|  |
| --- |
| FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY UNIVERZITA KOMENSKÉHO |
| ANALÝZA TECHNOLÓGIÍ, DEKOMPOZÍCIA A DÁTOVÝ MODEL |
| Web občianskeho združenia FMFI a exkurzie |
|  |
| **Tutifruty** |
| Štefan Horváth, Matej Toma, Martin Kloss, Milan Michalec, Zuzana Záhorszká |
| **20.10.2012** |

|  |
| --- |
| Tento dokument popisuje možné prístupy riešenia projektu z hľadiska použitých technológií a postupov. Okrem toho analyzuje a navrhuje rozdelenie projektu na jednotlivé časti, ktoré je možné riešiť nezávislé, popisuje ich vzájomné rozhrania a interakcie v diagramoch. Dokument obsahuje podrobný návrh dátového modelu (tabuľky, vzťahy medzi nimi a kľúče). |

Obsah

[1 Možné použité technológie 2](#_Toc338975536)

[1.1 Popis k jednotlivým technológiám 2](#_Toc338975537)

[1.1.1 PHP 2](#_Toc338975538)

[1.1.2 MySQL 2](#_Toc338975539)

[1.1.3 Codeigniter(PHP Framework) 2](#_Toc338975540)

[1.1.4 HTML 3](#_Toc338975541)

[1.1.5 CSS 3](#_Toc338975542)

[1.1.6 Java Script 3](#_Toc338975543)

[1.1.7 JQuery 3](#_Toc338975544)

[2 Komponentný diagram 4](#_Toc338975545)

[2.1 Komponent autentifikácia užívateľa 5](#_Toc338975546)

[2.2 Komponent správa členov 5](#_Toc338975547)

[2.3 Komponent schválovanie členov 5](#_Toc338975548)

[2.4 Komponent platby 5](#_Toc338975549)

[2.5 Komponent hromadná korešpondencia 5](#_Toc338975550)

[2.6 Komponent správa článkov 5](#_Toc338975551)

[2.6.1 Statické články 5](#_Toc338975552)

[2.6.2 Dynamické články 5](#_Toc338975553)

[2.7 Komponent import/export údajov 6](#_Toc338975554)

[2.8 Komponent kalendár 6](#_Toc338975555)

[2.9 Komponent správa exkurzií 6](#_Toc338975556)

[2.10 Komponent login/logout exkurzií 6](#_Toc338975557)

[2.11 Komponent správa udalosti 6](#_Toc338975558)

[3 Databázový model 7](#_Toc338975559)

[4 Deployment diagram 9](#_Toc338975560)

[5 Domain model diagram 10](#_Toc338975561)

[6 História dokumentu 11](#_Toc338975562)

# Možné použité technológie

Pre náš projekt sme si vybrali technológiu PHP v spojení s MySQL databázou, net framework (code igniter), HTML a CSS, java script a jquery. Na základe požiadaviek od zadávateľa sme zvolili túto technológiu pre jej dostupnosť a jednoduché nasadenie. Ďalší dôvod bol, aby aplikácia bola prístupná pre širokú verejnosť prostredníctvom internetu a to pomocou prehliadaču. Táto technológia nie je limitovaná operačným systémom serveru a preto je možné túto technológiu nasadiť na akýkoľvek server.

Aplikácia je rozdelená na jednotlivé triedy (class) vďaka čomu sme si mohli rozdeliť projekt na celky a pracovať na ňom nezávislé.

## Popis k jednotlivým technológiám

### PHP

PHP je serverový skriptovací jazyk, ktorý je od roku 1994 jedným z najpoužívanejších pri tvorbe dynamických web stránok. PHP bola pôvodne skratka „**Personal Home Page**“, dnes sa už používa **Hypertext Processor**. Oproti svojím rovesníkom Javasriptu a CSS je rozdielny predovšetkým tým, že sa skrypt vykoná mimo užívateľského počítača a na výstup monitora príde len výsledok.

