

# **Webový portál pro administraci projektů**

Web portal for project administration

Bc. Martin Šuránek

---

Diplomová práce  
2010



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Martin ŠURÁNEK**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Webový portál pro administraci projektů**

Zásady pro vypracování:

1. Analýza problematiky a vypracování literární rešerše na dané téma.
2. Návrh struktury databáze systému a webových stránek.
3. Vytvoření aplikace a popis postupu řešení.
4. Implementace systému.
5. Vytvoření podpory uživatelů – helpdesk.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. PROKOPOVÁ, Zdenka. Databázové systémy MySQL+PHP. FAI UTB Zlín, 126 s. , 2006, Vysokoškolská skripta. ISBN 80-7318-486-9.
2. JESUS, Castagnetto, et al. Programujeme profesionálně : PHP. Robert FE Shaw. 2004. autoriz. vyd. Praha : Computer Press, 2004. 656 s., CD-ROM. ISBN 80-7226-310-2.
3. JavaScript a Ajax : Moderní programování webových aplikací. 1. vyd. Praha: Computer Press, a.s., 2007. 360 s. ISBN 978-80-251-1824-5.
4. LACKO, Luboslav. SQL: hotová řešení. Praha : Computer press, 2003. 296 s. ISBN 80-7226-975-5.
5. HAVLENKA, J. a kol. Vytváříme WWW stránky a spravujeme moderní web site. Computer Press, 2006, ISBN: 80-251-0801-5.
6. KOSEK, J. HTML – tvorba dokonalých WWW stránek. Grada Publishing, 2006, ISBN: 80-7169-608-0.
7. VALADE, J. PHP5 for dummies. Wiley Publishing, 2004, ISBN: 0-7645-4166-8.

Vedoucí diplomové práce:

**doc. Ing. Zdenka Prokopová, CSc.**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

**19. února 2010**

Termín odevzdání diplomové práce:

**8. června 2010**

Ve Zlíně dne 19. února 2010

prof. Ing. Vladimír Vašek, CSc.

*děkan*



prof. Ing. Vladimír Vašek, CSc.

*ředitel ústavu*

## **ABSTRAKT**

Diplomová práce se v úvodu zabývá analýzou nově zřízené interní grantové agentury (IGA) na Univerzitě Tomáše Bati. Pro tyto potřeby bylo nutno vytvořit nový webový portál pro administraci projektů. Webový portál má za úkol usnadnit správu této interní grantové agentury. Uveden je i popis problematiky tvorby zadaného webového portálu. Vytvořený portál zachovává MVC model a k jeho tvorbě byl použit NetteFramework, který je v teoretické části práce popsán, stejně jako databázový layer Dibi. Nachází se zde i stručný rozbor software, který byl při tvorbě portálu používán. Dále je popsána struktura databáze a struktura několika tabulek v ní obsažených. Část práce je věnována implementaci a testování portálu. V diplomové práci je uveden manuál k vytvořenému webovému portálu, který má uživatelům práci s výsledným portálem usnadnit.

Klíčová slova: Webový portál, web, server, PHP, MySQL, JavaScript.

## **ABSTRACT**

Diploma thesis in the introduction deals with the analysis of the newly established Internal Grant Agency (IGA) at Tomas Bata University. For these purposes it was necessary to create a new web portal for project administration. The web portal is aimed at facilitating the management of the Internal Grant Agency. Is given a description of the problem creating the specified Web site. Web site created and maintains the MVC model was used to create his NetteFramework, which is in the theoretical part of the work described, as well as the database layer Dibi. There is also a brief analysis software that was used in creating the portal. Furthermore, the structure of the database structure and several tables contained in it. Part is devoted to implementation and testing site. The thesis is set to manual created a Web portal that has the users to work with the resulting portal easier.

Keywords: Web portal, web, server, PHP, MySQL, JavaScript.

Zde bych rád poděkoval vedoucí diplomové práce, paní doc. Ing. Zdence Prokopové CSc. za vstřícnost a rady, které mě poskytovala v průběhu vypracování této diplomové práce.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

**Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 ANALÝZA PROBLEMATIKY</b> .....	<b>12</b>
1.1 ZÁSADY STUDENTSKÉ GRANTOVÉ SOUTĚŽE UNIVERZITY TOMÁŠE BATI VE ZLÍNĚ (SMĚRNICE REKTORA) .....	12
1.1.1 Účelová podpora na specifický vysokoškolský výzkum .....	12
1.1.2 Studentská vědecká konference.....	12
1.1.3 Interní granty na podporu vědeckovýzkumné činnosti.....	13
1.1.4 Grantové přihlášky, jejich hodnocení a schválení.....	14
1.1.5 Komise IGA .....	17
1.1.6 Fakultní hodnotící komise.....	18
<b>2 POUŽITÝ SOFTWARE</b> .....	<b>19</b>
2.1 NETTE FRAMEWORK.....	19
2.1.1 Model-View-Presenter (MVP).....	19
2.2 DATABÁZOVÝ LAYER DIBI .....	20
2.3 EDITACE CSS A JS .....	21
2.3.1 PSPad .....	21
2.4 EDITACE PHP.....	22
2.4.1 Eclipse Galileo .....	22
2.5 LADĚNÍ SQL DOTAZŮ.....	23
2.5.1 SQLyog .....	23
2.6 PRÁCE S GRAFIKOU .....	24
2.6.1 PhotoFiltre.....	24
<b>II PRAKTICKÁ ČÁST</b> .....	<b>26</b>
<b>3 NÁVRH STRUKTURY</b> .....	<b>27</b>
3.1 DATABÁZE .....	27
3.1.1 Schéma relací v databázi .....	27
3.1.2 Tabulka s_setings .....	28
3.1.3 Tabulka faculties .....	29
3.1.4 Tabulka workplace .....	29
3.1.5 Tabulka student .....	30
3.1.6 Tabulka study_course.....	30
3.1.7 Tabulka study_program.....	30
3.1.8 Tabulka users.....	31
3.1.9 Tabulka project.....	32
3.1.10 Tabulka opponency .....	33
3.1.11 Tabulka committee_result .....	35
3.2 APLIKAČNÍ ČÁST.....	35
3.2.1 Presentery (Controls).....	35
3.2.2 Modely (Models).....	36

3.2.3	Pohledy (Views).....	36
3.3	ADMINISTRAČNÍ ČÁST .....	37
3.3.1	Presentery (Controls).....	37
3.3.2	Modely (Models).....	38
3.3.3	Pohledy (Views).....	38
3.4	PODPORA UŽIVATELŮ (HELPDESK) .....	40
3.4.1	Pracovní pokyny v systému.....	40
3.4.2	Manuál k systému.....	40
<b>4</b>	<b>POSTUP ŘEŠENÍ .....</b>	<b>41</b>
4.1	APLIKAČNÍ ČÁST.....	41
4.1.1	Registrace uživatelů .....	41
4.1.2	Přihlašování.....	41
4.1.3	Tvorba projektů .....	41
4.1.4	Správa projektů .....	42
4.1.5	Oponentura.....	42
4.1.6	Schvalování projektů.....	43
4.2	ADMINISTRAČNÍ ČÁST .....	43
4.2.1	Správa tabulek v databázi.....	44
4.2.2	Přihlašování.....	44
4.2.3	Správa administrátorů .....	44
4.2.4	Správa fakult .....	44
4.2.5	Správa pracovišť .....	45
4.2.6	Správa studijních programů a oborů .....	45
4.2.7	Správa registrovaných uživatelů .....	45
4.2.8	Správa komisí.....	45
4.2.9	Správa projektů .....	46
4.3	PODPORA UŽIVATELŮ – HELPDESK.....	46
4.3.1	Manuál k systému.....	46
4.3.2	Pracovní pokyny v systému.....	56
<b>5</b>	<b>IMPLEMENTACE A TESTOVÁNÍ SYSTÉMU.....</b>	<b>57</b>
5.1	IMPLEMENTACE SYSTÉMU.....	57
5.1.1	Requirements-Checker .....	57
5.2	TESTOVÁNÍ SYSTÉMU .....	58
5.2.1	Zachytávání chyb a výjimek.....	58
5.2.2	Logování chyb.....	59
	<b>ZÁVĚR .....</b>	<b>60</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>61</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>62</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK .....</b>	<b>63</b>
	<b>SEZNAM OBRÁZKŮ .....</b>	<b>64</b>
	<b>SEZNAM TABULEK.....</b>	<b>65</b>



## ÚVOD

Cílem diplomové práce a mým úkolem byla tvorba webového portálu pro správu projektů. Portál byl vytvořen pro usnadnění správy nově vzniklé interní grantové agentury na Univerzitě Tomáše Bati.

Nejdříve je potřeba provést analýzu problematiky proto se na úvod této diplomové práce seznámíme s plným zněním směrnice rektora, která pojednává o zásadách studentské grantové soutěže Univerzity Tomáše Bati ve Zlíně. Tato směrnice určuje průběh a pravidla interní grantové soutěže a je tedy proto pro tvorbu portálu stěžejním dokumentem. Součástí analýzy problematiky je také popis použitého software. Dále se dostaneme k popisu NetteFrameworku na kterém je vzniklý portál postaven, jedná se o framework pro tvorbu webových aplikací v PHP 5. Zaměřuje se na eliminaci bezpečnostních rizik, podporuje AJAX, DRY, KISS, MVC a znovupoužitelnost kódu. Využívá událostmi řízené programování a z velké části je založen na použití komponent. Původním autorem Nette Frameworku je David Grudl, o jeho další rozvoj se stará organizace Nette Foundation. Nette Framework je svobodný software, nabízený pod licencemi GNU GPL a licenci Nette. Další část analýzy je věnována PHP databázovému layeru Dibi, který se snaží zjednodušit zápis SQL příkazů a ulehčit rutiny, se kterými se programátor běžně setkává, např. získání výsledku jako dvourozměrné pole, import/export SQL souboru, vytvoření asociativního pole ze dvou sloupců apod. Dibi má implementovanou ochranu proti SQL injection, je napsána v objektovém PHP 5 a distribuuje se též v jednosouborové verzi. Obsahuje ovladače pro práci s 8 typy databází - MySQL, MySQLi, PostgreSQL, SQLite, ODBC a experimentálně MS SQL, Oracle a PDO. Následně přistoupíme k praktické části, která je rozdělená na tři hlavní kapitoly. První kapitolou je návrh struktury, druhou postup řešení a nakonec třetí kapitola, která pojednává o implementaci a testování vytvořeného systému.

Návrh struktury se zabývá strukturou databáze a několika tabulek v ní obsažených. Pro představu je zde uvedeno schéma relací v databázi, na kterém je krásně vidět propojení jednotlivých tabulek. Dále přejdeme ke struktuře aplikace, která je založena na MVC modelu. Jsou zde přehledně uvedeny jednotlivé části MVC modelu aplikace jak pro aplikační, tak i pro administrační část portálu. Tzn. Přehled modelů (Model), pohledů (View) a presenterů (Control). Kapitola návrhu struktury je zakončena popisem struktury podpory uživatelů tedy Helpdesku. Ten je rozdělen na dvě části, první z nich jsou pracovní

pokyny v systému, druhou částí je manuálu, který byl pro usnadnění práce s portálem vytvořen.

Postup řešení je rozdělen do tří kapitol. Jedná se o aplikační část, administrační část a podporu uživatelů, tedy Helpdesk. Aplikační část je věnována registraci uživatelů, kteří budou portál využívat a jejich přihlašování do portálu. Asi nejdůležitější částí je ale popis tvorby projektů a jejich správu. V neposlední řadě je popsán proces oponentury k projektům a také jejich schvalování komisí IGA. Administrační část popisuje postup řešení správy tabulek v databázi, dále je věnována pozornost správě administrátorských účtů a možnosti přidávání nových fakult a nových pracovišť jim podléhajícím. Na zapojení studentů doktorandského studia do grantové soutěže nebylo zapomenuto, a proto skýtá administrace možnost správy studijních programů a oborů na vytvořených fakultách. Samozřejmostí byla nutnost implementovat správu registrovaných uživatelů. Na závěr této části je uveden postup řešení správy komisí a to nejdůležitější, správa projektů vytvořených v systému. Kapitola pojednávající o postupu řešení je pak uzavřena manuálem k výslednému webovému portálu a objasněním řešení pracovních pokynů v systému.

Poslední kapitola diplomové práce je věnována implementaci a testování systému. V implementační části jsou popsány nároky výsledného portálu a popsán skript, který nám přehledně zobrazí, zda můžeme systém na vybraném serveru provozovat. V části testování systému jsem se věnoval popisu možností ladění systému jak ve vývojové, tak i v produkční fázi.

## **I. TEORETICKÁ ČÁST**

## 1 ANALÝZA PROBLEMATIKY

Vytvoření webového portálu bylo zadáno z důvodu jednodušší správy nově zřízené interní grantové agentury IGA. Z důvodu vyjasnění všech bodů zadání je v následující části uvedeno plné znění směrnice rektora [8], která zmíněné skutečnosti stanovuje.

### 1.1 Zásady studentské grantové soutěže Univerzity Tomáše Bati ve Zlíně (směrnice rektora)

#### 1.1.1 Účelová podpora na specifický vysokoškolský výzkum

- Účelová podpora na specifický vysokoškolský výzkum je poskytována Univerzitě Tomáše Bati ve Zlíně (dále jen UTB) na příslušný kalendářní rok podle zákona č. 130/2002 o podpoře výzkumu, experimentálního vývoje a inovací v platném znění a Pravidel pro poskytování účelové podpory na specifický vysokoškolský výzkum vydaná MŠMT podle tohoto zákona.
- Algoritmus rozdělení přidělených prostředků je popsán v Rozpočtu UTB pro příslušný kalendářní rok.
- Pro realizaci IGA je zřízena jako odborný, řídicí i výkonný orgán Komise Interní grantové agentury (dále jen Komise IGA).

#### 1.1.2 Studentská vědecká konference

Granty na podporu studentské vědecké konference (SVK) jsou určeny na úhradu způsobilých nákladů spojených s organizací studentské grantové soutěže.

- Aktivními účastníky studentské vědecké konference jsou studenti DSP a MSP.
- O konání SVK rozhoduje děkan fakulty, fakulta také hradí náklady spojené s její organizací. Souhrn nákladů na všechny SVK v rámci UTB nesmí přesáhnout 10% poskytnuté účelové podpory.

Hodnotící komisi, která stanovuje pořadí tří nejlepších prací, jmenuje děkan fakulty.

Do 31. ledna následujícího kalendářního roku předloží garant ustanovený děkanem Komisi IGA závěrečnou zprávu o průběhu studentské vědecké konference.

### 1.1.3 Interní granty na podporu vědeckovýzkumné činnosti

V rámci finanční podpory jsou udělovány interní granty (dále IG) na podporu studentských projektů (dále jen SP). Doba řešení projektů je 1 až 3 roky. Po finanční stránce musí být projekt ukončen do 30. listopadu příslušného kalendářního roku.

Granty na podporu SP jsou určeny vědeckovýzkumným a tvůrčím projektům.

Řešitelem SP je student doktorského studijního programu (dále jen DSP) nebo akademický pracovník.

- Členy řešitelského týmu jsou studenti DSP nebo magisterského studijního programu (dále jen MSP) nebo akademičtí pracovníci nebo výzkumní pracovníci UTB. V případě, že je řešitelem SP student DSP, je vždy členem řešitelského týmu jeho školitel, který je garantem projektu doktoranda po stránce odborné i finanční, případně jiný akademický pracovník UTB pověřený oborovou radou příslušné součásti, není-li školitel z UTB. Počet studentů v řešitelském týmu je alespoň roven počtu ostatních členů týmu. Garant SP může neúspěšné řešení projektu zastavit. V případě zastavení řešení projektu o využití zbývajících prostředků rozhodne Komise IGA.
- Za úspěšné řešení projektu se považuje publikace výsledků řešení započitatelných dle Metodiky Rady vlády pro vědu a výzkum (patent, článek, konferenční příspěvek apod.), případně v jiném odborném periodiku nebo plný text ve sborníku. Řešení projektu musí být doktorandem úspěšně ukončeno nejpozději k datu ukončení posledního roku studia, v závislosti na akreditačním rozhodnutí, nebo obhajoby disertační práce.

