

UNIVERZITA PARDUBICE

FAKULTA EKONOMICKO-SPRÁVNÍ

Příprava umělé sociální sítě pro předmět Sociální sítě

Jana Kremlová

Bakalářská práce

2011

Univerzita Pardubice
Fakulta ekonomicko-správní
Akademický rok: 2010/2011

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jana KREMLOVÁ**
Osobní číslo: **E08257**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Příprava umělé sociální sítě pro předmět Sociální sítě**
Zadávací katedra: **Ústav systémového inženýrství a informatiky**

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

1. Zpracování databáze a GUI pro sociální síť
2. Tvorba a úprava fotografií
3. Implementace a otestování navržené aplikace

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací: podrobný průvodce.* Praha: Grada, 1998. 492 s. ISBN 80-7169-373-1.

HLAVENKA, Jiří, et al. *Vytváříme WWW stránky: a spravujeme moderní web site.* Praha: Computer Press, 1997. 360 s. ISBN 80-251-0801-5.

NEFF, Ondřej, PODHAJSKÝ, Petr, BŘEZINA, Jan. *Fotografování s digitálním fotoaparátem,* Praha: Institut digitální fotografie, 2003. 104 s.

Vedoucí bakalářské práce:


Ing. Jan Panuš, Ph.D.

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **4. října 2010**

Termín odevzdání bakalářské práce: **6. května 2011**



doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.



doc. Ing. Jiří Křupka, Ph.D.

vedoucí ústavu

V Pardubicích dne 4. října 2010

Prohlášení

Tuto práci jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

Ve Slatiňanech dne 21. dubna 2011

Jana Kremlová

Poděkování

Tímto bych chtěla velmi poděkovat Ing. Janu Panušovi, Ph.D. za odbornou pomoc při tvorbě této práce

Anotace

Tato bakalářská práce se zabývá návrhem a realizací přípravy umělé sociální sítě pro předmět Sociální sítě, který se vyučuje na Fakultě ekonomicko-správní Univerzity Pardubice. Obsahem je vysvětlení pojmu sociální síť, hlavní metody jazyků PHP, JavaScript, HTML a SQL, návrh datového modelu a popis vytvořené aplikace.

Klíčová slova

sociální síť, programování, databáze, PHP, HTML, JavaScript, SQL

Title

Preparation of artificial social network for subject Social networks

Annotation

The bachelor thesis deals with the preparation and implementation of an artificial social network for the subject Social Networks, which is taught at the Faculty of Economics and Administration, University of Pardubice. It contains the explanation of the term social network, the main method of languages PHP, JavaScript, HTML and SQL, design of data model and description of the created application.

Keywords

social network, programming, database, PHP, HTML, JavaScript, SQL

Obsah

Úvod	8
1. Pojem sociální síť	9
2. HTML a JavaScript	10
2.1. Formulář v HTML	10
2.1.1. Metoda GET a POST	10
2.1.2. Tag input, textarea	11
2.2. Obrázek v HTML	11
2.3. JavaScript	12
2.3.1. Funkce	12
2.3.2. Větvení	13
2.3.3. Dialogová okna	13
2.3.4. Formuláře	14
3. Příkazy jazyka SQL	15
3.1. Vytvoření tabulky	15
3.2. Přidávání záznamu do tabulky	16
3.3. Výběr a prohlížení záznamů tabulky	16
3.4. Mazání záznamů	16
3.5. Změna záznamu v tabulce	17
4. Vybrané vlastnosti a metody systému PHP	18
4.1. Superglobální proměnné	18
4.2. Logické výrazy a operátory	20
4.3. Větvení, cykly v PHP	20
4.4. PHP a MySQL	22
5. Realizace praktické části	23
5.1. Definice požadavků na aplikaci	23

5.2. Návrh databáze	23
5.3.1. Konceptuální úroveň.....	23
5.3.2. Technologická úroveň	25
5.3.3. Implementační úroveň	27
5.3. Use case diagram	32
5.4. Grafické rozložení aplikace	34
5.5. Registrace uživatele	34
5.6. Přihlášení a odhlášení uživatele	37
5.7. Přidání kamaráda	38
5.8. Úprava osobních údajů	40
5.9. Vložení příspěvku, prohlédnutí a ohodnocení fotografie	42
5.10. Odzkoušení aplikace v různých internetových prohlížečích	43
Závěr.....	44
Seznam obrázků.....	45
Seznam tabulek.....	46
Použitá literatura.....	47

Úvod

Pojem sociální síť existoval ještě před vznikem internetu, ale s jeho příchodem dostal tento pojem o mnoho jiný smysl. Vzniklo mnoho internetových stránek sdružujících lidi se společným zájmem, které jsou známé jako sociální sítě. Počet jejich uživatelů roste velmi rychle a v dnešním světě si mnoho lidí neumí život bez sociálních sítí ani představit. S ohledem na tento fenomén dnešní doby se Fakulta ekonomicko-správní rozhodla do výuky zavést nový předmět „Sociální sítě“, ve kterém se studenti seznámí s pojmem sociální síť, možnostmi, které sociální sítě nabízejí a s riziky spojenými s užíváním těchto sítí.

Tato práce stručně vysvětlí pojem sociální síť, budou zde popsány hlavní metody programovacích jazyků PHP, HTML a JavaScript, příkazy jazyka SQL, které byly použity při tvorbě umělé sociální sítě. Dále práce obsahuje definici požadavků na aplikaci, návrh databáze na všech úrovních modelování a popis vytvořené aplikace.

Cílem této práce je vytvořit umělou sociální síť, která bude využívána při výuce předmětu „Sociální sítě“.

1. Pojem sociální síť

Pojem sociální síť má mnoho různých významů. Prvně byl tento pojem zaveden již v polovině 20. století, tzn. již před vznikem internetu a internetových sociálních sítí. Pojem původně vychází ze sociologie, sociální síť je určité propojení lidí, kde je člověk centrem sociální sítě a větve této sítě tvoří jeho rodina, kamarádi, známí, spolupracovníci, obchodní partneři apod. Vztahy v síti mohou být jednosměrné nebo obousměrné. [9] V dnešní době je pojem sociální síť chápán v souvislosti s rozšířenými internetovými stránkami sdružujícími uživatele, kteří mají něco společného - stejné zájmy, studovali spolu na škole, jsou fanoušky stejné skupiny, atp. V rámci této práce bude pojem sociální síť chápán jako takovýto web.

První sociální síť *www.classmates.com* vznikla v roce 1995, stvořil ji Randy Conrads a tato sociální síť je funkční dodnes. Jelikož byla tato sociální síť se svojí koncepcí velmi úspěšná, vzniklo následně velké množství podobných sociálních sítí jako je například MySpace, LinkedIn, Facebook, Spolužáci a mnoho dalších. [9]

Chceme-li efektivně využít některou ze sociálních sítí například pro účely podnikání, pak je dobré tuto síť podrobit důkladné analýze. Pro analýzu sociálních sítí existuje celá řada metodologií a metrik, pomocí nichž můžeme dostat odpovědi zejména na tyto důležité otázky [8]:

- „Jak silně je spojen určitý subjekt s ostatními subjekty v rámci sítě?“
- „Jaký je význam daného subjektu v síti?“
- „Jak rychle a kudy tečou informace v síti?“

Analýza sociálních sítí je metoda, která se používá v sociálních vědách a slouží ke znázornění vztahů mezi lidmi v síti. Používá se forma grafu, kde subjekt je jednotlivec nebo jiná entita, tento subjekt má určité vztahy s dalšími subjekty a tyto vztahy se znázorňují pomocí grafu (sítě). Entity jsou reprezentované nejčastěji vrcholy (body, uzly) a vztahy mezi nimi jsou reprezentované hranami (čarami). Pro popis těchto sítí se používá metodologie a terminologie z matematické teorie grafů. Matice sousednosti zobrazuje, zda jsou jednotlivé uzly spolu spojené hranou nebo ne a pro analýzu vztahů je důležitá míra - umístění uzlu v síti (tzv. centralita uzlu). Nejpoužívanější míry centrality jsou například „Degree centrality“ (centralita měřená stupněm uzlu), „Closeness centrality“ (blízkost polohy ve středu) nebo „Betwenness centrality“ (Středová mezipoloha uzlu). [1]

2. HTML a JavaScript

V této kapitole jsou popsány hlavní funkce a metody jazyků HTML a JavaScript, které byly použity při tvorbě praktické části práce.

2.1. Formulář v HTML

Formulář se zapisuje pomocí tagu `<form>`, který je párový, což znamená, že se za poslední položku formuláře musí uvést tag `</form>`. Atributy tohoto tagu mohou být například dle [7] následující:

- name - název formuláře
- action - obsahuje URL adresu skriptu, kterému jsou data z formuláře odeslána pro zpracování
- method - způsob předání dat z formuláře serveru, jsou metody GET nebo POST
- enctype - určuje způsob kódování dat z formuláře při jejich přenosu, pro přenos souborů se musí použít kódování „multipart/form-data“ a pro odeslání metoda POST

2.1.1. Metoda GET a POST

Při odesílání formuláře lze použít metodu GET nebo POST. Při použití metody GET jsou odesílaná data součástí URL adresy a jsou tedy vidět v adresním řádku internetového prohlížeče. Zapiší se za hlavní adresu stránky za otazník. Například z ukázkové adresy `www.stranka.cz?prom=5` je vidět, že obsahuje jednu proměnnou „prom“ ve které je uloženo číslo 5. Další proměnné se přidávají tak, že za hodnotu první proměnné připojíme „&“ a za něj napíšeme další proměnnou s její hodnotou. Takto můžeme odeslat několik proměnných. [4]

Metoda POST odesílá data odděleně od URL adresy, nejsou tedy v adresním řádku internetového prohlížeče vidět. Používá se při odesílání dat z rozsáhlejších formulářů, nebo když není vhodné, aby data odesílaná z formuláře byla vidět. [4]