**Výhody PHP:**

* Veľmi účinné a elegantné spracovanie formulárov, dát a pošty
* Možnosť prístupu k súborom aj cookies
* Veľmi široké možnosti použitia
* Spolupráca s databázami

**Nevýhody PHP:**

* Na niektorých serveroch (hlavne neplatených) nemusí vôbec fungovať
* Nutnosť komplikovanej inštalácie na počítač správcu
* Zložitosť

### MySQL

MySQL je systém na správu relačných databáz. Tento systém využívajú napr. Wikipédia, Facebook alebo Twitter. Pre potreby projektu budeme používať verziu 5.

### Codeigniter(PHP Framework)

Codeigniter je PHP framework využívajúci MVC (Model, View, Controler) pattern. Je to kompaktný framework, ktorý v základnom balíku poskytuje iba najdôležitejšiu funkcionalitu, preto sa radí do kategórie rýchlych na naučenie. Obsahuje iba základnú vrstvu pre správu databázy postavenú na active record patterne. Avšak programátor môže používať aj klasické SQL dopyty, active record však poskytuje lepšiu a pohodlnejšiu prácu. Má detailnú dokumentáciu, ktorá je dostupná aj v offline verzii v rar archíve frameworku. Obsahuje veľa užitočných príkladov, ktoré sú nápomocné. Ak by ani toto nestačilo tak má širokú celosvetovú komunitu, ktorú je možné požiadať o pomoc, keď si človek nevie s niečim pomôcť. Vzhľadom na hore uvedené fakty Codeigniter patrí medzi frameworky, ktoré je ľahké sa naučiť za krátku dobu.

### HTML

Značkovací jazyk pre tvorbu najjednoduchšej varianty web stránky. Pre účel tohto projektu využijeme xHTML 1.0 transitional a HTML 5.0.

### CSS

Kaskádové štýly alebo CSS (Cascading Style Sheets) je všeobecné rozšírenie x[HTML](http://sk.wikipedia.org/wiki/HTML). Konzorcium [W3C](http://sk.wikipedia.org/wiki/W3C) označuje CSS ako jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov.

### Java Script

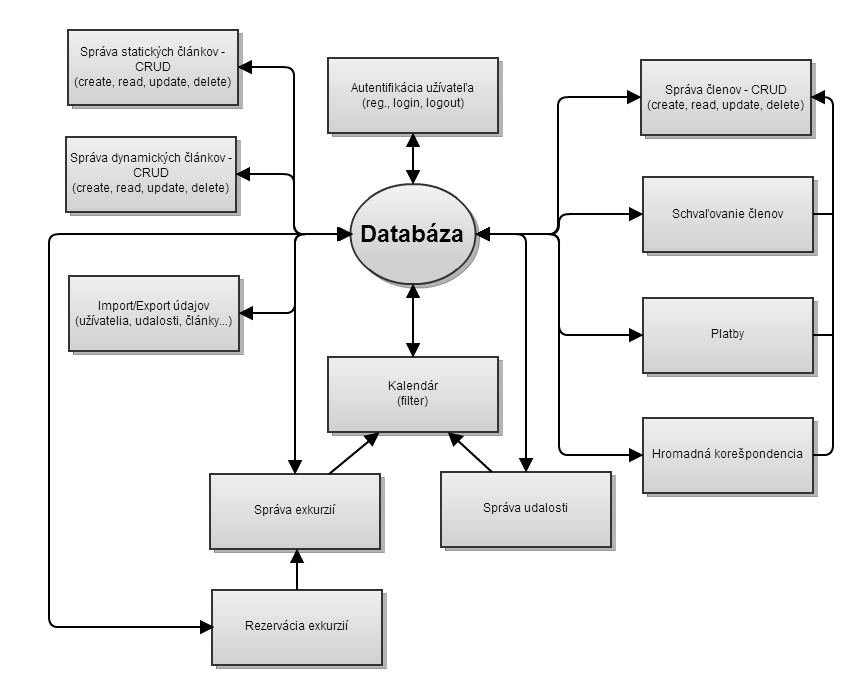
Java Script je jazyk, ktorý je súčasťou webových prehliadačov. Ako taký sa líši pre jednotlivé prehliadače a ich verzie. Používa sa najmä na animácie a zvýšenie interaktivity webových stránok. Existujú rôzne frameworky, my budeme využívať knižnicu JQuery.