Časový průběh studentské grantové soutěže:

- Zásady studentské grantové soutěže jsou stanoveny a zveřejněny do 30. listopadu kalendářního roku předcházejícího roku realizace soutěže.
- Žádosti o granty SP jsou podávány na fakultách do 30. ledna roku realizace soutěže na oddělení vědy a výzkumu příslušné fakulty. Tam proběhne jejich posouzení a s doporučením Fakultní hodnotící komise jsou postoupeny Komisi IGA. Komise IGA také do 28. února roku realizace soutěže rozhodne o podporovaných studentských projektech a o využití nákladů na studentské konference.

### 1.1.4 Grantové přihlášky, jejich hodnocení a schválení

Každý navrhovatel může podat v daném roce jen jednu přihlášku SP, a to za předpokladu, že:

- Navrhovaný projekt není součástí navrhovatelem řešeného projektu podpořeného externí grantovou agenturou a navrhovatel není ani takového projektu spoluřešitelem.
- Navrhovaný projekt není zároveň řešen formou doplňkové činnosti nebo není součástí zakázkové činnosti.
- Navrhovatel je v řešitelském týmu maximálně jednoho dalšího projektu, přihlašovaného do IGA.
- Projekt navrhovatele, podpořený IGA v předchozích třech letech, nebyl hodnocen jako nevyhovující. Pokud nevyhovující hodnocení minulého projektu následovalo až po podání nového, je to důvodem k vyřazení přihlášky.

Podávání přihlášek

- Přihlášky musí být podány na formuláři, který je pro daný rok uveden společně s komentářem k vyplnění v upřesňujících podmínkách soutěže o IG. Všechny požadované položky formuláře musí být vyplněny.
- Grantové přihlášky se předkládají prostřednictvím fakult v elektronické formě do termínu stanoveného harmonogramem při vyhlášení soutěže o IG. Fakulta je povinna předat přihlášky Komisi IGA i v případě negativního vyjádření.
- Grantová přihláška má tyto náležitosti:
  - *jméno řešitele projektu, případně spoluředitelů,*
  - *název projektu a zdůvodnění návrhu projektu,*
  - *předpokládaná finanční podpora projektu (ve struktuře: osobní náklady včetně stipendií; věcné náklady) a odůvodnění požadované podpory*
  - *krátká anotace,*
  - *jméno a vyjádření školitele – je nutné vyplnit u SP podávaných studenty,*

- *vyjádření fakulty (součásti) k projektu a k požadované finanční podpoře, potvrzení případné finanční spoluúčasti fakulty.*
- Přihlášky s formálními nedostatky budou ještě před hodnocením vyřazeny.
- V rámci soutěže o IG je možné požadovat pouze způsobilé náklady, které jsou definovány v § 2 odst. 2 písm. 1) zákona o podpoře výzkumu, experimentálního vývoje a inovací a to:
  - osobní náklady nebo výdaje, včetně stipendií na výzkum a vývoj a inovace podle zákona o vysokých školách,
  - další provozní náklady nebo výdaje,
  - náklady nebo výdaje na služby,
  - doplňkové náklady nebo výdaje.
- Podpora projektu na jeden kalendářní rok může činit nejvýše 1 000 000 Kč.

### **Osobní náklady**

Z podpory projektu lze hradit stipendia pro studenty, kteří studují v prezenční formě studia. Podíl osobních nákladů nebo výdajů (včetně stipendií), spojených s účastí studentů DSP nebo MSP jakožto řešitelů nebo dalších členů řešitelského týmu na řešení SP, na celkových osobních nákladech nebo výdajích (včetně stipendií), hrazených v rámci způsobilých nákladů SP, činí více než 60%. Osobní náklady na studenta nesmí přesáhnout 50 tis. Kč za rok.

Odměny zaměstnancům, kteří se podílejí na řešení projektu; dohody o pracích konaných mimo pracovní poměr, které byly uzavřeny výhradně v přímé souvislosti s řešením projektu.

### **Věcné náklady**

Provozní náklady – lze hradit provozní náklady vzniklé v přímé věcné a časové souvislosti s řešením projektu, tj. náklady na materiál, zásoby a drobný hmotný a nehmotný majetek, dále náklady na odborné knihy a časopisy, na výrobky vzniklé při řešení projektu, na provoz a udržování majetku využívaného při řešení projektu, náklady na povinné

zákonné odvody. Náklady na služby zahrnují: náklady nebo výdaje na služby využívané v přímé souvislosti s řešením projektu, mezi tyto výdaje patří např. náklady na konzultační a poradenské služby využívané v přímé souvislosti s řešením projektu. Cestovné domácí i zahraniční, lze žádat pouze pro studenty, není možné hradit cestovné akademickým pracovníkům podílejícím se na řešení projektu. Z podpory nelze hradit náklady na jazykové kurzy.

### **Doplňkové náklady a výdaje**

Doplňkové (režijní) náklady (např. administrativní náklady, náklady na energie a služby, vč. kopírování a veškerých nákladů na užívání telefonu, faxu), tyto náklady nesmí přesáhnout 20% z celkových věcných nákladů.

Všechny finanční požadavky na přihlášce musí být dostatečně zdůvodněny.

- Pokud dojde u navrhovatele ke změnám nebo se navrhovatel rozhodne přihlášku stáhnout, musí tak učinit neprodleně v období výběrového řízení. Stažení projektu po jeho udělení je považováno za nevyhovující řešení, pokud Komise IGA nerozhodne na základě závažných důvodů o výjimce.

O změny v průběhu řešení projektů lze žádat nejpozději do 15. září příslušného kalendářního roku.

V průběhu řešení, není možné navýšit položky „Osobní náklady“ a „Doplňkové náklady nebo výdaje“.

### Hodnocení grantových přihlášek

- Základní kritéria pro odborné posouzení přihlášek projektů jsou:
  - *původnost a vědecká závažnost projektu,*
  - *aktuálnost a perspektiva projektu,*
  - *očekávaný přínos řešení, uplatnitelnost výsledků,*
  - *zpracování návrhu projektu a cíle řešení,*



- *koncepce a metodika,*
- *časový plán prací,*
- *přiměřenost a zdůvodněnost požadovaných finančních prostředků.*
- Každý projekt hodnotí 2 oponenti, z nichž jeden musí být mimo příslušnou fakultu. Projekty jsou hodnoceny v bodovém rozmezí 1- 10.
- Oponentům je za jejich činnost vyplácena odměna.
- Prioritu budou mít projekty interdisciplinární s mezifakultní spoluprací.
- Fakultní hodnotící komise na základě bodového hodnocení sestaví pořadí projektů a předá svůj návrh Komisi IGA. V případě rovnosti bodů bude pořadí určeno losem. Projekty jsou financovány dle pořadí do vyčerpání limitu stanoveného pro projekty příslušné fakulty.

Hodnocení výsledků projektů:

Výsledky řešení každého projektu podpořeného IGA jsou hodnoceny z hlediska odborného i finančního na základě závěrečné nebo dílčí zprávy projektu interním oponentním řízením. Oponentní řízení organizuje Fakultní hodnotící komise. Řešitel projektu předloží závěrečnou nebo dílčí zprávu do 10. ledna následujícího kalendářního roku na oddělení vědy a výzkumu příslušné fakulty v elektronické formě. Do 20. ledna předloží fakultní hodnotící komise zprávu o řešení projektu a zprávu z oponentního řízení.

Každý projekt musí mít publikační výstup. V publikačních výstupech musí být uvedeno, že práce byla uskutečněna za finanční podpory IGA s uvedením čísla projektu. Číslo projektu je řešiteli sděleno v dopise o přijetí projektu k financování. V případě opětovné žádosti o grant musí navrhovatel doložit publikační výstupy z předchozího grantu IGA.

### **1.1.5 Komise IGA**

Předsedou komise je prorektor pro tvůrčí činnosti, členy komise jsou proděkaní pro vědu a výzkum jednotlivých fakult.

Agendu IGA a Komise IGA vede pověřený pracovník referátu vědy a výzkumu UTB.

Jednání Komise IGA jsou důvěrná. Veřejně přístupné jsou zápisy z jednání komise, navrhovatelé se mohou seznámit s hodnocením svých projektů (beze jmen posuzovatelů).

Základní úkoly Komise IGA jsou:

- organizační zabezpečení realizace IGA,
- organizační zabezpečení realizace soutěže o IG podle čl. 2, shromáždění přihlášek projektů, zhodnocení přihlášek, sestavení jejich pořadí na základě stanoviska Fakultní hodnotící komise a vypracování návrhu rektorovi na udělení IG. Zajištění zhodnocení výsledků řešení projektů, na které byly uděleny IG,
- řešení stížností,
- přerozdělení vrácených prostředků v průběhu řešení projektů.

#### **1.1.6 Fakultní hodnotící komise**

Předsedou fakultní hodnotící komise je proděkan pro vědu a výzkum příslušné fakulty. Členy komise jsou 1 zástupce Akademického senátu fakulty a 1 akademický pracovník mimo orgány fakulty.

Základní úkoly Fakultní hodnotící komise:

- shromáždění přihlášek a organizace oponentního řízení projektů a předání podkladů Komisi IGA;
- oponentní řízení a hodnocení projektů, na které byly přiděleny prostředky a předání výsledků Komisi IGA.

## 2 POUŽITÝ SOFTWARE

### 2.1 Nette Framework

Frameworky jsou knihovny, které mají ulehčit práci při programování aplikace. To znamená méně psaní, přehlednější kód a rychlejší vývoj. V případě webových frameworků se často zmiňuje MVC, což je architektura rozdělující aplikaci na nezávislé vrstvy.

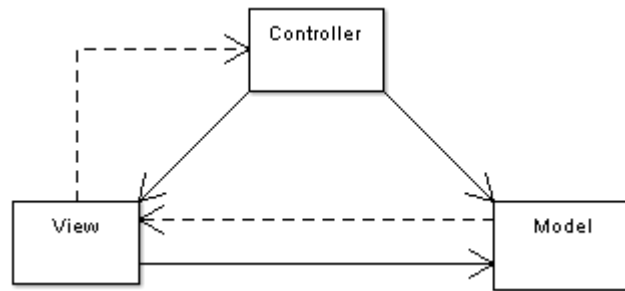
Nette Framework je napsaný v PHP 5 s plným využitím objektů (OOP). Ačkoliv vznikl už v roce 2004, teprve vloni byl uvolněn jako open source a zpřístupněn veřejnosti. Jeho licence, která vychází z BSD, patří k těm nejvolnějším. Vyrostla kolem něj jedna z nejaktivnějších komunit českých PHP vývojářů, ne-li nejaktivnější vůbec. Podle testu uveřejněném na serveru Root je jedním z nejvýkonnějších frameworků.

Framework je koncipován jako „otevřený“, je ho tedy možné používat i v primitivních webových aplikacích nebo společně s jiným otevřeným frameworkem, jako je například Zend Framework. [9]

#### 2.1.1 Model-View-Presenter (MVP)

V poslední době se neustále mluví o návrhovém vzoru MVC (Model-View-Controller). Co to vlastně je? A k čemu je to dobré? MVC je spíše než návrhový vzor softwarová architektura, která rozděluje aplikaci do tří vrstev: na datový model, uživatelské rozhraní a řídicí logiku. Přičemž modifikace některé z nich má pouze minimální vliv na ostatní.

Když se nad tím zamyslíte, každá část kódu webových aplikací skutečně spadá do jedné z těchto kategorií. MVC však říká, že tyto části je nutné oddělit do samostatných komponent nebo modulů. V praxi se ukázalo, že jde o velmi užitečný přístup. Vývojáři si ověřili, že tato separace je nezbytná pro udržení přehledného kódu – obzvláště v případech, kdy na jedné aplikaci pracuje více lidí.



Obrázek 1 - MVC model

MVC dále určuje vztah jednotlivých komponent, který je znázorněn na obrázku Obrázek 1 - MVC model:

- Model zajišťuje přístup k datům a manipulaci s nimi.
- View (pohled) převádí data reprezentovaná modelem do podoby vhodné k prezentaci uživateli.
- Controller (řadič) reaguje na události pocházející od uživatele a zajišťuje změny v modelu nebo v pohledu.

Tento princip poprvé popsal Trygve Reenskaug v roce 1979. Dnes je velmi populární právě u webových aplikací, jenže často jde o tvrzení pramenící z jeho pochopení. Ve své původní podobě jej vlastně nepoužívá nikdo. Role a vztahy jednotlivých vrstev se často chápou velmi volně. To je také důvod, proč se Nette Framework hlásí k MVC jen jako k duševně spřízněné architektuře.

Logice Nette Frameworku daleko lépe odpovídá nepříliš známý vzor MVP, tedy Model-View-Presenter. Zjednodušeně tak můžeme říct, že presenter v Nette je totéž, co controller v jiných frameworkcích. [10]

## 2.2 Databázový layer Dibi

Hlavním účelem layeru Dibi je maximálně ulehčit práci programátorům a zjednodušit zápis SQL příkazů, co to jen půjde. Dalším důvodem proč Dibi použít je snadný přístup k metodám, i bez globálních proměnných. Obsahuje funkce pro několik rutinních úkonů.

Navíc můžeme eliminovat výskyt chyby například pomocí přehledného zápisu SQL příkazů. Obrovskou výhodou je automatická podpora konvencí (escapování/slashování, uvozování identifikátorů), automatické formátování speciálních typů, např. datum, řetězec.

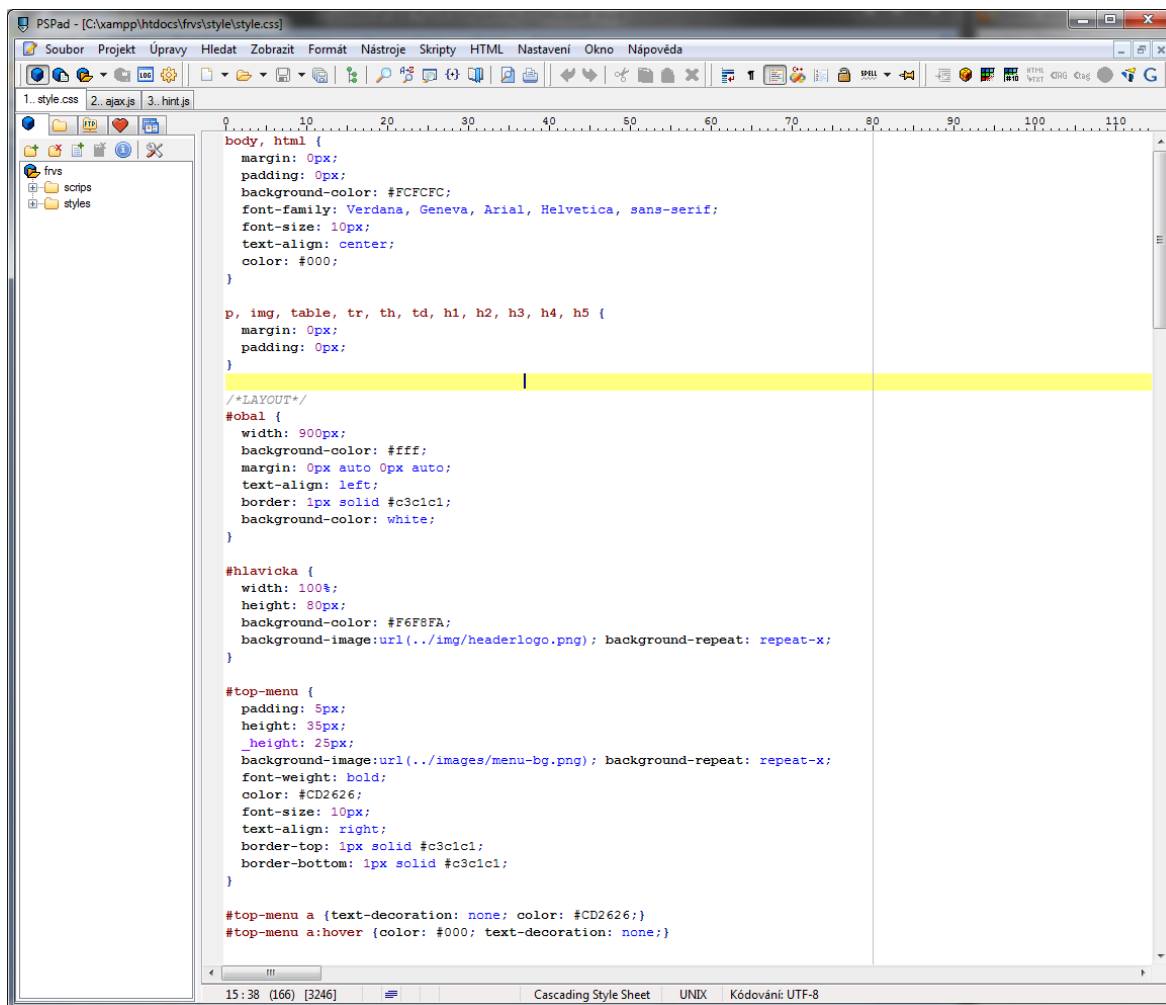
Obsahuje sjednocení základních funkcí (připojení k db, vykonání příkazu, získání výsledku) a především se snaží zachovat maximální jednoduchost (raději jeden geniální nápad, než 10.000 hloupých řádků kódu).