2.1.2. Tag input, textarea

Tag <input> má mnoho podob. Může být například vstupním polem, zaškrtnutím políčkem, přepínačem či mazacím tlačítkem. To, jakým typem daný input bude, určuje atribut „type“. Dalšími atributy tagu <input> jsou například „name“, „value“, „size“, „checked“, „src“. Které z těchto atributů jsou v daném případě použitelné, závisí na typu daného *inputu*. Možnosti atributu „type“ pro tag <input> jsou dle [5] následující:

- text - je „normální“ vstupní pole,
- password - vstupní pole u kterého jsou vepsané hodnoty skryty
- radio - přepínač, který dovolí aktivovat pouze jednu hodnotu
- checkbox - zaškrtnutí políčko, uživatel může vybrat několik hodnot
- submit - odesílací tlačítko formuláře
- reset - tlačítko obnovující původní hodnoty ve formuláři
- hidden - pole není vidět, pouze se z něj odesílají informace
- image - odesílací tlačítko, které zároveň posílá souřadnice kliknutí na obrázek
- file - slouží k přenesení souboru (například fotografie)
- accept - určujeme jím MIME typy souborů, které je přístupné vybrat

Pro vložení delšího textu slouží tag <textarea>, jeho atributy jsou například „rows“, ve kterém se nachází počet řádků, „cols“, který určuje délku řádku, „wrap“ určující zalamování textu a samozřejmě atribut „name“. [5]

2.2. Obrázek v HTML

Pro zobrazení obrázku slouží tag s atributy „src“, který udává cestu k obrázku, atribut „alt“ zachycuje alternativní popis, který se zobrazí, pokud má uživatel ve svém internetovém prohlížeči vypnuté zobrazování obrázků. Rozměry obrázku se zadávají pomocí „width“ a „height“ nebo přes kaskádové styly.

2.3. JavaScript

JavaScript se zpracovává na straně klienta, což nám umožňuje zkontrolovat data vkládaná do formuláře ještě před jejich odesláním. Kód skriptu se zapisuje mezi tagy `<script type="text/JavaScript">` a `</script>`.

Charakteristiky jazyka JavaScript dle [6]:

- interpretovaný - nemusí se kompilovat
- objektový - využívá objektů prohlížeče a zabudovaných objektů
- závislý na prohlížeči - v některém prohlížeči nemusí fungovat nebo jej může mít uživatel zakázaný
- case sensitivní - v zápisu záleží na velikosti písmen

2.3.1. Funkce

V JavaScriptu, jako i v jiných jazycích, existují dva druhy funkcí - funkce bez argumentu a funkce s argumentem. U funkcí s argumentem lze přímo ovlivnit provádění úlohy, což u funkce bez argumentu nelze. V praktické části byly použity pouze funkce bez argumentu, proto zde je uveden její tvar:

```
function nazev_funkce()
{
    tělo funkce
}
```

Tělo funkce představuje souhrn příkazů, které se mají po vyvolání této funkce provést. Funkce mohou vypsát nějaký text pomocí příkazu „`document.write()`“ nebo mohou vracet hodnotu pomocí příkazu „`return`“. Tato hodnota může být číselná, textový řetězec nebo logické hodnoty. [11]

Funkce se volají pomocí příkazu nebo události. Ovladače událostí, které byly použity v praktické části práce, jsou následující:

- `onsubmit` - událost nastane při odesílání formuláře
- `onreset` - událost nastává ve chvíli, kdy uživatel klepne na restartovací tlačítko formuláře [11]

2.3.2. Větvení

JavaScript má, stejně jako ostatní programovací jazyky, několik typů rozhodovacích konstrukcí, s jejichž pomocí můžeme zabezpečit větvení skriptu. Jsou to přiřazovací podmínka „? :“, podmínka „if...else“ a příkaz „switch“. V praktické části byla využita podmínka „if...else“, která je nejpoužívanějším typem rozhodovací konstrukce. Obecně lze tuto podmínku zapsat ve tvaru [11]:

```
if (výraz) {  
    proved' příkazy;  
}  
else {  
    proved' příkazy;  
}
```

Porovnávací operátory dle [11]:

a == b	hodnoty proměnných a, b jsou stejné
a != b	hodnoty proměnných a, b jsou rozdílné
a < b	hodnota a je menší než hodnota b
a > b	hodnota a je větší než hodnota b
a >= b	hodnota a je větší nebo rovna hodnotě b
a <= b	hodnota a je menší nebo rovna hodnotě b

Dle [11] logické operátory vrací hodnotu true pokud:

a && b	jsou výrazy a, b pravdivé
a b	je pravdivý alespoň jeden z výrazů a, b
!a	je výraz a nepravdivý

2.3.3. Dialogová okna

Dialogová okna patří mezi nejzákladnější způsoby komunikace mezi uživatelem a stránkou. Existují systémová dialogová okna a uživatelská dialogová okna. Systémová dialogová okna nemůžeme ovlivnit v JavaScriptu, zahrnují všechny druhy chybových dialogových oken a bezpečnostních varování. Uživatelská dialogová okna mohou být výstražné informační okno, potvrzovací okno (Ok/Storno) a vstupní okno. Tato okna

můžeme JavaScriptem ovládat a pracovat s nimi. V praktické části je využito výstražné informační okno, které se zapisuje ve tvaru: `window.alert(„Text okna“)`. [11]

2.3.4. Formuláře

Pomocí JavaScriptu lze kontrolovat data, která uživatel vyplnil do formuláře. Můžeme zkontrolovat, zda uživatel zadal všechny povinné položky nebo zda zadal správný formát e-mailu. K jednotlivým formulářům a prvkům formuláře se nejlépe přistupuje pomocí jejich pojmenování. Například k formulářovému prvku `<input type="text" name="jmeno">`, který je součástí formuláře „form1“ a do kterého uživatel wpisuje svoje jméno, přistupujeme pomocí příkazu: `document.form1.jmeno.value`. Takovýmto způsobem můžeme přistupovat ke všem prvkům ve formuláři.

3. Příkazy jazyka SQL

Tato kapitola se bude zabývat nejčastěji používanými a nejdůležitějšími příkazy, které se používají při práci s databází MySQL, čili příkazy jazyka SQL. Jednotlivé podkapitoly se budou zabývat vytvořením tabulky, přidáváním záznamů do tabulky databáze, výběrem a prohlížením záznamů v tabulkách a také mazáním a úpravou záznamů.

3.1. Vytvoření tabulky

Pro vytvoření nové tabulky v databázi slouží SQL-příkaz CREATE TABLE. Jeho syntaxe je následující:

```
CREATE TABLE jméno_tabulky
(
    jméno1.sloupce typ,
    jméno2.sloupce typ,
    ...,
    jmenon.sloupce typ
)
```

Datové typy použitelné v jazyce SQL se moc neliší od typů jiných jazyků, k dispozici jsou typy pro celá i reálná čísla, pro text, datum, čas, speciální typy pro binární data apod. Při vytváření tabulek lze nastavit u každé položky i další vlastnosti jako je například vlastnost NOT NULL, což znamená, že takováto položka nesmí obsahovat prázdnou hodnotu. Další vlastností, kterou lze nastavit je například primární klíč - PRIMARY KEY. [7]

Při vytváření tabulek a jejich sloupců je dobré mít na paměti, že některé servery nepodporují české znaky a některé mají problém i s mezerami a pomlčkami, proto se při nazývání tabulek a sloupců doporučuje používat pouze písmena a podtržítka. [7]

3.2. Přidávání záznamu do tabulky

K přidávání nových záznamů do tabulky slouží SQL-příkaz INSERT INTO a jeho základní syntaxe je:

```
INSERT INTO jmeno_tabulky VALUES(  
    „hodnota1.sloupce“,  
    „hodnota2.sloupce“,  
    ...  
    „hodnota n.sloupce“) [7]
```

3.3. Výběr a prohlížení záznamů tabulky

Zřejmě nejpoužívanějším SQL-příkazem je příkaz SELECT, který umožňuje výběr a prohlížení záznamů z tabulky. Vypsání všech záznamů z tabulky je velmi snadné:

```
SELECT * FROM jmeno_tabulky
```

Tento příkaz je nejzákladnějším a nejjednodušším dotazem na databázi, ale jeho syntaxe může být mnohem složitější, jak je vidět na následujícím schématu [7]:

```
SELECT seznam položek, které chceme vypsát  
FROM seznam tabulek  
WHERE podmínka/y (další se přidávají pomocí AND)  
GROUP BY seznam položek  
HAVING skupinová podmínka  
ORDER BY kritéria třídění
```

3.4. Mazání záznamů

Pro smazání záznamu z tabulky slouží SQL-příkaz DELETE FROM, který má takovouto syntaxi:

```
DELETE FROM jméno_tabulky WHERE podmínka
```

Pozor se musí dát na to, že pokud se v příkazu nezadá podmínka, vymažou se všechny záznamy z tabulky. [7]

3.5. Změna záznamu v tabulce

Ke změně položek v záznamu tabulky slouží SQL-příkaz UPDATE. Jeho zápis vypadá dle [7] následovně:

```
UPDATE jméno_tabulky
SET jméno_položky = „nová_hodnota_položky“,
    jméno_položky = „nová_hodnota_položky“,
    ...
    jméno_položky = „nová_hodnota_položky“
WHERE podmínka
```

4. Vybrané vlastnosti a metody systému PHP

V této kapitole jsou popsány hlavní části jazyka PHP, jeho hlavní funkce a metody, které byly použity při tvorbě praktické části práce.

4.1. Superglobální proměnné

Superglobální proměnná je předdefinované pole obsahující hodnoty proměnných získaných ze serveru či od uživatele. Zde jsou dle [3] vypsány superglobální proměnné, které byly použity v praktické části práce:

- `$_GET[]`
zpracovává hodnoty, které získáme z URL adresy - hodnoty jednotlivých proměnných jsou v adresním řádku vidět. Například `www.stranka.cz?a=5&b=7` znamená, že v proměnné „a“ je uložena hodnota 5 a v proměnné „b“ je uloženo číslo 7. PHP automaticky vytvoří superglobální pole `$_GET[]`, které obsahuje všechny hodnoty předané v URL adrese. Hodnoty získáme použitím pole `$_GET["proměnná"]`, v našem případě tedy `$_GET["a"]` a `$_GET["b"]`.
- `$_POST[]`
zpracovává hodnoty získané metodou POST z formulářů. PHP vytvoří superglobální pole `$_POST[]`, které obsahuje hodnoty proměnných z formuláře, nejčastěji indexované pomocí HTML atributu „name“. Pokud máme například takovýto formulář:

```
<form method="post" action="zpracujform.php">  
    Jméno: <input type="text" name="jmeno">  
    Příjmení: <input type="text" name="prijmeni">  
    <input type="submit" value="Odeslat">  
</form>
```

Hodnoty z tohoto formuláře získáme ve skriptu „zpracujform.php“ použitím superglobálního pole `$_POST[“proměnná“]`, v našem případě konkrétně `$_POST[“jmeno“]` a `$_POST[“prijmeni“]`.

