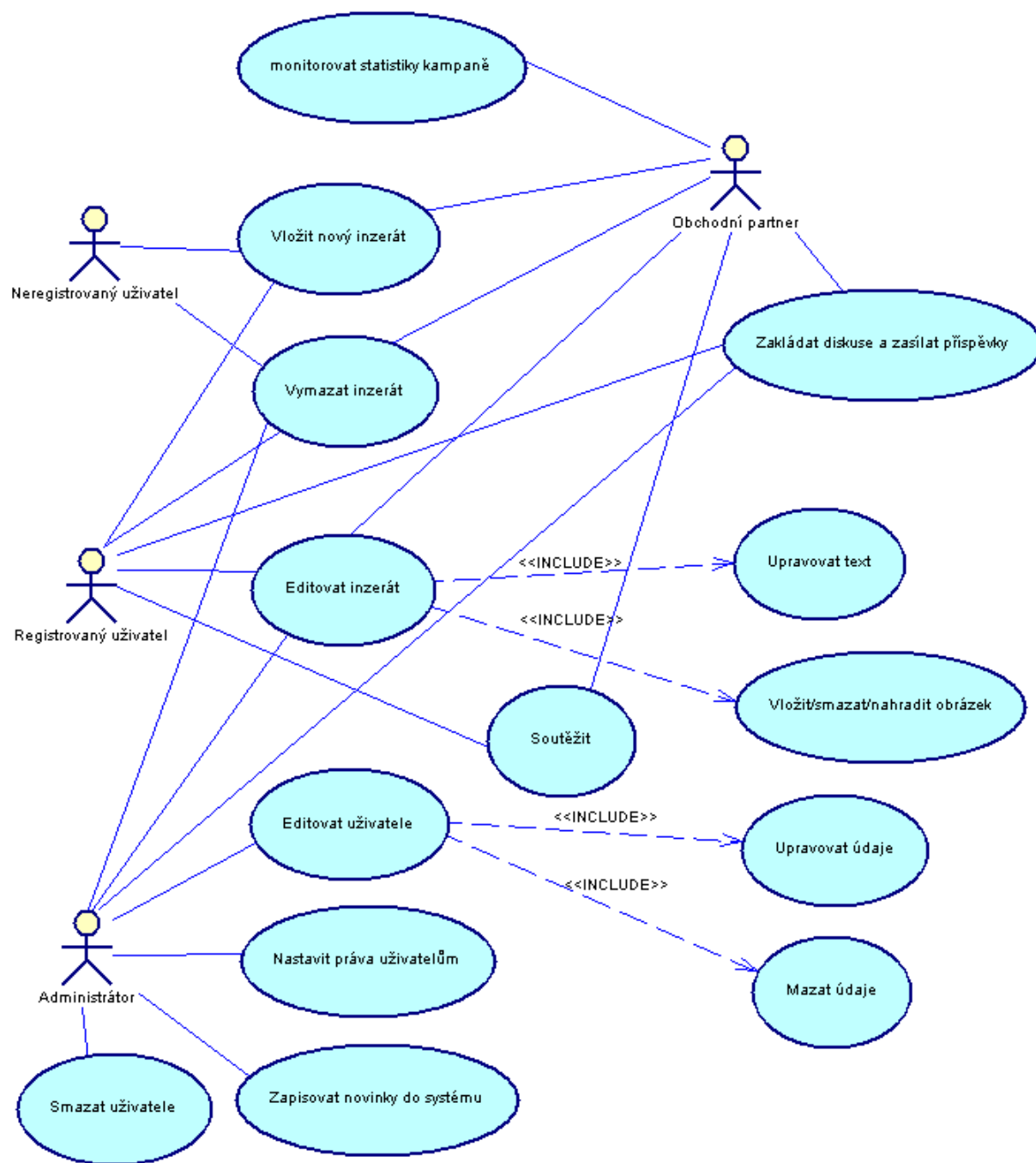
Use Case diagram znázorňuje, jaké jsou hlavní funkce budoucí aplikace včetně popisu práv jednotlivých uživatelů systému na používání výše uvedených funkcí. Na obrázku (obrázek č. 55) můžeme vidět všechny čtyři typy uživatelů našeho vzorového systému s jednotlivými možnostmi využití jejich práv.

Neregistrovaný uživatel je základním typem uživatele. Má právo na vkládání nových inzerátů (bez jakéhokoliv omezení) a jejich mazání.

Registrovaný uživatel má pak k těmto právům navíc možnost svůj inzerát editovat, což znamená upravovat text a nahrát, přehrát nebo vymazat fotografie k inzerátu. Dále má registrovaný uživatel právo zakládat diskuse a v rámci nich (nejen jím založených) posílat své příspěvky. Možnost zapojit se do soutěže patří k dalšímu právu tohoto typu uživatele.

Další role patří obchodním partnerům. Ti budou mít identická práva jako mají registrovaní uživatelé a navíc mohou kdykoliv sledovat vývoj jejich reklamních kampaní.

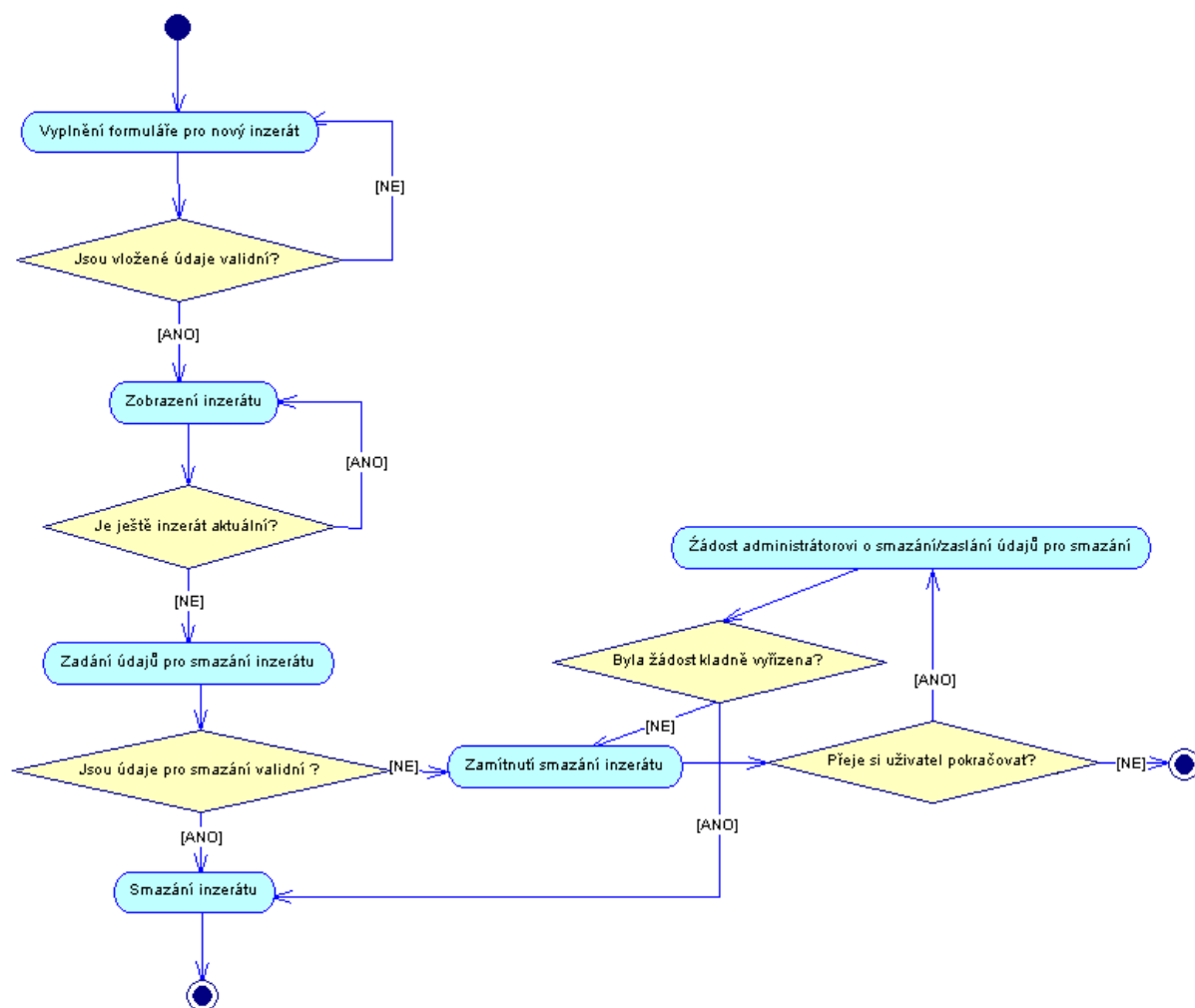
Poslední druh uživatele je administrátor systému. Administrátor má právo editovat i mazat jednotlivé inzeráty ostatních uživatelů. Kromě inzerátů může také editovat/mazat všechny uživatelské účty. Dále může administrátor nastavovat uživatelům práva k používání určitých systémových funkcí. Mezi pravidelnou činnost administrátora systému spadá i vypisování novinek z oblasti aplikace.



Obr. č. 55: Use Case diagram

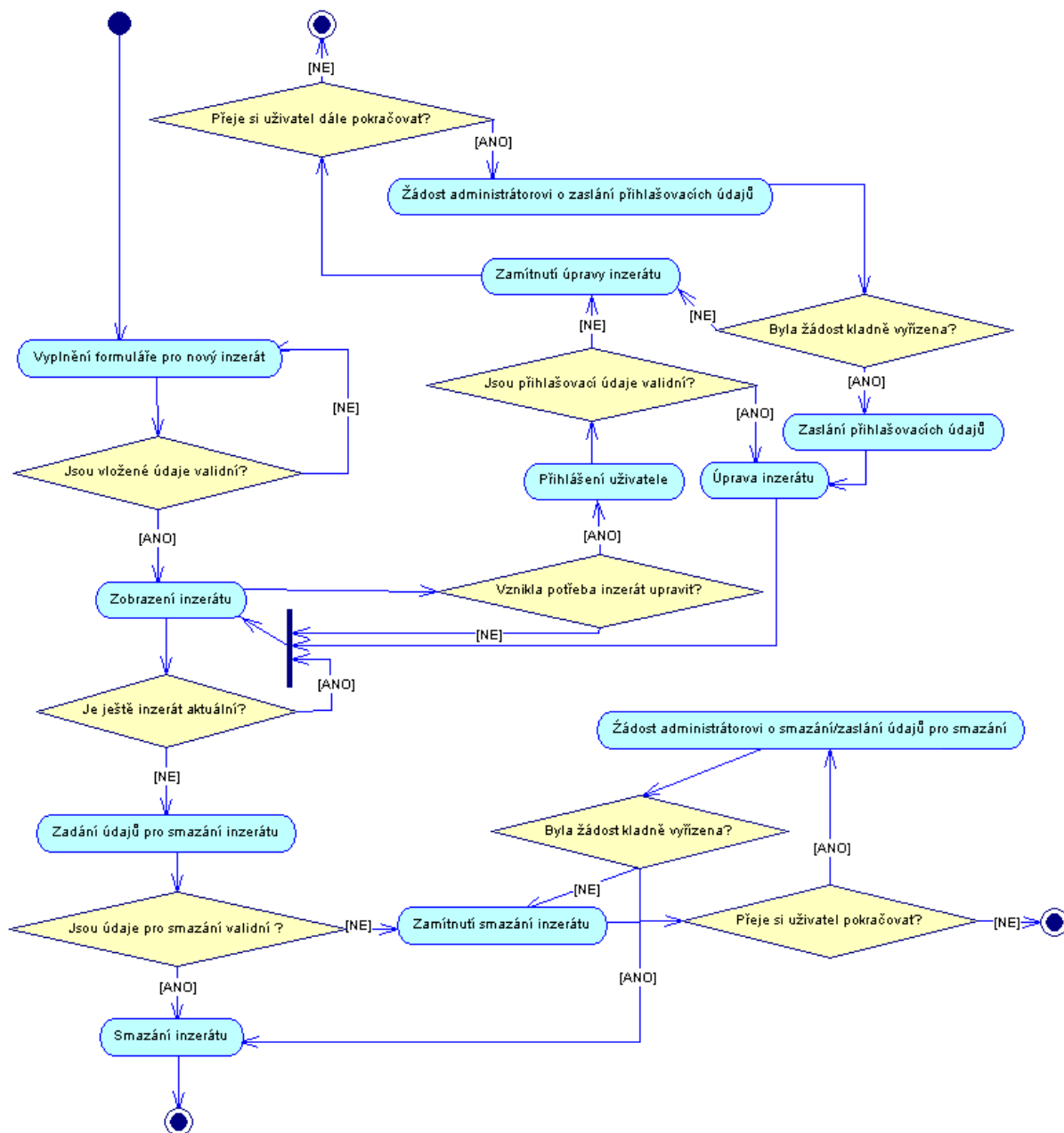
## Diagram aktivit

Tento diagram reprezentuje jednotlivé procesy v systému prostřednictvím aktivit (modré ovály) a přechodů (žluté kosočtverce) mezi nimi. Níže uvedený obrázek reprezentuje životní cyklus inzerátu podaného neregistrovaným uživatelem.



Obr. č. 56: Diagram aktivit: neregistrovaný uživatel

Registrovaný uživatel může kromě podávání a mazání také jednotlivé inzeráty upravovat. Pokud se tedy přihlásí (pomocí přihlašovacího jména a hesla), může libovolně upravovat text a spravovat fotografie svých inzerátů. Obrázek č. 57 demonstruje životní cyklus inzerátu, který podal uživatel registrovaný.



Obr. č. 57: Diagram aktivit: registrovaný uživatel

Záleží na řešitelské firmě, jaké diagramy a modely ve svém projektovém návrhu vypracuje (nemusí tedy vždy použít všechny druhy diagramů). Zpravidla záleží na charakteru a rozsahu

projektu. Vybrané modely by však měly zajistit kompletní grafický podklad pro tvorbu budoucí aplikace.