## 2.3 Editace CSS a JS

### 2.3.1 PSPad

PSPad je navržen jako univerzální editor pro editaci prostých textů a zdrojových kódů mnoha programovacích, skriptovacích a značkovacích jazyků. Umožňuje správu projektů, manipulaci s více soubory najednou pomocí moderního rozhraní MDI (Multiple Document Interface), průzkumník kódů a zvýrazňování syntaxe (včetně uživatelské možnosti implementace dalších jazyků) vytvořené pomocí komponenty SynEdit. Obsahuje také značné množství interních nástrojů např. FTP klient, HEX Editor, Lorem ipsum generátor, průzkumník kódu a další.

Možnosti zpracování textu jsou v PSPadu poměrně široké – editor podporuje makra, sloupcové bloky, vyhledávání a nahrazování s podporou regulárních výrazů, různé operace s označeným blokem textu (např. převod na velká/malá písmena, slučování nových řádků nebo pevné zarovnávání na nastavitelnou šířku řádky) a další. V nastavení nechybí pokročilé možnosti jako chytré, skutečné nebo mezerami nahrazované tabulátory, u UTF-8 kódování vkládání BOM sekvence.



Obrázek 2 - PSPad

## 2.4 Editace PHP

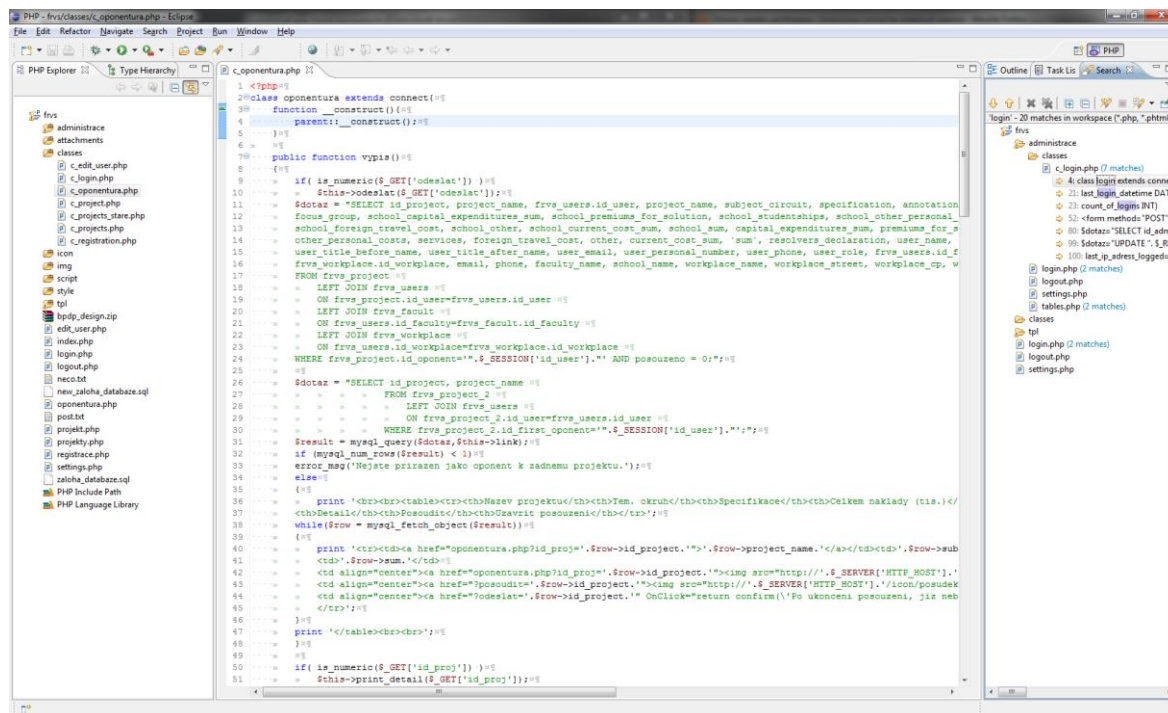
### 2.4.1 Eclipse Galileo

Eclipse je open source vývojová platforma, která je pro většinu lidí známa jako vývojové prostředí (IDE) určené pro programování v jazyce Java. Flexibilní návrh této platformy dovoluje rozšířit seznam podporovaných programovacích jazyků za pomoci pluginů, například o C++ nebo PHP. Právě pluginy umožňují toto vývojové prostředí rozšířit například o návrh UML, či zápis HTML nebo XML.

Oproti ostatním vývojovým prostředím v Javě, jako například Netbeans, je filozofie Eclipse úzce svázána právě s rozšiřitelností pomocí pluginů. V základní verzi obsahuje Eclipse pouze integrované prostředky pro vývoj standardní Javy jako kompilátor, debugger atd., ale neobsahuje například nástroj pro vizuální návrh grafických uživatelských rozhraní

desktopových aplikací nebo aplikační server – všechna taková rozšíření je potřeba dodat formou pluginů. Z tohoto důvodu přímo pod křídly Eclipse vznikly takzvané subprojekty.

Eclipse Galileo je jedním z nich.



Obrázek 3 - Eclipse Galileo

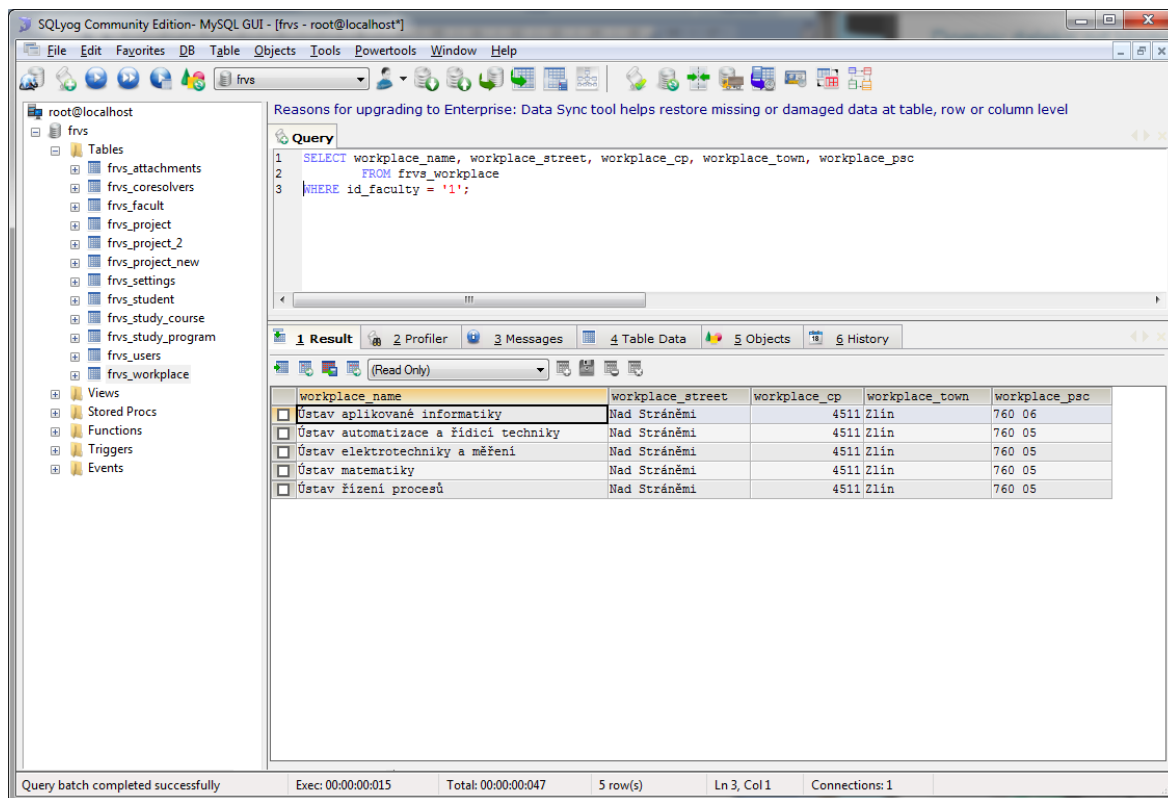
## 2.5 Ladění SQL dotazů

### 2.5.1 SQLyog

Program podporuje databáze MySQL včetně verze 5. x. Nabízí množství možností jak spravovat databázi, provádět dotazy, testovat atd.

Obsahuje Query editor a Result editor, umožňuje velké množství exportů (dump, XML, HTML, CSV) a importů dat, zvládá práci s velkými dump soubory.

Připojit k databázi se lze klasicky nebo pomocí SSH a HTTP tunelů.



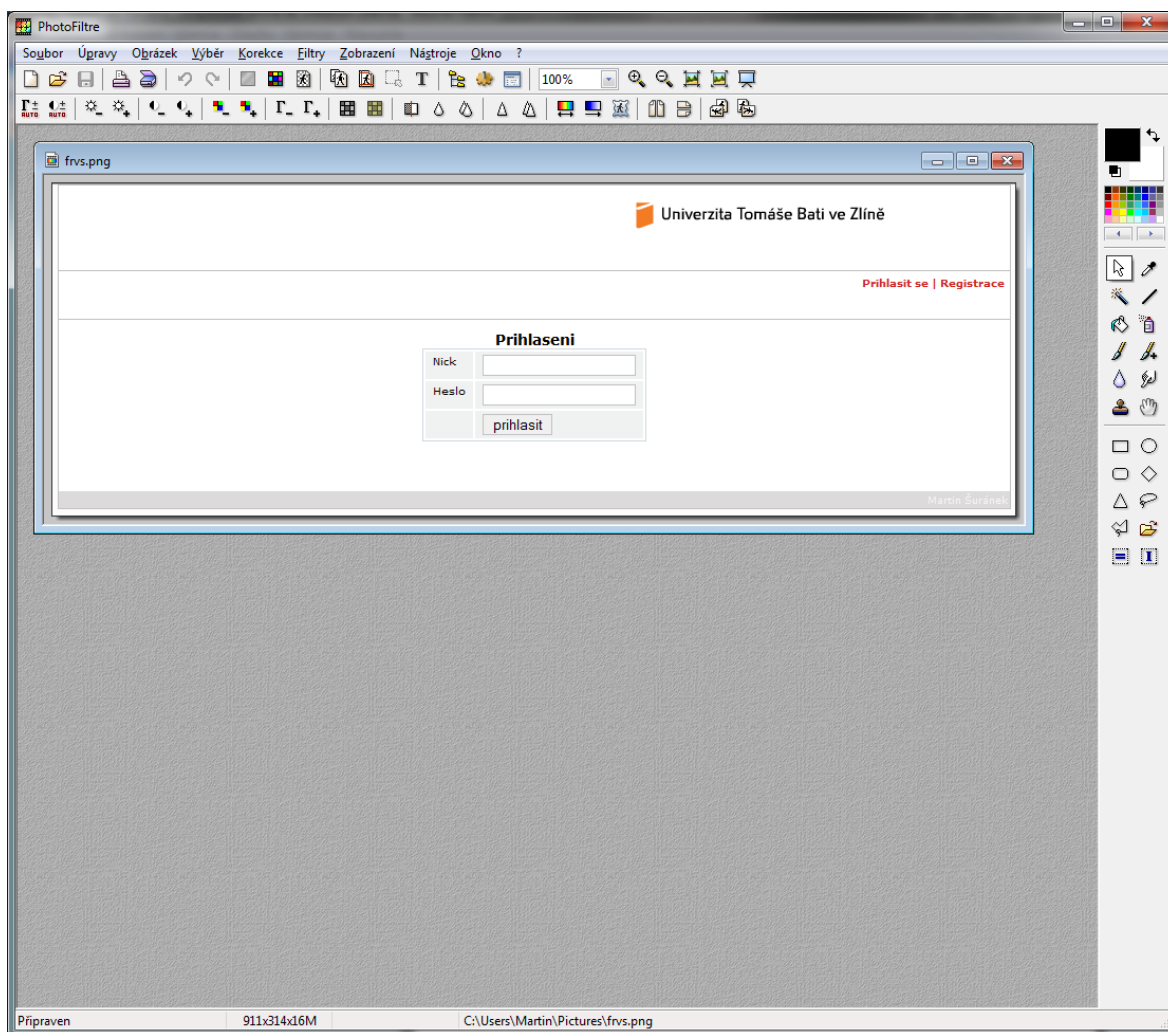
Obrázek 4 - SQLyog

## 2.6 Práce s grafikou

### 2.6.1 PhotoFiltre

Pro jednodušší práce s grafikou byl používán klasický pomocník z windows a to malování, ale občas došlo k tomu, že bylo potřeba udělat něco, na co malování nestačilo. Proto bylo nutné hledat nějakého jednoduchého pomocníka pro práci s grafikou, který by byl volně dostupný. Nakonec byl zvolen PhotoFiltre - je to jednoduchý freeware program, se kterým se dají mnohdy velmi rychle a snadno upravovat grafické objekty. Nabízí také velké množství nástrojů, které při práci oceníme. Po základní instalaci tento software neobsahuje češtinu, a proto si musíme stáhnout plugin z domovských stránek tohoto softwaru.