- `$_SESSION[]`

session je vlastně globální asociativní pole. Session by se dalo přeložit jako relace, představuje prostředek k zachování a předávání dat mezi jednotlivými přístupy ke stránkám. V PHP jsou session podporovány od verze 4.0. Když PHP obdrží příkaz k započetí session, zjistí nejdříve, jestli již session neběží a pokud ne, tak ji vytvoří. Když session již běží, tak se k ní připojí. Session se ukončí zavřením prohlížeče nebo lze ukončit manuálně. Session se spouští příkazem `session_start()` a registrace proměnných v session se provádí funkcí `session_register(“promenna“)`. [12] Proměnné do tohoto automaticky vytvořeného pole zapíšeme tímto způsobem: `$_SESSION[“promenna“] = hodnota`

- `$_FILES`

slouží k získávání souborů od uživatele prostřednictvím formuláře. PHP vytváří superglobální dvourozměrné pole, kde první index představuje atribut „name“ z formuláře a druhý index jednotlivé hodnoty týkající se vloženého souboru. Pokud bude mít atribut „name“ ve formuláři hodnotu „fotka“, k jednotlivým indexům (Tabulka 1).

Tabulka 1 - Přístupy k jednotlivým indexům pole `$_FILES` [2]

Položka	Význam
○ <code>\$_FILES[“fotka“][“name“]</code>	původní jméno souboru
○ <code>\$_FILES[“fotka“][“size“]</code>	velikost v bytech
○ <code>\$_FILES[“fotka“][“type“]</code>	MIME typ souboru
○ <code>\$_FILES[“fotka“][“tmp_name“]</code>	dočasné jméno souboru, pod kterým byl uložen na serveru
○ <code>\$_FILES[“fotka“][“error“]</code>	chybový kód

4.2. Logické výrazy a operátory

Porovnávací operátory dle [7]:

<code>\$_a == \$_b</code>	hodnoty proměnných a, b jsou stejné
<code>\$_a != \$_b</code>	hodnoty proměnných a, b jsou rozdílné
<code>\$_a <> \$_b</code>	hodnoty proměnných a, b jsou rozdílné
<code>\$_a < \$_b</code>	hodnota a je menší než hodnota b
<code>\$_a > \$_b</code>	hodnota a je větší než hodnota b
<code>\$_a >= \$_b</code>	hodnota a je větší nebo rovna hodnotě b
<code>\$_a <= \$_b</code>	hodnota a je menší nebo rovna hodnotě b

Tabulka 2 obsahuje logické operátory.

Tabulka 2 - Logické operátory [11]

Operátor	Vrací hodnotu true pokud:
<code>\$_a && \$_b</code> <code>\$_a AND \$_b</code>	jsou výrazy a, b pravdivé (logický součin)
<code>\$_a \$_b</code> <code>\$_a OR \$_b</code>	je pravdivý alespoň jeden z výrazů a, b (logický součet)
<code>\$_a XOR \$_b</code>	pravdivý je právě jeden z výrazů a, b
<code>! \$_a</code>	provede negování výrazu

4.3. Větvení, cykly v PHP

V PHP jsou dva způsoby větvení kódu a to „if“ a „switch“. V případě použití „switch“ je nejprve vyhodnocen výraz a pak jsou procházeny jednotlivé větve programu uvedené za slovem „case“, dokud není nalezena hodnota shodná s hodnotou vyhodnocovaného výrazu. Následně jsou pak vykonány příkazy dané větve „switch“.

Větvení pomocí „if“, „if...else“, „if...else if...else“ slouží k podmíněnému provedení příkazu za splnění určité podmínky. Podmínka se zadává jako výraz, který vrací logickou hodnotu („true“, „false“). [7]

Zde je uveden příklad:

```
if ($_a<$_b) {  
    echo ("a je menší než b");  
    další příkazy;}  
else if ($_a == $_b) {  
    echo ("a je rovno b");  
    další příkazy;}  
else  
    echo ("a je větší než b")
```

Tento příklad znázorňuje, že pokud hodnota v proměnné `$_a` bude menší než hodnota proměnné `$_b`, provedou se všechny příkazy ve složené závorce, v našem případě program vypíše „a je menší než b“ a provedou se další příkazy zapsané před ukončující složenou závorkou. Pokud první tvrzení neplatí, vstoupí program do druhé části podmínky a zjistí, zda je proměnná `$_a` rovna proměnné `$_b`, pokud ano, provede příkazy ve složené závorce a pokud ne, vstoupí do poslední části podmínky a vypíše, že „a je větší než b“. [7]

V PHP máme k dispozici cykly „while“, „do-while“ a „for“. Nejjednodušším cyklem je cyklus „while“, kterému se též říká cyklus s podmínkou na začátku. Znamená to, že dokud platí určitá podmínka, provádí se příkazy. Cyklus „do-while“ je rozdílný od cyklu „while“ tím, že podmínka je umístěna až na konci cyklu, takže příkazy v těle cyklu se provedou alespoň jednou, což v cyklu „while“ neplatí, pokud na začátku není splněna podmínka pro provedení cyklu. Cyklus „for“ je z cyklů nejvýkonnější a zároveň i nejsložitější. Jeho syntaxe je následující:

```
for (výraz1;výraz2;výraz3) příkaz;
```

„výraz1“ je vyhodnocen ještě před začátkem provádění cyklu, poté je vyhodnocen „výraz2“. Pokud je jeho hodnota „true“, provede se tělo cyklu a na konci každého průchodu tělem cyklu se vyhodnocuje „výraz3“. Tělo cyklu se provádí tak dlouho, dokud má výraz hodnotu „true“. [7]

4.4. PHP a MySQL

PHP podporuje několik databázových systémů, ale z důvodu využití MySQL databáze v praktické části práce bude zde popsána pouze spolupráce PHP s tímto databázovým systémem. [7]

Práci s databází zahájíme připojením k serveru pomocí funkce `MySQL_Connect()`. Tato funkce má jako parametr jméno počítače, na kterém běží MySQL. Pokud je přístup k databázi podmíněn jménem a heslem, zadávají se také jako parametr této funkce. Tato funkce vrací číslo spojení, které identifikuje připojení k danému serveru a lze jej použít v dalších funkcích. [7] V praxi vypadá tento příkaz například takto:

```
MySQL_Connect(„jméno serveru“, „přihlašovací jméno“, „přihlašovací heslo“)
```

Po přihlášení k serveru se vybere databáze, se kterou se bude pracovat, k čemuž se využívá funkce `MySQL_SELECT_DB`. Parametry této funkce jsou jméno databáze a číslo spojení s databází. [7]

Poté už lze na databázi klást dotazy v jazyce SQL, k čemuž slouží funkce `MySQL_Query()`, která má jako parametry SQL-příkaz a číslo spojení. Výsledkem funkce je číslo identifikující výsledek dotazu. [7]

Pro čtení jednotlivých záznamů výsledku dotazu `SELECT` slouží funkce `MySQL_Fetch Array()`, která čte výsledky dotazu a zapisuje je do asociativního pole. [7]

Na konci práce s databází je vhodné ukončit spojení s MySQL databází pomocí funkce `MySQL_Close()` s parametrem spojení. Pokud tak neučiníme, nic se nestane, protože PHP na konci každého skriptu automaticky ukončuje spojení se všemi SQL-servery. [7]

5. Realizace praktické části

Tato kapitola se bude zabývat realizací praktické části, tj. přípravě prostředí umělé sociální sítě pro předmět Sociální sítě. Průběh činností, které byly třeba pro úspěšné vytvoření umělé sociální sítě zapotřebí, jsou popsány v jednotlivých podkapitolách této kapitoly.

5.1. Definice požadavků na aplikaci

Aplikace má několik základních požadavků jako je jednoduchost, přehlednost, nemá být náročný na hardwarové ani softwarové prostředky. Aplikace musí být funkční v nejpoužívanějších internetových prohlížečích. Aplikace musí obsahovat registraci nových uživatelů, bezpečné přihlašování již registrovaných uživatelů, možnost přidání příspěvku a fotografie na zeď sociální sítě, možnost přidání přátel do seznamu přátel, úpravu osobních údajů a samozřejmě i prohlížení příspěvků a fotografií, jejich hodnocení a prohlížení profilů svých kamarádů.

Vzhledem k povaze aplikace a ke studiu bakalářského oboru Informatika ve veřejné správě a absolvovaných předmětech Tvorba WWW stránek, Technologie internetu a Databázové systémy I a II byla jako forma zvolena webová aplikace HTML spojená s jazykem PHP a JavaScript. Jako databázový systém je použit MySQL a jsou použity kaskádové styly CSS. Aplikace je určena pro webový server podporující PHP a databázový systém MySQL, k jejímu spuštění postačí webový prohlížeč.

5.2. Návrh databáze

Tato podkapitola obsahuje proces návrhu databázového modelu na všech úrovních a jeho realizaci. Jednotlivé úrovně návrhu jsou popsány v následujících podkapitolách.

5.3.1. Konceptuální úroveň

V první části návrhu databázového systému musíme určit entity a vztahy mezi nimi. Každá entita má své atributy, podtržené atributy jsou primárními klíči, atributy napsané kurzívou jsou cizími klíči. V sociální síti jsou zjištěny následující entity a jejich atributy:

Uživatel

Uživatel je základní entitou v sociální síti, touto entitou jsou reprezentováni všichni lidé, kteří se zaregistrují k používání sociální sítě.

