UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Aplikace na správu rozvrhů*

Seminární práce z předmětu Databázové systémy II

Pracovní tým: Tomáš Křičenský, Michal Struna

V Pardubicích dne 18. 12. 2018

Obsah

[Úvod 10](#_Toc439267758)

[Zadání 11](#_Toc439267759)

[Uživatelská dokumentace 12](#_Toc439267760)

[Základní popis používané aplikace 13](#_Toc439267761)

[Instalace 13](#_Toc439267762)

[Přístupová oprávnění 13](#_Toc439267763)

[Použití aplikace 13](#_Toc439267764)

[Programová dokumentace 14](#_Toc439267765)

[Datová část 15](#_Toc439267766)

[Analýza 15](#_Toc439267767)

[Fyzický model dat 15](#_Toc439267768)

[Číselníky 15](#_Toc439267769)

[Pohledy 15](#_Toc439267770)

[Funkce 15](#_Toc439267771)

[Uložené procedury 15](#_Toc439267772)

[Spouště 15](#_Toc439267773)

[Indexy 15](#_Toc439267774)

[Sekvence 15](#_Toc439267775)

[Systémový katalog 15](#_Toc439267776)

[Aplikace 16](#_Toc439267777)

[Použité prostředí 16](#_Toc439267778)

[Řízení uživatelských účtů 16](#_Toc439267779)

[Moduly 16](#_Toc439267780)

[Formuláře 16](#_Toc439267781)

[Orientace ve zdrojovém kódu 16](#_Toc439267782)

[Závěr 17](#_Toc439267783)

Úvod

Obsahuje základní seznámení s organizací, procesy uvnitř organizace a způsob současného využívání IT.

# Zadání

Obsahuje popis prostředí, v němž bude aplikace nasazena (stručná charakteristika organizace a dosavadní využívání IT), a požadavky na nově tvořený systém (sledované údaje, vstupy a výstupy, přístupová práva, ...)

Uživatelská dokumentace

## Základní popis používané aplikace

Webová aplikace slouží ke správě rozvrhů a úvazků na vysoké škole. Aplikace eviduje předměty, vyučující včetně jejich role v předmětu, studijní plány i jednotlivé rozvrhové akce. Aplikace dokáže vypočítat úvazky jednotlivých vyučujících podle jejich rozvrhových akcí. Vyučující si také může navrhnout vlastní rozvrh, který mu poté schválí administrátor.

Aplikace neumožňuje správu zkouškových termínů nebo studijních materiálů. Studenti (nepřihlášení uživatelé) si pouze mohou zobrazit veřejné informace o předmětech a mohou si zobrazit rozvrh svého studijního plánu.

## Instalace

Uveďte informace o instalaci aplikace.

## Přístupová oprávnění

Aplikace má 5 stupňů přístupových oprávnění: neregistrovaný uživatel, registrovaný uživatel, vyučující, vyučující administrátor a administrátor. Neregistrovaný uživatel si může zobrazit skoro všechny dostupné informace (např. rozvrhové akce, informace o vyučujících, o předmětech apod.), ale nemůže v systému nic změnit. Registrovaný uživatel si navíc může upravit přihlašovací e-mail a heslo.

Dalším stupněm oprávnění je vyučující. Vyučující může vše, co registrovaný uživatel a také si může navrhnout vlastní rozvrh, upravovat svůj profil učitele nebo se podívat na vlastní úvazek.

Nejvyšší oprávnění má administrátor, který může spravovat uživatele, schvalovat rozvrhové akce nebo upravovat data ve všech tabulkách (předměty, studijní plány aj.).

Uživatel s oprávněním "vyučující administrátor" může v systému dělat vše stejně jako administrátor, ale má přístup i ke všem učitelským možnostem.

|  |  |  |
| --- | --- | --- |
| Role | Email | Heslo |
| Registrovaný uživatel | test@test.cz | test |
| Učitel | teacher@teacher.cz | teacher |
| Učitel administrátor | teacher@admin.cz | admin |
| Administrátor | admin@admin.cz | admin |

## Použití aplikace

Sepište jednoduchý návod na používání aplikace, rozdělte jej na jednotlivé moduly aplikace. Z této části musí být jasné, jak je možné program využívat.

Programová dokumentace

# Datová část

Zde uveďte řešení zadavatelské dokumentace z pohledu datové části, postupujte dle zadání projektu.

## Analýza

Bude obsahovat entitně-vztahový diagram (ERD) a datový model popisující datové prostředí aplikace a dále stručnou charakteristiku uživatelského rozhraní (typy formulářů, tiskových sestav, .), které bude nad zmíněným datovým prostředím pracovat.

## Fyzický model dat

Uveďte logický datový model a datový slovník. Je možné vložit printscreeny datových struktur s popisem (informacemi) o jednotlivých tabulkách.

## Číselníky

Uveďte jaké číselníky Váš projekt využívá a zdroj čerpání těchto číselníků.

## Pohledy

V projektu je vytvořeno celkem 10 pohledů.