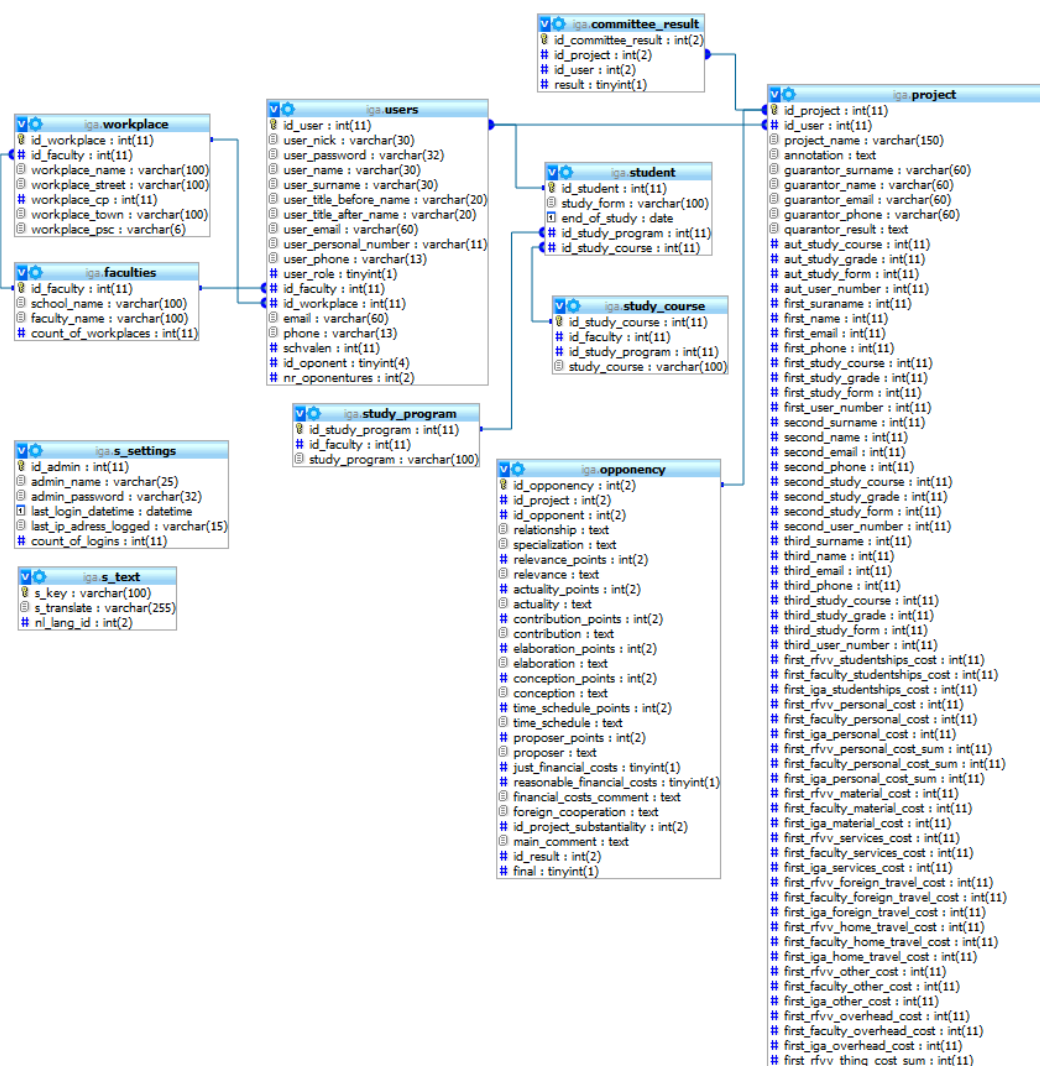
Obrázek 5 - PhotoFiltre

## **II. PRAKTICKÁ ČÁST**

## 3 NÁVRH STRUKTURY

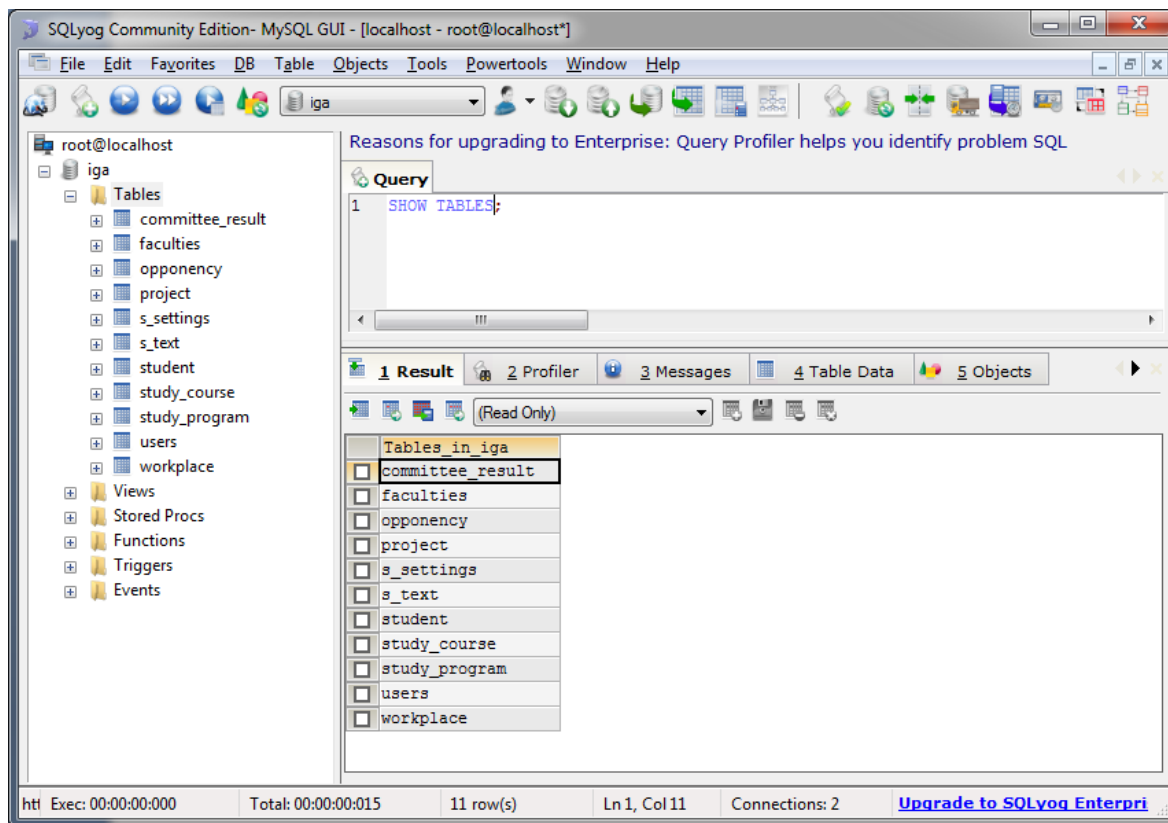
### 3.1 Databáze

#### 3.1.1 Schéma relací v databázi



Obrázek 6 - schéma relací v databázi

Databáze nese název „iga“ a obsahuje jedenáct tabulek. Do těchto tabulek se ukládají data, která jsou pak zobrazována uživatelům. Níže si popíšeme několik tabulek. Vysvětlíme si, v jakém formátu se data do tabulek vkládají a k čemu slouží.



Obrázek 7 - Databáze iga

### 3.1.2 Tabulka s\_settings

Tato tabulka slouží k udržení informací o administrátorech systému. Je nutná též k instalaci systému, při instalaci je tabulka vytvořena, a pokud ještě není vytvořen žádný administrátor, pak je vytvořen administrátor defaultní, který by po prvním přihlášení měl neprodleně změnit své údaje kvůli bezpečnosti systému.

Tabulka 1 - tabulka s\_settings

Sloupec	Hodnota
<b>id_admin</b>	INT, NOT NULL, AUTO_INCREMENT
<b>admin_name</b>	VARCHAR (25)
<b>admin_password</b>	VARCHAR (32)
<b>last_login_datetime</b>	DATETIME
<b>last_ip_adress_logged</b>	VARCHAR (15)

<b>count_of_logins</b>	INT(2)
------------------------	--------

### 3.1.3 Tabulka faculties

Tato tabulka obsahuje seznam fakult, které je možno přidávat v administrační části webu. Tato tabulka je provázána s tabulkou users.

Tabulka 2 - tabulka faculties

Sloupec	Hodnota
<b>id_faculty</b>	INT, NOT NULL, AUTO_INCREMENT
<b>school_name</b>	VARCHAR (100)
<b>faculty_name</b>	VARCHAR (100)
<b>count_of_workplaces</b>	INT (2)

### 3.1.4 Tabulka workplace

Zde najdeme informace o pracovištích fakult, ať už jde o ústavy, nebo detašované pracoviště. Jsou zde uloženy adresy těchto pracovišť. Pracoviště můžeme podobně jako fakulty měnit v administračním rozhraní, takže tak snadno přidáme, odstraníme nebo upravíme záznamy o pracovištích.

Tabulka 3 - tabulka wokplace

Sloupec	Hodnota
<b>id_workplace</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_faculty</b>	INT (2)
<b>workplace_name</b>	VARCHAR (100)
<b>workplace_street</b>	VARCHAR (100)
<b>workplace_cp</b>	INT (2)
<b>workplace_town</b>	VARCHAR (100)
<b>workplace_psc</b>	VARCHAR (6)

### 3.1.5 Tabulka student

Tabulka sloužící k uchování informací o uživateli, kteří plní roli studenta. Je provázána s tabulkami „users“, „study\_course“ a „study\_program“.

Tabulka 4 - tabulka student

Sloupec	Hodnota
<b>id_student</b>	INT, NOT NULL, AUTO_INCREMENT
<b>study_form</b>	VARCHAR (30)
<b>end_of_study</b>	VARCHAR (32)
<b>id_study_program</b>	VARCHAR (30)
<b>id_study_course</b>	VARCHAR (30)

### 3.1.6 Tabulka study\_course

Obsahem této tabulky jsou studijní obory na všech fakultách a ve všech studijních programech. Správa této tabulky je v administraci a umožňuje tak rozšířit studijní obory na fakultách, případně neplatné obory odstranit.

Tabulka 5 - tabulka study\_course

Sloupec	Hodnota
<b>id_study_course</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_faculty</b>	VARCHAR (30)
<b>id_study_program</b>	VARCHAR (32)
<b>study_course</b>	VARCHAR (30)

### 3.1.7 Tabulka study\_program

Předchozí tabulka study\_course obsahovala informace o studijních programech, tato tabulka obsahuje informace o studijních programech, které jsou dostupné na všech fakultách. A podobně jako předchozí tabulka je i tato upravovatelná v administračním rozhraní a můžeme tak studijní programy v případě změn na fakultách upravovat.

Tabulka 6 - tabulka study\_program

Sloupec	Hodnota
<b>id_study_program</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_faculty</b>	VARCHAR (30)
<b>study_program</b>	VARCHAR (32)

### 3.1.8 Tabulka users

Tabulka slouží k uložení informací o uživateli. Ukládá se id uživatele, přezdívka, heslo, jméno, příjmení, tituly před jménem, tituly za jménem, email, osobní číslo uživatele (stag), telefon, uživatelská role (zaměstnanec, student, účastník komise), id fakulty pod kterou spadá, a id pracoviště které navštěvuje, kontaktní email na pracoviště, kontaktní telefon na pracoviště, identifikátor stavu schválení uživatele, počet projektů ke kterým uživatel dělá oponenturu.

Tabulka 7 - tabulka users

Sloupec	Hodnota
<b>id_user</b>	INT, NOT NULL, AUTO_INCREMENT
<b>user_nick</b>	VARCHAR (30)
<b>user_password</b>	VARCHAR (32)
<b>user_name</b>	VARCHAR (30)
<b>user_surname</b>	VARCHAR (30)
<b>user_title_before_name</b>	VARCHAR (20)
<b>user_title_after_name</b>	VARCHAR (20)
<b>user_email</b>	VARCHAR (60)
<b>user_personal_number</b>	VARCHAR (11)
<b>user_phone</b>	VARCHAR (13)
<b>user_role</b>	TINYINT(1)

<b>id_faculty</b>	INT(2)
<b>id_workplace</b>	INT(2)
<b>email</b>	VARCHAR(60)
<b>phone</b>	VARCHAR(13)
<b>confirmed</b>	TINYINT(1)
<b>id_oponent</b>	INT(2)
<b>nr_oponentures</b>	INT(2)

### 3.1.9 Tabulka project

Tato tabulka slouží k uložení informací o projektech, které vytváří zaměstnanci nebo studenti doktorandského studia na univerzitě. Do tabulky se ukládá id projektu, id uživatele, který projekt vytvořil, název projektu, informace o spolu-navrhovatelích, anotaci projektu, informace o garantu projektu, finanční náklady projektu, zdůvodnění finančních nákladů projektu, zdůvodnění návrhu v dané struktuře a spoustu dalších informací o projektu.

Tabulka 8 – část tabulky project

Sloupec	Hodnota
<b>id_project</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_user</b>	INT(2)
<b>project_name</b>	VARCHAR(150)
<b>annotation</b>	TEXT
<b>guarantor_surname</b>	VARCHAR(60)
<b>guarantor_name</b>	VARCHAR(60)
<b>guarantor_email</b>	VARCHAR(60)
<b>guarantor_phone</b>	VARCHAR(60)
<b>guarantor_result</b>	TEXT



<b>aut_study_course</b>	INT(2)
<b>aut_study_grade</b>	INT(2)
<b>aut_study_form</b>	INT(2)
<b>aut_user_number</b>	INT(2)
<b>first_frvv_studentships_cost</b>	INT(2)
<b>first_faculty_studentships_cost</b>	INT(2)
<b>first_iga_studentships_cost</b>	INT(2)
<b>first_frvv_personal_cost</b>	INT(2)
<b>first_faculty_personal_cost_sum</b>	INT(2)
.....	.....
<b>personal_cost_first_year</b>	INT(2)
<b>thing_cost_first_year</b>	INT(2)
.....	.....
<b>reason_of_financial_offers</b>	INT(2)
<b>reason_concept</b>	INT(2)
<b>final</b>	INT(4)
<b>id_first_oponent</b>	TEXT
<b>id_second_oponent</b>	TINYINT(1)
<b>ready_to_committee</b>	INT(2)

### 3.1.10 Tabulka opponency

Do této tabulky se ukládají posudky k vytvořeným projektům (chcete-li přihláškám ke grantové soutěži). Tabulka je provázána s tabulkou „users“ a „project“. Část informací o posudku je po finalizaci oponentury (oponent musí označit posudek jako finální) zobrazena autorovi projektu, který je hodnocen.

Tabulka 9- tabulka opponency

Sloupec	Hodnota
<b>id_opponency</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_project</b>	INT(2)
<b>id_opponent</b>	INT(2)
<b>relationship</b>	TEXT
<b>specialization</b>	TEXT
<b>relevance_points</b>	INT (2)
<b>relevance</b>	TEXT
<b>aktuality_points</b>	INT (2)
<b>aktuality</b>	TEXT
<b>contribution_points</b>	INT(2)
<b>contribution</b>	TEXT
<b>elaboration_points</b>	INT(2)
<b>elaboration</b>	TEXT
<b>conception_points</b>	INT(2)
<b>conception</b>	TEXT
<b>time_schedule_points</b>	INT(2)
<b>time_schedule</b>	TEXT
<b>proposer_points</b>	INT(2)
<b>proposer</b>	TEXT
<b>just_financial_costs</b>	TINYINT(1)
<b>reasonable_financial_costs</b>	TINYINT(1)
<b>financial_costs_comment</b>	TEXT

<b>foreign_cooperation</b>	TEXT
<b>id_project_substantiality</b>	INT(2)
<b>main_comment</b>	TEXT
<b>id_result</b>	INT (2)
<b>final</b>	TINYINT(1)

### 3.1.11 Tabulka `committee_result`

Tato tabulka slouží k ukládání výsledků hodnotící komise.

Tabulka 10 - tabulka `committee_result`

Sloupec	Hodnota
<b>id_committee_result</b>	INT, NOT NULL, AUTO_INCREMENT
<b>id_project</b>	INT (2)
<b>id_user</b>	INT (2)
<b>result</b>	TINYINT (1)

## 3.2 Aplikační část

### 3.2.1 Presentery (Controls)

Aplikační část je rozdělena do 4 základních presenterů.

- DefaultPresenter – základní presenter
- AuthPresenter - zajišťuje přihlašování uživatelů
- AccountPresenter – slouží ke správě uživatelských účtů
- RegistrationPresenter – obsluhuje registraci nových uživatelů
- ProjectPresenter – spravuje přihlášky do grantové soutěže
- OppnencyPresenter – presenter pro posuzování projektů

### 3.2.2 Modely (Models)

K výše zmíněným presenterům patří následující modely, které zajišťují komunikaci s databázovou vrstvou.

- Model – základní model, který obsluhuje připojení k databázi (od něj pak ostatní modely dědí připojení k databázi)
- Users – model který předává potřebná data AuthPresenteru
- AccountModel – předává data AccountPresenteru
- RegistrationModel – model patřící RegistrationPresenteru
- ProjectModel – vybírá a ukládá data z ProjectPresenteru
- OponnencyModel – obsluhuje OponnencyPresenter a předává mu potřebná data
- ConnivancePresenter – presenter pro správu projektů připravených k posouzení komisí

### 3.2.3 Pohledy (Views)

Pohledy obstarávají prosté zobrazení dat uživatelům. Ke každému presenteru může být přiděleno libovolně mnoho pohledů, které se aktivují v presenteru reakcí na změny.