### JQuery

jQuery je ľahká [cross-browser](http://sk.wikipedia.org/w/index.php?title=Cross-browser&action=edit&redlink=1) [JavaScript knižnica](http://sk.wikipedia.org/w/index.php?title=JavaScript_kni%C5%BEnica&action=edit&redlink=1), ktorá kladie dôraz na interakciu medzi JavaScriptom a HTML. jQuery je aktuálne najpoužívanejšia knižnica.

Je [slobodný](http://sk.wikipedia.org/wiki/Slobodn%C3%BD_softv%C3%A9r) a [otvorený software](http://sk.wikipedia.org/wiki/Open_source_software) pod [MIT](http://sk.wikipedia.org/wiki/MIT_licencia) licenciou. Jej syntax je navrhnutá pre jednoduchšiu navigáciu dokumentu, výber [DOM](http://sk.wikipedia.org/wiki/Document_Object_Model) elementov, vytváranie [animácií](http://sk.wikipedia.org/w/index.php?title=Animation&action=edit&redlink=1), spracovanie udalostí, a vývoj [Ajax aplikácií](http://sk.wikipedia.org/wiki/Ajax_(informatika)). jQuery tiež poskytuje možnosti pre vývojárov na vytváranie pluginov postavených na tejto JavaScript knižnici.

# Komponentný diagram



Obrázok 1 – Komponent diagram

## Komponent autentifikácia užívateľa

Komponent dôležitý pre registráciu, prihlásenie a odhlásenie užívateľa. Najprv treba registrovať do systému a čakať na schválenie, komponent kontroluje, či všetky povinné polia sú správne vyplnené. Po schválení je možné prihlásiť. Úlohou komponentu je overovanie hesla, ktoré je zadane pri prihlásovaní do systému. Ak bolo heslo správne zadané, použivateľovi je povolený prístup k možnostiam, ktoré sú prístupné iba po prihlásení. Po prihlásení je aktívná možnosť odhlásenie.

## Komponent správa členov

Komponent je prístupný pre prihláseného člena.

* Create: Vytvoriť člena môže návštevník s registráciou, alebo administrátor môže manuálne pridať člena do databázy.
* Read: Z databázy sa vyberú uložené informácie, ktoré sa zobrazia.
* Update: Je možné meniť údaje. Zobrazí sa formulár ako pri registrácií, z databázy sa vyberú uložené informácie. Po potvrdení zmeny údajov a ich skontrolovaní, sa uložia do databázy.
* Delete: Každého člena je možné odstrániť z databázy.

## Komponent schválovanie členov

Komponent je prístupný pre administrátora. Pri registrácii položka „schválený“ sa automaticky nastaví na „neschválený“. Administrátor môže, ale nemusí schváliť člena.

## Komponent platby

Platba môže byť: registračný, členský poplatok alebo dobrovoľný príspevok.Komponent kontroluje, či pri registrácií bolo odoslané registračný poplatok (v ďalších rokoch to bude členský poplatok), ktorý sa ukladá do databázy. Administrátor na základe variabilného symbolu priradí člena.

## Komponent hromadná korešpondencia

Po vyplnení formulára administrátor môže posielať informačné e-maily konkrétnym členom a potenciálnym členom. Podľa formulára budú vybraté vhodné e-mailové adresy z databázy.

## Komponent správa článkov

Úloha komponentu:

* Create: Administrátor môže vytvoriť článok, po správne vyplnení formulára údaje sa ukladajú do databázy.
* Delete: Adminstrátor môže zmazať článok.
* Read: Každý člen môže zobraziť článok. Po vybraní článka sa vypíše text z databázy.

### Statické články

Sú to články, ktoré sú pevne dané. Sú to hlavne články informačného charakteru s veľmi dlhou periódou zobrazenia, napr. stanovy, dôležité oznamy.

### Dynamické články

Sú to tipy článkov, ktoré vytvára administrátor a majú informačný charakter, napr. článok o nadchádzajúcich stretnutiach, novinky na webe, novinky so sveta a iné.