**CREATE OR REPLACE VIEW** sem\_p\_katedra **AS  
SELECT** sem\_katedra.*\**, sem\_fakulta.zkratka **AS** "fakulta"  
**FROM** sem\_katedra  
**JOIN** sem\_fakulta **ON** sem\_katedra.fakulta\_id = sem\_fakulta.id  
**ORDER BY** sem\_katedra.nazev;

*Zdrojový kód 1: Pohled pro výpis kateder a jejich fakult.*

**CREATE OR REPLACE VIEW** SEM\_P\_UCITEL **AS  
SELECT** sem\_ucitel.*\**, (titul\_pred || **' '** || jmeno || **' '** || prijmeni || **' '** || titul\_za) **as** "dlouhe\_jmeno", sem\_katedra.zkratka **as** "katedra"  
**FROM** sem\_ucitel  
**JOIN** SEM\_KATEDRA **ON** sem\_katedra.id = SEM\_UCITEL.katedra\_id  
**ORDER BY** sem\_ucitel.prijmeni, sem\_ucitel.jmeno;

*Zdrojový kód 2: Pohled pro výpis učitelů, jejich titulů a kateder.*

**CREATE OR REPLACE VIEW** SEM\_P\_PREDMET **AS  
SELECT** sem\_predmet.*\**, sem\_zpus\_zak.nazev **as** "zpusob\_zakonceni", sem\_forma\_vyuky.nazev **as** "forma\_vyuky"  
**FROM** SEM\_PREDMET  
**JOIN** sem\_zpus\_zak **ON** sem\_zpus\_zak.ID = sem\_predmet.zpusob\_zakonceni\_id  
**JOIN** sem\_forma\_vyuky **ON** SEM\_FORMA\_VYUKY.id = sem\_predmet.forma\_vyuky\_id  
**ORDER BY** sem\_predmet.nazev;

*Zdrojový kód 3: Pohled pro výpis předmětů, jejich způsobů zakončení a forem výuky.*

**CREATE OR REPLACE VIEW** SEM\_P\_PREDM\_PLAN **AS  
SELECT** SEM\_PREDM\_PLAN.*\**, sem\_kategorie.nazev **as** "kategorie", SEM\_STUD\_PLAN.nazev **as** "plan", sem\_predmet.nazev **as** "predmet", sem\_semestr.nazev **as** "semestr"  
**FROM** SEM\_PREDM\_PLAN  
**JOIN** SEM\_KATEGORIE **ON** SEM\_KATEGORIE.ID = SEM\_PREDM\_PLAN.KATEGORIE\_ID  
**JOIN** SEM\_STUD\_PLAN **ON** SEM\_STUD\_PLAN.ID = SEM\_PREDM\_PLAN.STUDIJNI\_PLAN\_ID  
**JOIN** SEM\_SEMESTR **ON** SEM\_SEMESTR.ID = SEM\_PREDM\_PLAN.SEMESTR\_ID  
**JOIN** sem\_predmet **ON** sem\_predmet.ID = SEM\_PREDM\_PLAN.predmet\_id  
**ORDER BY** SEM\_STUD\_PLAN.nazev, sem\_predmet.nazev, sem\_kategorie.nazev, rocnik;

*Zdrojový kód 4: Pohled pro výpis předmětů v plánu, jejich semestrů a kategorií.*

**CREATE OR REPLACE VIEW** SEM\_P\_ZPUS\_PREDM **AS  
SELECT** SEM\_ZPUS\_PREDM.*\**, SEM\_ZPUS\_VYUKY.nazev **as** "zpusob\_vyuky", (SEM\_STUD\_PLAN.NAZEV || **' '** || SEM\_PREDMET.NAZEV) **as** "PREDM\_PLAN"  
**FROM** SEM\_ZPUS\_PREDM  
**JOIN** SEM\_ZPUS\_VYUKY **ON** SEM\_ZPUS\_VYUKY.id = SEM\_ZPUS\_PREDM.zpusob\_vyuky\_id  
**JOIN** SEM\_PREDM\_PLAN **ON** SEM\_PREDM\_PLAN.id = SEM\_ZPUS\_PREDM.PREDM\_PLAN\_id  
**JOIN** SEM\_STUD\_PLAN **ON** SEM\_STUD\_PLAN.id = SEM\_PREDM\_PLAN.STUDIJNI\_PLAN\_ID  
**JOIN** SEM\_PREDMET **ON** SEM\_PREDMET.id = SEM\_PREDM\_PLAN.predmet\_id  
**ORDER BY** SEM\_STUD\_PLAN.nazev, sem\_predmet.nazev, sem\_zpus\_vyuky.nazev;