### 3.4 WebML

Výše uvedené modelování prostřednictvím jazyka UML lze brát jako zcela profesionální řešení při návrhu softwarové aplikace, nicméně UML postrádá definici uživatelského rozhraní (rozvržení entit budoucího systému jako je menu, zobrazené záznamy, formuláře, stránkovací objekty atd.) a jejich vzájemné hypertextové propojení. Tyto jednotlivé prvky si zadavatel musí připravit a popsat zvlášť.

WebML (Web Modeling Language) je metodologie, která byla navržena přímo pro tvorbu webových aplikací [21]. Hlavní výhodou jazyka je možnost při modelování oddělit informační obsah stránek (datový model) od jejich struktury a navigace (hypertextový model). Můžeme využít i možnosti sestavení prezentačního modelu, což znamená transformovat předchozí modely do konkrétní podoby budoucí aplikace. Jako podklad pro vytvoření prezentačního modelu slouží XML formát<sup>6</sup> datového a hypertextového modelu. WebML dokáže dále specifikovat i dynamickou stránku webové prezentace (manipulace s daty). Výstup je generován XSL transformací<sup>7</sup> a je tvořen šablonami jednotlivých stránek včetně kódu značkovacího jazyka jako je např. ASP.NET.

WebML lze samozřejmě (stejně jako UML) využívat prostřednictvím CASE nástrojů. V našem případě je použit software *WebRatio* [44].

#### 3.4.1 Datový model

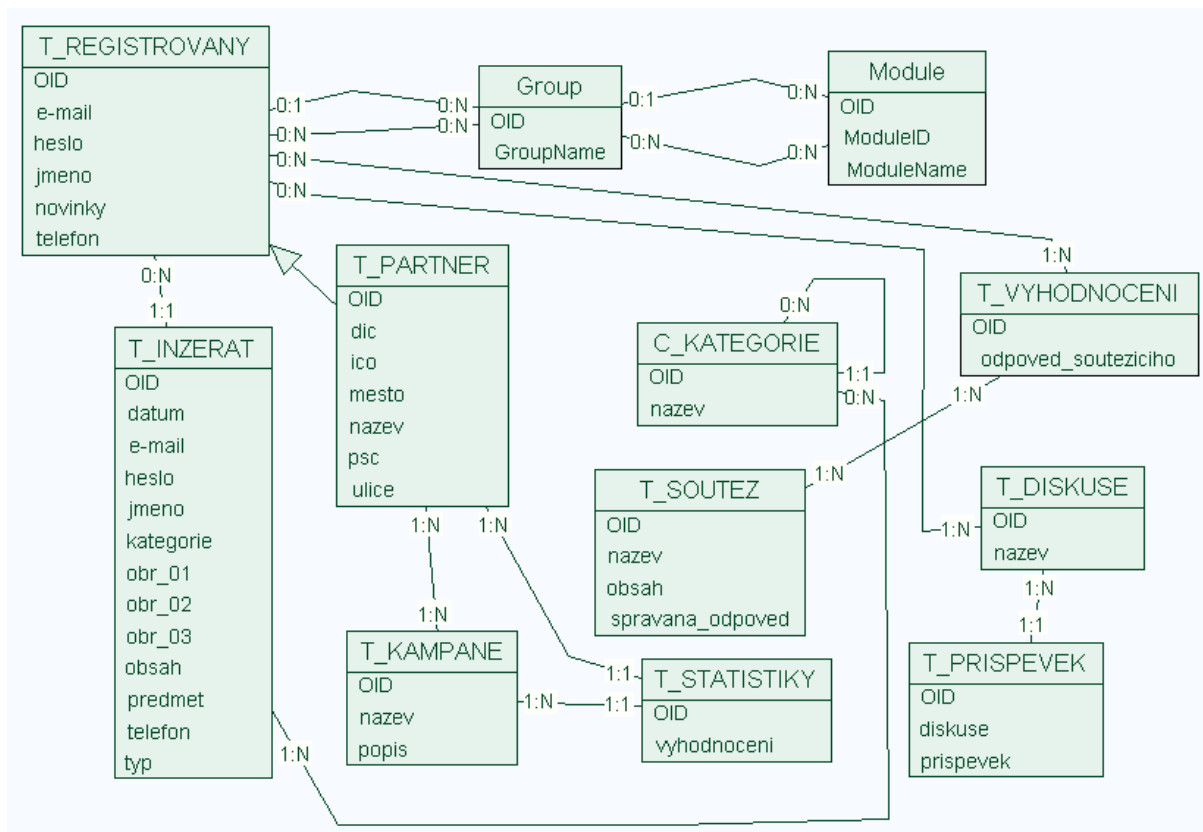
Datový model je určen (stejně jako v rámci UML modelování) pro návrh datové struktury, ze které pak bude budoucí internetová aplikace informačně vycházet. Model je tvořen sadou entit, které jsou mezi sebou vzájemně relačně provázány.

---

<sup>6</sup> XML: značkovací jazyk, který je určen především pro výměnu dat mezi aplikacemi

<sup>7</sup> XSL: rodina jazyků, umožňujících popsat způsob, jakým se mají XML soubory formátovat a předávat

Na obrázku č. 58 můžeme všechny entity a jejich vztahy názorně vidět. Jednotlivé atributy (e-mail, heslo, jméno apod.) mají (jako v rámci UML diagramů) určen datový typ, který je však implicitně uložen v samotném systému.



Obr. č. 58: Datový model v rámci WebML

### 3.4.2 Hypertextový model

Abychom mohli vytvořit hypertextový model, musíme mít sestavený model datový, ze kterého hypertextový vychází. Cílem modelu je stanovit strukturu (kompozici) a navigaci.

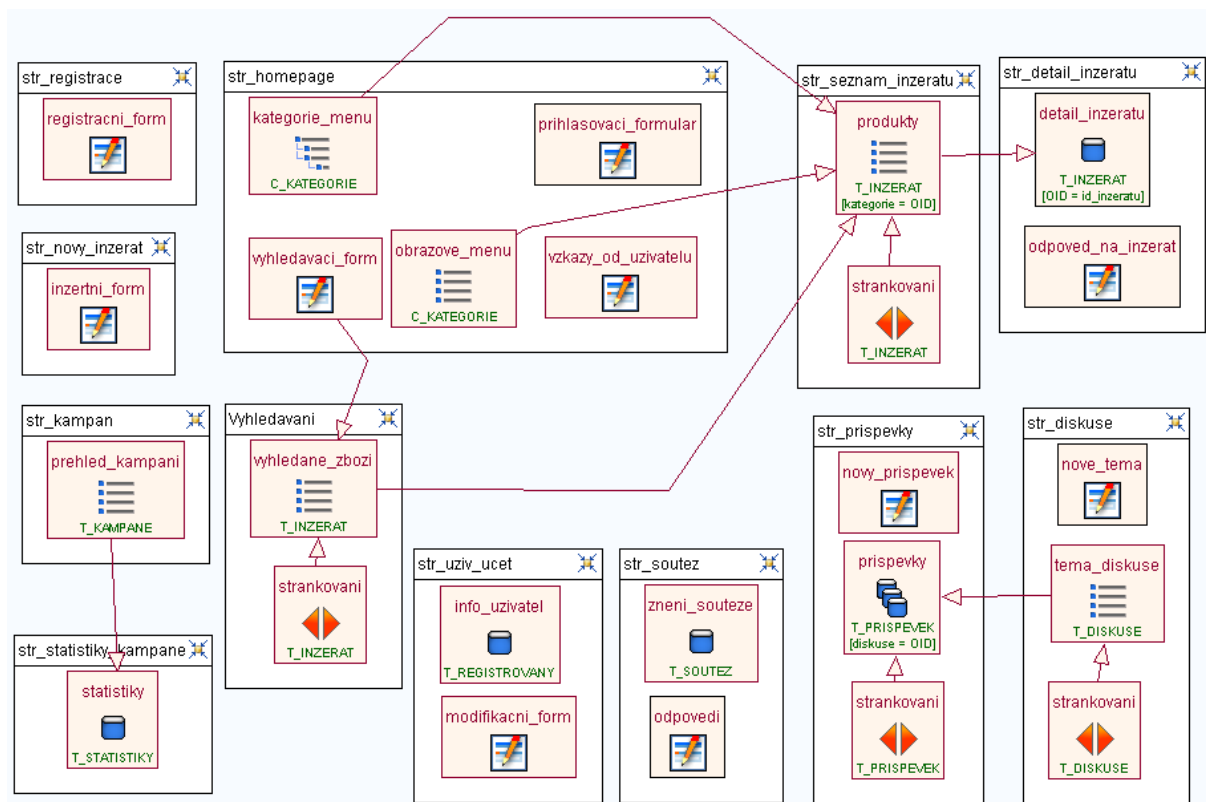
Nejprve musíme definovat logickou strukturu webové aplikace. Každá aplikace se skládá z určitého počtu webových stránek, které obsahují základní elementy. WebML tyto elementy nazývá *Units* a popisuje tyto základní typy:

- **Data units:** obsahují informace o jedné instanci zvolené entity (v našem případě se bude jednat například o jeden inzerát)

- **Multidata units:** obsahují informace o skupině instancí zvolené entity (v našem případě všechny příspěvky v rámci založených diskusí)
- **Index units:** představují seznamy instancí entity za účelem navigace (v našem případě půjde o zobrazení všech inzerátů spadající pod určitou kategorii)
- **Scroller units:** představují prostředek přístupu k jednotlivým instancím nebo skupinám instancí zvolené entity (v našem případě se bude jednat např. o stránkování na další inzeráty)
- **Entry units:** slouží pro modelování vstupů od uživatele (v našem případě se bude jednat např. o formulář pro podávání inzerátů)

Naše vzorová aplikace se bude skládat z 12 webových stránek, obsahujících jednotlivé instance pro zapisování, zobrazování, vyhledávání a mazání informací (viz obrázek č. 59).

Každá instance, která je určena pro zobrazení dat, obsahuje informaci o zdrojové tabulce, ze které bude čerpat. Jednotlivé instance jsou mezi sebou propojeny v rámci jedné nebo více webových stránek prostřednictvím čar.



Obr. č. 59: Hypertextový model v rámci WebML



## 4 Implementace projektu

Pokud používáme některý z CASE nástrojů, který umožňuje podporu celého životního cyklu projektu od analýzy až po implementaci (jako například software *WebRatio*), nemusíme se již shánět pod jiných nástrojích. Implementaci provádíme tedy přímo v daném CASE nástroji, který sám generuje programový kód a podklady pro databázový systém. V opačném případě využijeme dalších specializovaných „implementačních“ aplikací.

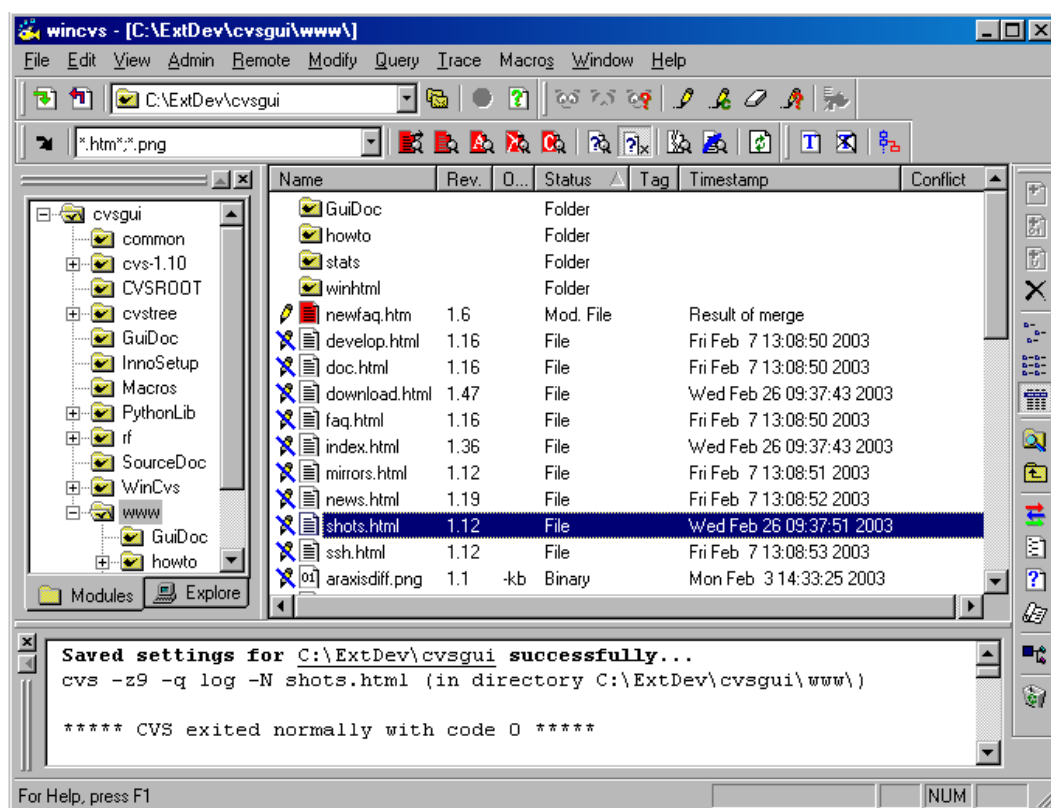
### 4.1 Týmová práce

Jestliže bude na implementaci pracovat více lidí najednou (obvykle tomu tak je), je dobré využívat software, který má funkci umožňující správu verzí pro software. *WebRatio* tento nástroj k dispozici má. V jiném případě vzniká potřeba se po nějakém nástroji poohlédnout. Existuje například software WinCvs [43], který je v rámci GNU<sup>8</sup> licence volně k dispozici a navíc je velice uživatelsky přívětivý (viz obrázek 60).

Pomocí takovýchto nástrojů bychom tedy měli zabránit tomu, že si budou programátoři vytvořené skripty navzájem přepisovat.

---

<sup>8</sup> Licence umožňující libovolné zacházení se softwarem



Obr. č. 60: WinCvs: software pro správu verzí

## 4.2 Tvorba databáze

Prvním krokem v rámci fáze implementace projektu by měl být výběr databázového serveru a tvorba samotné databáze. Jako podklad pro vytvoření databáze jsou ideální datové modely (konceptuální a fyzický model), které byly prezentovány v kapitole o UML.