Atributy: id, jméno, příjmení, pohlaví, nick, e-mail, heslo, sport, tanec, hudba, filmy, posezení s přáteli, internet, hry, divadlo, památky, přírodní památky, zvířata, automobily, jiné, fotografie, datum registrace.

Příspěvek

Příspěvek přidává uživatel, je to základní entita komunikace mezi uživateli. Do příspěvku se ukládá text a může se přiložit i fotografie nebo obrázek.

Atributy: id_příspěvku, id_uživatele, text_příspěvku, datum_a_čas_vložení, fotografie

Akce

Entita akce zastupuje akce, které uživatel na sociální síti provedl. Je to prohlédnutí a ohodnocení fotky, přihlášení či odhlášení uživatele, prohlížení osobní strany kamaráda.

Atributy: id_akce, druh_akce, čas_akce, id_uživatele, akce (její popis), název_fotky_id_kamaráda (název fotky při prohlédnutí, nebo id kamaráda, na jehož osobní stranu uživatel klikl), likes (hodnocení u fotky)

Mezi jednotlivými entitami jsou definované vztahy, které si zde popíšeme:

Je přidán

Tento vztah je vztahem mezi uživatelem a příspěvkem. Reprezentuje přidání příspěvku uživatelem. Uživatel může vložit několik příspěvků a příspěvek musí být přidán jedním uživatelem.

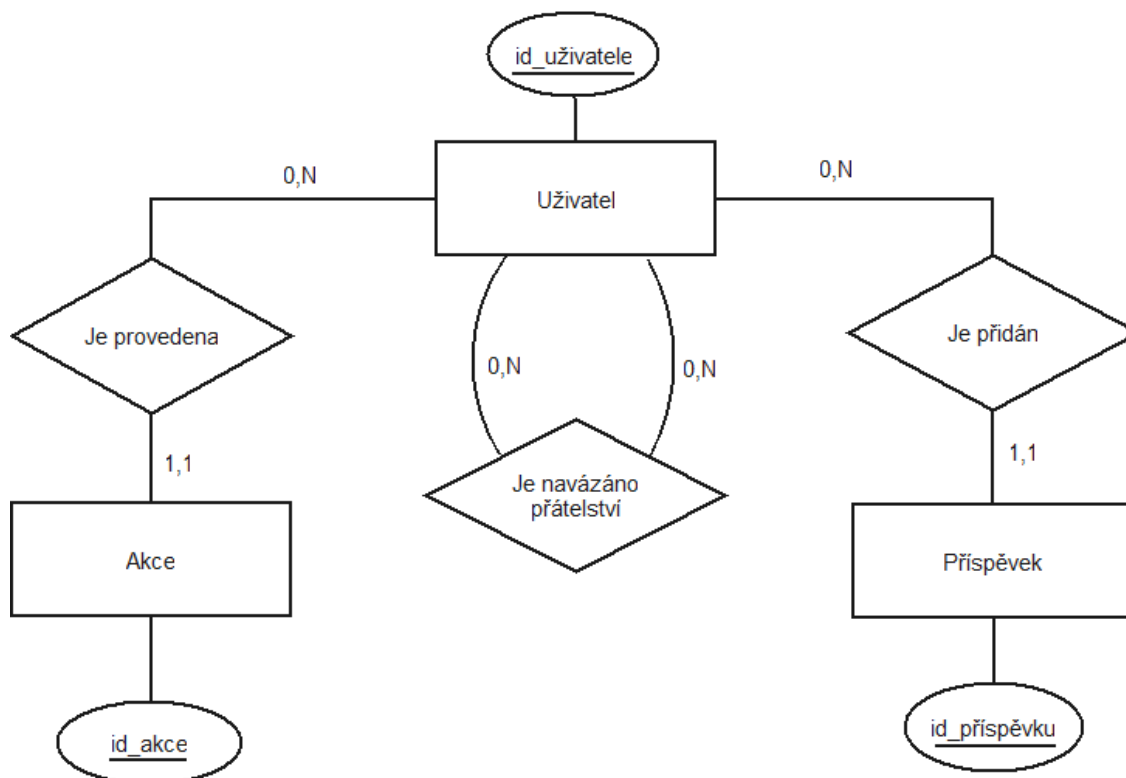
Je provedena

Vztah mezi akcí a uživatelem, zachycuje akce, které daný uživatel provedl v sociální síti. Protože již při registraci se uživatel „přihlásí“, je zde povinnost ve vztahu u obou entit ve vazbě 1:N. Uživatel musí provést alespoň jednu akci a akci musí provést jeden uživatel.

Je navázáno přátelství

Vztah mezi dvěma uživateli, zaznamenává, že se daní uživatelé přátelí. Každý uživatel se může přátelit s N uživateli, oba uživatelé se účastní vztahu nepovinně.

Obrázek 1 zachycuje výše popsané entity a vztahy mezi nimi v E-R diagramu (Entity-Relationship diagram).



Obrázek 1 - E-R diagram [zdroj: vlastní]

5.3.2. Technologická úroveň

Z důvodu použití databázového systému MySQL na implementační úrovni, se zde provede relační model dat, ve kterém jsou údaje uspořádány do relací - tabulek. Tabulka má své jméno a zpravidla obsahuje údaje (atributy) o jednom druhu entity. Řádky představují jednotlivé záznamy a sloupce jejich atributy. Každá tabulka by měla obsahovat primární klíč. Primárním klíčem je myšlen atribut, jehož hodnota je pro každý záznam jedinečná. Pro každý atribut v tabulce musíme určit typ dat, které může obsahovat. Tabulky mezi sebou mohou být propojené. [7] Tabulky vzniknou transformací entit a vztahů.

Obrázek 2 znázorňuje relace vzniklé transformací entit z konceptuální úrovně do datového modelování.

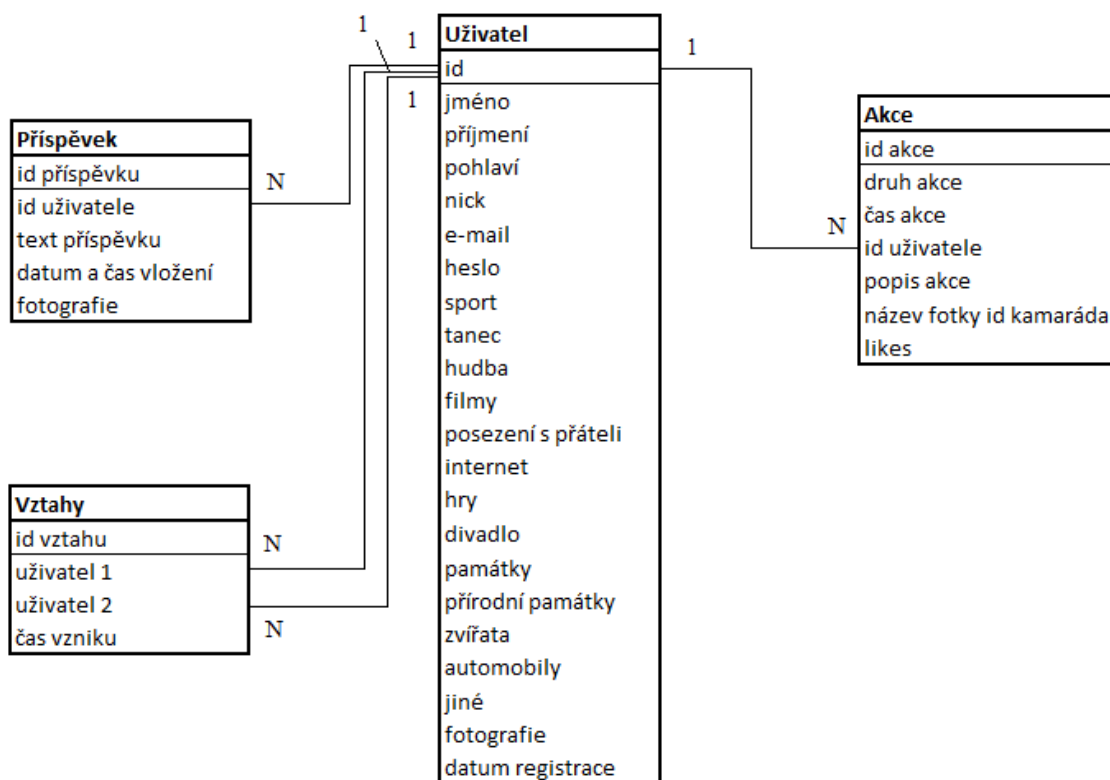
Uživatel	Příspěvek	Akce
id	id příspěvku	id akce
jméno	id uživatele	druh akce
příjmení	text příspěvku	čas akce
pohlaví	datum a čas vložení	id uživatele
nick	fotografie	popis akce
e-mail		název fotky id kamaráda
heslo		likes
sport		
tanec		
hudba		
filmy		
posezení s přáteli		
internet		
hry		
divadlo		
památky		
přírodní památky		
zvířata		
automobily		
jiné		
fotografie		
datum registrace		

Obrázek 2 - Relace vzniklé z entit [zdroj: vlastní]

Transformaci vztahů mezi entitami v našem případě provádíme pouze tehdy, pokud je mezi entitami vazba M:N, což splňuje pouze vazba Je navázáno přátelství mezi jednotlivými uživateli. Z tohoto důvodu vznikne relace Vztahy, která bude zachycovat přátelství mezi uživateli. Atributy této relace budou „id_vztahu“, a „čas vzniku“. K těmto atributům se ještě připojí jednoznačné identifikátory uživatelů: *uživatel_1*, *uživatel_2*, které budou obsahovat jejich ID. Obrázek 3 zachycuje vzniklou relaci vztahu. Obrázek 4 zobrazuje celé relační schéma.

Vztahy
id vztahu
uživatel 1
uživatel 2
čas vzniku

Obrázek 3 - Relace vztahu [zdroj: vlastní]



Obrázek 4 - Relační schéma [zdroj: vlastní]

5.3.3. Implementační úroveň

Na této úrovni modelování je vybrán konkrétní databázový systém, jsou zde popsány konkrétní nástroje a aplikace, které jsou při tvorbě databáze využity.