- Layout & Plugins
  - header.phtml
  - menu.phtml
  - footer.phtml
  - VisualPaginator.phtml
- DefaultPresenter
  - default.phtml
- AuthPresenter
  - login.phtml
- AccountPresenter
  - edit.phtml

- RegistrationPresenter
  - regform.phtml
- ProjectPresenter
  - all.phtml
  - new.phtml
- OponnencyPresenter
  - print.phtml

### 3.3 Administrační část

#### 3.3.1 Presentery (Controls)

Aplikační část je rozdělena do 4 základních presenterů.

- DefaultPresenter – základní presenter
- AuthPresenter - zajišťuje přihlašování administrátorů
- AdminPresenter – slouží ke správě administrátorských účtů
- FacultiesPresenter – slouží ke správě fakult
- WorkplacesPresenter – slouží ke správě pracovišť na fakultách
- StudyProgramCoursesPresenter – presenter pro správu studijních programů na jednotlivých fakultách a jejich příslušných oborů
- UsersPresenter – slouží ke správě registrovaných uživatelů
- CommitteePresenter – slouží ke správě schvalovací komise
- ProjectPresenter – spravuje přihlášky do grantové soutěže
- OponnencyPresenter – presenter pro přidělování oponentů k projektům
- ConnivancePresenter – presenter pro správu projektů připravených k posouzení komisí

### 3.3.2 Modely (Models)

K výše zmíněným presenterům patří následující modely, které zajišťují komunikaci s databázovou vrstvou.

- Model – základní model, který obsluhuje připojení k databázi (od něj pak ostatní modely dědí připojení k databázi)
- Users – model který předává potřebné data AuthPresenteru
- AdminModel – předává data AdminPresenteru
- FacultiesModel – model patřící FacultiesPresenteru
- WorkplacesModel – patří WorkplacesPresenteru
- ProjectModel – vybírá a ukládá data z ProjectPresenteru
- StudyProgramCoursesModel – slouží StudyProgramCoursesPresenteru
- UsersModel – patří UsersPresenteru
- CommitteeModel – patří CommitteePresenteru
- ProjectModel – slouží ProjectPresenteru
- OponnencyModel – obsluhuje OponnencyPresenter a předává mu potřebné data
- ConnivanceModel – patří ConnivancePresenteru

### 3.3.3 Pohledy (Views)

Pohledy obstarávají prosté zobrazení dat uživatelům. Ke každému presenteru může být přiděleno libovolně mnoho pohledů, které se aktivují v presenteru reakcí na změny.

- Layout & Plugins
  - header.phtml
  - menu.phtml
  - footer.phtml
  - VisualPaginator.phtml
- DefaultPresenter
  - default.phtml

- AuthPresenter
  - login.phtml
- AdminPresenter
  - edit.phtml
  - add\_admin.phtml
- FacultiesPresenter
  - faculties.phtml
  - edit\_faculty
- WorkplacesPresenter
  - workplaces.phtml
  - edit\_workplace
- ProjectPresenter
  - all.phtml
  - new.phtml
- StudyProgramCoursesPresenter
  - print.phtml
  - edit.phtml
- UsersModel
  - print.phtml
- CommitteeModel
  - print.phtml
- ConnivanceModel
  - print.phtml
- OponnencyPresenter
  - print.phtml

## 3.4 Podpora uživatelů (Helpdesk)

### 3.4.1 Pracovní pokyny v systému

Pracovní pokyny v systému jsou realizovány pomocí vykreslení dialogového okna s nápovědou po najetí myši na vyznačená místa, které jsou určeny k dostatečnému informování uživatele o akci, kterou se pokouší provést.

### 3.4.2 Manuál k systému

Manuál k systému byl vytvořen jako návod k používání systému a je uveden v praktické části. Jsou zde popsány základní úkony patřící do Aplikační části a následně i Administrační části.

Aplikační část:

- Základní rozložení
- Registrace uživatele
- Vytváření projektů
- Správa projektů

Administrační část:

- Správa fakult
- Správa pracovišť
- Správa studijních programů a oborů
- Správa uživatelů
- Správa komise
- Správa projektů



## 4 POSTUP ŘEŠENÍ

### 4.1 Aplikační část

Aplikační část slouží pro registrované uživatele. Ti zde mohou vytvářet a spravovat svoje projekty, jejichž prostřednictvím žádají následně o grant v grantové soutěži. V aplikační části také probíhá tvorba posudků. Tuto činnost vykonávají uživatelé, jež jsou k projektu v administraci přiděleni jako oponenti (ke každému projektu dva, jeden z toho musí být mimo fakultu uživatele, jenž projekt vytvořil).

#### 4.1.1 Registrace uživatelů

K registraci uživatelů, bylo potřeba vytvořit registrační formulář. Ten sem realizoval pomocí třídy *AppForm* v presenteru *RegistrationPresenter*. Třída *AppForm* umožňuje jednoduchou a přehlednou tvorbu formulářů. Další výhodou použití této třídy je, že můžeme rovnou definovat validační pravidla a filtry pro jednotlivé položky formuláře. Validace probíhá na straně klienta, tedy pomocí JavaScriptu. Podle mého názoru je toto řešení pro uživatele velmi přívětivé. Po úspěšném vyplnění toho formuláře stačí už jen předat metodě *insert\_user(\$data)* v modelu *RegistrationModel* data, které má uložit do databáze.

#### 4.1.2 Přihlašování

Přihlašování je realizováno třídou *AuthPresenter*, ve které se nachází komponenta *LoginForm()*, ta s pomocí třídy *AppForm* vytvoří přihlašovací formulář. Zpracování formuláře probíhá pomocí obslužného handleru *loginFormSubmitted(\$form)*. Ten pomocí metody *authenticate()*, obsažené v modelu *Users*, provede přihlášení. V případě neúspěchu nám vyhodí tato metoda výjimku. Buď uživatel nebyl nalezen, nebo špatně zadané heslo.

#### 4.1.3 Tvorba projektů

Tvorba projektu je obsažena v třídě *ProjectPresenter*. Nejdříve bylo potřeba promyslet jak podat uživatelům tak rozsáhlý formulář jako přívětivý a neodradit je ještě před jeho vyplněním. Pro tvorbu a validaci formuláře sem opět použil třídu *AppForm*, to ale neřešilo rozsáhlost tohoto formuláře. Rozhodl jsem se ten to formulář rozdělit do více kroků. Zde jsem zjistil že odesílání formuláře po částech má své úskalí, proto jsem musel přijít na způsob jak podat uživateli formulář po částech se snadnou možností přecházení mezi

částmi toho formuláře. Nakonec jsem to vyřešil pomocí JavaScriptu. Formulář je rozdělen na čtyři části, každá z nich je umístěna zvlášť v *div*ech s jedinečnými *id*. Uživatel se pak pohybuje ve formuláři tlačítky zpět a další, na poslední stránce formuláře se tlačítko další nahradí tlačítkem odeslat. Při procházení formuláře se pomocí JavaScriptu nastaví viditelnost stránky formuláře (tedy části formuláře která je obsažena v *divu*), ostatní části formuláře jsou uživateli skryty. Výhodou je, že při procházení mezi stránkami formuláře nedochází k refreshi stránky a uživatel se tak může snadno vracet na předchozí stránky formuláře. Kontrola formuláře probíhá na každé stránce zvlášť, tedy průběžně. K přechodu na další stránku formuláře je nutností validně vyplněná aktuální stránka formuláře. Odeslání formuláře proběhne až po kliknutí na tlačítko odeslat na poslední stránce. Uložení pak proběhne pomocí metody *insert\_project(\$\_POST)*, která se nachází ve třídě *ProjectModel*. Takto vytvořený projekt je ale stále považován jako rozpracovaný, aby byl projekt zařazen do grantové soutěže, musí být jeho autorem označen jako finální. To znamená, že už nebude moct na projektu provádět žádné změny. A projekt bude poslán k posouzení.

#### 4.1.4 Správa projektů

Správa projektů je realizována v třídě *ProjectsPresenter*, která poskytuje přihlášenému uživateli seznam projektů které vlastní. Projekty se dělí na rozpracované a finální. Finální projekt pak může být pro každého uživatele jenom jeden. Na finálním projektu uživatel nemá možnost provádět žádné úpravy, může si jen projekt prohlédnout, pokud jsou posudky k projektu ve finálním stavu, může si část posudku určeného pro autory projektů zobrazit. Dále vidí autor projektu výsledek schvalovacího procesu členů komise. Třída *ProjectsPresenter* obsahuje metodu *print\_projects()*, *print\_detail\_project(\$id)*, *finalize(\$id)*, *edit\_project(\$id)*, *delete\_project(\$id)*. K těmto metodám jsou v modelové třídě *ProjectsModel* obslužné metody pro správu dat se kterými metody pracují.

#### 4.1.5 Oponentura

Oponenturu projektů provádí registrovaní uživatelé, kteří byli v administraci k projektům přiděleni. Oponent má po přihlášení do systému a zvolení položky oponentura z horního menu k dispozici výpis projektů, ke kterým je přidělen jako oponent. U těchto projektů si může oponent zobrazit detail projektu a po důkladném přezkoumání projektu pak může pomocí formuláře pro posudek projektů vytvořit posudek. Při tvorbě formuláře pro

opponentské posudky jsem postupoval podobně jako u formulářů pro přidávání projektů, opět jsem použil třídu *AppForm* a jelikož byl formulář opět docela rozsáhlý, rozdělil jsem jej na tři části. Procházení těchto částí je realizováno obdobně jako u tvorby projektů pomocí JavaScriptu. Oponent má možnost svůj posudek měnit do chvíle než jej označí za finální. Pak má možnost svůj posudek jenom prohlížet. Část výsledného posudku je pak zobrazena autorovi projektu.

#### 4.1.6 Schvalování projektů

Schvalování projektů má na starosti komise, která je sestavena v administračním rozhraní. Složení komise by mělo být:

- Předseda komise: prorektor pro tvůrčí činnosti
- 1. člen komise: proděkan pro vědu a výzkum
- 2. člen komise: proděkan pro vědu a výzkum

Komise se skládá z uživatelů, kteří jsou zaregistrováni jako normální uživatelé s tím, že mají nastavena vyšší práva. Tím jsou rozlišeni od ostatních uživatelů a po přihlášení do systému mají nabídku horního menu rozšířenu o položku „Projekty ke schválení“. Zde jsou jim zobrazeny projekty, které jsou finální a jsou k nim finální posudky od obou oponentů. K těmto projektům pak mají možnost zobrazit si jejich detail, i detail jejich obou posudků. Po přezkoumání pak mají možnost se rozhodnout pro schválení nebo naopak zamítnutí projektu. Schvalování projektů je realizováno pomocí třídy *ConnivancePresenter*. Třída obsahuje metodu *print\_projects()* pro výpis seznamu projektů, které jsou připraveny pro schválení, dále metodu *print\_project\_detail(\$id)* pro výpis detailu projektu, metodu *print\_opponency(\$id)* pro výpis posudku, metodu *save\_result(\$result)* pro zpracování procesu schválení a nakonec metodu *show\_project\_status(\$id)* pro výpis informací o stavu schválení projektu od všech členů komise. Všechny tyto metody mají v třídě *ConnivanceModel* svoje obslužné metody pro výběr / zápis dat z / do databáze.

## 4.2 Administrační část

Administrační část slouží převážně ke správě uživatelů a projektů. Dále k určení účastníků komise, přidělování oponentů k projektům. V neposlední řadě poskytuje administrační

rozhraní možnost spravovat údaje o fakultě, jejich ústavech (pracovištích) a studijních oborech.

#### 4.2.1 Správa tabulek v databázi

Ke správě tabulek v databázi slouží třída *create\_tables*, která obsahuje pole, ve kterém jsou obsaženy dotazy pro vytvoření tabulek potřebných k chodu systému. Administrátorovi jsou vypsány dva sloupce tabulek, v prvním sloupci se nachází tabulky, které je potřeba vytvořit a ve druhém tabulky které vykazují konflikt (uvedeny jsou zde i ty tabulky, které již existují). Administrátor pak může chybějící tabulky vytvořit, nebo stávající tabulky smazat a znovu vytvořit (pouze u prázdných tabulek), pro případ že došlo ve změnách struktury tabulky.

#### 4.2.2 Přihlašování

Přihlašování je realizováno obdobně jako v aplikační části. Opět je použita třída *AuthPresenter*, ve které se nachází komponenta *LoginForm()*, ta s pomocí třídy *AppForm* vytvoří přihlašovací formulář. Zpracování formuláře probíhá opět pomocí obslužného handleru *loginFormSubmitted(\$form)*. Ten pomocí metody *authenticate()*, obsažené v modelu *Users*, provede přihlášení. S tím rozdílem, že se nevybírají uživatelé s tabulkou *users*, ale administrátoři s tabulkou *s\_settings*. V případě neúspěchu nám opět vyhodí tato metoda výjimku. Buď uživatel nebyl nalezen, nebo špatně zadané heslo.

#### 4.2.3 Správa administrátorů

O správu administrátorů se stará třída *AdminPresenter*, která obsluhuje tvorbu nových administrátorských účtů a úpravu vlastního administrátorského účtu. Tyto funkce zajišťují metody *edit\_admin(\$id)* a *add\_admin()*. Tvorba formuláře a jeho následná validace je opět zajištěna za pomoci třídy *AppForm*. Po úspěšném vyplnění formuláře jsou následně upravena data v databázi pomocí metod *update\_admin(\$id)* a *insert\_admin* z modelu *AdminModel*.

#### 4.2.4 Správa fakult

Správa fakult je obsluhována třídou *FacultiesPresenter*, která obsahuje metodu *show\_faculties()* pro výpis fakult nacházejících se v systému, *insert\_faculty(\$data)* pro

vložení nové fakulty do systému, `edit_faculty($id)` pro úpravu údajů o fakultě a `delete_faculty($id)` pro odstranění fakulty ze systému.

#### 4.2.5 Správa pracovišť

Ke správě pracovišť slouží `WorkplacesPresenter`, pomocí metod obsažených v této třídě může administrátor přidávat nové pracoviště k příslušné fakultě. Tuto funkci zajišťuje metoda `add_workplace()`. Pracoviště se dají následně upravovat a odstraňovat za pomoci metod `edit_workplace($id)` a `delete_workplace($id)`.

#### 4.2.6 Správa studijních programů a oborů

Ke správě pracovišť slouží `StudyProgramCoursesPresenter`, administrátor zde může vkládat studijní programy a k nim přiřazené studijní obory. Tato data slouží pro výběr studijních programů a oborů při registraci uživatele v roli studenta doktorandského studia.

#### 4.2.7 Správa registrovaných uživatelů

Zde pomocí třídy `UsersPresenter` administrátor schvaluje registrované uživatele, přehledně se mu zobrazují registrovaní uživatelé a navíc má administrátor možnost sám uživatelům vytvářet uživatelské účty (jsou jim generovány univerzální hesla, které si uživatelé po přihlášení mohou změnit). K tomu slouží metody `show_users()`, `confirm_user($id)`, `create_user($data)` a samozřejmě jim přísluší metody z třídy `UsersModel`, které zajišťují komunikaci s databází.

#### 4.2.8 Správa komisí

Ke správě komisí slouží třída `ConnivancePresenter` administrátorovi je zde vypsán přehled členů komise s odkazem pro přidání nových členů komise. Jak již bylo zmíněno, komise by měla být ve složení:

- Předseda komise: prorektor pro tvůrčí činnosti
- 1. člen komise: proděkan pro vědu a výzkum
- 2. člen komise: proděkan pro vědu a výzkum

### 4.2.9 Správa projektů

Správa projektů je obstarána třídou ProjectsPresenter. Z pohledu administrátora je zde rozdělena do tří částí:

- 1.) Projekty – zde jsou administrátorovi vypsány všechny projekty, které jsou v systému obsaženy, ať už finální, nebo rozpracované, schválené či neschválené. U každého z těchto projektů je mu pro přehlednost zobrazen stav projektu a stav posouzení. Administrátor je pomocí metody `show_detail_project($id)` může prohlížet.
- 2.) Projekty k posouzení – tady administrátor nalezne projekty, které jsou ve finální fázi a jeho povinností je k takovému projektu přidělit dva oponenty, z nichž jeden musí být mimo fakultu autora projektu. O to se ale administrátor nemusí starat, oponenty vybírá ze seznamu, který již obsahuje vhodné uživatele pro oponenturu.
- 3.) Projekty ke schválení – zde administrátor uvidí seznam projektů, které již mají posudky od obou oponentů ve finálním stavu. Administrátor si tak může na jednom místě přehledně zobrazit detail projektu a detaily obou oponentských posudků v nezkrácené podobě jak je tomu u zobrazení posudků u autora projektu.