Nejpoužívanější databázové servery jsou následující:

**MSSQL:** jedná se o systém, který využívá řada podnikových informačních systémů. Databázová platforma je dostupná pouze pro operační systémy Microsoft Windows. Systém plně podporuje standardy jazyka SQL<sup>9</sup> (pohledy, procedury, trigger<sup>10</sup> apod.). Tento systém vyvíjí firma Microsoft [54].

<sup>9</sup> SQL: standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích

<sup>10</sup> Trigger: v databázi definuje činnosti, které se mají provést v případě definované události nad databázovou tabulkou

**Oracle (Oracle databáze management systém - DBMS):** moderní multiplatformní databázový systém s velice pokročilými možnostmi zpracování dat, vysokým výkonem a snadnou škálovatelností. Systém je schopen tvořit uložené procedury, uživatelské funkce, programové balíky a triggery. Tento systém vyvíjí firma Oracle Corporation [55].

**PostgreSQL:** tento databázový systém je následovníkem systému POSTGRES. Jedná se o plnohodnotný relační databázový systém s otevřeným zdrojovým kódem. Funguje na všech rozšířených operačních systémech, podporuje cizí klíče, operace JOIN<sup>11</sup>, pohledy, triggery a uložené procedury [56].

**MySQL:** vznikla jako reakce na nedostatečnou flexibilitu a rychlost již existující databáze *mSQL*. Předností tohoto databázového serveru je rychlost, přenositelnost, dobrá podpora propojení s okolním světem a mnoho vestavěných SQL funkcí. Systém funguje téměř na všech operačních systémech. Serveru chybí transakce, triggery nebo referenční integrita. Bez těchto funkcí a relací se dá však obejít [57].

Pokud nedisponujeme již hotovým vygenerovaným skriptem pro vytvoření projektové databáze (v rámci využití některého z CASE nástrojů), začínáme databázi tvořit vkládáním jednotlivých tabulek podle vzoru fyzického datového modelu. Vytvoření tabulek není složité, neboť všechny výše uvedené databázové systémy mají velice přívětivé uživatelské rozhraní, pomocí něhož můžeme zakládat nové tabulky zcela bez problémů.

Po založení všech tabulek včetně vyplnění jejich položek a datových typů můžeme začít utvářet mezi tabulkami relace (např. referenční integrita).

### 4.3 Vývojové prostředí

Implementaci můžeme samozřejmě provést i bez pomoci CASE nástroje, který nám na základě vstupních údajů/modelů vygeneruje programový kód. Podlé mých zkušeností se většinou vyplatilo napsat si zdrojový kód sám, nežli k tomu využívat funkci automatického generování.

---

<sup>11</sup> SQL příkaz pro spojování tabulek

Při tvorbě webové prezentace se například používají tzv. *WYSIWYG* editory. Jedná se o nástroje, které pracují na základě vizuálního rozhraní, kde stránky tvoříte v podstatě vkládáním objektů a jejich formátováním (jako tomu je např. v textovém editoru MS Word). Nevýhoda je v tom, že se při kompilaci ve zdrojovém kódu často vyskytují redundantní znaky, které ve finále způsobí zmatek a ztrátu validity celého dokumentu. Další nevýhoda je v tom, že po zobrazení zdrojového kódu často ani nevíte, co který blok zdrojového kódu reprezentuje, tudíž do něho můžete jen těžko zasahovat, aniž byste nepodstoupili riziko ztráty funkčnosti. Poslední nevýhodou je fakt, že editory typu *WYSIWYG* jsou dosti omezené, co se možností týká. Složitější požadavky editor prostě neumí.

Profesionální přístup při implementaci však vyžaduje profesionální editory a programové prostředí. Níže jsou uvedeny nejpoužívanější prostředí pro tvorbu webových aplikací (a nejen webových). Všechna prostředí jsou určena pro operační systém MS Windows.

- **.NET:** jedná se o šablonu (vývojové prostředí) pro tvorbu aplikací, kde základní komponentou je platforma .NET Framework. Na Framework lze nahlížet jako na obrovskou kolekci funkcí pro prakticky libovolný programátorský problém. Tvůrcem tohoto prostředí je společnost Microsoft [58].
- **Homesite:** je koncipován jako nevizuální editor HTML stránek, ale již od verze 4.0 umožňuje i tvorbu vizuální. Homesite umožňuje například kontrolovat správnost syntaxe dokumentu, odkazy nebo barevně odlišovat tagy<sup>12</sup>. Dále program podporuje klávesové zkratky, což velice usnadní práci. FTP klient je u této aplikace samozřejmostí. Výrobcem tohoto produktu je známá společnost Adobe [59].
- **PSPad:** je český volně šiřitelný (freeware) univerzální editor, určený pro operační systém Microsoft Windows. V tomto editoru můžete pracovat se soubory v různých formátech, kódováních apod. Jednotlivá syntaxe je barevně zvýrazněna, což velice usnadňuje orientaci v kódu. Nástroj je velice jednoduchý, jehož funkce jsou zcela transparentní. Autorem tohoto editoru je Jan Fiala [60].
- **Golden HTML Editor:** je programový editor české produkce, jehož počátek se datuje k roku 1996. Dlouholeté zkušenosti vývojářů tohoto softwaru zajišťují jeho výbornou

---

<sup>12</sup> Tagy: HTML značky pomocí kterých se formátuje celý webový dokument

funkčnost a přehlednost. Mezi hlavní rysy programu patří rychlé (intuitivní) vkládání HTML tagů a jejich parametrů (prostřednictvím tzv. CodeAssistant). Dále program umožňuje číst a upravovat stránky napsané v PHP včetně nápovědy pro zadávání jednotlivých příkazů a funkcí [61].

## 4.4 Programovací jazyk

Pomocí programovacího jazyka (někdy jazyků) de facto převádíme jednotlivé návrhy (modely) do reality. Jestliže máme k dispozici kvalitní podklady v podobě textových a obrazových specifikací, fáze programování by pro nás neměla představovat žádný větší problém.

Hlavní programovací jazyky, které slouží pro tvorbu webových aplikací, jsou následující:

- **ASP.NET:** je součástí .NET Frameworku (viz kapitola 4.3) a je určen pro tvorbu webových aplikací a služeb. Podporuje připojení na databázové servery MSSQL, PostgreSQL atd. Aplikace založené na ASP.NET jsou rychlejší, protože jsou (na rozdíl od ryze skriptovacích jazyků) překompilovány do jednoho či několika málo DLL souborů (knihoven).

ASP.NET ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je tedy možné používat ovládací prvky jako tlačítko (Button), nápis (Label) a další. Těmto prvkům lze přiřazovat určité vlastnosti, zachytávat na nich události, atd. Tak, jako se ovládací prvky pro Windows samy kreslí do formulářů na obrazovku, webové ovládací prvky produkují HTML kód, který tvoří část výsledné stránky poslané do klienta prohlížeče.

- **PHP** (PHP: Hypertext Preprocessor): je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek [62]. Nejčastěji se zakomponuje přímo do struktury jazyka HTML (XHTML). PHP skripty jsou

prováděny na straně serveru a ke klientovi (uživateli) je přenášén již zkompileovaný výsledek. PHP je nezávislý na platformě a díky tomu je hojně využíváný a oblíbený. PHP umožňuje bezproblémový přístup k většině databázových serverů (MySQL, ODBC, Oracle, PostgreSQL, MSSQL).

- **JSP** (JavaServer Pages): technologie, založená na jazyku Java, která umožňuje vývoj dynamických webových stránek [63]. JSP soubory jsou vlastně HTML stránky, do kterých jsou vloženy speciální tagy (značky) obsahující javovský zdrojový kód.

## 4.5 Projektová dokumentace

Pokud je webová aplikace hotova (databáze a programování), měl by zhotovitel vypracovat tzv. projektovou dokumentaci. Dokumentace musí obsahovat zdrojové soubory ke všem obrázkům, dokumentaci dynamických skriptů, databázový exportní soubor a hlavně podrobný popis funkčnosti celé aplikace včetně jejích jednotlivých modulů. Kompletní technický popis projektu slouží pro případné budoucí modifikace nebo inovace systému.

Často se stává, že zhotovitel vytvoří velice podprůměrnou dokumentaci nebo dokonce žádnou. Tomuto stavu může zadavatel předejít již při sestavování smlouvy o dílo stanovením přesných pravidel.

## 4.6 Případová studie implementace

V rámci tvorby databáze pro náš inzertní systém si zvolíme databázový systém *MySQL Admin*. Tento systém je pro tvorbu aplikace naprosto dostačující a hlavně ho později oceníme pro jeho jednoduchou formu komunikace se zdrojovými kódem.

Nejprve si vytvoříme požadované tabulky. K tomu využijeme již několikrát zmiňovaného fyzického datového modelu. Abychom vytvořili jednotlivé tabulky, nepotřebujeme používat přímo jazyk SQL. Stačí nám k tomu přívětivé uživatelské grafické rozhraní, pomocí něhož nám vytvoření tabulky nedá žádnou práci.

Na obrázku č. 61 můžeme vidět tabulku *T\_INZERECE* včetně jejích položek, datových typů, a primárního klíče. Dané položky nebo jejich datové typy můžeme prostřednictvím uživatelského rozhraní kdykoliv měnit či mazat. K tomu slouží sloupce, které na obrázku spadají pod výraz *Akce*.

Obdobně vytvoříme i ostatní tabulky.

	Sloupec	Typ	Porovnávání	Vlastnosti	Ilulový	Výchozí	Extra	Akce					
<input type="checkbox"/>	id_inzerce	int(11)			Ne	0		Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	id_kategorie	int(11)			Ne	0		Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	id_podkategorie	int(11)			Ne	0		Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	id_registrovani	int(11)			Ne	0		Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	predmet	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	obsah	text	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	email	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	telefon	varchar(15)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	uzivatel_jmeno	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	uzivatel_heslo	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	obr_01	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	obr_02	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	obr_03	varchar(50)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	datum	datetime			Ne	0000-00-00 00:00:00		Změnit	Odstranit	Primární	Index	Unikátní	Fulltext
<input type="checkbox"/>	typ_inzeratu	char(1)	latin2_general_ci		Ne			Změnit	Odstranit	Primární	Index	Unikátní	Fulltext

Zaškrtnout vše /  Odškrtnout vše    Zaškrtnuté:  nebo  nebo  nebo  nebo  nebo

---

**Náhled k vytištění**   **Navrhnout strukturu tabulky** ?  
 Přidat  položek    Na konci tabulky    Na začátku tabulky    Po   

---

Indexy: ?					Využití místa:		Statistika řádků:	
Klíčový název	Typ	Mohutnost	Akce	Sloupec	Typ	Používá	Údaj	Hodnota
PRIMARY	PRIMARY	0	<input type="button" value="Upravit"/> <input type="button" value="Odstranit"/>	id_inzerce	Data	0 bajtů	Formát	dynamický
Vytvořit index na <input type="text" value="1"/> sloupcích <input type="button" value="Proved'"/>					Index	1 024 bajtů	Porovnávání	latin2_general_ci
					Celkem	1 024 bajtů	Řádků	0
							Vytvoření	Neděle 05. srpna 2007, 23:19
							Poslední změna	Neděle 05. srpna 2007, 23:19

Obr. č. 61: Zobrazení tabulky T\_INZERCE v databázovém systému MySQL

Poté, co jsme vytvořili databázi, můžeme započít s programováním zdrojového kódu aplikace. Pro náš příklad inzertního systému použijeme programovací jazyk PHP, který budeme využívat v editoru *PSPad*.

Vzhled grafického rozhraní editoru můžeme vidět na obrázku č. 62. Na první pohled by se mohlo zdát, že se jedná o poznámkový blok (standardní komponentu operačního systému Windows). On to takový poznámkový blok vlastně je, nicméně jen po grafické stránce. Po stránce funkční se jedná o velice silný a efektivní nástroj pro tvorbu webových prezentací/aplikací.

Na obrázku můžeme vidět zvýrazněnou syntaxi, což napomáhá k lepší orientaci ve zdrojovém kódu (obzvlášť pokud se jedná o objemnější kód).



```

PSPad - [C:\MICHAL\WWW\WWWprezentaceARMYBURZA\index.php]
Soubor Projekt Úpravy Hledat Zobrazit Formát Nástroje HTML Nastavení Okno Nápověda
Nový.txt index.php
0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160
<?php
session_start();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<meta name="generator" content="PSPad editor, www.pspad.com">
<meta name="copyright" content="Copyright C 2004 Michal Pivonka">
<meta name="robots" content="follow">
<meta name="revisit-after" content="7 days">
<meta name="keywords" content="army, burza, armyshop, army inzerce, inzerce, burza, militarie, nabídka, poptavka, vojensky, valecny">
<meta name="description" content="ARMYBURZA: nabídka/poptávka uniform, výstroje, výzbroje, vyznamenání, vozidel, literatury, airsoftových produktů.">
<meta name="Author" content="Michal Pivonka, email: webmaster@armyburza.cz">
<!--if IE--<script type="text/javascript" src="http://www.armyburza.cz/js/fix_eolas.js" defer="defer"></script><!--endif-->

<?php
$page = $_REQUEST["page"];
if ($page == "detailInzeratu")
{
include "connectDB.php";
$idInzeratu = $_REQUEST["idInzeratu"];
$uplnyNazevDB = mysql_query("SELECT predmet FROM t_inzerce where id = $idInzeratu");
$rowUplynNazev = mysql_fetch_assoc($uplnyNazevDB);
$uplnyNazevProdukt = $rowUplynNazev["predmet"];
$title = "Armyshop / Armyburza : $uplnyNazevProdukt";
}
else {$title = "Armyshop / Armyburza : obchod a burza s army zbožím / military";}
?>
<title><?php echo $title ?></title>
<link rel="stylesheet" type="text/css" href="styly.css">
</head>

<body>
<div id="prouzek" class="odKasProuzek"><table border="0" cellpadding="0" cellspacing="0" align="left" valign="top" width="100%"><tr><td width=226><img src='\"Images/Reklama/Armyprodej/armyprodej135x220.jpg\"' border='\"0\"' alt='\"\"'></a\" ";
if ($page == "inzeraty" || $page == "vyhledavani" || $page == "diskuse") {$vyskaTabulky = "height=2000" };

//$reklamaArmyzet = "<br><a href='\"Static/proklik.php?id=61\"' target='\"_blank\"'><img src='\"Images/Reklama/Elacena/armyzet135x190.jpg\"' border='\"0\"' alt='\"\"'></a\"
$reklamaArmykauke = "<br><object classid='\"clsid:d27c66e-ae6d-11cf-96b8-444553540000\"' codebase='\"http://fpdownload.macromedia.com/pub/shockwave/cabs/flash/sw
<param name='\"allowScriptAccess\"' value='\"sameDomain\"' />
<param name='\"movie\"' value='\"Images/Reklama/2darma/sadal_120x600.swf?timeout=60&amp;clickthru=Static/proklik.php?id=58\"' /><param nam
</object> ";}

else {$vyskaTabulky = "height=800" ; $reklamaArmykauke = "" ;}

```

Obr. č. 62: Zvýrazněná syntaxe v editoru PSPad

Základem programování je HTML stránka, do které se poté vkládají jednotlivé PHP skripty (popřípadě skripty jiných programovacích jazyků). Každá HTML stránka podléhá určité konvenci uspořádání jednotlivých značek (tagů) a jejich atributů.