## Komponent import/export údajov

* Import údajov: Z kompatibilného súboru je možné importovať údaje do databázy.
* Export údajov: Po zvolení adresy na vytvorenie súboru je možné exportovať údaje z databázy.

## Komponent kalendár

Do kalendára sa ukladajú udalosti, aj exkurzie ako udalosti. Je možné filtrovať, napr. odkedy-dokedy chceme udalosti zobraziť a z databázy budú vyťahované vhodné udalsoti so všetkými údajmi.

## Komponent správa exkurzií

Úloha komponentu:

* Create: Administrátor môže vytvoriť exkurziu, po správne vyplnení formulára údaje sa ukladajú do databázy. Aj v kalendári sa objaví nová exkurzia.
* Update: Administrátor môže meniť údaje exkurzie. Z databázy sa zobrazia uložené údaje a uložia sa nové údaje.
* Delete: Adminstrátor môže zmazať exkurziu. Z databázy tým odstráni všetky údaje o exkurzie. Odstráni exkurziu aj z kalendára.
* Read: Každý kto ma prístup na stránku združenia môže zobraziť údaje o exkurzie alebo konkrétnej udalosti v kalendári. Po vybraní exkurzie/udalosti sa vypíšú vhodné údaje z databázy.

## Komponent login/logout exkurzií

Každý prihlásený používateľ sa môže prihlásiť alebo odhlásiť z exkurzie a spolu so svojou osobou nahlásiť aj počet sprevádzaných ľudí. Pri prihlásení sa skontroluje obsadenosť. Ak je tam ešte požadovaný počet voľných miest, tak prihlásenie sa uskutoční a do databázy uložíme nové údaje o obsadenosti. Pri odhlásení tak isto meníme obsanenosť v datebáze.

## Komponent správa udalosti

Úloha komponentu:

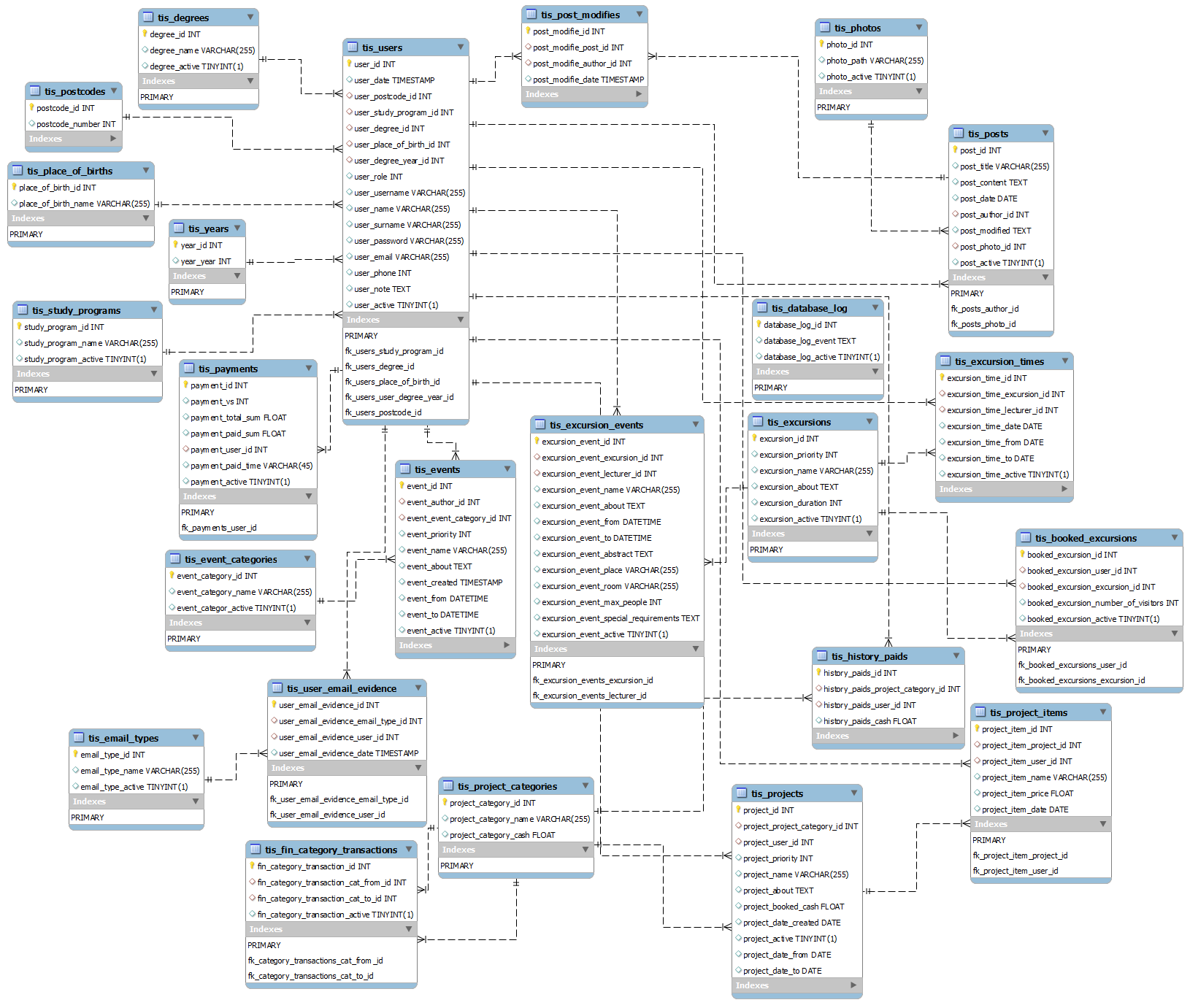
* Create: Administrátor môže vytvoriť udalosť, po správne vyplnení formulára údaje sa ukladajú do databázy. Aj v kalendári sa objaví nová udalosť.
* Update: Administrátor môže meniť údaje udalosti. Z databázy sa zobrazia uložené údaje a uložia sa nové údaje.
* Delete: Adminstrátor môže zmazať udalosť. Z databázy tým odstráni všetky údaje o udalosti. Odstráni udalosť aj z kalendára.
* Read: Každý návštevník stránky môže zobraziť údaje o udalosti. Po vybraní udalosti sa vypíšú vhodné údaje z databázy.

# Databázový model

Databázový model bol vyhotovený tak aby reálne pokryl všetky situácie, ktoré môžu nastať počas behu aplikácie a zohľadnuje vzťahy medzi udalosťami, ktoré počas behu aplikácie môžu nastať a je na ne pripravený. Za základnú tabuľku možno označiť tabuľku „users“, ktorá uchováva všetky informácie o konkrétnom používateľovi. Za zaujímavé tabuľky možno označiť tabuľky zaoberajúce sa správou projektov a vecí ohľadne projektov. Projekty sú členené do kategórií, podľa toho kvôli čomu boli vytvorené. Projekt sa môže nachádzať vo viacerých stavoch. A to napríklad schválený, vytvorený ale zatiaľ neschválený, ukončený. Tieto stavy sa zaznamenávaju podľa hodnoty v stĺpci „active“. Každý projekt vedie o sebe informácie za akým účelom bol vytvorený, kto ho vytvoril a taktiež predpokladané finančné náklady na jeho realizáciu a časové obdobie na ktoré bol vytvorený. Pre každý projekt sa vedie evidencia nákupov jednotlivých čiastkových položiek, ktoré sú potrebné pre jeho celkovú realizáciu. Tieto položky sa viažu na konkrétny projekt, je známe meno položky, celková suma a ID používateľa, ktorý takýto nákup uskutočnil. Tieto finančné prostriedky sa čerpajú z peňazí, ktoré boli na tento projekt vyčlenené. Po ukončení projektu sa zráta celková suma, ktorá bola vynaložena na daný projekt a ak zostanú nevyčerpané peniaze, tak tieto peniaze sa vrátia späť do spoločných peňazí určených na danú kategóriu pod ktorú patrí daný projekt. Peniaze medzi kategóriami je možné presúvať, keď nastane potreba združenia presunúť akútne peniaze na inú kategóriu z rôznych dôvodov. Tieto tranzakcie sa evidujú takže je jednoduché zistiť z kade dané peniaze prišli, aby bolo jednoduché ich vrátiť späť. Taktiež za zmienku , kvôli svojej významnosti stojí tabuľka „database\_log“, lebo táto tabuľka obsahuje všetky zmeny, ktoré boli vykonané v databázi od insertov, updatov až po delete operácie.