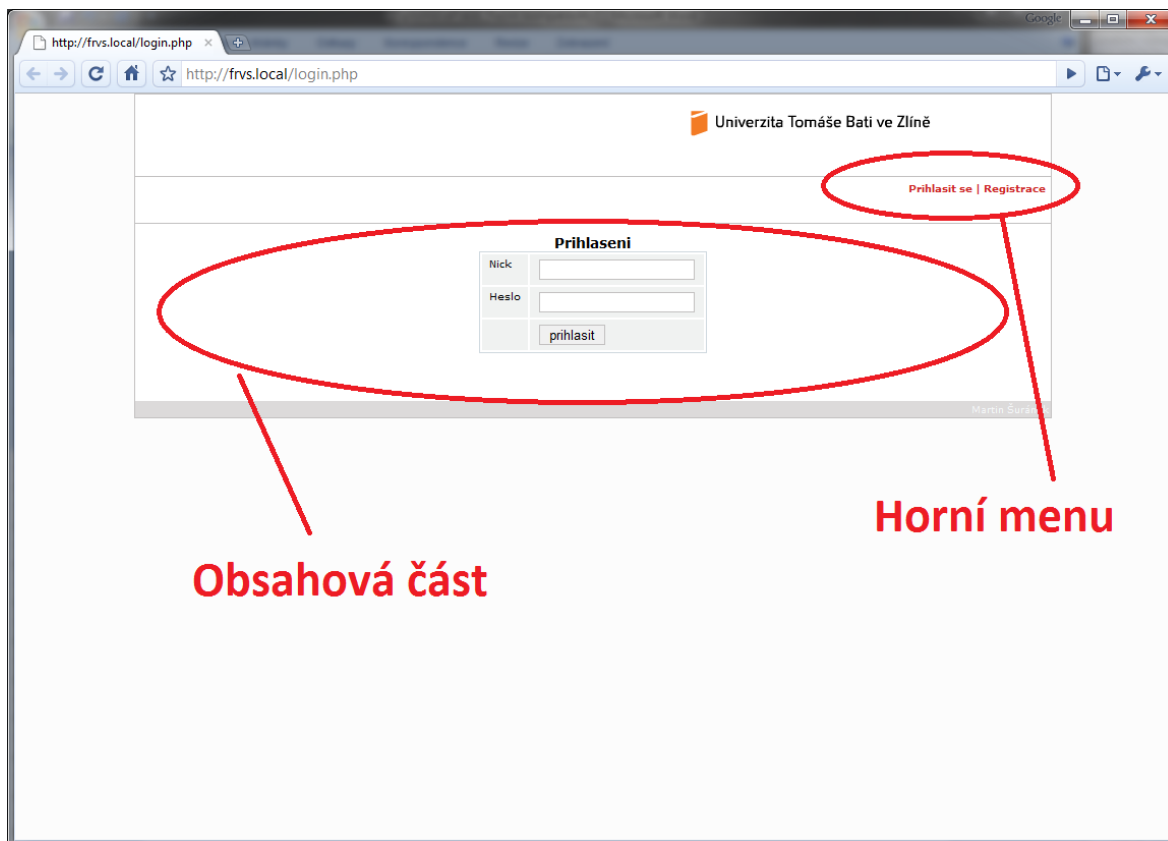
## 4.3 Podpora uživatelů – Helpdesk

### 4.3.1 Manuál k systému

#### 1. Aplikační část

Aplikační část slouží pro registrované uživatele. Ti zde mohou vytvářet a spravovat svoje projekty, jejichž prostřednictvím žádají následně o grant.

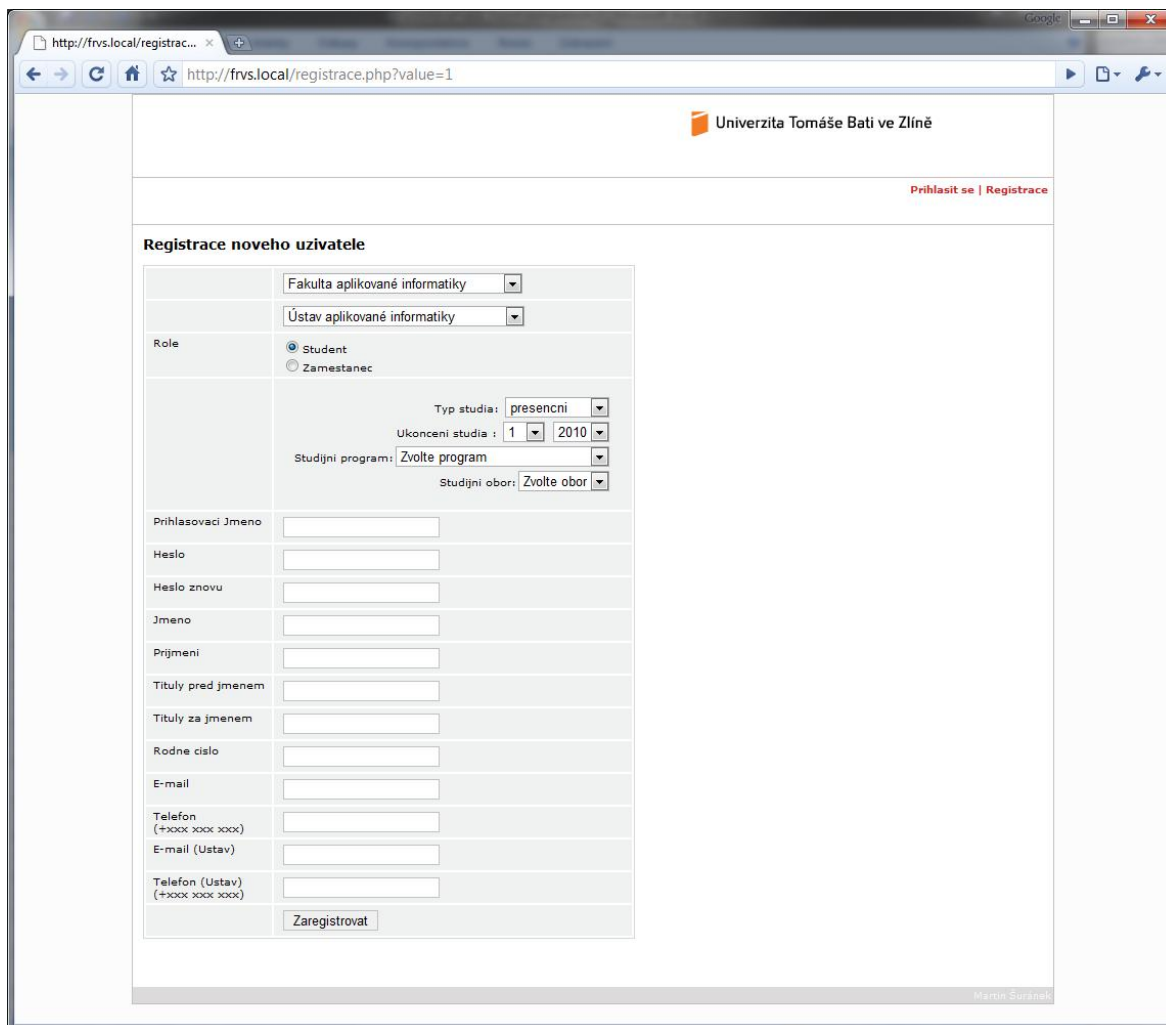
##### 1.1. Základní rozložení



Obrázek 8 - Základní rozložení

## 1.2. Registrace uživatele

Pro používání portálu pro administraci projektů se musí v první řadě uživatel registrovat. Registrace probíhá pomocí registračního formuláře. Při registraci si uživatel zároveň volí svou roli. Má na výběr ze dvou možností: zaměstnanec fakulty, nebo student univerzity. Student pak navíc vyplňuje informace o typu studia, čas ukončení studia, studijní program a studijní obor. Všechny položky registračního formuláře jsou povinné, vyplňování dlouhých formulářů bývá sice „nepohodlné“, ale v budoucnu ušetří uživatelům spoustu práce při vytváření jejich projektů, kde bude většina jejich údajů automaticky vyplněna.



The screenshot shows a web browser window with the URL `http://fivs.local/registrac...` and `http://fivs.local/registrace.php?value=1`. The page header includes the logo and name of the University of Tomáš Bati in Zlín, along with links for 'Přihlásit se' and 'Registrace'. The main content area is titled 'Registrace nového uživatele' and contains a registration form. The form has several sections: 1. Selections: Faculty (Fakulta aplikované informatiky), Department (Ústav aplikované informatiky), Role (Student selected, Zamestnanec), Typ studia (presencni), Ukončení studia (1, 2010), Studijní program (Zvolte program), and Studijní obor (Zvolte obor). 2. Personal Information: Fields for Přihlasovací Jméno, Heslo, Heslo znovu, Jméno, Příjmení, Tituly před jménem, Tituly za jménem, Rodné číslo, E-mail, Telefon (+xxx xxx xxx), E-mail (Ustav), and Telefon (Ustav) (+xxx xxx xxx). 3. Action: A 'Zaregistrovat' button at the bottom. The footer of the page mentions 'Martin Šuranek'.

Obrázek 9 - Registrace uživatele

### 1.3. Vytváření projektů

Jedná se o nejdůležitější část portálu. Pro vytvoření nového projektu, musí vyplnit formulář, který je kvůli jeho velikosti rozdělen na čtyři kroky. Během vyplňování údajů a přepínáním mezi jednotlivými kroky probíhá kontrola právě vyplněných údajů pomocí JavaScriptu, což má řadu výhod. Během vyplňování formuláře a přepínáním mezi jednotlivými kroky nedochází k obnovování (refreshi) stránky, tudíž nemusíme mít strach, že při vracení se o krok zpět přijdeme o již vyplněná data. Navíc je tento způsob nesmírně pohodlný a přehledný. Kontrola hodnot ve formuláři probíhá pro každý krok zvlášť, nemusíme mít proto strach z toho, že po dlouhém, úmorném vyplňování formuláře zjistíme, kolika chyb jsme se dopustili a nemáme již chuť ani sílu je opravovat. U posledního kroku formuláře se nám zobrazí tlačítko pro uložení projektu, které při správném vyplnění všech údajů provede zápis do databáze.



http://frvs.local/projekt.p... x

http://frvs.local/projekt.php

Univerzita Tomáše Bati ve Zlíně

Nový Projekt | Moje Projekty | Oponentura | Jste přihlášen jako: zdenka Odhlásit | Upravit účet |

### Nový projekt

#### Žádost o udělení interního grantu - A

Datum převzetí žádosti oddělením VaV:  Reg. číslo\*

Název projektu:

**Navrhovatel:**

Příjmení  Jméno  Kontakt: (e-mail, tel.)

Prokopová  Zdenka  prokopova@fai.utb.cz

+420576035186

Studovaný obor  Ročník  Forma  Osobní číslo

**1 Spolunavrhovatel:**

Příjmení  Jméno  Kontakt: (e-mail, tel.)

Studovaný obor  Ročník  Forma  Osobní číslo

**2 Spolunavrhovatel:**

Příjmení  Jméno  Kontakt: (e-mail, tel.)

Studovaný obor  Ročník  Forma  Osobní číslo

**3 Spolunavrhovatel:**

Příjmení  Jméno  Kontakt: (e-mail, tel.)

Studovaný obor  Ročník  Forma  Osobní číslo

*(Další členy týmu uveďte na samostatném listu)*

**Anotace projektu**

**Garant projektu:**

Příjmení  Jméno  Kontakt: (e-mail, tel.)

**Vyjádření garanta:**

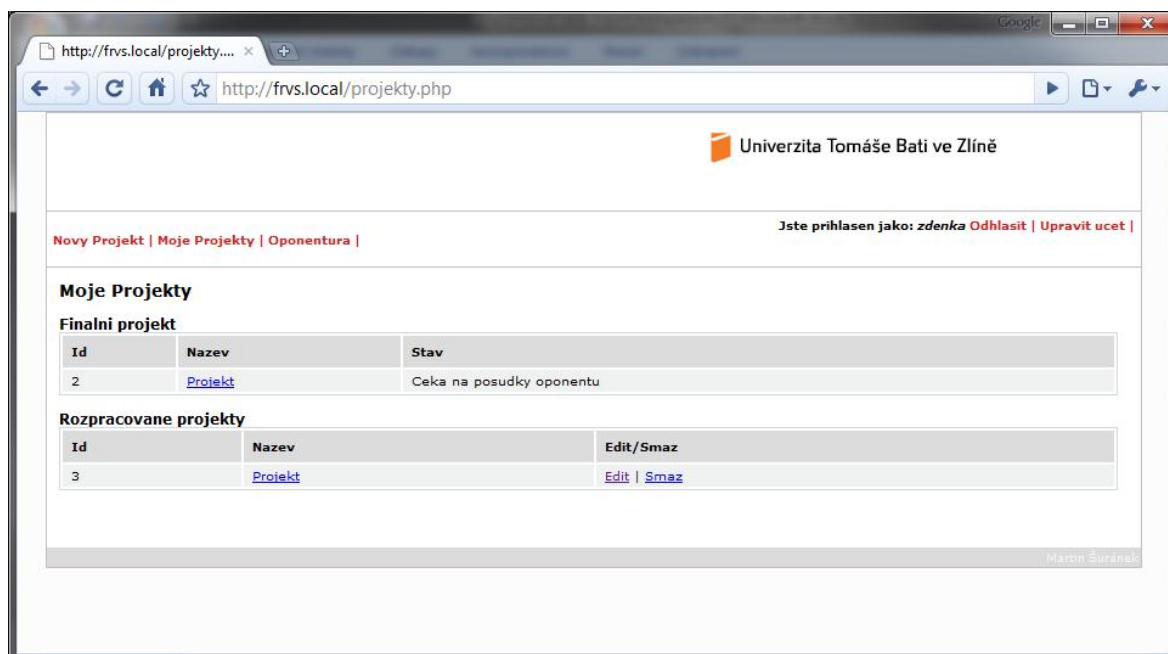
\* Reg. číslo přiděluje rektorát

Zpět Další Strana: 1/4

Obrázek 10 - Vytváření projektů

## 1.4. Správa projektů

Každý z vytvořených projektů má pak uživatel k dispozici pod odkazem „Moje projekty“, kde může projekty různě upravovat, mazat a nakonec nastavit projekt jako finální. Takto zvolený projekt již nemůže být nadále upravován či smazán. Tato možnost je pouze u jednoho projektu, to je totiž maximální počet žádostí o grant v jednom roce od jednoho uživatele. Není mu však bráněno v rozpracování dalších projektů, které si tak může chystat na další rok. Ty už ale nebudou moci být označeny jako finální.