HTML dokument se skládá ze specifikace typu dokumentu (*!DOCTYPE*), hlavičky (*HEAD*) a těla (*BODY*). Specifikace udává verzi dokumentu. V našem případě se bude jednat o *HTML 4.01 Transitional*.

Po definici dokumentu následuje značka *<HTML>*, která určuje začátek vlastního HTML dokumentu. Tato značka je párová (jako většina HTML značek), což znamená, že pokud jsme vytvořili počáteční značku (například *<HTML>*), musíme k ní vytvořit i značku ukončovací (*</HTML>*). Ukončovací značka je velice podobná počáteční, jen má navíc před názvem značky lomítko. To, co je uvedeno mezi počátečním a ukončovacím tagem *HTML*, je obsahem dokumentu.

Uvnitř obsahu dokumentu se uvádí značka označující hlavičku dokumentu. Označuje se tagem `<HEAD>`, který je, stejně jako u značky pro dokument, párový.

Uvnitř hlavičky se nachází tzv. metatagy, což jsou zjednodušeně řečeno informace o informacích. Pomocí metatagů určíme například znakovou sadu (*charset*), ve které bude dokument psán. Jedná se o velice důležitý krok při vytváření HTML dokumentu, jelikož pokud si zvolíme špatnou znakovou sadu, mohou se nám některá písmena zobrazovat špatně. Pro podporu češtiny se využívají dva typy standardů *iso-8859-2* nebo *windows-1250*<sup>13</sup>. Tento metatag je povinný. Dále můžeme prostřednictvím metatagů zadat autorská práva, autora, klíčová slova dokumentu nebo jeho popis.

Součástí hlavičky je také titulek dokumentu. Ten určuje, co se bude při otevření dokumentu zobrazovat v horní liště okna prohlížeče.

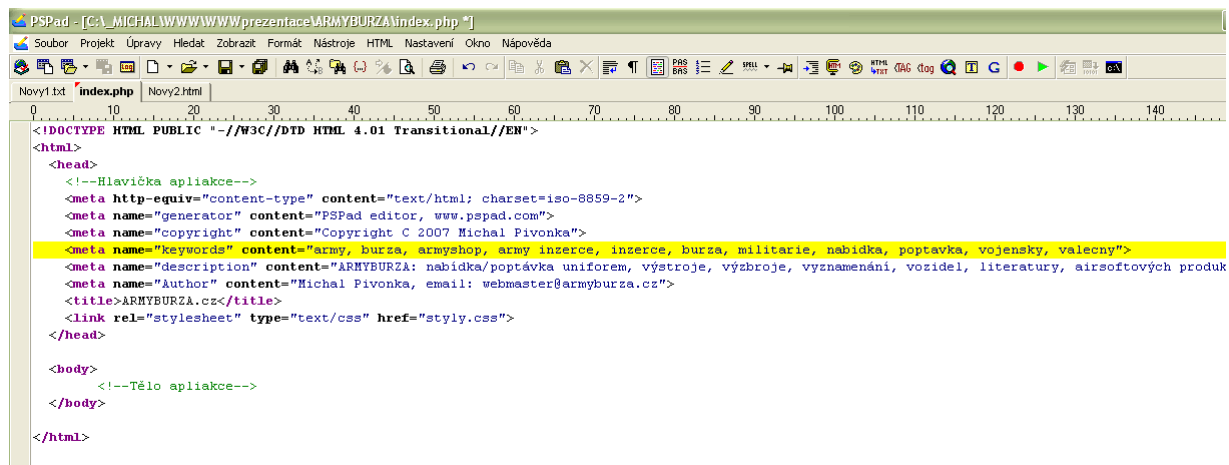
Posledním tagem hlavičky dokumentu je zpravidla odkaz na externí stylopis (`<link>`), pomocí kterého se budou strukturovat a formátovat jednotlivé objekty dokumentu. Tento tag není povinný.

Po ukončovacím tagu hlavičky (`</HEAD>`) následuje tělo dokumentu. To se označuje tagem `<BODY>`. Mezi tento počáteční tag a jeho ukončovací tag (`</BODY>`) se píše celý obsah dokumentu.

Posledním tagem je ukončovací značka celého dokumentu (`</HTML>`). Pro lepší pochopení popisu slouží obrázek č. 63.

---

<sup>13</sup> Jedná se o kódování, které podporuje diakritiku

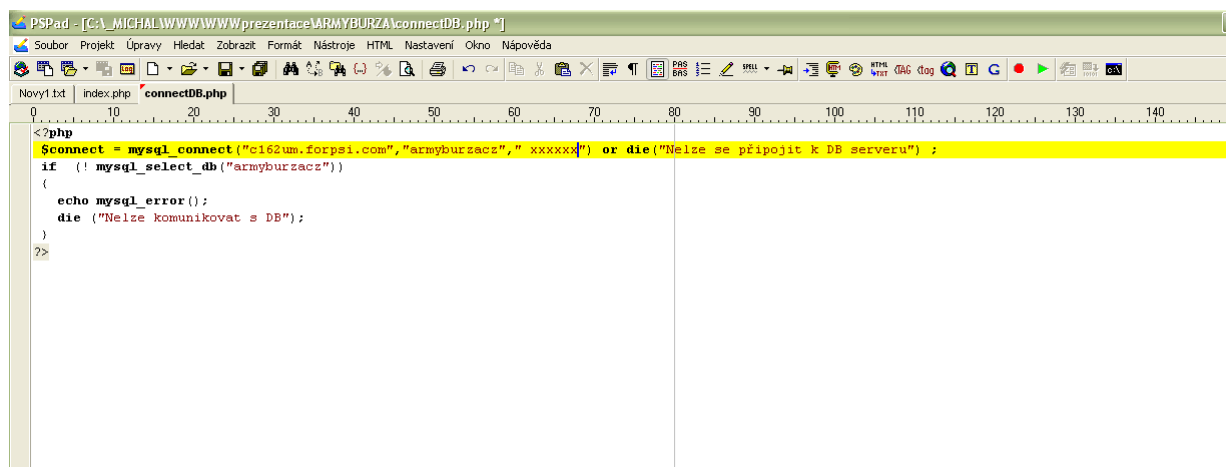


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<!--Hlavička aplikace-->
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<meta name="generator" content="PSPad editor, www.pspad.com">
<meta name="copyright" content="Copyright C 2007 Michal Pivonka">
<meta name="keywords" content="army, burza, armyshop, army inzerce, inzerce, burza, militarie, nabídka, poptavka, vojensky, valecny">
<meta name="description" content="ARMYBURZA: nabídka/poptávka uniform, výstroje, výzbroje, vyznamenání, vozidel, literatury, airsoftových produk">
<meta name="author" content="Michal Pivonka, email: webmaster@armyburza.cz">
<title>ARMYBURZA.cz</title>
<link rel="stylesheet" type="text/css" href="styly.css">
</head>
<body>
<!--Tělo aplikace-->
</body>
</html>
```

Obr. č. 63: Základní definice HTML dokumentu v editoru PSPad

Jednotlivé skripty, které jsou obsahem dokumentu, zde vypisovány samozřejmě nebudou, nicméně ještě uvedu skript pro navázání komunikace s databázovým serverem a skript obsahující dotaz nad vybranou tabulkou.

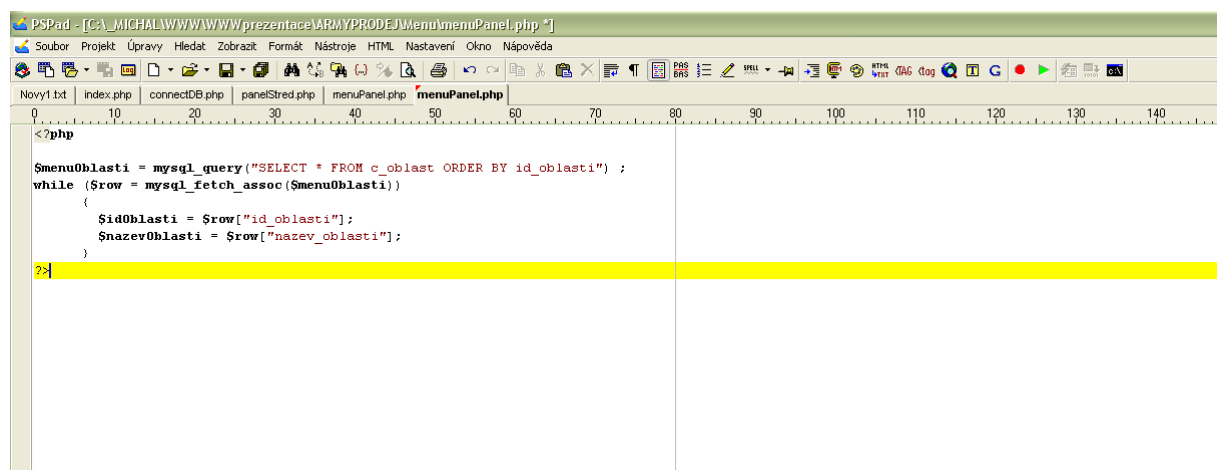
Jak již bylo uvedeno výše, PHP dokáže navázat spojení s databázovým serverem velice snadno. Pro spojení stačí použít PHP funkci `mysql_connect()`, která má tři povinné parametry: adresu databázového serveru, login a heslo. Poté, co se úspěšně připojíme, si musíme zvolit databázi, nad kterou budeme chtít provádět dotazy. Výběr databáze a následné spojení zprostředkovává PHP funkce `mysql_select_db()`. Tato funkce má pouze jediný parametr, kterým je název databáze. Na obrázku č. 64 můžeme vše vidět v praxi.



```
<?php
$connect = mysql_connect("c162um.forpsi.com", "armyburzacz", "xxxxxx") or die("Nelze se připojit k DB serveru");
if (!mysql_select_db("armyburzacz"))
{
    echo mysql_error();
    die("Nelze komunikovat s DB");
}
?>
```

Obr. č. 64: PHP skript popisující spojení s databázovým serverem MySQL v editoru PSPad

Nyní, když máme spojení s databází, můžeme provést vlastní dotazování. V našem případě budeme (za účelem zobrazení menu se všemi kategoriemi inzerátů) vytvářet dotaz s cílem vypsát veškerý obsah tabulky C\_KATEGORIE. Pro stanovení dotazu slouží funkce *mysql\_query()*, která má jediný parametr v podobě SQL dotazu. Další funkce *mysql\_fetch\_assoc()* zajistí načtení výsledného pomyslného řádku tabulky. Hodnotu určitého sloupce v rámci jednoho řádku zjistíme zavoláním konkrétního jména sloupce tabulky (viz obrázek č. 65).

The image shows a screenshot of the PSPad text editor. The title bar indicates the file path: [C:\MICHAL\WWW\WWW\prezentace\ARMYPRODEJ\Menu\menuPanel.php \*]. The menu bar includes options like Soubor, Projekt, Úpravy, Hledat, Zobrazit, Formát, Nástroje, HTML, Nastavení, Okno, and nápověda. The toolbar contains various icons for file operations and editing. The main editing area shows PHP code for a database query. The code is as follows:

```
<?php
$menuOblasti = mysql_query("SELECT * FROM c_oblast ORDER BY id_oblasti") ;
while ($row = mysql_fetch_assoc($menuOblasti))
{
    $idOblasti = $row["id_oblasti"];
    $nazevOblasti = $row["nazev_oblasti"];
}
?>
```

The code is enclosed in PHP tags. The first line uses *mysql\_query()* to execute a SQL query that selects all columns from the *c\_oblast* table, ordered by *id\_oblasti*. The second line starts a *while* loop that iterates over the results returned by *mysql\_fetch\_assoc()*. Inside the loop, two variables are assigned: *\$idOblasti* is assigned the value of the *id\_oblasti* column, and *\$nazevOblasti* is assigned the value of the *nazev\_oblasti* column. The code ends with a closing PHP tag. A yellow highlight is visible under the closing PHP tag.

Obr. č. 65 PHP skript popisující výběr databáze z databázového serveru MySQL v editoru PSPad

Obsáhnout zde celý popis skriptovacího jazyka PHP není předmětem této práce, nicméně jsem považoval za nutnost zmínit alespoň základní definici struktury HTML dokumentu a způsob spolupráce mezi skriptovacím jazykem a databází.

## 5 Testování a optimalizace prototypu

Tato fáze vývoje projektu je velice důležitá z hlediska nalezení a odstranění případných chyb a nesrovnalostí. Úroveň testování a optimalizace je často nedostačující nebo zcela opomíjena.

Během testování bychom měli analyzovat míru funkčnosti a použitelnosti aplikace.

Použitelnost webových stránek určuje, jak snadno se na nich uživatelé orientují, jak rychle pochopí jejich uspořádání, ovládání a jaký dojem si z nich odnesou. Dobře použitelné stránky jsou přehledné, srozumitelné a intuitivní. Na špatně použitelných stránkách uživatelé tápou, nedokáží dosáhnou svého cíle (v naší případové studii se může jednat například o neschopnost podat inzerát) a ze stránek odcházejí zklamaní.

### 5.1 Programátorské testování funkčnosti

Prvotní testování, týkající se funkčnosti aplikace, by měl provádět sám programátor. Předmětem testování je především kontrola bezchybného ukládání, úpravy, zobrazování a mazání dat. Nesmí se stát, že se bude ukládat, zobrazovat nebo mazat pouze část záznamu (způsobená např. špatně zadaným SQL dotazem).

Programátor by měl také ošetřit bezchybnou funkčnost případných omezení v rámci vyplňování formulářových polí. Pokud má textové pole akceptovat pouze vkládané hodnoty numerického charakteru (například při vkládání identifikačního čísla podnikatele), nesmí se stát, že by uživatel vložil jinou hodnotu (např. text). To samé platí u omezení počtů znaků (např. při zadávání PSČ), kde by se uživateli nemělo podařit zadat více nebo méně znaků, než je požadováno.

Dalším požadavkem při testování je bezchybné a plně funkční provázání jednotlivých stránek prostřednictvím hypertextových odkazů.

## 5.2 Všeobecný rozbor

Všeobecný rozbor stránek se skládá ze dvou hlavních částí: **analýzy nedostatků** a **doporučení úprav stránek** a je obvykle prováděn odbornou poradenskou firmou [13].

### 5.2.1 Analýza nedostatků

Úspěšnost webových stránek je ovlivňována mnoha faktory. Analýza tvořící první část všeobecného rozboru stránek postihuje ty nejdůležitější, které jsou uspořádané do níže uvedených čtyř skupin:

- **Přístupnost stránek:** předmětem této části analýzy je zjištění přístupnosti obsahu stránek. Špatná přístupnost stránek může způsobit pokles potenciálních návštěvníků až o třetinu.
- **Použitelnost stránek:** zde dochází k prověření skutečnosti, jak rychle se uživatel na stránkách zorientuje.
- **Obsah stránek:** testuje srozumitelnost a přehlednost obsahu stránek. Dochází také ke kontrole stylistiky, gramatiky, informační hodnoty a přesvědčivosti.
- **Viditelnost stránek:** určuje, jak snadno jsou stránky na internetu k nalezení. Testuje se zejména přítomnost odkazů ve vyhledávačích, katalozích a dalších internetových zdrojích.

### 5.2.2 Doporučení úprav

Doporučení úprav je druhou hlavní částí všeobecného rozboru stránek a obsahuje již konkrétní kroky vedoucí k nápravě chyb zjištěných předchozí analýzou. Jednotlivé kroky jsou uspořádány do tří skupin:

- **1. skupina:** v ní se zpravidla objevují nápravy zjištěných chyb, které způsobují zásadní škody a jejich odstranění má nejvyšší prioritu. Platí zde pravidlo, že pokud se

chyby v této skupině nenapraví, nemohou testované stránky plnit cíle, pro které byly vytvořeny.

- **2. skupina:** obsahuje úpravy odstraňující méně významné nedostatky a úpravy, které lze provést rychle a s minimálními finančními náklady.
- **3. skupina:** v této skupině jsou uvedeny typy úprav, jejichž přínos je menšího rázu či dokonce nejistý a náklady na jejich realizaci nemusí být vždy přímo úměrné dosaženému efektu.

### **5.3 Audit přístupnosti**

Přístupné stránky se vyznačují tím, že respektují uživatele. Respektují jeho technické vybavení, zdravotní dispozice, znalosti, zkušenosti a vzdělání. Stránky nijak neomezují uživatele méně běžných internetových prohlížečů (např. Opera), barvoslepé uživatele, dyslektiky apod.

Přístupnost se testuje různými metodami v rámci světově uznávaných konvencí WCAG [48], Section 508 [49], Blind Friendly Web [50] nebo podle předpisu k Zákonu č. 365/2000 Sb. O informační systémech veřejné správy [47].

### **5.4 Uživatelské testování použitelnosti (user testing)**

Použitelnost stránek lze odborně odhadnout jen do určité míry. Některé zásady sice platí vždy, ale rozhodující je především testování s věrohodným vzorkem potenciálních uživatelů (tzv. user testing). I ty nejmenší detaily hrají svou roli - volba slov v navigačním menu, grafické odlišení odkazů, rozmístění obsahu na stránce, velikost písma a desítky dalších. Každý uživatel je jiný, a proto se nelze spoléhat jen na osobní zkušenosti webdesignéra.

Použitelnost/transparentnost se testuje na lidech, kteří daný web neznají. Důsledkem pozorování této skupiny lidí často odhalíme největší slabiny systému.

Při plnění úkolů je uživatel pečlivě pozorován a jsou zaznamenávány všechny jeho reakce, úspěchy i klopýtnutí.

## 5.5 Heuristická analýza použitelnosti

Další formou testování je tzv. heuristická analýza použitelnosti. Ta spočívá v prohlížení webu a hodnocení jeho použitelnosti na základě známých pouček, pravidel a výsledků výzkumů.

Tato analýza je svým pojetím jedinečná. Nejedná se totiž pouze o hodnocení jasně daných ukazatelů, ale o prověření webu v kontextu jeho zaměření a poslání.

Analýzu zpravidla nabízejí specializované poradenské firmy.

## 5.6 Rozbor stránek z pohledu SEO

SEO (Search Engine Optimization) je metodologie vytváření a upravování webových stránek takovým způsobem, aby byly ve výsledcích hledání v internetových vyhledávacích zobrazeny na nejlepších pozicích (v místech, kde je hledající vyhledává). Cílem je získat co největší přístup návštěvníků (potenciálních zákazníků).

Mezi nejdůležitější prvky ovlivňující míru úspěšnosti v jednotlivých vyhledávacích patří:

- **Výběr hodných klíčových slov:** klíčová slova by měla co nejpřesněji reprezentovat tematický obsah webové prezentace/aplikace. Podle klíčových slov se pak jednotlivé vyhledávače orientují a posléze zobrazují odkazy na základě uživatelem zadaných dotazů.

Pokud bychom hledali vhodná klíčová slova pro inzertní systém v rámci naší případové studie, nejspíše bychom vybrali následující výrazy: *Inzerce, Burza, Nabídka, Poptávka, Armáda, Military, Vojenský* apod.

- **Kontrola přístupnosti stránek:** odhalí případné chyby bránící vyhledávačům stránky procházet a zařazovat do svých databází.
- **Navigační struktura:** testuje se struktura a způsob provázání stránek.
- **Zajištění zpětných odkazů:** pro úspěch ve vyhledávacích hraje velmi důležitou roli počet a relevance zpětných odkazů. Zpětné odkazy jsou odkazy směřující na naše stránky. Platí zde přímá úměrnost – čím více relevantních odkazů, tím lépe. Odkazy



by měly vést pouze z kvalitních a tematicky příbuzných webů. Opačné odkazy míru úspěchu ve vyhledávacích snižují.

- **Používání zakázaných postupů:** pokud vyhledávače zjistí, že na našich stránkách používáme některý ze zakázaných postupů zvýšení bonity stránek, bude těmto stránkám sníženo hodnocení, nebo může dokonce dojít k vymazání stránek z databáze. Příkladem zakázaného postupu může být vkládání skrytého textu (textu napsaného v barvě pozadí) obsahujícího klíčová slova. Dalším reprezentantem zakázaného postupu jsou klamná slova, což jsou slova, která nemají s obsahem prezentace žádnou souvislost, ale zároveň jsou internetovými uživateli často zadávaná.

## 6 Prezentace projektu

Jestliže vývoj aplikace úspěšně prošel fázemi testování a optimalizace, může se plynule přejít do úrovně prezentace.

Abychom umožnili uživatelům neustálý přístup k vytvořené webové prezentaci/aplikaci, musí být daný projekt umístěn na internet. Každé webové stránky ve světě internetu splňují dvě základní kritéria. Mají své specifické doménové jméno a webhosting.

### 6.1 Doména

Doména je obdobou klasické poštovní adresy. Tak jako je každá adresa unikátní, je i každá doména na internetu unikátní (jedná se vlastně o slovní překlad IP adresy<sup>14</sup>). Abychom si nemuseli pamatovat a na internetu vypisovat složité IP adresy serverů, na které se chceme podívat, zadáme jejich jmenný ekvivalent - doménu. O překlad doménového jména na IP adresu se starají tzv. DNS servery (jmenné servery).

Doménové jméno je tvořeno posloupností několika částí oddělených tečkami.

Domény se rozdělují podle úrovně na následující:

- **Nejvyšší domény 1. úrovně:** někdy označované také jako tzv. top-level domény (.TLD). V současné době existuje nepřehledné množství těchto domén [51]. Téměř každý stát má svoji .TLD (včetně Evropské Unie). .TLD doména pro Českou republiku je *.cz*.
- **Domény 2. úrovně:** jsou hned nalevo za národní (top-level) doménou a tvoří podstatu doménového jména. Například doména *armyburza.cz* je již kompletní doménové jméno, prostřednictvím kterého se můžeme prezentovat na internetu. Tato doména je složena z domény 2. úrovně (*armyburza*) a domény 1. úrovně (*cz*) je tedy plnohodnotná.

---

<sup>14</sup> Jednoznačná identifikace konkrétního zařízení (typicky počítače) v prostředí internetu

- **Domény 3. až n-té úrovně:** reprezentují další úrovně doménového jména (například `soutez.armyburza.cz`) a mohou sloužit např. pro přímý přístup do určité sekce webové prezentace/aplikace. Po napsání výše uvedené domény 3. úrovně do adresního řádku prohlížeče, budeme odkázáni přímo do určité části webových stránek. Alternativní (delší) cesta k dosažení stejného cíle by vedla přes zadání domény 2. úrovně a následné využití hypertextového odkazu do příslušné kategorie.

Pro podnikání na internetu je důležité mít vlastní doménu 2. úrovně a to zejména z prestižních důvodů. Měla by být dobře zapamatovatelná, krátká, ale výstižná. Potenciální uživatelé/zákazníci si budou jen těžko pamatovat rozsáhlý řetězec znaků nebo vágní názvy.

V doménových jménech lze používat pouze znaky anglické abecedy, číslice a pomlčku (každá část jména však musí začínat písmenem a nesmí končit pomlčkou). Jména nejsou citlivá na velikost písmen (`www.armyburza.cz` popisuje stejný počítač jako `WWW.ArMyBuRzA.cZ`) a každá část jména smí být maximálně 63 znaků dlouhá [53]. Délka celého jména může být maximálně 255 znaků.

Teoreticky můžeme vlastnit neomezený počet domén, avšak v praxi si společnosti většinou registrují jednu či dvě. Doména se kupuje na dobu určitou. Pokud si chcete doménu po uplynutí daného období prodloužit, musíte zaplatit tzv. udržovací poplatek.

Všechny internetové domény se registrují v národních registrech, které vystupují jako koordinátoři a zabraňují používání stejného názvu více subjekty.

V rámci naší republiky byl výhradním registrátorem CZ.NIC [52]. Od 13.10. 2003 již existuje více registrátorů domén a tudíž existuje i možnost výběru (viz tabulka č.1).

Provozovatel	Název	URL
Czech On Line, a.s.	Czech On Line	domeny.col.cz
GENERAL REGISTRY, s.r.o.	DomainMaster	www.domainmaster.cz
ACTIVE 24, s. r. o.	ACTIVE 24	www.active24.cz
ha-vel internet s.r.o.	ha-vel	domeny.ha-vel.cz
IGNUM, s.r.o.	DOMENA.CZ	www.domena.cz
INTERNET CZ, a.s.	Velkoobchod domén	domeny.velkoobchod.cz