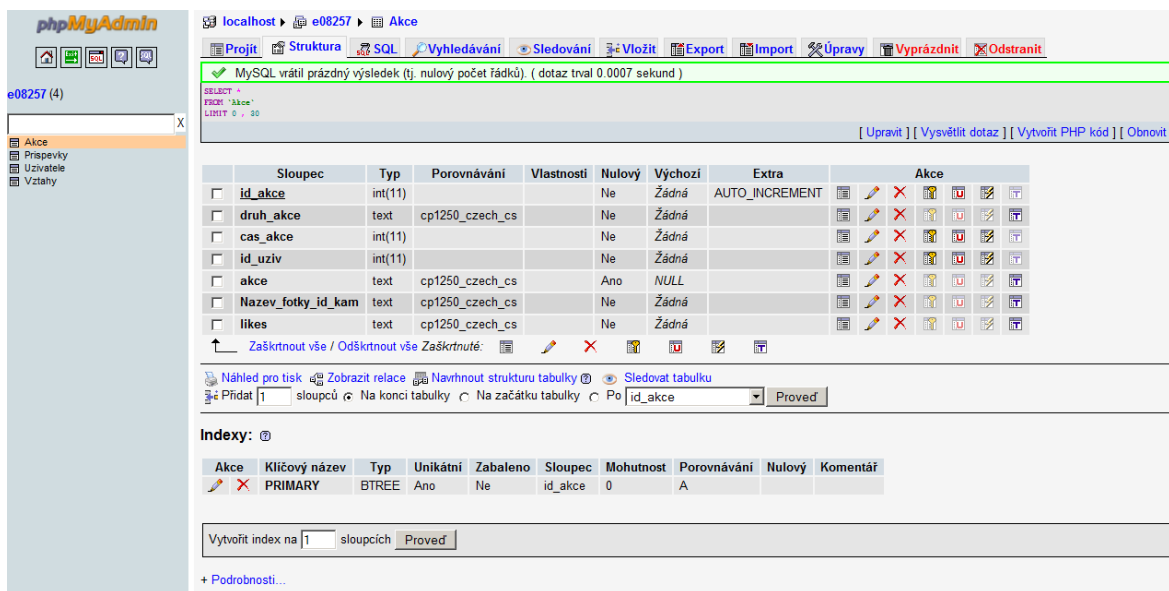
Při tvorbě byl vybrán databázový systém MySQL, konkrétně ve verzi MySQL 5.1.51. Databáze je umístěna na univerzitním serveru Apache a přístup k ní se provádí pomocí phpMyAdmin, což je nástroj pro správu databáze MySQL. Použitá byla verze

phpMyAdmin 3.3.8.1. V databázi jsou vytvořeny 4 tabulky, které vzniknou z relací popsaných na technologické úrovni modelování. Tabulky jsou vytvořeny pomocí phpMyAdmin (Obrázek 5).



Obrázek 5 - Vytvoření tabulky v phpMyAdmin [zdroj: vlastní]

Po potvrzení vytvoření tabulky se zobrazí budoucí struktura tabulky, do které je potřeba zadat hodnoty. U každého atributu zadáváme jeho název (sloupec), typ, a volitelně můžeme zvolit i délku, výchozí hodnotu, porovnání, vlastnosti, nulovost, index, auto increment, komentáře, mime typ, transformaci při prohlížení a parametry transformace. Obrázek 6 zobrazuje práci v phpMyAdmin, jsou na něm v levé části vidět již vytvořené tabulky a v pravé části struktura tabulky *Akce*.



Obrázek 6 - Práce v phpMyAdmin [zdroj: vlastní]

Jednotlivé vytvořené tabulky, jejich atributy, datové typy a popisy atributů jsou zachyceny v následujících čtyřech tabulkách.

Tabulka akce

Tabulka 3 „Akce“ vznikla z relace „*Akce*“ definované na technologické úrovni modelování, obsahuje informace o uživatelem prováděných akcích jako je přihlášení, odhlášení, prohlížení fotografií či prohlížení profilů kamarádů.

Tabulka 3 - Databázová tabulka Akce [zdroj: vlastní]

AKCE		
Atribut	Datový typ	Popis
id_akce	int(11)	Identifikační číslo akce, vkládá se pomocí auto increment.
druh_akce	text	Ukládá druh akce: P - přihlášení uživatele, O - odhlášení uživatele, K - prohlédnutí osobní strany kamaráda, F - prohlédnutí a případné ohodnocení fotografie, která byla vložena na zeď.
cas_akce	int(11)	Čas provedení akce, uvedeno ve formátu počtu uplynulých vteřin od 1. 1. 1970.
id_uziv	int(11)	Identifikační číslo uživatele, který danou akci provedl. Cizí klíč.
akce	text	Popis akce.
Nazev_fotky_id_kam	text	Při přihlášení a odhlášení uživatele neobsahuje nic, při prohlédnutí profilu kamaráda obsahuje identifikační číslo kamaráda a při prohlédnutí a ohodnocení fotografie obsahuje cestu k dané fotografii.
likes	text	Pouze u fotografie, zachycuje, zda se danému uživateli fotografie líbí nebo nelíbí.

Tabulka uživatelé

Tabulka 4 „Uživatelé“ vznikla z relace „*Uživatel*“ definované na technologické úrovni modelování, obsahuje důležité informace o uživateli.

Tabulka 4 - Databázová tabulka Uživatelé [zdroj: vlastní]

UŽIVATELÉ		
Atribut	Datový typ	Popis
ID	int(11)	Identifikační číslo uživatele, je primárním klíčem, nemá žádný informační význam, důležitý pouze pro dotazování na databázi. Vypĺňováno pomocí auto increment.
Jméno	text	Jméno uživatele, které vyplnil do registračního formuláře.
Prijmeni	text	Příjmení uživatele, které vyplnil do registračního formuláře. Lze podle něj vyhledávat přátele.
Pohlavi	text	Pohlaví uživatele získané z radio inputu formuláře.
Nick	varchar(20)	Uživatelská přezdívka, získává se z formuláře a je jedinečná (jeden nick může v databázi existovat jen jednou). Používá se například pro vyhledávání přátel.
Email	varchar(30)	E-mail uživatele získaný z formuláře. Je jedinečný, lze použít pro vyhledávání přátel.
Heslo	text	Heslo slouží pro přihlášení uživatele. Je zvoleno uživatelem a zapsáno do databáze.
Sport	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Tanec	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Hudba	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Filmy	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Posezeni	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Internet	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Hry	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Divadlo	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Vystavy	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Pamatky	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
PrirodniPamatky	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Zvirata	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Automobily	text	Záliba uživatele. Zde je uvedeno, zda zálibu má, nebo ne.
Jine	text	Zde může uživatel zadat případnou další zálibu.
Fotografie	text	Obsahuje cestu k profilové fotografii uživatele, pokud byla vložena.
Datum_registrace	int(11)	Datum zaregistrování uživatele ve formátu počtu uplynulých vteřin od 1. 1. 1970.

Tabulka vztahy

Tabulka 5 „Vztahy“ vznikla z relace „*Vztahy*“ definované na technologické úrovni modelování, obsahuje informace o přátelství mezi jednotlivými uživateli.

Tabulka 5 - Databázová tabulka Vztahy [zdroj: vlastní]

VZTAHY		
Atribut	Datový typ	Popis
uz1	int(11)	Id prvního uživatele, který se přátelí s druhým uživatelem, cizí klíč.
uz2	int(11)	Id druhého uživatele, který je s prvním uživatelem v přátelství. Cizí klíč.
id_vztahu	int(11)	Identifikační číslo vztahu, vyplňováno pomocí auto increment.
Cas_vzniku	int(11)	Čas vzniku přátelství, uvedeno ve formátu počtu uplynulých vteřin od 1. 1. 1970.

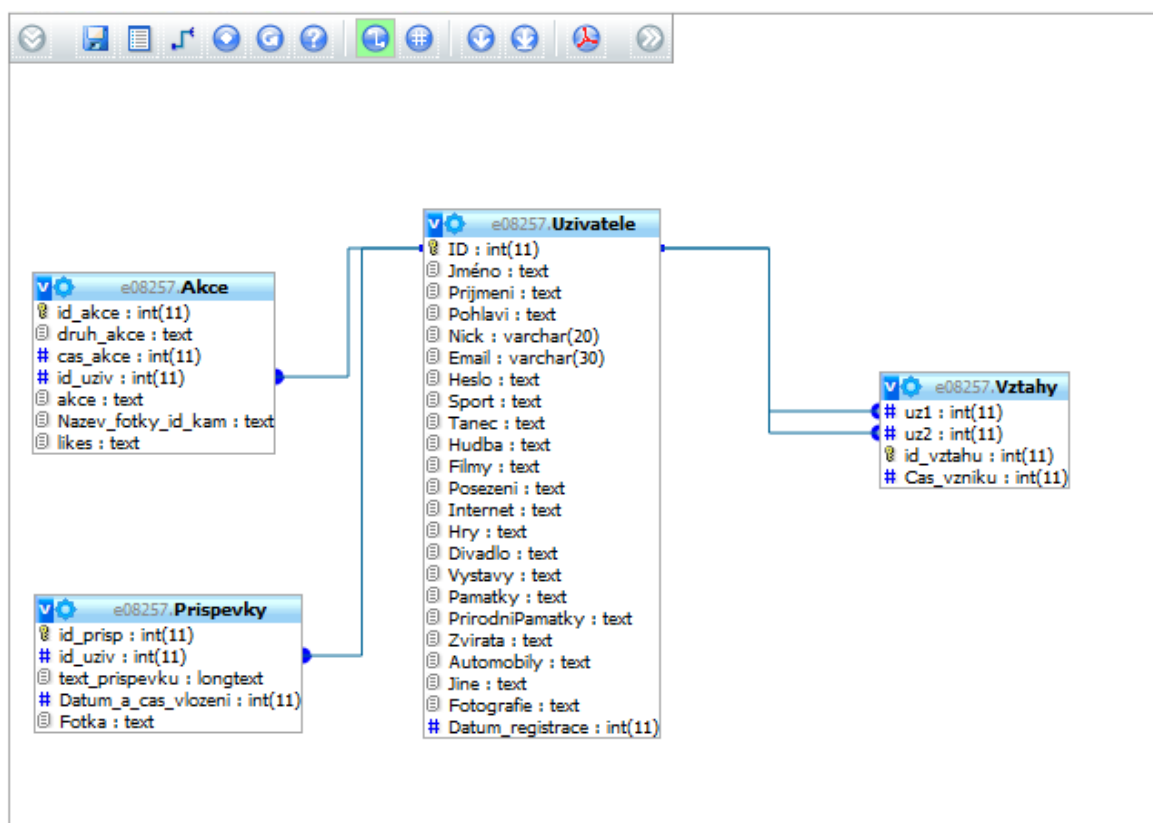
Tabulka příspěvky

Tabulka 6 „Příspěvky“ vznikla z relace „*Příspěvek*“ definované na technologické úrovni modelování, všechny vložené příspěvky uživatelů.