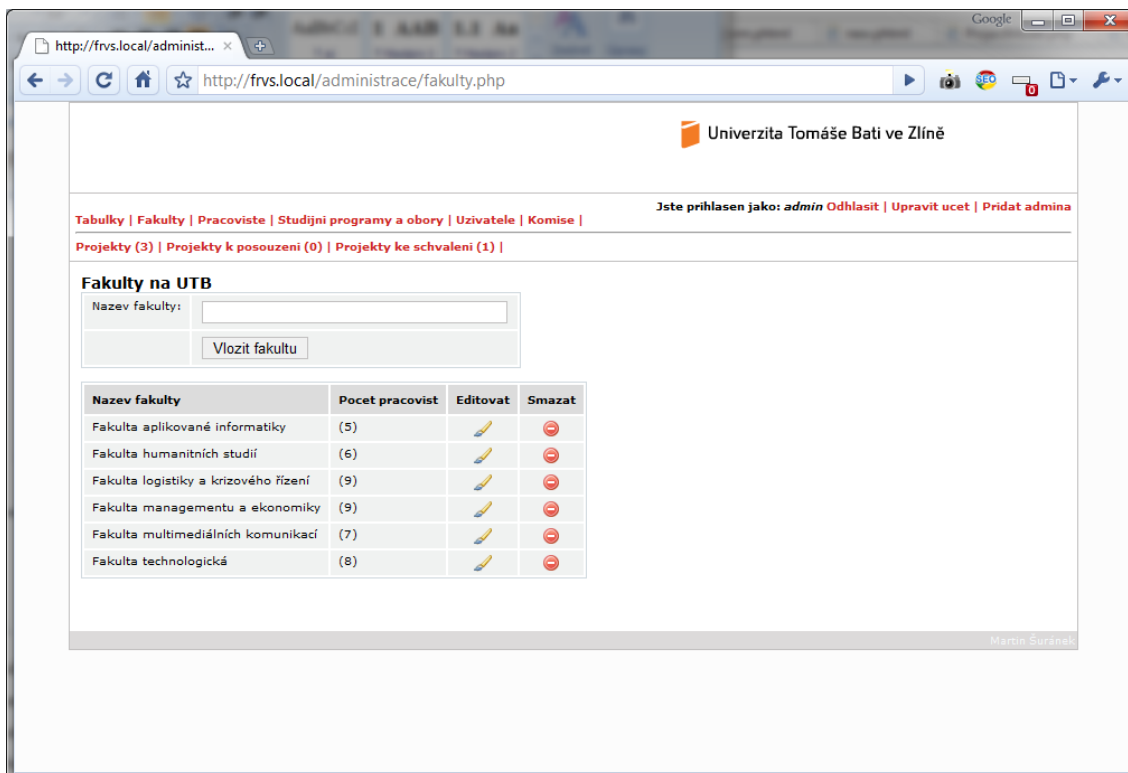
Obrázek 11 - Správa projektů

## 2. Administrační část

Administrační část slouží převážně ke správě uživatelů a projektů. Dále k určení účastníků komise, přidělování oponentů k projektům. V neposlední řadě poskytuje administrační rozhraní možnost spravovat údaje o fakultě, jejich ústavech (pracovištích) a studijních oborech.

### 2.1. Správa fakult

Správa fakult je v menu umístěna pod položkou „Fakulty“. Kde má administrátor možnost přidávat nové fakulty, případně je upravovat, nebo je může úplně odstranit. Pro přehlednost je zde administrátorovi zobrazen u každé fakulty počet pracovišť, které pod aktuální fakultu spadají.



Obrázek 12 - Správa fakult

## 2.2. Správa pracovišť

Tuto akci nalezneme v menu pod položkou „Pracoviště“. Zde administrátor přiřazuje ke zvoleným fakultám pracoviště. Pojmem pracoviště jsou zde míněny ústavy na fakultě a detašované pracoviště fakult. Z informací o pracovištích se ukládá: název pracoviště, město, ulice, čp a psč. Takto vytvořené pracoviště zde může administrátor libovolně upravovat či přímo odstraňovat.

Univerzita Tomáše Bati ve Zlíně

Jste přihlazen jako: *admin* [Odhlásit](#) | [Upravit účet](#) | [Přidat admina](#)

[Tabulky](#) | [Fakulty](#) | [Pracoviste](#) | [Studijní programy a obory](#) | [Uzivatele](#) | [Komise](#) |

[Projekty \(3\)](#) | [Projekty k posouzení \(0\)](#) | [Projekty ke schválení \(1\)](#) |

### Pracoviste na fakultach UTB

Zvolte fakultu:

Nazev pracoviste:

Mesto:

Ulice:

CP:

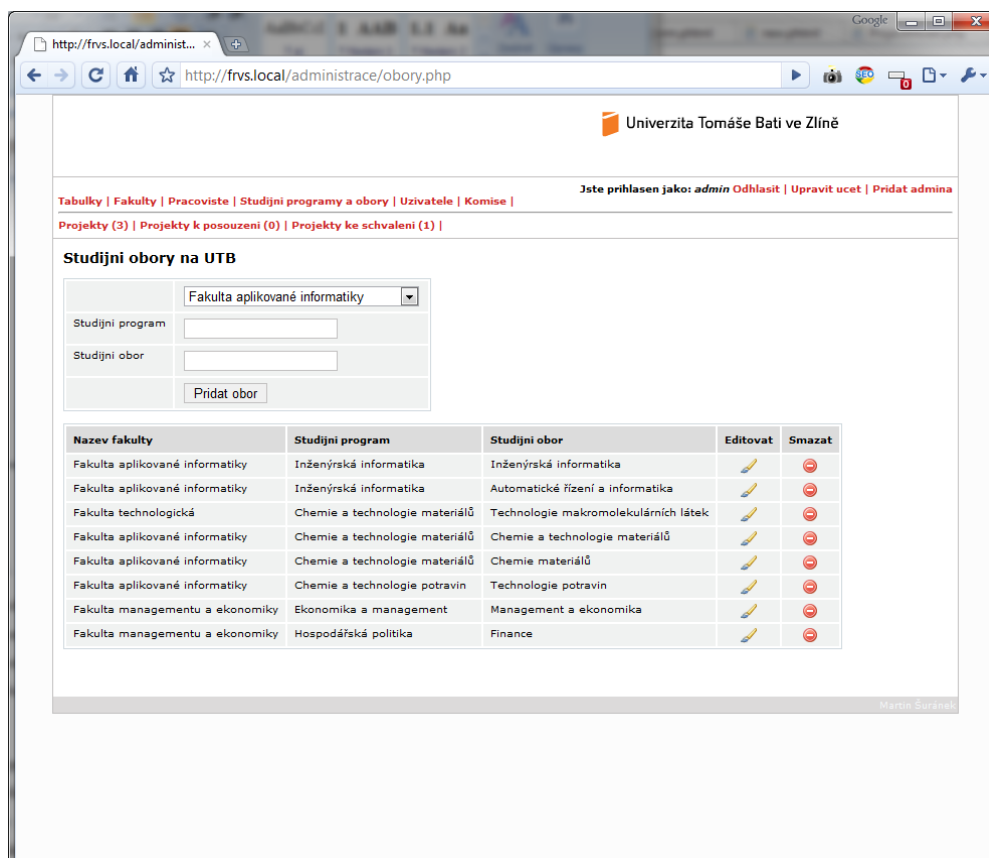
PSC:

Fakulta	Pracoviste	Adresa	Editovat	Smazat
Fakulta logistiky a krizového řízení	detašované pracoviště Praha - Vinoř	Vinořské náměstí 22 Praha 190 17		
Fakulta logistiky a krizového řízení	detašované pracoviště RVC Prostějov	Husovo náměstí 91 Prostějov 796 04		
Fakulta logistiky a krizového řízení	detašované pracoviště Zlín, UTB	náměstí T. G. Masaryka 588 Zlín 760 01		
Fakulta multimediálních komunikací	Kabinet teoretických studií	nám. T. G. Masaryka 1279 Zlín 760 01		
Fakulta humanitních studií	Ústav anglistiky a amerikanistiky	Mostní 5139 Zlín 760 05		
Fakulta multimediálních komunikací	Ústav animace a audiovizí	Štefánikova 2431 Zlín 760 01		
Fakulta aplikované informatiky	Ústav aplikované informatiky	Nad Stráněmi 4511 Zlín 760 06		
Fakulta humanitních studií	Ústav aplikovaných společenských věd	Mostní 5139 Zlín 760 01		
Fakulta aplikované informatiky	Ústav automatizace a řídicí techniky	Nad Stráněmi 4511 Zlín 760 05		

Obrázek 13 - Správa pracovišť

### 2.3. Správa studijních programů a oborů

Tato možnost je skryta pod položkou menu s názvem „Studijní programy a obory“. Administrátor zde vytváří studijní programy a studijní obory na zvolených fakultách. Samozřejmě je následná možnost úpravy a odstraňování nevhodných, nebo neplatných fakult.



Obrázek 14 - Správa studijních programů a oborů

## 2.4. Správa uživatelů

Volba je umístěna pod položkou menu „Uživatelé“, administrátor zde může vytvářet účty pro uživatele, takto vytvořené účty se vytváří s přednastaveným heslem, které je vytvořenému uživateli sděleno a je upozorněn na to, aby svůj účet co nejdříve upravil změnou hesla. Dále je administrátorovi vypsán seznam již zaregistrovaných uživatelů, které má možnost filtrovat podle rolí uživatelů. Administrátor zde plní funkci schvalování vytvořených účtů, tím se předchází situaci, kdy by mohl kdokoliv po úspěšné registraci vytvářet projekty (příhlášky do grantové soutěže).

Univerzita Tomáše Bati ve Zlíně

Jste přihlazen jako: **admin** [Odhlásit](#) | [Upravit účet](#) | [Přidat admina](#)

[Tabulky](#) | [Fakulty](#) | [Pracovisté](#) | [Studijní programy a obory](#) | [Uživatelé](#) | [Komise](#) |

[Projekty \(3\)](#) | [Projekty k posouzení \(0\)](#) | [Projekty ke schválení \(1\)](#) |

### Registrování uživatele frvs na UTB

Uživatelum, které takto vytvoříte bude vytvořeno defaultní heslo: "heslo" po přihlase si jej mohou samozřejmě změnit.

Zvolte fakultu

Role

Student

Zaměstnanec

Učastník komise

Přihlasovací jméno

Jméno

Příjmení

Tituly před jménem

Tituly za jménem

Rodné číslo

E-mail

Telefon (+xxx xxx xxx)

E-mail (Ustav)

Telefon (Ustav) (+xxx xxx xxx)

Zaregistrovat

Studenti:  Zaměstnanci:  Učastníci komise:  Filtrovat

Uživatel	Role	E-mai	Tel.	R.C.	Schvalit / Zamítnout
Ing. Karel Jonatan Phd.	Zaměstnanec	k@jonatan.cz	+420576035186	840815/4260	✘
doc. RNDr. Vojtěch Křesálek CSc.	Učastník komise	kresalek@fai.utb.cz	+42057603203	840517/9090	✘
Ing. Michal Proháčka	Student	mprochazka@fai.utb.cz	+420576035186	840517/9090	✔
doc., Ing. Zdenka Prokopová CSc.	Zaměstnanec	prokopova@fai.utb.cz	+420576035186	656120/2340	✘
Ing. Martin Sysel Phd.	Zaměstnanec	msysel@fai.utb.cz	+420576035186	840815/4260	✘
Ing. Pavel Varacha Phd.	Učastník komise	varacha@fai.utb.cz	+420576035186	840517/4260	✘

Obrázek 15 - Správa uživatelů

## 2.5. Správa komise

Správa komise je umístěna v menu „Komise“. Administrátor zde vytváří komisi, která bude rozhodovat o udělování grantů vybraným projektům. Komise by měla být tříčlenná, její složení by mělo být podle interního manuálu následující:

- Předseda komise: prorektor pro tvůrčí činnosti
- 1. člen komise: proděkan pro vědu a výzkum
- 2. člen komise: proděkan pro vědu a výzkum

Nicméně toto uspořádání komise není v systému striktně povinné.

## 2.6. Správa Projektů

Správa projektů je zde rozdělena mezi tři následující položky menu: „Projekty“, „Projekty k posouzení“, „Projekty ke schválení“. První položka (Projekty) slouží k výpisu všech projektů, které se v systému nachází. Administrátor zde může prohlížet jak rozpracované tak dokončené projekty. Druhá položka (Projekty k posouzení) obsahuje výpis projektů, které jsou ve finálním stavu a čekají tak na posudky oponentů. Zde musí administrátor tyto oponenty přidělit k projektům, tato akce je realizována pomocí výběrového HTML prvku SELECT, do kterého jsou automaticky vybráni uživatelé, kteří odpovídají podmínkám pro oponenturu daného projektu, na administrátorovi je už jen zvolit si z vyhovujících uživatelů. Třetí položka je věnována projektům, které mají již oba dva posudky od oponentů ve finálním stavu a jsou tak připraveny k předání komisi. Účastníkům komise pak budou tyto projekty zobrazeny po přihlášení do systému a budou je tak moci schválit nebo zamítnout.

Univerzita Tomáše Bati ve Zlíně

Jste přihlášen jako: **admin** [Odhlásit](#) | [Upravit účet](#) | [Přidat admina](#)

[Tabulky](#) | [Fakulty](#) | [Pracoviste](#) | [Studijní programy a obory](#) | [Uzivatele](#) | [Komise](#) |

[Projekty \(3\)](#) | [Projekty k posouzení \(0\)](#) | [Projekty ke schválení \(1\)](#)

### Vypis projektu

Vypis projektu, které již byly posouzeny.

Nazev projektu	Detail	Autor	Stav dokončení	Stav posouzení	První posudek	Druhy posudek
nevim	<a href="#">i</a>	Ing. Michal Proházka	Dokončen	Posouzen	<a href="#">Zobraz</a>	<a href="#">Zobraz</a>

### Posudek grantové přihlášky

Jméno: Ing. Michal Proházka Reg. c.

**Posuzovatel:**

Příjmení:  Pracoviste:

Jméno:

Tituly:  Telefon:  Email:

Popište Váš profesionální vztah k navrhovateli:

Uvedte prosím Vaše odborné zaměření a užší specializaci:

Prohlašuji, že při posuzování tohoto projektu jsem se řídil(a) výhradně objektivními hledisky. Jsem si vědom(a) toho, že posuzovaný návrh a podkladové materiály jsou důvěrné povahy a nepředkládám(a) jsem je třetím osobám, nepoživoval(a) z nich kopie a spolu s posudkem je v úplnosti vracím. Při posuzování jsem dodržel(a) zásadu anonymity posuzovatele.

Body **1. Původnost a vědecká závažnost projektu.** Vysoce originální projekt, přináší zcela nové myšlenkové pohledy a přístupy v oboru, nebo pouze prohlubuje existující znalosti, anebo nepřináší nic nového pro výzkum. V čem spatřujete originalitu navrženého způsobu řešení?

Původnost

Body **2. Aktuálnost a perspektivnost projektu.** Hodnocení jako vysoce aktuální projekt, spadající do perspektivní vědní oblasti, nebo projekt spadá do relativně probírané oblasti, ale aktuálnost je omezená, anebo výzkum nepovažují za účelný. V čem spatřujete aktuálnost a perspektivnost projektu, příp. proč lze pochybovat o účelnosti tohoto výzkumu?


Aktualnost

Obrázek 16 - Správa projektů k posouzení

### 4.3.2 Pracovní pokyny v systému

Pracovní pokyny v systému jsou realizovány pomocí vykreslení dialogového okna s nápovědou po najetí myši na vyznačená místa, které jsou určeny k dostatečnému informování uživatele o akci, kterou se pokouší provést. Toto je řešeno pomocí jednoduchého JavaScriptu pro zobrazení tipů, nápovědy.

#### Nový projekt

Žádost o udělení interního grantu - A			
Datum převzetí žádosti oddělením VaV:	<input type="text"/>	Reg. číslo*	<input type="text"/>
Název projektu 	<input type="text"/>		
Navrhovatel:	Zde zadejte název projektu.		Kontakt: (e-mail, tel.)
Příjmení Prokopová	Jméno Zdenka	prokopova@fai.utb.cz +420576035186	
1 Spolunavrhovatel:	Kontakt: (e-mail, tel.)		

Obrázek 17 – Ukázka pracovních pokynů v systému



## 5 IMPLEMENTACE A TESTOVÁNÍ SYSTÉMU

### 5.1 Implementace systému

Implementace systému se dá rozdělit do několika částí:

- kontrola konfigurace serveru a PHP pomocí Requirements-Checker
- vytvoření databáze na serveru
- úprava konfiguračního souboru (informace pro připojení k databázi atd.)
- nakopírování souborů na FTP server
- přístup do administrace (při prvním přihlášení je automaticky vytvořen administrátorský účet, administrátor si jej pak v administraci může změnit)
- tvorba tabulek v administraci

#### 5.1.1 Requirements-Checker

Requirements-Checker je nástroj, který slouží ke kontrole, zda váš server a PHP konfigurace splňuje požadavky pro běh Nette Frameworku. Po spuštění tohoto skriptu se nám dostane přehledného zobrazení případných konfliktů, které brání tomu, aby aplikace na Nette Frameworku běžela. Máme tak ihned přehled o tom, co máme v konfiguraci serveru, případně PHP změnit aby aplikace postavená na Nette Frameworku mohla bez problémů fungovat.