MIRAMO spol. s r.o.	9net.cz	www.9net.cz
ZONER software, s.r.o.	RegZone!	www.regZone.cz
Telefónica O2 Czech Republic a.s.	O2	www.cz.o2.com/domeny
KRAXNET s.r.o.	XNET	www.xnet.cz
Media4web s.r.o.	Media4web	www.media4web.cz
ONE.CZ s.r.o.	REGISTRATOR.CZ	www.registrator.cz
Web4U s.r.o.	Správa domén	www.spravadomen.cz
GTS NOVERA a.s.	GTS Novera	www.gtsnovera.cz
IPEX a.s.	IPEX	www.ipex.cz
SkyNet, a.s.	SkyNet	www.skynet.cz
P.E.S. consulting, s.r.o.	FORPSI	www.forpsi.com
CZ.NIC, z.s.p.o.	LRR	www.lrr.cz

Tab. č. 1: Seznam všech registrátorů domén v ČR

## 6.2 Hosting

Hosting je zjednodušeně výraz pro zpřístupnění webové prezentace/aplikace na internet. Zpřístupnění můžeme dosáhnout například pronájmem webového prostoru (webhosting) u některého z poskytovatelů (viz příloha č. 3), pronájmem celého serveru (serverhosting), nebo si můžeme pronajmout pouze vysokorychlostní připojení k pátevní síti poskytovatele. K této síti pak umístíme vlastní server nebo servery. Vše záleží na rozsahu, technologii a náročnosti aplikace, kterou chceme zpřístupnit světu.

### 6.2.1 Webhosting

Webhosting můžeme jinými slovy označit jako pronájem www prostoru. Využití webhostingu představuje nejnadanější a neekonomičtější variantu pro vstup na internet. Pokud se majitel webové aplikace chce prezentovat na internetu nebo prodávat své zboží/služby on-line přes internet, musí buď vlastnit výkonný www server s určitým softwarovým vybavením včetně permanentního připojení k internetu (což je dosti finančně nákladné), nebo může využít právě služby webhostingu, která spočívá pouze v pronájmu určitého webového prostoru na serveru poskytovatele. Tím odpadá nutnost investovat do vlastního hardwaru a připojení k internetu. Platíme tedy pouze za nájem webového prostoru, jehož specifikaci si určíme výběrem vhodného webhostingového programu. Navíc webhostingové servery, na kterých si můžeme pronajmout www prostor, mají většinou velice kvalitní, tj. extrémně rychlé připojení

k internetu. To zajišťuje rychlou odezvu a snadnou dostupnost našich www prezentací a aplikací.

Na server provozovatele se přenáší jak databáze celé aplikace, tak i jednotlivé programové skripty (viz kapitola 4). Majitel prezentace/aplikace má většinou neomezený přístup ke svým www stránkám i databázi prostřednictvím uživatelsky přívětivého grafického rozhraní (vstup do administrátorského systému je zajištěn heslem, které majitel zpravidla obdrží ihned po zakoupení příslušného webhostingu).

Při volbě vhodného webhostingu musíme především dbát na to, aby pronajímaný prostor podporoval databázový server a programovací jazyk, ve kterém byla naše aplikace vytvořena. Škála podpory programových technologií se odvíjí od použitého operačního systému webhostingového serveru poskytovatele. K nejrozšířenějším operačním systémům patří *Microsoft Windows NT* a *Linux*.

Výše uvedeným typům pronajímaných prostor podporujících určité technologie se říká webhostingový program. Každý webhostingový provozovatel většinou nabízí několik programů (viz obrázek č. 66), ze kterých si zákazník může zvolit ten, který mu bude nejvíce vyhovovat. Pro náš případ inzertního serveru budeme potřebovat webhosting podporující databázový server MySQL a technologii PHP, ve které je aplikace napsána.

			
od <b>19 Kč</b> / měsíc	od <b>19 Kč</b> / měsíc	od <b>100 Kč</b> / měsíc	od <b>499 Kč</b> / měsíc
<b>EMAILhosting</b>	<b>PIDIhosting</b>	<b>NORMALhosting</b>	<b>EXCLUSIVEhosting</b>
<ul style="list-style-type: none"> <li>- 10 e-mailových schránek</li> <li>- <b>neomezený přenos dat</b></li> <li>- plná zákaznická podpora</li> <li>- POP3, SMTP, IMAP4, web rozhraní</li> <li>- volitelně neomezený počet schránek</li> <li>- volitelně schránka o velikosti až 1 GB</li> </ul> <p>*Akce: "Léto u FORPSI"</p>	<ul style="list-style-type: none"> <li>- 300 MB prostoru</li> <li>- 10 e-mailových schránek (typ PIDI+mail)</li> <li>- <b>neomezený přenos dat</b></li> <li>- plná zákaznická podpora</li> <li>- HTML - pouze statické stránky</li> <li>- operační systém Windows</li> </ul> <p>*Akce: "Léto u FORPSI"</p>	<ul style="list-style-type: none"> <li>- 3000 MB prostoru</li> <li>- 100 e-mailových schránek</li> <li>- <b>neomezený přenos dat</b></li> <li>- podpora dle výběru</li> <li>- ASP.NET 2.0, ASP, PHP</li> <li>- FTP / FrontPage</li> <li>- MySQL, volitelně MSSQL</li> <li>- POP3, SMTP, IMAP</li> <li>- Windows 2003 / Linux</li> </ul> <p>*Akce: "Léto u FORPSI"</p>	<ul style="list-style-type: none"> <li>- 15000 MB prostoru</li> <li>- 200 e-mailových schránek</li> <li>- <b>neomezený přenos dat</b></li> <li>- plná zákaznická podpora</li> <li>- ASP.NET 2.0, ASP, PHP</li> <li>- MySQL, MSSQL</li> <li>- FTP</li> <li>- POP3, IMAP, SMTP</li> <li>- Windows 2003 / Linux</li> </ul>
<a href="#">více info</a> <a href="#">objednat</a>	<a href="#">více info</a> <a href="#">objednat</a>	<a href="#">více info</a> <a href="#">objednat</a>	<a href="#">více info</a> <a href="#">objednat</a>

Obr. č. 66: Ukázka nabídky různých webových hostingů

Součástí webhostingu je i zřízení a správa e-mailových adres, které si majitel aplikace sám určí. E-mailová adresa se skládá ze dvou částí. První polovinu adresy tvoří jméno e-mailové schránky (např. *info*). Druhou část tvoří název naší domény (např. *armyburza.cz*). Název schránky od doménového jména vždy odděluje znak @. Pokud bychom chtěli vytvořit pro náš inzertní server e-mailovou schránku, do které by nám mohli uživatelé zasílat své připomínky či dotazy, zvolili bychom si například název [info@armyburza.cz](mailto:info@armyburza.cz).

Počet možných e-mailových schránek se odvíjí od zakoupené formy webhostingu, zpravidla však stačí založit cca 5 adres. V rámci zakoupení hostingu majitel aplikace obdrží přístupové údaje do administračního systému za účelem kompletní správy všech e-mailových schránek.

### **6.2.2 Serverhosting**

Pokud majiteli webové aplikace nestačí nebo nevyhovují webhostingové služby, popřípadě chce provozovat aplikace náročnějšího charakteru, může si u poskytovatele pronajmout celý www server, na kterém bude svůj projekt prezentovat. Tomuto druhu hostingu se říká dedikovaný serverhosting (dále jen serverhosting).

Zákazník má zpravidla plný přístup ke "svému" serveru a s nikým se nemusí dělit o systémové prostředky (např. procesor serveru). Přístup ke stránkám je možný přes administrativní systém (obdobně jako u webhostingu), který je k dispozici 24 hodin denně. Provozovatel serverhostingu často nabízí klientovi plnou správu serveru, zálohování a další doprovodné služby.

Serverhosting představuje výhodnou variantu pro provozování náročných www aplikací, kde odpadá nutnost investovat do vlastního hardwaru a připojení k internetu.

Serverhosting nabízí také připojení vlastního serveru (PC) dodaného zákazníkem v serverovně k páteřní síti internetu. Zákazník poté platí pouze za připojení resp. za přenesená data. Tato služba se zpravidla nazývá *privateserver*.

## 6.3 Správa

Dokončením vývojových prací na webu a jejich následným spuštěním na „ostrý“ web naše práce zdaleka nekončí. V této etapě životního cyklu projektu musí být zajištěn bezproblémový chod aplikace včetně její permanentní údržby a rozvoje. Stagnace je jedna z nejhorších věcí, která může náš projekt potkat. Návštěvníci jsou velice citliví na absenci jakýchkoliv inovací a aktualizací, na kterou reagují zavržením celého projektu.

Správu webové aplikace může vykonávat sám vlastník, zpravidla se však o tuto činnost dělí s externí specializovanou firmou, například s poskytovatelem hostingu (viz kapitola 6.2).

Správou webu se rozumí:

- pravidelné a občasné obsahové aktualizace webu
- návrhy na rozvoj a následná realizace webové prezentace
- uživatelská podpora (technické i netechnické záležitosti)
- podpora pro reklamních kampaních (např. tvorba bannerů, zajištění statistik apod.)
- monitoring umístění webové prezentace ve vyhledávacích
- pravidelné zálohování celé databáze a programových skriptů, archivace
- komunikace s případnými dalšími projektovými partnery

## 7 Podpora a propagace projektu

Poslední fází vývoje webového projektu je jeho podpora a propagace. Webové aplikace, které jsou na internetu prezentovány čerstvě, určitě nebudou mít příliš vysokou návštěvnost. Je to zcela přirozené, jelikož potenciální uživatelé popř. zákazníci ještě nevědí, že aplikace existuje. Abychom se dostali do podvědomí internetových uživatelů, musíme sestavit kvalitní reklamní kampaň.

### 7.1 Registrace v katalozích a vyhledávačích

Mezi první kroky vedoucí ke zviditelnění nového webového projektu patří bezpochyby registrace v elektronických katalozích (viz obrázek č. 67). Tato služba je zpravidla zcela zdarma (základní zápis) a zajišťuje stálý přísun uživatelů. Nejdůležitějším krokem registrace je správná volba příslušné kategorie, ve které se bude odkaz na náš projekt zobrazovat. Kategorie by měla co nejpřesněji reprezentovat předmět a účel naší aplikace. V rámci naší případové studie bychom tedy hledali kategorii co nejbližší inzerci a zároveň armádní tématice.



Obr. č. 67: Ukázka virtuálního katalogu



Při registraci webových stránek se vyplňují následující údaje:

- **Název stránek:** většinou jej reprezentuje název domény nebo provozovatele. Maximální počet znaků se pohybuje kolem 25. Název nesmí být celý napsán velkými písmeny (pokud se nejedná o placený zápis) a nesmí být použita předpona „nej“.
- **Popisek stránek:** blíže určuje obsah webové prezentace. Popisek musí být tvořen smysluplnými větami. Maximální délka popisku je většinou 250 znaků.
- **Kategorie:** hlavním cílem je umístit odkaz do tematicky správné kategorie a (v případě, že je to možné) co nejhluběji ve stromové struktuře kategorií katalogu.
- **Odkaz na prezentaci:** obsahuje hypertextový odkaz, prostřednictvím něhož dojde k přesměrování na naši prezentaci.
- **Adresa provozovatele:** zobrazuje adresní údaje provozovatele aplikace včetně kontaktu. Možnost vyplnění těchto údajů je již zpoplatněna.
- **Logo:** registrovaný odkaz může obsahovat i logo aplikace nebo společnosti. Umístění loga je zpoplatněno.

Pro lepší ilustraci slouží obrázek č. 68, který zobrazuje formát a obsah vzorku registrovaných webových projektů ve virtuálním katalogu.



#### [Aukro.cz - Vízitka](#)

Zlín, Vavrečkova

Aukční portál nabízí denně 650.000 nových i použitých položek od auto-moto přes počítače, mobilní telefony, kosmetiku či dětské oblečení. Prodejci nabízejí pohodlné nástroje pro vystavení zboží v aukcích.

Zařazeno v: Bazary a zastavárny, Nakupování na internetu a on-line služby, Inzertní servery

[im.aukro.cz](http://im.aukro.cz)



#### [HYPERINZERCE - Vízitka](#)

Praha 2, Záhřebská

Provozujeme inzertní portál a on-line bazar. Inzerce je soukromá i firemní s fotografiemi. Inzerujeme auto-moto, práci, bydlení, nemovitosti, počítače, mobilní telefony, domácí spotřebiče, nábytek, elektro, seznámení, oblečení, zvířata a fotoaparáty.

Zařazeno v: Bazary s elektro zbožím, Bazary a zastavárny, Bazary s nábytkem

[www.hyperinzerce.cz](http://www.hyperinzerce.cz)



#### [GIGACOMPUTER, s.r.o. - Vízitka](#)

Brno-Zábřehovice, Cejl

Profesionál na trhu repasované výpočetní techniky, notebooky, počítače, monitory za bazarové ceny.

Krátkodobě použitá výpočetní technika světových značek převážně HP, IBM, Dell, Toshiba, Acer. Vše skladem k okamžitému odběru i pro obchodníky.

Zařazeno v: Bazary s elektro zbožím, Bazary s výpočetní technikou, Prodej výpočetní techniky

[www.gigacomputer.cz](http://www.gigacomputer.cz)

Obr. č. 68: Ukázka prezentace registrovaných zápisů do virtuálního katalogu

Počtu registrací projektu do jednotlivých katalogů se meze nekladou, je však pravdou, že registrace do méně známých nebo nekvalitních katalogů nám moc užítka nepřinese.

Svoji prezentaci bychom měli určitě zaregistrovat na následujících serverech:

- **Seznam.cz** ([www.seznam.cz](http://www.seznam.cz))
- **Centrum.cz** ([www.centrum.cz](http://www.centrum.cz))
- **Atlas.cz** ([www.atlas.cz](http://www.atlas.cz))
- **Shopy.cz** ([www.shopy.cz](http://www.shopy.cz))
- **TipNaNakup** ([www.tipnanakup.cz](http://www.tipnanakup.cz))
- **InternetObchody** ([www.internet-obchody.cz](http://www.internet-obchody.cz))