Tabulka 6 - Databázová tabulka Příspěvky [zdroj: vlastní]

PŘÍSPĚVKY		
Atribut	Datový typ	Popis
id_prisp	int(11)	Identifikační číslo příspěvku, vyplňováno auto increment. Je primárním klíčem.
id_uziv	int(11)	Identifikační číslo uživatele, který příspěvek vložil, cizí klíč.
text_prispevku	longtext	Obsahuje text příspěvku.
Datum_a_cas_vlozeni	int(11)	Čas vložení příspěvku, uvedeno ve formátu počtu uplynulých vteřin od 1. 1. 1970.
Fotka	text	Pokud byla k příspěvku přiložena fotografie, zde je uvedena cesta k souboru.

Po vytvoření všech potřebných tabulek je ještě třeba jednotlivé tabulky propojit mezi sebou přes primární a cizí klíč. V phpMyAdmin se jednotlivé tabulky propojí mezi sebou tak, že se klikne na název databáze (v tomto případě e08257), zvolí se položka návrhář, v návrhář se zobrazí všechny tabulky a jejich atributy a pomocí funkce „Vytvořit relaci“ vytvoří se jednotlivé požadované vazby mezi atributy tabulek. Obrázek 7 zachycuje výsledné propojení tabulek v naší databázi.

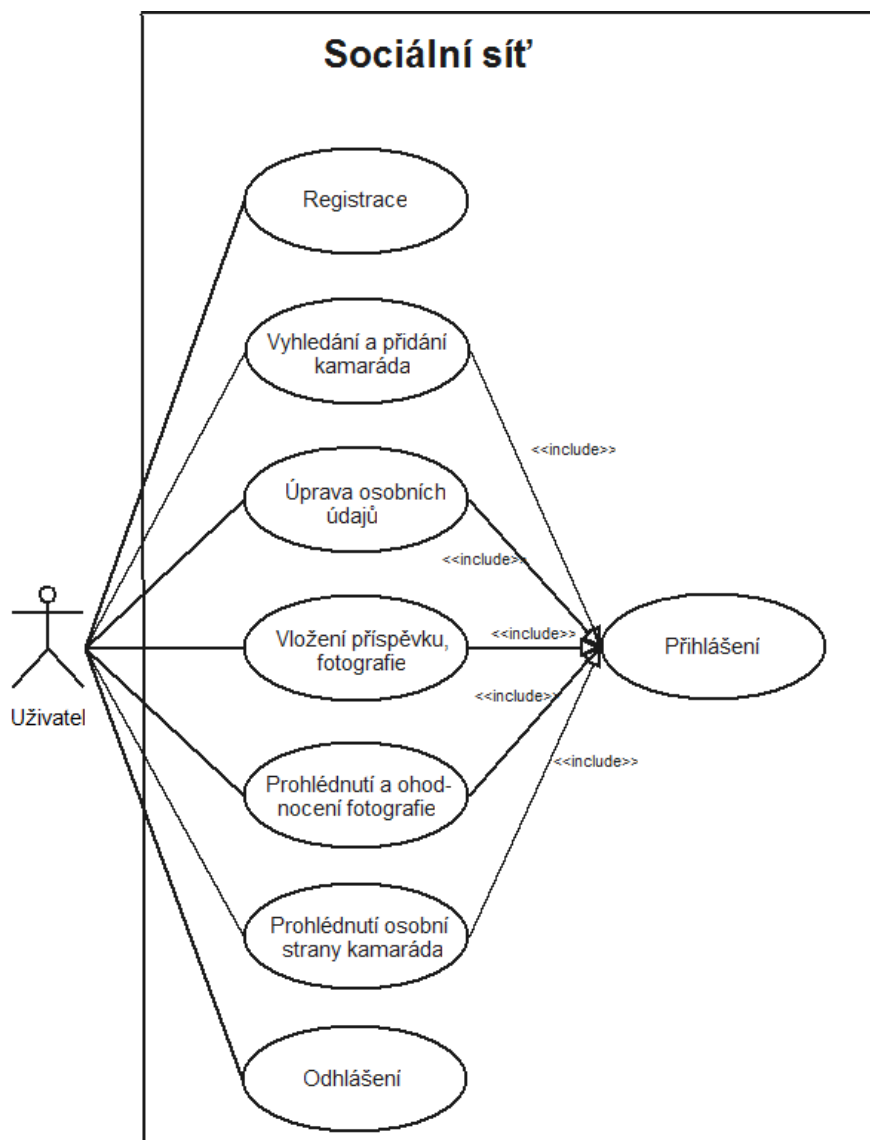


Obrázek 7 - Propojení tabulek v *phpMyAdmin* [zdroj: vlastní]

5.3. Use case diagram

Use Case diagram (případy užití) dokumentuje možné případy použití systému vyvolané událostmi, na které musí systém reagovat. Případy užití přesně zachycují funkčnost, která bude systémem pokryta, a tím jednoznačně vymezují rozsah prací. [10]

Aktér (zobrazen panáčkem) představuje uživatelskou roli. Hranice systému (zobrazena obdélníkem) představuje vymezení hranice mezi systémem a okolím systému. Příklad užití (zobrazen elipsou) představuje službu, kterou musí systém poskytnout jako aktivitu vyvolanou podnětem zvenčí. Příklad užití je vždy iniciován aktérem. [10] Obrázek 8 zobrazuje use case diagram této umělé sociální sítě.



Obrázek 8 - Use Case diagram [zdroj: vlastní]

5.4. Grafické rozložení aplikace

Pro vytvoření základu grafického prostředí aplikace byl využit program pro generování šablon Artisteer 3. Tento základ byl upraven dle potřeb sociální sítě. Obrázek 9 zobrazuje výsledné grafické rozložení sociální sítě.



Obrázek 9 - Grafické rozložení aplikace, osobní strana [zdroj: vlastní]

A - Menu aplikace

B - Profilová fotografie a nick přihlášeného uživatele

C - Přátelé přihlášeného uživatele (jejich profilové fotografie a nicky)

D - Prostor pro vkládání příspěvků a fotografií

E - Prostor pro zobrazení textových příspěvků a fotografií uživatele a jeho přátel za poslední týden řazené od nejnovějšího k nejstaršímu příspěvku

5.5. Registrace uživatele

Při zobrazení stránky *index.php* se uživateli zobrazí stránka, kde má uživatel možnost se buď přihlásit do sociální sítě, nebo v případě, že je na této sociální síti poprvé, může se zde zaregistrovat, kliknutím na položku menu „Registrace“, která ho přesměruje na stránku *registrace.htm*.

Registrace nového uživatele

Jméno*

Příjmení*

Pohlaví* Muž Žena

Nick*

E-mail*

Heslo*

Potvrzení hesla*

Zájmy

Sport

Tanec

Hudba

Filmy

Posezení s přáteli

Internet

Hry - počítačové i ostatní

Divadlo

Výstavy

Hradý, zámky a podobné památky

Přírodní památky a turistika

Zvířata

Automobily, motoroky

Jiné

* Takto označené položky jsou povinné

Copyright © 2011. All Rights Reserved.

Powered by [Artisteer.net](#) and Template created with [Artisteer](#).

Obrázek 10 - Registrace nového uživatele [zdroj: vlastní]

Na této stránce uživatel vyplní registrační formulář (Obrázek 10) a poté klikne na tlačítko „Kontrola nicku, e-mailu a přidání fotografie“. Při kliknutí na toto tlačítko se aktivuje JavaScriptová funkce *kontrola* uložená v externím JavaScriptovém souboru „*script.js*“, která zjistí, zda uživatel vyplnil všechny povinné položky, jestli je e-mail ve správném tvaru, jestli se shodují zadaná hesla a zda není nick nebo e-mail již registrován. Kód funkce *kontrola* je následující:

```
function kontrola(){
x=pohlavi();
z=mail();
if (document.form1.jmeno.value=="    ||
    document.form1.prijmeni.value=="    ||
    document.form1.nick.value=="    ||
    document.form1.mail.value=="    ||
    document.form1.heslo1.value=="    ||
    document.form1.heslo2.value=="    ||
```

```

        x==false) {
window.alert("Nejsou vyplněny všechny povinné položky!");
zmen(); return false;}
else if(document.form1.heslo1.value != document.form1.heslo2.value)
    {window.alert("Neshodují se hesla!");
zmen(); return false;}
    else if(z==false){window.alert("Špatně zadaná e-mailová adresa!");
zmen(); return false;}
        else{ return true; }

```

Tato funkce využívá dalších funkcí jako je funkce *pohlaví*, která určuje, zda bylo vybráno pohlaví, funkce *mail*, která zajišťuje kontrolu správné formy vloženého e-mailu, či funkce *zmen*, která se aktivuje v případě, že bylo něco špatně vyplněno a změní barvu písma u těch položek, které buď nebyly vyplněny, nebo byly vyplněny špatně. Zde je uveden kód funkce *pohlaví* a kód funkce *mail*:

```

function pohlavi(){
if(document.form1.pohlavi[0].checked || document.form1.pohlavi[1].checked){
return true; }
else{ return false; }}

function mail(){
if (window.RegExp){
znagy=new RegExp("^[^.]+(\\.[^.]*)*@([^.]+[.])+[a-z]{2,3}$");
if (!znaky.test(document.form1.mail.value)){
document.form1.mail.focus();
return false;}}}}

```

Pokud vše proběhne v pořádku, uživateli se zobrazí druhý formulář s předvyplněnými daty, která zadal uživatel v prvním formuláři, a uživatel zde může ještě přiložit svou profilovou fotografii. Při stisknutí tlačítka „Registrace“ se opět provede funkce *kontrola* pro případ, že by uživatel v tomto formuláři něco změnil nebo smazal. Následně stránka „*zpracuj.php*“ zjistí, zda byla k formuláři přiložena i fotografie uživatele a pokud ano, zjistí její „MIME“ typ a podle něj do proměnné „*přípona*“ uloží příslušný

formát fotografie. Dále přesune uloženou fotografii z dočasného umístění na serveru do složky „ikony“ pod názvem složeným z e-mailové adresy uživatele a přípony, která je uložena v proměnné „přípona“. Pokud uživatel svou fotografii nepřiloží, bude se místo jeho profilové fotografie zobrazovat obrázek „smile“. Po úspěšné registraci se uživateli zobrazí jeho „Osobní strana“ (Obrázek 9).