## Nette Framework Requirements Checker

This script checks if your server and PHP configuration meets the requirements for running [Nette Framework](#).

**Congratulations! Server configuration meets the minimum requirements for Nette Framework.**

Please see the warnings listed below.

### Details

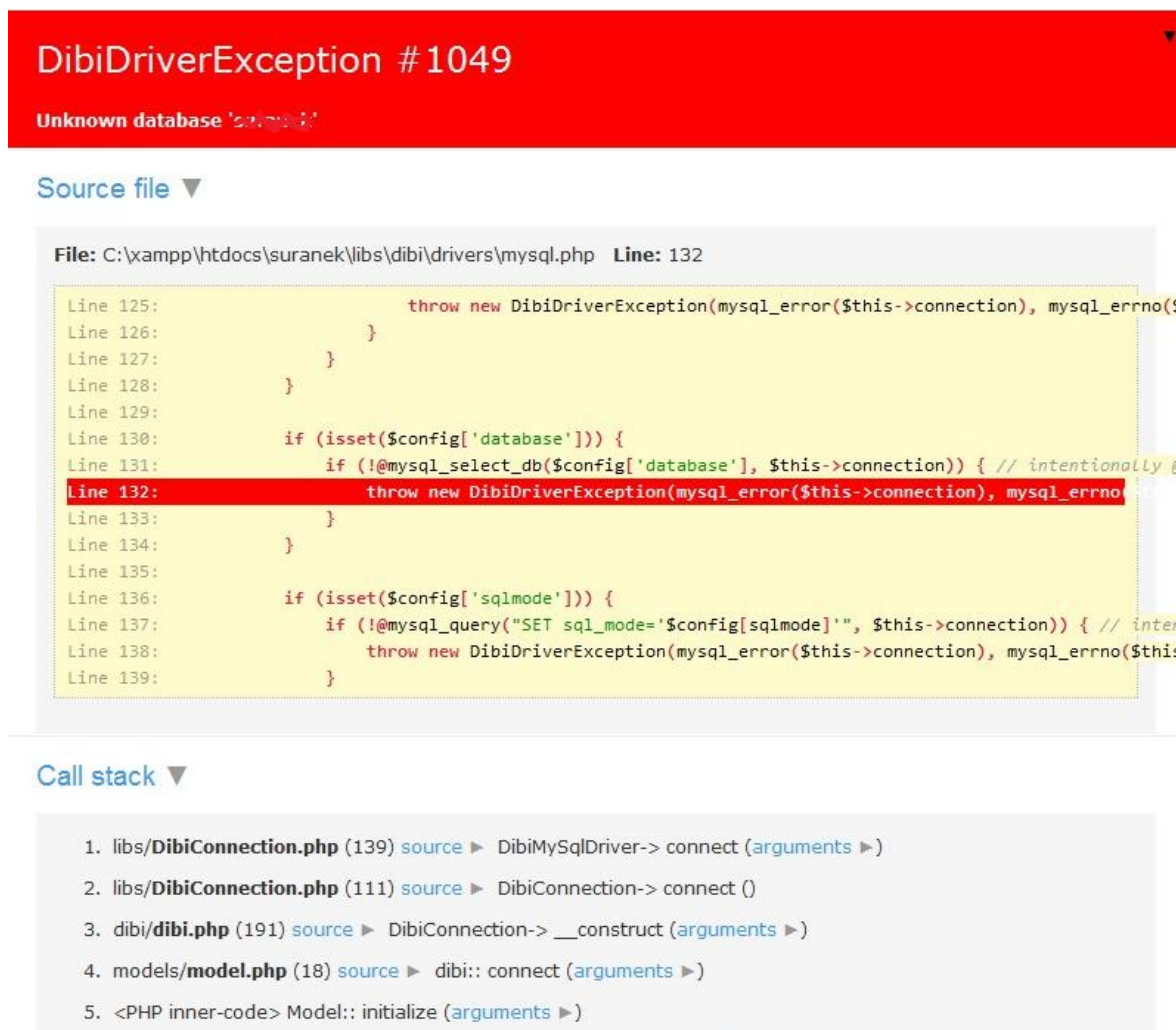
	<b>Web server</b>	Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14 OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
	<b>PHP version</b>	5.3.1
	<b>Memory limit</b>	128M
	<b>.htaccess file protection</b>	
	<b>Function ini_set</b>	Enabled
	<b>Magic quotes</b>	Disabled
	<b>Register_globals</b>	Disabled
	<b>Zend.ze1_compatibility_mode</b>	Disabled
	<b>Variables_order</b>	Enabled
	<b>Reflection extension</b>	Enabled
	<b>Reflection phpDoc</b>	Enabled
	<b>SPL extension</b>	Enabled
	<b>PCRE extension</b>	Enabled
	<b>ICONV extension</b>	Enabled and works properly

Obrázek 18 - Requirements-Checker

## 5.2 Testování systému

### 5.2.1 Zachytávání chyb a výjimek

Ladění systému se dá rozdělit do dvou částí. Ladění na vývojovém serveru a ladění produkčním serveru. V Nette Frameworku se ladění aktivuje přidáním řádku `Debug::Enable()` do zaváděcího souboru `bootstrap.php`. Tím aktivujeme ladící nástroj tzv. *laděnku*, která sama detekuje, zda běží na vývojovém nebo produkčním serveru. Navíc je s pomocí *laděnky* odladění systému mnohem příjemnější záležitost.



**DibiDriverException #1049**

Unknown database 'suranek'

Source file ▼

File: C:\xampp\htdocs\suranek\libs\dibi\drivers\mysql.php Line: 132

```
Line 125:         throw new DibiDriverException(mysql_error($this->connection), mysql_errno($
Line 126:     }
Line 127: }
Line 128: }
Line 129:
Line 130:     if (isset($config['database'])) {
Line 131:         if (!@mysql_select_db($config['database'], $this->connection)) { // intentionally e
Line 132:             throw new DibiDriverException(mysql_error($this->connection), mysql_errno
Line 133:         }
Line 134:     }
Line 135:
Line 136:     if (isset($config['sqlmode'])) {
Line 137:         if (!@mysql_query("SET sql_mode='".$config['sqlmode']."' ", $this->connection)) { // inten
Line 138:             throw new DibiDriverException(mysql_error($this->connection), mysql_errno($this
Line 139:     }
```

Call stack ▼

1. libs/DibiConnection.php (139) source ► DibiMySqlDriver-> connect (arguments ►)
2. libs/DibiConnection.php (111) source ► DibiConnection-> connect ()
3. dibi/dibi.php (191) source ► DibiConnection-> \_\_construct (arguments ►)
4. models/model.php (18) source ► dibi:: connect (arguments ►)
5. <PHP inner-code> Model:: initialize (arguments ►)

Obrázek 19 – Vzhled ladícího nástroje laděnka

## 5.2.2 Logování chyb

Na produkčním serveru je ale *laděnka* spíše na škodu. Mohla by totiž říct návštěvníkovi portálu mnohem víc než je potřeba. Proto se na produkčním serveru logují chyby do souboru, nebo se dají posílat mailem administrátorovi portálu, samozřejmě je možnost obojího logování. Tyto logy mají opět přehledný vzhled jako u *laděnky*.

## ZÁVĚR

V této diplomové práci jsem se zabíral problematikou tvorby webového portálu pro správu projektů. Po nastudování směrnice rektora Zásady studentské grantové soutěže Univerzity Tomáše Bati ve Zlíně, jsem si udělal představu o způsobu zpracování přihlášek do grantové soutěže. Na základě těchto informací jsem si navrhl role pro uživatele, které budou systém používat. Následně jsem si vyžádal formuláře, které se používají pro tvorbu přihlášek do grantové soutěže a pro posudky k těmto přihláškám. Tyto formuláře bylo nutno přepsat do HTML, aby je bylo možné použít pro potřeby portálu. Formuláře byly hodně rozsáhlé, a proto bylo jejich vyplňování pro uživatele zpříjemněno tím, že se nevyplňují najednou, ale jsou rozděleny do několika kroků. Tímto způsobem jsem postupoval jak při vyplňování tak i u prohlížení přihlášek a jejich posudků. Přihlášky jsou ve vytvořeném systému pojmenovány jako projekty.

Životní cyklus projektu je následující. Po vytvoření projektu uživatelem je projekt veden jako rozpracovaný, uživatel má tak možnost svůj projekt měnit či jej úplně odstranit. Takto rozpracovaných projektů má možnost uživatel vytvořit tolik, kolik sám potřebuje. Ale jen jeden jediný projekt může být označen jako finální. Ten už uživatel nebude mít možnost nijak měnit, bude mít pouze možnost sledovat stav projektu. Finální projekt je pak poslán k přidělení oponentů (to má na starosti administrátor systému). Oponenti pak mají za úkol k projektu vytvořit posudky. Vytváření těchto posudků má podobný průběh jako tvorba projektů. Posudek může být též libovolně upravován, dokud není označen jako finální. Část posudku ke svému projektu si pak může autor projektu zobrazit v přehledu svých projektů. V situaci kdy má projekt oba posudky od obou oponentů ve finálním stavu, je projekt předán k posouzení komisí (ta je vytvořena v administraci), která rozhodne, zda se projekt schválí, či zamítne. Výsledek tohoto posouzení má pak autor projektu k dispozici obdobně jako posudky oponentů.

Dalším úkolem bylo vytvoření podpory uživatelů. Tento úkol jsem rozdělil do dvou částí. Nejdříve jsem podporu uživatelů naimplementoval do portálů pomocí informačních oken, které se zobrazují po najetí myši nad vybrané prvky, které by mohli uživateli dělat problém. Poté jsem vytvořil stručný manuál k portálu, který by měl sloužit k zasvěcení uživatelů pro používání systému.

Výstupem této diplomové práce je funkční webový portál, který naleznete na adrese <http://iga.suranek.cz> a na přiloženém CD.

## ZÁVĚR V ANGLIČTINĚ

In this thesis, I undertake the problems of creating a Web portal for project management. After preparation of the Rector Principles student grant competition, Tomas Bata University in Zlin, I made an idea about how to process applications for the grant competition. Based on this information, I have proposed a role for users who will use the system. Subsequently, I requested the forms to be used for creating applications for the grant competition and the opinions of those applications. These forms had to be rewritten into HTML so they can be used for the needs of the portal. The forms were very extensive, and it was filling in for their users make good time that shall not be completed all at once, but they are divided into several steps. In this way, I proceeded to fill in the applications and also in view of their opinions. Applications are in a system named as projects.

Project life cycle is as follows. After creating a project by project is being managed as in-process, the user has the option to change your project, or completely remove it. Thus unfinished projects user has the option to create as much as he needs. But just one project can be marked as final. The user will no longer be able to not change, will only have the ability to monitor project status. The final project is then sent to the allocation of opponents (which is responsible for the system administrator). Opponents then have the task to create project reports. Creating these opinions has followed a similar pattern as the creation of projects. No one may also be arbitrarily adjusted until it is marked as final. Part of their assessment of the project can then view the author of the project in the list of their projects. In situations where a project has both expertise from both opponents in the final state, the project is submitted for consideration by the Commission (which is made up of administration), which decides whether the project is approved or rejected. The result of this assessment is the author of the project are available like the opinions of opponents.

The next task was to support users. This task has been divided into two parts. First I put in user support portal with information windows that appear after hovering over the selected elements, which could make users of the problem. Then I created a brief guide to the portal, which should serve for the dedication of the users use the system.

The outcome of this thesis is a functional Web portal, which can be found at <http://iga.suranek.cz> and on CD.

**SEZNAM POUŽITÉ LITERATURY**

- [1] PROKOPOVÁ, Zdenka. Databázové systémy MySQL+PHP. FAI UTB Zlín, 126 s. , 2006, Vysokoškolská skripta. ISBN 80-7318-486-9.
- [2] JESUS, Castagnetto, et al. Programujeme profesionálně : PHP. Robert FE Shaw. 2004. autoriz. vyd. Praha : Computer Press, 2004. 656 s., CD-ROM. ISBN 80-7226-310-2.
- [3] JavaScript a Ajax : Moderní programování webových aplikací. 1. vyd. Praha: Computer Press, a.s., 2007. 360 s. ISBN 978-80-251-1824-5.
- [4] LACKO, Luboslav. SQL: hotová řešení. Praha : Computer press, 2003. 296 s. ISBN 80-7226-975-5.
- [5] HAVLENKA, J. a kol. Vytváříme WWW stránky a spravujeme moderní web site. Computer Press, 2006, ISBN: 80-251-0801-5.
- [6] KOSEK, J. HTML - tvorba dokonalých WWW stránek. Grada Publishing, 2006, ISBN: 80-7169-608-0.
- [7] VALADE, J. PHP5 for dummies. Wiley Publishing, 2004, ISBN: 0-7645-4166-8.
- [8] HONZA Ignác, rektor. Zásady studentské grantové soutěže Univerzity Tomáše Bati ve Zlíně. In *Zásady studentské grantové soutěže Univerzity Tomáše Bati ve Zlíně*. SR/21/2009. Univerzita Tomáše Bati ve Zlíně : UTB, 2009. s. 6.
- [9] GRUDL, David. Nette Framework: zvyšte svoji produktivitu. *Zdroják.cz* [online]. 10. 3. 2009, n. 1, [cit. 2010-05-19]. Dostupný z WWW: <<http://zdrojak.root.cz/clanky/nette-framework-zvyste-svoji-produktivitu/>>.
- [10] Model-View-Presenter (MVP). *Nettephp.com* [online]. 2009, n.1, [cit. 2010-05-19]. Dostupný z WWW: <<http://doc.nettephp.com/cs/model-view-presenter>>.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

HTML HyperText Markup Language.

CSS Cascading Style Sheet.

PHP Hypertext Preprocessor.

HTTP Hypertext Transfer Protokol.

SQL Structured query language.

URL Uniform Resource Locator.

MVP Model-View-Presenter.

FTP File Transfer Protokol

**SEZNAM OBRÁZKŮ**

Obrázek 1 - MVC model .....	20
Obrázek 2 - PSPad .....	22
Obrázek 3 - Eclipse Galileo .....	23
Obrázek 4 - SQLyog .....	24
Obrázek 5 - PhotoFiltre.....	25
Obrázek 6 - schéma relací v databázi .....	27
Obrázek 7 - Databáze iga.....	28
Obrázek 8 - Základní rozložení.....	47
Obrázek 9 - Registrace uživatele .....	48
Obrázek 10 - Vytváření projeků.....	49
Obrázek 11 - Správa projektů .....	50
Obrázek 12 - Správa fakult .....	51
Obrázek 13 - Správa pracovišť .....	52
Obrázek 14 - Správa studijních programů a oborů .....	53
Obrázek 15 - Správa uživatelů.....	54
Obrázek 16 - Správa projektů k posouzení .....	55
Obrázek 17 – Ukázka pracovních pokynů v systému .....	56
Obrázek 18 - Requirements-Checker.....	58
Obrázek 19 – Vzhled ladícího nástroje laděnka .....	59



**SEZNAM TABULEK**

Tabulka 1 - tabulka s_settings.....	28
Tabulka 2 - tabulka faculties.....	29
Tabulka 3 - tabulka wokplace.....	29
Tabulka 4 - tabulka student.....	30
Tabulka 5 - tabulka study_course.....	30
Tabulka 6 - tabulka study_program.....	31
Tabulka 7 - tabulka users.....	31
Tabulka 8 – část tabulky project.....	32
Tabulka 9- tabulka opponency.....	34
Tabulka 10 - tabulka committee_result.....	35