- **České shopy** ([www.ceskeshopy.cz](http://www.ceskeshopy.cz))

- **I-obchody** ([www.i-obchody.com](http://www.i-obchody.com))
- **Adresář firem** ([www.adresarfirem.cz](http://www.adresarfirem.cz))
- **Onlineobchody** ([www.onlineobchody.cz](http://www.onlineobchody.cz))

Aplikaci můžeme kromě katalogů registrovat také do fulltextových vyhledávacích serverů. V tomto případě stačí jen zadat URL<sup>15</sup> aplikace do příslušného formuláře a vyhledávací robot si již veškeré informace stáhne přímo ze zdrojového kódu.

Nejdůležitější vyhledávací servery jsou:

- **Google** ([www.google.com](http://www.google.com))
- **Jyxo** (<http://jyxo.cz/>)
- **Zoohoo** ([www.zoohoo.cz](http://www.zoohoo.cz))
- **Seznam-fulltext** (<http://fulltext.seznam.cz/url.py/addScreen>)

Seznam několika dalších významných českých katalogů a vyhledávacích serverů je uveden v příloze č. 4.

## 7.2 Statistické služby

Statistické služby lze popsat jako velké servery, které podle návštěvnosti jednotlivých zaregistrovaných serverů tvoří žebříček návštěvnosti těchto serverů. Velký počet lidí tyto servery používá jako náhradu za servery vyhledávací, proto je žádoucí mít své www stránky zařazené i zde. Přidanou hodnotu v podobě monitoringu naší návštěvnosti lze využít například pro porovnání s ostatními zaregistrovanými servery.

---

<sup>15</sup> URL: jednoznačné určení zdroje. Způsob, jak jednoznačně zapsat umístění souboru na internetu/intranetu

Princip statistických služeb spočívá v tom, že po registraci obdržíme zdrojový kód, který umístíme na naše stránky. Daný kód zajišťuje sledování počtu příchozích návštěvníků, který se (spolu s dalšími statistikami) posílá statistickému serveru k vyhodnocení.

Na českém internetu existují dvě významné veřejné statistické služby:

- **Navrcholu** ([www.navrcholu.cz](http://www.navrcholu.cz))
- **TopList** ([www.toplist.cz](http://www.toplist.cz))

### 7.3 Bannery

Banner (neboli reklamní proužek) je statický, dynamický nebo interaktivní obrázek o určitých rozměrech, který se umísťuje na partnerské servery s cílem získání nových návštěvníků.

Při umísťování reklamního proužku bychom měli věnovat velkou pozornost výběru vhodných internetových portálů či serverů. Pokud chceme, aby byla reklama co nejúčinnější, měli bychom zjistit, kým jsou cílové stránky nejvíce navštěvovány a tento typ návštěvníků následně porovnat s naší cílovou skupinou. Čím je daný typ návštěvníků bližší naší cílové skupině, tím lépe.

Reklamní proužek můžeme umístit také na tematicky neutrální internetové portály nebo elektronická periodika. Cena za umístění je však neúměrně drahá a výsledný efekt není nijak valný.

Platba za umístění banneru může nabýt následujících variant:

- **Časově placená prezentace:** cena za pronájem reklamního prostoru a umístění banneru se vztahuje k časovému úseku, po který bude banner vystaven.
- **Platba za počet impresí:** cena za pronájem reklamního prostoru a umístění banneru je dána počtem zobrazení (impresí) reklamy na daném serveru.

- **Platba za příchozí návštěvníky:** cena za umístění banneru je dána počtem příchozích návštěvníků, který na daný banner klikli.
- **Vzájemná výměna bannerů:** v rámci této formy bannerové kampaně je reklamní proužek odkazující na naše webové stránky umístěn na určitém serveru (předem vybraném). Na druhou stranu je rozměrově podobný proužek, který odkazuje na stránky partnerského serveru, umístěn v naší prezentaci. Pro tuto formu kampaně existují specializované systémy (viz kapitola 7.4).

## 7.4 Reklamní síť

Reklamní sítě fungují na základě vložení speciálního HTML kódu (obdržíme jej po registraci) na své stránky, který zajistí, že se na našem serveru budou objevovat reklamní bannery jiných serverů registrovaných v této reklamní síti. Náš banner se zas bude objevovat na ostatních zaregistrovaných serverech. Součástí reklamních sítí jsou i přehledné statistiky, monitorující servery, kde a s jakou frekvencí se náš banner zobrazoval.

Na českém internetu patří mezi neznámější systém reklamní sítě **Billboard.cz** ([www.billboard.cz](http://www.billboard.cz))

## 7.5 Affiliate programy

Affiliate programy jsou většinou programy, v rámci kterých na předem vybraný web necháme umístit nějaký formulář či odkaz. Návštěvník těchto stránek poté na následující odkaz klikne a dostane se na naši stránku, jejímž cílem je zpravidla si něco objednat či koupit. Pokud se tak stane, obdrží partnerský server poměrnou či fixní částku peněz.

V rámci těchto programů platíme tedy pouze za reálné výsledky v podobě prokliků na naše stránky, vyplnění formuláře, uskutečněný prodej apod.

Mezi nejznámější affiliate programy patří:

- **Potenza** ([www.potenza.cz](http://www.potenza.cz))
- **Etarget** ([www.etaarget.cz](http://www.etaarget.cz))
- **Google AdWords** (<https://adwords.google.com/select/Login>)

## 8 Závěr

Při tvorbě internetových aplikací a informačních systémů je více než žádoucí si vše předem promyslet, specifikovat a připravit. Platí zde totiž pravidlo „dvakrát měř a jednou řež“. Samozřejmě není potřeba tvořit komplexní specifikaci požadavků a analýzu včetně různých modelů při tvorbě jednoduchých aplikací, jakou může být například softwarová evidence domácí sbírky hudebních titulů. Nicméně u větších a velkých komerčních internetových projektů nám projektování pomáhá určit přesné funkce a cíle budoucí aplikace.

Projektování systémů napomáhá k lepší komunikaci mezi zadavatelem/budoucím uživatelem a zhotovitelem. Tak, jako se již od pradávna stavějí budovy a stavby na základě podrobných plánů, jež zhotovily profesionální architekti, by měly být na stejném principu zhotovovány i elektronické aplikace. Při tvorbě webových projektů se velmi často využívá jazyka UML, který pomocí grafických modelů přesně popisuje jednotlivé objekty a jejich relace v systému.

V neposlední řadě nám příprava a projektování budoucího systému může ušetřit mnoho peněz, které bychom vydali na případné změny požadované již ve fázi vlastní implementace.

Velice důležitou roli hraje kvalitní projektová dokumentace, která je zapotřebí při různých změnách nebo inovacích stávající aplikace. Jen těžko se navazuje na systém, který není řádně popsán.

Profesionální hosting, správa a propagace hotového internetového projektu je stejně důležitá jako jeho příprava a projektování. Můžeme vytvořit sebelepší internetovou aplikaci, nicméně bez jejích uživatelů je aplikace k ničemu. Přísunu nových uživatelů dosáhneme především bezchybností chodu aplikace, její neustálou aktuálností a dobrou propagací.

Jednotlivé postupy při projektování, správě a propagaci webových aplikací, které byly předmětem této práce, mohou sloužit jako inspirace pro zadavatele i zhotovitele budoucích profesionálních internetových projektů.

## Seznam použité literatury a zdrojů

1. POLÁK, Jiří, MERUNKA, Vojtěch a CARDA, Antonín. *Umění systémového návrhu : Objektivě orientovaná tvorba informačních systémů pomocí původní metody BORM*. Praha : Grada, 2003. 195 s. ISBN 80-247-0424-2.
2. KUČEROVÁ, Helena. *Projektování informačních systémů II* [online]. [cit. 2007-2-16]. Dostupný z WWW: <[http://info.sks.cz/users/ku/PRI/pri\\_i\\_2.htm](http://info.sks.cz/users/ku/PRI/pri_i_2.htm)>.
3. KUČEROVÁ, Helena. *Projektování informačních systémů*. Sylaby ke kurzu. Praha : Vyšší odborná škola informačních služeb, 2005. 110 s.
4. RICHTA, Karel. *Unifikovaný modelovací jazyk UML* [online]. [cit. 2007-02-20]. [8 s.]. Dostupný z WWW: <[http://si.vse.cz/archiv/clanky/2003/08\\_richta.pdf](http://si.vse.cz/archiv/clanky/2003/08_richta.pdf)>.
5. KOPŘIVA, Petr. *Vyberte správný CASE - Stopařův průvodce CASE nástroji* [online]. [cit. 2007-02-21]. Dostupný z WWW: <[http://www.komix.cz/Tisk/Clanky/Historie/Spravny\\_CASE.aspx](http://www.komix.cz/Tisk/Clanky/Historie/Spravny_CASE.aspx)>.
6. *Objektová analýza, návrh a programování*. Praha: Vysoká škola ekonomická, Fakulta informatiky a statistiky. Luboš Pavlíček (správce webu). Poslední aktualizace 21. 6. 2005. Dostupný z WWW: <<http://objekty.vse.cz>>.
7. BUCHALCEVOVÁ, Alena a PAVLÍČKOVÁ, Jarmila. *Základy softwarového inženýrství - objektivě orientovaný přístup*. 1. vyd. Praha : Vysoká škola ekonomická, 2002. 216 s. ISBN 80-245-0268-2.
8. ŘEPA, Václav. *Podnikové procesy : Procesní řízení a modelování*. Praha: Grada, 2006. 265 s. ISBN 80-247-1281-4.



9. HANÁČEK, Petr a STAUDEK, Jan. *Bezpečnost informačních systémů : Metodická příručka zabezpečování produktů a systémů budovaných na bázi informačních technologií*. Praha : Úřad pro státní informační systém, 2000. [127 s.]. Dostupný z WWW: <<http://www.micr.cz/files/479/uvis-Bezpecnost-20000701.pdf>>.
10. RYDVAL, Slávek. *Sybase Power Designer 9.0* [online]. [cit. 2007-02-25]. Dostupný z WWW: <<http://www.rydval.cz/slavek/PowerDesigner9.php>>.
11. ŘEPA, Václav a KÖPPEL, Adam. *Programování ve velkém. Srovnání: Nástroje CASE*. [online]. [cit. 2007-02-25]. [str. 18-27]. Dostupný z WWW: <<http://info.sks.cz/users/ku/dokumenty/swn-2003-05.pdf>>.
12. SMIL, Petr. *Úvod do objektového modelování a jazyka UML*. [online]. [cit. 2007-02-27]. Dostupný z WWW: <[http://www.komix.cz/Tisk/Clanky/Historie/Uvod\\_UML.aspx](http://www.komix.cz/Tisk/Clanky/Historie/Uvod_UML.aspx)>.
13. H1.cz s.r.o. *Analýza klíčových slov* [online]. [cit. 2007-07-16]. Dostupný z WWW: <<http://www.h1.cz/analyza-klicovych-slov>>.
14. BENÁK, Karel. *Složené klíče* [online]. [cit. 2007-07-16]. Dostupný z WWW: <[http://www.benak.net/slozene\\_klice](http://www.benak.net/slozene_klice)>.
15. *W3C Markup Validation Service* [online aplikace]. Version 0.7.4. W3C, c 1994-2007 [2007-06-21]. Dostupný z WWW: <<http://validator.w3.org/>>.
16. *W3C CSS Validation Service* [online aplikace]. W3C, c 1994-2003 [2007-06-21]. Dostupný z WWW: <<http://jigsaw.w3.org/css-validator/>>.
17. *Wikipedie* [online]. [cit. 2007-07-17]. Dostupný z WWW: <<http://cs.wikipedia.org>>.
18. LinuxEXPRES. *Srovnání databázových serverů* [online]. [cit. 2007-07-17]. Dostupný z WWW: <<http://www.linuxexpres.cz/software/srovnani-databazovych-serveru>>.
19. PAZDZIORA, Jan. *MySQL databázový server* [online]. [cit. 2007-07-17]. Dostupný z WWW: <<http://www.linux.cz/noviny/1999-11/clanek06.html>>.

20. Czech ICT Alliance. *Počet uživatelů internetu se zvýšil* [online]. [cit. 2007-07-21].  
Dostupný z WWW: <<http://www.czechict.cz/?funct=513&id=175>>.
21. ZELENKA, Petr. *WebML - projektování webových aplikací* [online]. [cit. 2007-07-21].  
Dostupný z WWW: <<http://interval.cz/clanky/webml-projektovani-webovych-aplikaci/>>.
22. Tvorba-webu.cz. *PHP /základy/* [online]. [cit. 2007-06-13]. Dostupný z WWW:  
<<http://www.tvorba-webu.cz/php/>>.
23. MALO, Roman. *Webové projekty* [online]. [cit. 2007-07-22]. Dostupný z WWW:  
<<http://martin.feld.cvut.cz/~molhanec/VaV/files/publik/2003/WebML.pps#2>>.
24. TIŠNOVSKÝ, Pavel. *Nástroje pro tvorbu UML diagramů* [online]. [cit. 2007-07-22].  
Dostupný z WWW: <<http://www.root.cz/clanky/nastroje-pro-tvorbu-uml-diagramu/>>.
25. Interval.cz. *Návrh aplikací v jazyce UML* [online]. [cit. 2007-07-23]. Dostupný z WWW:  
<<http://interval.cz/serialy/navrh-aplikaci-v-jazyce-uml/>>.
26. Owebu.cz. *SEO - metoda optimalizace webu* [online]. [cit. 2007-06-19]. Dostupný  
z WWW: <<http://www.owebu.cz/serialy.php?serial=38>>.
27. 27 P.E.S. consulting, s.r.o. *Doménové centrum* [online]. [cit. 2007-06-15]. Dostupný  
z WWW: <<https://www.forpsi.com/domain/>>.
28. TOPlist s.r.o. *TOPlist* [online aplikace]. [cit. 2007-06-15]. Dostupný z WWW:  
<<http://www.toplist.cz>>.
29. Internet Info, s.r.o. *NAVRCHOLU.cz* [online aplikace]. [cit. 2007-06-15]. Dostupný  
z WWW:  
<<http://navrcholu.cz>>.
30. Seznam.cz, a.s. *Seznam : Firmy* [online katalog]. 1996 – 2006 [cit. 2007-06-18]. Dostupný  
z WWW: <<http://firmy.seznam.cz/>>.

31. Asociace.biz. *Asociace.biz* [online]. [cit. 2007-07-25]. Dostupný z WWW: <http://asociace.biz>.
32. PETROUTSOS, Evangelos. *Myslíme v jazyku Visual Basic .NET, 1. díl*. Praha : Grada, 2003. 676 s. ISBN 80-247-0371-8.
33. PETROUTSOS, Evangelos. *Myslíme v jazyku Visual Basic .NET, 2. díl*. Praha : Grada, 2003. 540 s. ISBN 80-247-0372-6.