5.6. Přihlášení a odhlášení uživatele

Uživatel se při každé návštěvě sociální sítě musí přihlásit. Po stisknutí tlačítka „přihlásit“ dojde k přesměrování na stránku „*prihlaseni.php*“, kde dojde k dotazu na databázi, kterým vybereme heslo z databáze podle zadaného *nicku*. Po zkontrolování, že se zadané heslo shoduje s heslem vybraným z databáze, se zahájí session, která udržuje v superglobální proměnné `$_SESSION["nick"]` *nick* přihlášeného uživatele. Zahájení session, registrace proměnné *nick* a uložení *nicku* do této proměnné se provede následujícím kódem:

```
session_start();
session_register("nick");
if(!isset($_SESSION["nick"]))
$_SESSION["nick"]=$_nick;
```

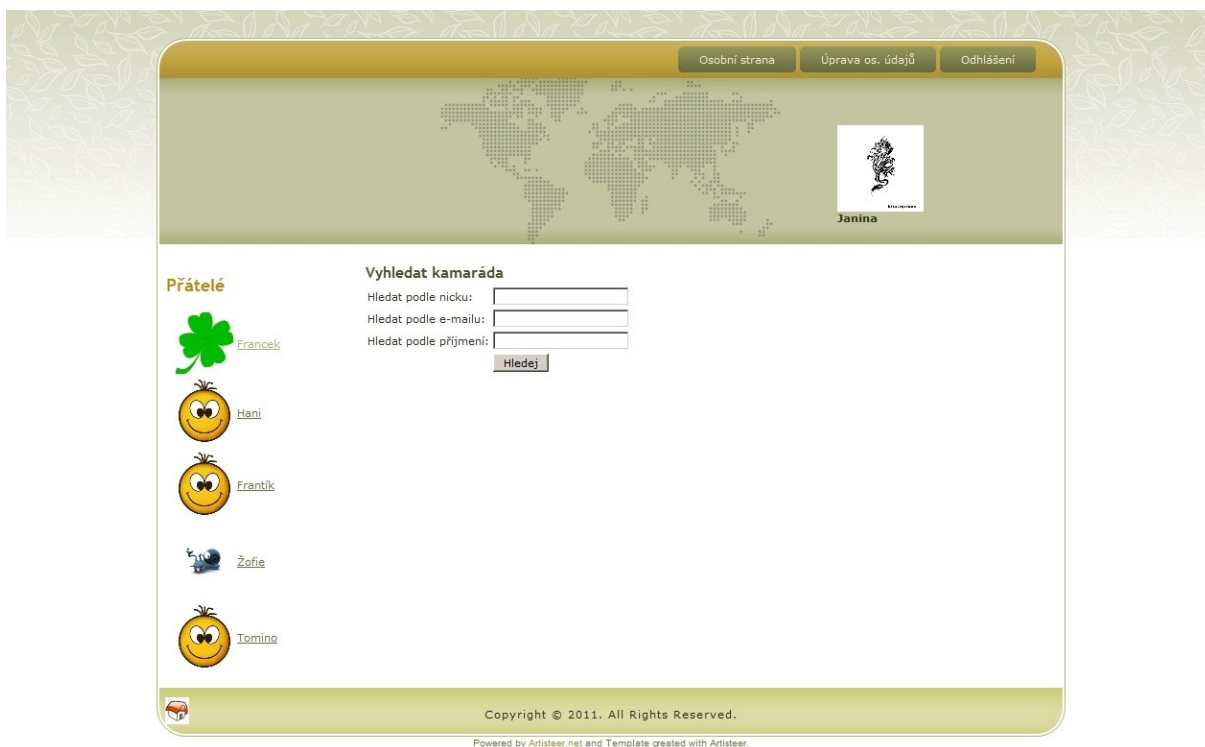
Pokud ale uživatel zadá špatné heslo nebo nick, zobrazí se stránka s informací, že byly zadány špatné přihlašovací údaje. Na této stránce je také přihlašovací formulář, kde uživatel může ihned zkusit přihlašování znovu.

Pokud by se nepřihlášený uživatel snažil otevřít některou ze stránek, ke kterým je zapotřebí být přihlášený, tato stránka ho přesměruje na stránku, kde se mu zobrazí hlášení: „Neoprávněný přístup! Prosím, přihlaste se.“

Při stisknutí odkazu „Odhlášení“ v menu se tato proměnná vymaže pomocí funkce `unset($_SESSION["nick"])`, tím se uživatel odhlásí od sociální sítě.

5.7. Přidání kamaráda

Přidání kamaráda uživatel provede tak, že na „Osobní straně“ vybere položku „Přidat kamaráda“ z menu. Tím se uživatel dostane na stránku „*pridej.php*“, kde může vyhledávat kamaráda podle jeho „nicku“, e-mailu nebo příjmení (Obrázek 11).



Obrázek 11 - Vyhledání kamaráda [zdroj: vlastní]

Vyhledávání probíhá tak, že se z formuláře odešle nick, e-mail nebo příjmení hledané osoby, a v souboru „*hledani.php*“ se vyhledává uživatel dotazem na databázi. Pokud například uživatel bude hledat kamaráda pomocí příjmení, kód pro vyhledání bude v následující podobě:

```
if ($_hledp!=""){
    $_dotaz = "SELECT ID, Jméno, Prijmeni, Nick, Email, Fotografie FROM Uzivatele
              WHERE Prijmeni='$_hledp'";

    $_vrati = MySQL_Query($_dotaz);
    while($_hodn = MySQL_Fetch_Array($_vrati)){ ... zde vypíšeme všechny uživatele
vyhovující zadanému příjmení...}}
```

Vyhledávání podle nicku a e-mailu vrátí maximálně jednoho člověka odpovídajícího zadanému kritériu, protože „nick“ i e-mail jsou jedinečné. Hledání podle příjmení může vrátit několik uživatelů, ze kterých si uživatel vybere toho, kterého hledal. Při výběru podle příjmení by uživateli měly pomoci zobrazované sloupce Jméno, Nick, E-mail, Fotografie. Po vybrání svého kamaráda uživatel klikne na tlačítko „Přidat kamaráda“ u vybraného člověka a tím se přesměruje na stranu „ulozkam.php“, která zjistí, zda uživatel nepřidává jako kamaráda sám sebe nebo někoho, koho již v přátelích má. Pokud ano, vypíše mu chybovou hlášku a pokud ne, tak kamaráda uloží do databáze a uživatele přesměruje na jeho osobní stranu, kde se mu v přátelích již zobrazí právě přidáný kamarád a v poli příspěvků se zobrazí kamarádovy vložené příspěvky. Kód pro výpis všech kamarádů daného uživatele je takovýto:

```
<table>
<? $_sql = "SELECT * FROM Vztahy WHERE uz1='$_id' or uz2='$_id'";
$_vysl = MySQL_Query($_sql);
while($_napln = MySQL_Fetch_Array($_vysl)){
    if ($_napln != ""){
        if ($_id == $_napln["uz1"]){
            $_kam = $_napln["uz2"];
            $_dotaz2 = "SELECT Nick, Fotografie FROM Uzivatele WHERE ID='$_kam'";
            $_vysledek = MySQL_Query($_dotaz2);
            $_vysled = MySQL_Fetch_Array($_vysledek); ?>
<tr><td><?if ($_vysled["Fotografie"] != ""){?><img alt="Profilová fotografie"
style='width:60px' src='<? echo $_vysled["Fotografie"];?>' /><?}

        else {?><img alt="Smile" style='width:60px' src='ikonka.jpg' /> <? ;} ?></td>
<td><a href="strkamarada.php?kam=<?echo $_vysled["Nick"];?>">
<?echo $_vysled["Nick"];?></a></td></tr>
<? ;} else if ($_id == $_napln["uz2"]){
    $_kam = $_napln["uz1"];
    $_dotaz2 = "SELECT Nick, Fotografie FROM Uzivatele WHERE ID='$_kam'";
    $_vysledek = MySQL_Query($_dotaz2);
    $_vysled = MySQL_Fetch_Array($_vysledek);
    ?><tr><td><?if ($_vysled["Fotografie"] != ""){?><img alt="Profilová fotografie"
```

```

        style='width:60px' src='<? echo $_vysled["Fotografie"];?>' /><?>
    else {?><img alt="smile" style='width:60px' src='ikonka.jpg' /> <? ;} ?></td>
<td><a href="strkamarada.php?kam=<?echo $_vysled["Nick"];?>">
<?echo $_vysled["Nick"];?></a></td></tr>
    <? ;}}?>
</table>

```

Kliknutím na nick kamaráda v seznamu přátel se uživateli zobrazí strana kamaráda, kde uživatel vidí kamarádovo jméno a příjmení, kontaktní údaje, záliby, profilovou fotografii a jeho přátele (Obrázek 12).



Obrázek 12 - Strana kamaráda [zdroj: vlastní]

5.8. Úprava osobních údajů

Vzhledem k tomu, že v průběhu času se uživatelovy zájmy mění, má možnost upravit své osobní údaje pomocí položky menu „*Úprava osobních údajů*“. Zde uživatel může pomocí formuláře změnit své zájmy, fotografii a heslo (Obrázek 13).