Príklad relácií:

* Projekt je v relácií ku kategórií vo vzťahu N:1, lebo n-projektov môže patriť pod jednu kategóriu ale zároveň každý projekt môže byť začlenený maximálne v jednej kategórií
* Projekt v relácii ku položkám projektu je vo vzťahu 1:N, lebo práve n-položiek bolo vyúčtovaných na daný, konkrétny projekt.
* Údaje typu miesto narodenie, poštové smerové číslo, štúdijný program a ďaľšie je vo vzťahu ku tabuľke „users“ vo vzťahu 1:N. Lebo daný používateľ sa narodil na práve jednom mieste, absolvoval práve jeden štúdijný program alebo jeho adresa obshauje práve jedno poštové smerové číslo a na strane druhej práve n-používateľov sa narodilo na práve jednom mieste. Keďže takéto údaje sa s veľkou pravdepodobnosťou opakujú tak sú vyčlenené do samostatných tabuľiek, čo zabezpečí ich rýchlejšiu úpravu ak je potrebná, lebo stačí nám zmeniť jeden údaj a zmenia sa všetky požadované, keďže iné tabuľky sa na tieto hodnoty odkazujú pomocou cudzích kľučov.



Obrázok 2 - Dátový model aplikácie

Konvencie:

* Jazyk: všetky názvy tabuliek a názvy stĺpcov sú uvedené v anglickom jazyku
* Názvy tabuliek: tabuliek začínajú prefixom "tis\_“ a následne pokračujú pomenovaním v množnom čísle, ktoré hovorí o tom čo daná tabuľka zabezpečuje
* Názvy stĺpcov: názvy stĺpcov začínajú prefixom, ktorý je odvodený z názvu tabuľky ale v jednotnom čísle a nasledujú pomenovaním stĺpca
* Cudzie kľúče: začínajú prefixom „fk\_“ pokračujú názvom tabuľky a končia názvom stĺpca z tabuľky odkial pochádza
* Názvy s postfixom „\_active“ – daný element v tab. Sa zobrazuje, alebo nezobrazuje