34. KOSEK, Jiří. *PHP, tvorba interaktivních internetových aplikací*. Praha : Grada, 1999. 492 s. ISBN 80-7169-373-1.
35. DELISLE, Marc. *phpMyAdmin - efektivní správa MySQL*. Praha : Zoner Press, 2004. 264 s. ISBN 80-86815-09-9.
36. SPELL, Brett. *Java, programujeme profesionálně*. Praha : Computer Press, 2002. 1040 s. ISBN 80-7226-667-5.
37. MASLAKOWSKI, Mark. *Naučte se MySQL za 21 dní*. Praha : Computer Press, 2001. 506 s. ISBN 80-7226-448-6.
38. KRUG, Steve. *Web design: Nenuťte uživatele přemýšlet!* Praha : Computer Press, 2003. 152 s. ISBN 80-7226-892-9.
39. TONDR, Luděk. *Podnikáme s Internetem*. Praha : Computer Press, 2002. 118 s. ISBN 80-7226-729-9.
40. PROKOP, Marek. *Analýza návštěvnosti vašeho webu* [online]. [cit. 2007-07-19]. Dostupný z WWW: <http://www.zive.cz/default.aspx?article=30388>.
41. ŠTRUPL, Václav. *Testování webových stránek* [online]. [cit. 2007-07-19]. Dostupný z WWW: <http://interval.cz/clanky/testovani-webovych-stranek/>.

42. KORBEL, Jan. *Návrh databáze* [online]. [cit. 2007-07-20]. Dostupný z WWW: <<http://jankorbel.cz/2007/04/25/web-aplikace-tisk-faktury-databaze>>.
43. CVSGui [online aplikace]. [cit. 2007-07-24]. Dostupný z WWW: <<http://www.wincvs.org/>>.
44. WebRatio [online aplikace]. [cit. 2007-07-24]. Dostupný z WWW: <<http://www.webratio.com/Home.do?link=oln489d.redirect>>.
45. Zákon č. 101/2000 Sb., o ochraně osobních údajů. *Sbírka zákonů České republiky*. 2000, částka 32, s. 1521 Dostupný z WWW: <<http://www.mvcr.cz/sbirka/2000/sb032-00.pdf>>.
46. Sybase Inc. *PowerDesigner* [online]. [cit. 2007-06-28]. Dostupný z WWW: <<http://www.sybase.com/products/modelingmetadata/powerdesigner>>.
47. Zákon č. 365 ze dne 14. září 2000 o informačních systémech veřejné správy a o změně některých zákonů. *Sbírka zákonů České republiky*. 2000, částka 99, s. 4666. Dostupný z WWW: <<http://www.mvcr.cz/sbirka/2000/sb099-00.pdf>>.
48. W3C. *Web Content Accessibility Guidelines 1.0* [online]. [cit. 2007-07-25]. Dostupný z WWW: <<http://www.w3.org/TR/WAI-WEBCONTENT/>>.
49. Section508. *508 Law* [online]. [cit. 2007-07-25]. Dostupný z WWW: <<http://www.section508.gov/index.cfm?FuseAction=Content&ID=3>>.
50. SONS ČR. *Blind friendly web* [online]. [cit. 2007-07-25]. Dostupný z WWW: <<http://www.blindfriendly.cz/>>.
51. 50 P.E.S. consulting, s.r.o. *Přehled nabízených domén* [online]. [cit. 2007-06-15]. Dostupný z WWW: <<https://www.forpsi.com/domain/offer.php?sess=bb7e3a0ab56f73728ebe59d7fbe8686f>>.
52. NIC.cz. *Nic.cz* [online]. [cit. 2007-07-30]. Dostupný z WWW: <<http://www.nic.cz/cz/>>.

53. Wikipedie. *Internetová doména* [online]. [cit. 2007-07-28]. Dostupný z WWW: [http://cs.wikipedia.org/wiki/Internetov%C3%A1\\_dom%C3%A9na](http://cs.wikipedia.org/wiki/Internetov%C3%A1_dom%C3%A9na).
54. Microsoft Corporation. *Microsoft SQL server* [online]. [cit. 2007-07-29]. Dostupný z WWW: <http://www.microsoft.com/cze/windowsserversystem/sql/default.msp>.
55. Oracle Czech s.r.o. *Databáze ORACLE* [online]. [cit. 2007-07-29]. Dostupný z WWW: <http://www.oracle.com/global/cz/database/index.html>.
56. PostgreSQL Global Development Group. *PostgreSQL* [online]. [cit. 2007-07-29]. Dostupný z WWW: <http://www.postgresql.org/>.
57. MySQL AB. *MySQL* [online]. [cit. 2007-07-29]. Dostupný z WWW: <http://www.mysql.com/>.
58. Microsoft Corporation. *.NET Framework Developer Center* [online]. [cit. 2007-07-29]. Dostupný z WWW: <http://msdn2.microsoft.com/en-us/netframework/default.aspx>.
59. Adobe Corporation. *Macromedia HomeSite 5.5* [online]. [cit. 2007-07-30]. Dostupný z WWW: <http://www.adobe.com/products/homesite/>.
60. FIALA, Jan. *PSPad* [online]. [cit. 2007-07-30]. Dostupný z WWW: <http://www.pspad.com/cz/>.
61. POLÁK, Pavel. *Golden HTML editor* [online]. [cit. 2007-07-30]. Dostupný z WWW: <http://www.golden-html.com/>.
62. The PHP Group. *PHP* [online]. [cit. 2007-07-30]. Dostupný z WWW: <http://cz.php.net/>.
63. Sun Microsystems, Inc. *JavaServer Pages Technology* [online]. [cit. 2007-07-30]. Dostupný z WWW: <http://java.sun.com/products/jsp/>.

## **Seznam příloh:**

1. Definice smlouvy o dílo
2. Struktura smlouvy o dílo
3. Přehled předních českých poskytovatelů webhostingu
4. Přehled předních českých vyhledávačů a elektronických katalogů
5. Evidence výpůjček

## **Přehled předních českých poskytovatelů webhostingu**

<a href="http://www.1990.cz">www.1990.cz</a>	<a href="http://www.GZK.cz">www.GZK.cz</a>	<a href="http://www.Pueblo.cz">www.Pueblo.cz</a>
<a href="http://www.1Webhosting.cz">www.1Webhosting.cz</a>	<a href="http://www.Hosting.cz">www.Hosting.cz</a>	<a href="http://www.SAVVY.cz">www.SAVVY.cz</a>
<a href="http://www.Active24.cz">www.Active24.cz</a>	<a href="http://www.Hosting4U.cz">www.Hosting4U.cz</a>	<a href="http://www.SPIDER.CZ">www.SPIDER.CZ</a>
<a href="http://www.Adaptic.cz">www.Adaptic.cz</a>	<a href="http://www.IGNUM.cz">www.IGNUM.cz</a>	<a href="http://www.Stable.cz">www.Stable.cz</a>
<a href="http://www.Aurem.cz">www.Aurem.cz</a>	<a href="http://www.InterHosting.cz">www.InterHosting.cz</a>	<a href="http://www.SuperHosting.cz">www.SuperHosting.cz</a>
<a href="http://www.Banan.cz">www.Banan.cz</a>	<a href="http://www.INTERWEB.cz">www.INTERWEB.cz</a>	<a href="http://www.SvetHostingu.cz">www.SvetHostingu.cz</a>
<a href="http://www.Bestwebhosting.cz">www.Bestwebhosting.cz</a>	<a href="http://www.iSOL.cz">www.iSOL.cz</a>	<a href="http://www.tojeono.cz">www.tojeono.cz</a>
<a href="http://www.Bravenet.cz">www.Bravenet.cz</a>	<a href="http://www.Klenot.cz">www.Klenot.cz</a>	<a href="http://www.Webhosting51.cz">www.Webhosting51.cz</a>
<a href="http://www.CEBIN.cz">www.CEBIN.cz</a>	<a href="http://www.Keep.cz">www.Keep.cz</a>	<a href="http://www.WebhostingC4.cz">www.WebhostingC4.cz</a>
<a href="http://www.Cool-Hosting.cz">www.Cool-Hosting.cz</a>	<a href="http://www.levne-domeny.cz">www.levne-domeny.cz</a>	<a href="http://www.Narovinu.cz">www.Narovinu.cz</a>
<a href="http://www.CyberHost.cz">www.CyberHost.cz</a>	<a href="http://www.Levnyhosting.com">www.Levnyhosting.com</a>	<a href="http://www.WEBY.CZ">www.WEBY.CZ</a>
<a href="http://www.CzechHosting.cz">www.CzechHosting.cz</a>	<a href="http://www.m-hosting.cz">www.m-hosting.cz</a>	<a href="http://www.xHOSTING.CZ">www.xHOSTING.CZ</a>
<a href="http://www.CZECHIA.cz">www.CZECHIA.cz</a>	<a href="http://www.Mediahost.cz">www.Mediahost.cz</a>	<a href="http://www.Xline.cz">www.Xline.cz</a>
<a href="http://www.ceskeweby.cz">www.ceskeweby.cz</a>	<a href="http://www.MediaWeb.cz">www.MediaWeb.cz</a>	<a href="http://www.Xsite.cz">www.Xsite.cz</a>
<a href="http://www.cesky-hosting.cz">www.cesky-hosting.cz</a>	<a href="http://www.Moonlake.cz">www.Moonlake.cz</a>	<a href="http://www.webhosting-c4.cz">www.webhosting-c4.cz</a>
<a href="http://www.decent.cz">www.decent.cz</a>	<a href="http://www.move.cz">www.move.cz</a>	
<a href="http://www.Eurohosting.cz">www.Eurohosting.cz</a>	<a href="http://www.mujhost.cz">www.mujhost.cz</a>	
<a href="http://www.Eurohosting.name">www.Eurohosting.name</a>	<a href="http://www.mujhost.net">www.mujhost.net</a>	
<a href="http://www.Expert.cz">www.Expert.cz</a>	<a href="http://www.Novyhosting.cz">www.Novyhosting.cz</a>	
<a href="http://www.FORPSI.cz">www.FORPSI.cz</a>	<a href="http://www.Onebit.cz">www.Onebit.cz</a>	
<a href="http://www.G-hosting.cz">www.G-hosting.cz</a>	<a href="http://www.OnlineHosting.cz">www.OnlineHosting.cz</a>	
<a href="http://www.Gatompark.cz">www.Gatompark.cz</a>	<a href="http://www.PIPNI.cz">www.PIPNI.cz</a>	
<a href="http://www.Gigaserver.cz">www.Gigaserver.cz</a>	<a href="http://www.ProFiTux.cz">www.ProFiTux.cz</a>	
<a href="http://www.Gigaweb.cz">www.Gigaweb.cz</a>	<a href="http://www.ProfiWH.cz">www.ProfiWH.cz</a>	

## Přehled předních českých vyhledávačů a el. katalogů

katalog.pcsvet.cz	www.1000webu.com	www.detektiv.cz
www.yo.cz	www.1x1.info	infotip.cz
katalog.wz.cz	www.aaainternet.cz	www.1.cz
www.enachod.cz	www.cent.cz	www.zdroj.cz
www.klikni.cz	www.jednorozec.cz	www.inform.cz
www.seeker.cz	www.napoveda.cz	www.czprima.cz
www.morfeo.cz	www.opendir.cz	www.kudy.cz
www.saspark.eu	www.quick.cz	www.hledejte.cz
references.xf.cz	www.startpage.cz	www.infoweb.cz
www.odkazy-linky.cz	www.tiscali.cz	www.morfeo.cz
www.divoch.net	www.vokno.cz	www.redbox.cz
www.allytrade.cz	www.zacatek.cz	www.globalsearch.cz
reklama.euweb.cz	www.kdenajdu.cz	www.najduvse.cz
toplink.miliweb.net	www.lister.cz	www.chces.info
katalog-stranek.sukvos.com	www.ukazto.eu	katalog.over.cz
www.zona.cz	www.krizovatka.cz	www.citysearch.cz
<a href="http://www.hradubicko.cz">www.hradubicko.cz</a>	<a href="http://klikni.idnes.cz">klikni.idnes.cz</a>	www.netcz.net
<a href="http://www.jahho.net">www.jahho.net</a>	<a href="http://verejne.nazory.cz">verejne.nazory.cz</a>	www.bezvaportal.cz
www.katedrala.cz	www.infoweb.cz	www.sestava.cz
www.dasi.cz	www.katalogfirem.com	www.odkazy-linky.cz
www.oblibena.cz	hledej-firmu.unas.cz	www.atlasfirem.info
www.webpark.cz	hledej.priluky.cz	hledani.tiscali.cz
najduto.zde.cz	snajpr.cz	alfa.elchron.cz
seznam.poutnici.com		



## **Smlouva o dílo - náležitosti**

Smlouvou o dílo se zavazuje zhotovitel k provedení určitého díla a objednatel se zavazuje k zaplacení ceny za jeho provedení.

Dílem se rozumí zhotovení určité věci, pokud nespadá pod kupní smlouvu, montáž určité věci, její údržba, provedení dohodnuté opravy nebo úpravy určité věci nebo hmotně zachycený výsledek jiné činnosti. Dílem se rozumí vždy zhotovení, montáž, údržba, oprava nebo úprava stavby nebo její části.

Cena musí být ve smlouvě dohodnuta nebo v ní musí být alespoň stanoven způsob jejího určení, ledaže strany ve smlouvě projeví vůli uzavřít smlouvu bez tohoto určení.

Z uvedené definice vyplývají základní povinnosti nezbytné pro uzavření smlouvy o dílo, které lze charakterizovat jako podstatné náležitosti smlouvy. Jde o určení účastníků smlouvy (smluvních stran) a následně o stanovení závazku zhotovitele dílo provést a závazku objednatele dílo zaplatit. Tyto podstatné náležitosti lze zkráceně vymežit jako:

- kdo komu ( určení stran )
- co ( určení předmětu )
- za kolik ( určení ceny nebo způsobu jejího stanovení )

Ostatní náležitosti smlouvy jsou sice nezbytné pro právní jistotu obou stran, ale záleží vždy na vůli každé z nich, v jakém rozsahu a jak budou ve smlouvě definovány. Jedná se zejména o tyto náležitosti:

- kdy ( doba provedení díla )
- kde ( místo provedení díla )
- jak ( podmínky provádění díla )
- způsob kontroly prováděného díla
- podmínky předání a převzetí díla
- rozsah a délka záruky
- způsob zaplacení ceny
- nároky z odpovědností

Forma smlouvy o dílo není obchodním zákoníkem upravena. Není tedy předepsána písemná forma smlouvy a smlouva může vzniknout i ústně. Vzhledem k povaze díla a v důsledku právní jistoty obou smluvních stran je však vhodnější uzavírat smlouvu vždy písemnou formou a stejně tak postupovat i v případě změn smlouvy. Ujednání o smluvní pokutě či ujednání o záruce písemnou formu mít musí.

## **Smlouva o dílo - struktura**

Struktura smlouvy má výhradně doporučující charakter a vychází z běžných zvyklostí současných smluv.

- 1) Označení smluvních stran objednatel
  - zhotovitel
  - oprávnění zástupci ve věcech smluvních
  - oprávnění zástupci ve věcech technických
  - vymezení pravomocí
- 2) Vymezení předmětu smlouvy o dílo
  - definice předmětu díla a technický popis
  - podmínky změn rozsahu předmětu díla
- 3) Sjednání ceny za dílo
  - výše ceny
  - podmínky změny ceny
- 4) Vymezení času plnění
  - termín zahájení
  - termín dokončení
- 5) Sjednání podmínek úhrady ceny za dílo - platební podmínky
  - způsob a forma úhrady
  - splatnost daňových dokladů
- 6) Záruka za jakost a záruční podmínky
  - délka poskytnuté záruční lhůty
  - způsob a obsah reklamace
  - termíny a povinnosti stran při odstranění reklamovaných vad
- 7) Podmínky provedení díla
- 8) Podmínky předání a převzetí díla
- 9) Smluvní pokuty
- 10) Vyšší moc
- 11) Ostatní ustanovení
- 12) Závěrečná ustanovení

## Evidence výpůjček

Prohlášení:

Dávám svolení k půjčování této diplomové práce. Uživatel potvrzuje svým podpisem, že bude tuto práci řádně citovat v seznamu použité literatury.

V Praze, 8.8. 2007.

Michal Pivoňka

Jméno	Katedra / Pracoviště	Datum	Podpis