Osobní strana Odhlášení

Janina

Úprava osobních údajů


Jméno: Jana
Příjmení: Kremlová
Pohlaví: Žena
Nick: Janina
E-mail: janina@seznam.cz

Změna hesla:
Nové heslo:
Potvrzení nového hesla:
Původní heslo:

Zájmy:

- Sport
- Tanec
- Hudba
- filmy
- Posezení s přáteli
- Internet
- Hry - počítačové i ostatní
- Divadlo
- Výstavy
- Hrady, zámky a podobné památky
- Přírodní památky a turistika
- Zvířata
- Automobily, motorcky

Jiné:

Fotografie: 

Smaž fotografii

* Takto označené položky jsou povinné!

Copyright © 2011. All Rights Reserved.
Powered by [Artisteer.net](#) and Template created with [Artisteer](#).

Obrázek 13 - Úprava osobních údajů [zdroj: vlastní]

Svou profilovou fotografií uživatel změní tak, že dříve vloženou fotografii smaže a poté uloží novou fotografii. Smazání původní fotografie probíhá tak, že pokud stránka „*update_save.php*“ zjistí, že uživatel chce smazat původní fotografii tak z databáze vymaže cestu k fotografii a na serveru smaže danou fotografii pomocí funkce „*unlink(cesta k fotografii)*“. Z bezpečnostních důvodů heslo uživatel změní pouze v případě, že zadá správné dosavadní heslo. Změna jména, příjmení, pohlaví, nicku a e-mailu není uživateli povolena, změnit tyto položky může jedině tak, že požádá administrátora sociální sítě. Před odesláním upravených údajů se provede kontrola zadaných dat, v tomto případě se jedná pouze o shodu hodnot vyplněných do položek inputů „Nové heslo“ a „Potvrzení nového hesla“.

5.9. Vložení příspěvku, prohlédnutí a ohodnocení fotografie

Vkládání příspěvku se provádí na „Osobní straně“, kde může uživatel přidat textový příspěvek a k němu může přiložit fotografii (Obrázek 9). Příspěvek i s fotografií se zobrazuje autorovi a všem jeho kamarádům po dobu jednoho týdne. Kód formuláře pro vkládání příspěvků je následující:

```
<form name="pridaniprispevku" enctype="multipart/form-data" action="ulozprisp.php"
method="post">
<table>
<tr><td>Vložit příspěvek:<br /></td><td>Přiložit fotografii:</td></tr>
<tr><td rowspan="2"><input type="hidden" name="ID" value="<?echo $_id ;?>" />
<textarea name="prispevek" cols="70" rows="3" wrap="soft"></textarea></td><td>
<input type="file" name="fotka" /></td></tr>
<tr><td><input type="submit" value="Vložit" /></td></tr>
</table>
</form>
```

Každý uživatel může zobrazit a ohodnotit fotografii, kterou vložili jeho přátelé nebo on. Hodnocení jednotlivých fotografií je ve tvaru „Líbí se mi/Nelíbí se mi“. Pokud uživatel ohodnotí některou fotografii a poté si ji znovu zobrazí, již nemůže znovu fotografii ohodnotit, ale má pod fotografií zobrazeno své hodnocení.

Dále se v práci s fotografiemi pracuje tak, že uživatelé vkládají své profilové fotografie a vkládají i fotografie do příspěvků. Tyto fotografie se ukládají na server a následně jsou fotografie při jejich zobrazování upraveny v kódu aplikace na požadovanou zobrazovací velikost. Dále se s fotografiemi provádí operace mazání ze serveru v případě, že uživatel odstraní svou profilovou fotografii.

5.10. Odzkoušení aplikace v různých internetových prohlížečích

Aplikace byla zkoušena v průběhu celého vývoje v různých internetových prohlížečích za účelem zajištění její funkčnosti v nejčastěji používaných prohlížečích. Problémy byly průběžně odstraňovány. Po dokončení vývoje byla aplikace znovu odzkoušena v následujících prohlížečích a je v nich plně funkční a bezpečná:

- Mozilla Firefox (verze 3.6.16)
- Internet Explorer (verze 9.0.8112.16421)
- Opera (verze 11.10)
- Safari (verze 5.0.5)
- Google Chrome (verze 10.0.648.205)

Závěr

Cílem práce bylo vytvořit umělou sociální síť, která bude využívána při výuce předmětu Sociální síť na Fakultě ekonomicko-správní Univerzity Pardubice. Práce zahrnuje popis nejdůležitějších částí jazyků PHP, JavaScriptu, SQL a HTML, které byly využity při tvorbě praktické části práce. Při tvorbě sociální sítě byla navržena databáze, do které se ukládají důležité informace o uživateli, o jejich aktivitě, příspěvky které uživatelé vložili a také přátelské vztahy, které mají jednotliví uživatelé mezi sebou. Tato databáze byla navržena na všech třech úrovních modelování a její implementace byla provedena na databázi MySQL.

Sociální síť byla vytvořena jako internetová stránka, ke které je možno přistupovat z jakéhokoli počítače s připojením k internetu a s nainstalovaným internetovým prohlížečem. Funkcemi této sociální sítě jsou registrace a přihlašování uživatelů, vyhledávání a přidávání kamarádů, vkládání příspěvků a fotografií, hodnocení vložených fotografií a další. Práce je doplněna obrázky z aplikace, diagramy, tabulkami a částmi kódu, kterým je aplikace tvořena. Vytvořená aplikace je plně funkční v pěti nejpoužívanějších internetových prohlížečích a je přiložena k této práci na CD.

Seznam obrázků

Obrázek 1 - E-R diagram [zdroj: vlastní]	25
Obrázek 2 - Relace vzniklé z entit [zdroj: vlastní]	26
Obrázek 3 - Relace vztahu [zdroj: vlastní]	27
Obrázek 4 - Relační schéma [zdroj: vlastní]	27
Obrázek 5 - Vytvoření tabulky v phpMyAdmin [zdroj: vlastní].....	28
Obrázek 6 - Práce v phpMyAdmin [zdroj: vlastní]	28
Obrázek 7 - Propojení tabulek v phpMyAdmin [zdroj: vlastní].....	32
Obrázek 8 - Use Case diagram [zdroj: vlastní].....	33
Obrázek 9 - Grafické rozložení aplikace, osobní strana [zdroj: vlastní]	34
Obrázek 10 - Registrace nového uživatele [zdroj: vlastní].....	35
Obrázek 11 - Vyhledání kamaráda [zdroj: vlastní]	38
Obrázek 12 - Strana kamaráda [zdroj: vlastní].....	40
Obrázek 13 - Úprava osobních údajů [zdroj: vlastní].....	41

Seznam tabulek

Tabulka 1 - Přístupy k jednotlivým indexům pole \$_FILES [2].....	19
Tabulka 2 - Logické operátory [11].....	20
Tabulka 3 - Databázová tabulka Akce [zdroj: vlastní].....	29
Tabulka 4 - Databázová tabulka Uživatelé [zdroj: vlastní].....	30
Tabulka 5 - Databázová tabulka Vztahy [zdroj: vlastní].....	31
Tabulka 6 - Databázová tabulka Příspěvky [zdroj: vlastní]	31

Použitá literatura

- [1] BERÁNEK, Ladislav. *Síťová analýza v marketingu* [online]. 2008, [cit. 4. 4. 2011]. Dostupné z <<http://znanosti2008.fiit.stuba.sk/download/articles/znanosti2008-Beranek.pdf>>
- [2] BRÁZA, Jiří. *PHP 5 začínáme programovat*. První vydání. Praha: GRADA Publishing, 2005. 244s. ISBN 80-247-1146-X
- [3] DUDEK, Jan. *Interval.cz* [online]. 26. 04. 2003, [cit. 3. 4. 2011]. Superglobální proměnné v PHP. Dostupné z <<http://interval.cz/clanky/superglobalni-promenne-v-php/>>
- [4] GRUDL, David. *La Trine* [online]. 31. 8. 2004, [cit. 2. 4. 2011]. Jak na formuláře: POST a GET. Dostupné z <<http://latrine.dgx.cz/jak-na-formulare-post-a-get>>
- [5] JANOVSÝ, Dušan. *Jak psát web* [online]. [cit. 2. 4. 2011]. Formuláře v HTML. Dostupné z <<http://www.jakpsatweb.cz/formulare-html.html>>
- [6] JANOVSÝ, Dušan. *Jak psát web* [online]. [cit. 2. 4. 2011]. Úvod do JavaScriptu. Dostupné z <<http://www.jakpsatweb.cz/javascript/javascript-uvod.html>>
- [7] KOSEK, Jiří. *PHP - tvorba interaktivních internetových aplikací*. Podrobný průvodce. První vydání. Praha: GRADA Publishing, 1999. 492 s. ISBN 80-7169-373-1
- [8] MOLNÁR, Zdeněk. *Možnosti využití sociálních sítí v Competitive Intelligence* [online]. 2010, [cit. 18. 3. 2011]. Dostupné z <<http://si.vse.cz/archive/proceedings/2010/molnar-moznosti-vyuziti-socialnich-siti-v-competitive-intelligence.pdf>>
- [9] *Sociální sítě* [online]. 25. 11. 2008, [cit. 30. 10. 2010]. Historie sociálních sítí. Dostupné z <http://socialnisite.cz/info/historie_socialnich_siti>
- [10] ŠIMONOVÁ, Stanislava; PANUŠ, Jan. *Databázové systémy I*. První vydání. Pardubice: Univerzita Pardubice, 2007. 106 s. ISBN 978-80-7194-988-6

[11] ŠKULTÉTY, Rastislav. JavaScript. Programujeme internetové aplikace. 2. aktualizované vydání. Brno: Computer Press, 2004. 224 s. ISBN 80-251-0144-4

[12] ZAJÍC, Petr. *Linuxsoft.cz* [online]. 8. 10. 2004, [cit. 8. 4. 2011]. PHP (53) - Sessions. Dostupné z <http://www.linuxsoft.cz/article.php?id_article=440>