- pre usera: 0 – v databáze sa vyskytuje, ale nemá možnosť sa prihlásiť

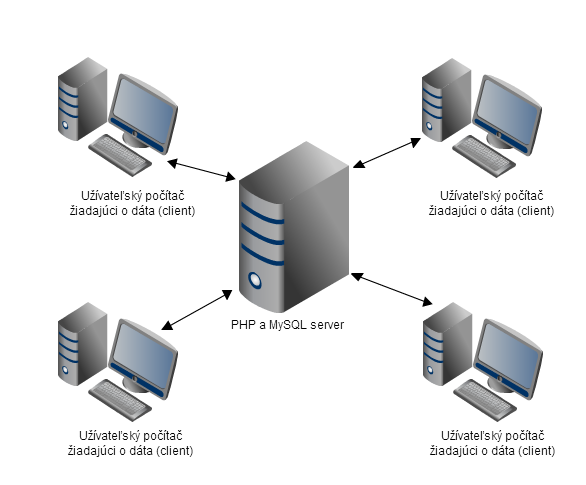
1 – shválený, prihlasuje sa

2 – expiračná doba, nachádza sa v čase kedy si môže znova obnoviť členské

- project: 0 – neschválený

1 – schválený

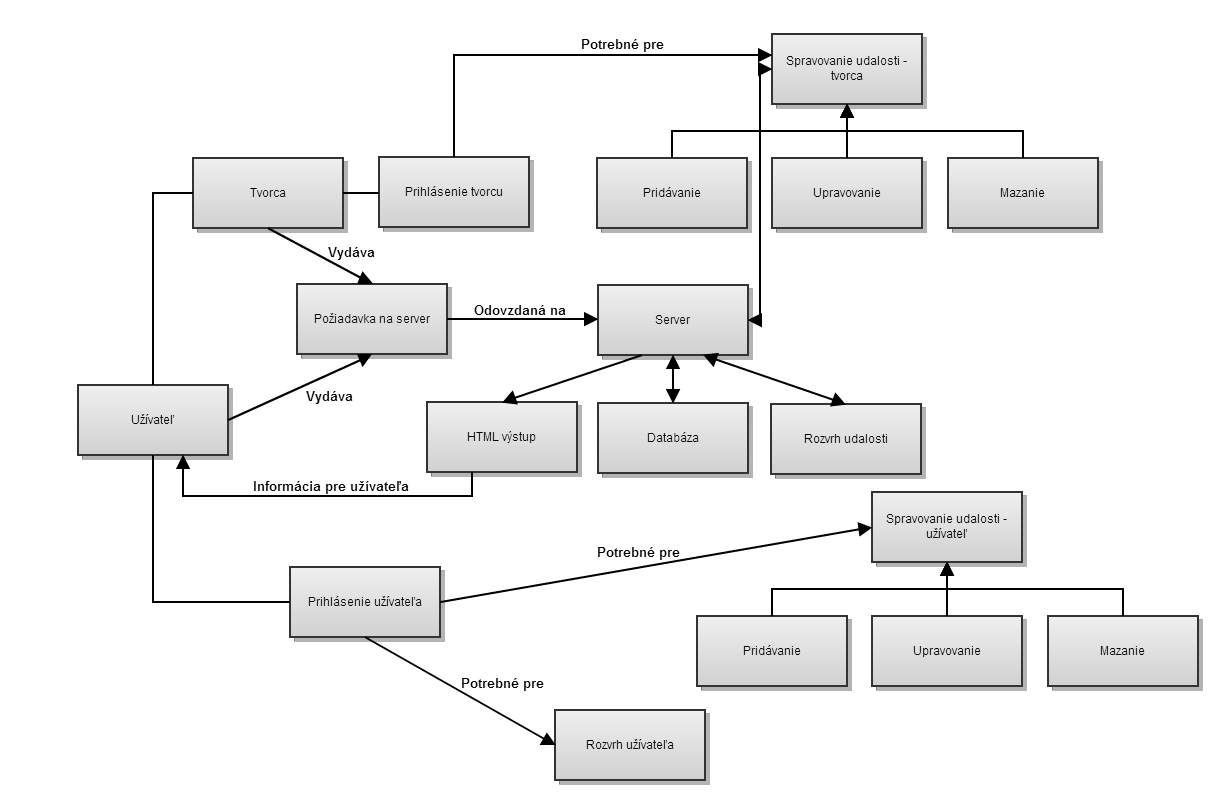
# Deployment diagram



Obrázok 3 – Deployment diagram

Aplikácia beží na serveri, ktorý podporuje PHP a MySQL. Používateľ má k aplikácií prístup pomocou webového prehliadaču, ktorý vykonáva dopyty na aplikáciu – http GET a POST. Pre úplné fungovanie stránky musí mať používateľ povolený javascript, v prípade blokovania javascriptu sa zobrazí upozornenie, že stránka nemusí pracovať korektne.

# Domain model diagram



Obrázok 4 – Domain model diagram

(Model odkopírovaný s minuloročného projektu v inej grafike)

# História dokumentu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verzia | Dátum | Autor | Popis zmeny Schválil | |
| 0.1 | 20.10.2012 | Matej Toma, Štefan Horváth, Martin Kloss, Milan Michalec | Úvodná verzia dokumentu |  |
| 0.2 | 21.10.2012 | Zuzana Záhorszká | Doplnenie prvej verzie dokumentu. Doplnený popis komponentného diagramu, dátového modelu |  |
| 0.3 | 23.10.2012 | Martin Kloss, Matej Toma | Úprava komponent diagramu a dátového modelu, pridanie domain model diagramu, pridané popisy obrázkov |  |