*Zdrojový kód 5: Pohled pro výpis způsobů předmětů a jejich výuky.*

**CREATE OR REPLACE VIEW** SEM\_P\_UCI **AS  
SELECT** SEM\_UCI.*\**, (SEM\_UCITEL.JMENO || **' '** || SEM\_UCITEL.PRIJMENI) **as** "ucitel", SEM\_ROLE.NAZEV **as** "role", (SEM\_STUD\_PLAN.NAZEV || **' - '** || SEM\_PREDMET.NAZEV) **as** "predmet"  
**FROM** SEM\_UCI  
**JOIN** SEM\_UCITEL **ON** SEM\_UCITEL.ID = SEM\_UCI.UCITEL\_ID  
**JOIN** SEM\_ROLE **ON** SEM\_ROLE.ID = SEM\_UCI.ROLE\_ID  
**JOIN** SEM\_PREDM\_PLAN **ON** SEM\_PREDM\_PLAN.ID = SEM\_UCI.PREDM\_PLAN\_ID  
**JOIN** SEM\_PREDMET **ON** SEM\_PREDMET.ID = SEM\_PREDM\_PLAN.PREDMET\_ID  
**JOIN** SEM\_STUD\_PLAN **ON** SEM\_STUD\_PLAN.ID = SEM\_PREDM\_PLAN.STUDIJNI\_PLAN\_ID  
**ORDER BY** "predmet", "ucitel", "role";

*Zdrojový kód 6: Pohled pro výpis učitelů a jejich předmětů.*

**CREATE OR REPLACE VIEW** SEM\_P\_UZIVATEL **AS  
SELECT** SEM\_UZIVATEL.ID **as** "id", email, **admin**, (SEM\_UCITEL.JMENO || **' '** || SEM\_UCITEL.PRIJMENI) **as** "ucitel", SEM\_BOOL\_TO\_STRING(**admin**) **as** "je\_admin"  
**FROM** SEM\_UZIVATEL  
**LEFT JOIN** SEM\_UCITEL **ON** SEM\_UCITEL.ID = SEM\_UZIVATEL.UCITEL\_ID  
**ORDER BY** email;

*Zdrojový kód 7: Pohled pro výpis uživatelů včetně označení administrátorů.*

**CREATE OR REPLACE VIEW** SEM\_P\_STUD\_PLAN **AS  
SELECT** SEM\_STUD\_PLAN.*\**, SEM\_OBOR.NAZEV **AS** "obor"  
**FROM** SEM\_STUD\_PLAN  
**JOIN** SEM\_OBOR **ON** SEM\_OBOR.ID = SEM\_STUD\_PLAN.OBOR\_ID  
**ORDER BY** SEM\_STUD\_PLAN.NAZEV;

*Zdrojový kód 8: Pohled pro studijních plánů.*

**CREATE OR REPLACE VIEW** sem\_p\_rozvrh **AS  
SELECT** sem\_rozvrh.*\**, sem\_zpus\_predm.kapacita **AS** "predmet\_kapacita", sem\_mistnost.kapacita **AS** "mistnost\_kapacita", sem\_stud\_plan.id **AS** "plan\_id", sem\_stud\_plan.nazev **AS** "plan", sem\_zpus\_vyuky.nazev **AS** "zpusob\_vyuky", sem\_predm\_plan.rocnik **AS** "rocnik", *TO\_CHAR*(sem\_rozvrh.presne\_datum, **'DD. MM. YYYY'**) **AS** "datum", sem\_ucitel.id **AS** "ucitel\_id", sem\_ucitel.jmeno || **' '** || sem\_ucitel.prijmeni **AS** "ucitel", sem\_zpus\_predm.pocet\_hodin **AS** "pocet\_hodin", sem\_predmet.zkratka **AS** "predmet", sem\_semestr.nazev **AS** "semestr", sem\_predm\_plan.semestr\_id **AS** "semestr\_id", sem\_mistnost.nazev **AS** "mistnost", sem\_mistnost.id **AS** "mistnost\_id", sem\_mistnost.kapacita **AS** "kapacita"  
**FROM** sem\_rozvrh  
**JOIN** sem\_mistnost  
**ON** sem\_rozvrh.mistnost\_id = sem\_mistnost.id  
**JOIN** sem\_zpus\_predm  
**ON** sem\_rozvrh.zpusob\_zakonceni\_predmetu\_id = sem\_zpus\_predm.id  
**JOIN** sem\_predm\_plan  
**ON** sem\_zpus\_predm.predm\_plan\_id = sem\_predm\_plan.id  
**JOIN** sem\_zpus\_vyuky  
**ON** sem\_zpus\_predm.zpusob\_vyuky\_id = sem\_zpus\_vyuky.id  
**JOIN** sem\_predmet  
**ON** sem\_predm\_plan.predmet\_id = sem\_predmet.id  
**JOIN** sem\_uci  
**ON** sem\_rozvrh.uci\_id = sem\_uci.id  
**JOIN** sem\_ucitel  
**ON** sem\_uci.ucitel\_id = sem\_ucitel.id  
**JOIN** sem\_semestr  
**ON** sem\_semestr.id = sem\_predm\_plan.semestr\_id  
**JOIN** sem\_stud\_plan  
**ON** sem\_stud\_plan.id = sem\_predm\_plan.studijni\_plan\_id;

*Zdrojový kód 9: Pohled pro výpis rozvrhových aktivit.*

**CREATE OR REPLACE VIEW** sem\_p\_uvazky **AS  
SELECT** SEM\_UCI.UCITEL\_ID **as** "ucitel", SEM\_PREDMET.NAZEV **as** "predmet", SEM\_ROLE.NAZEV **as** "role", *sum*(*NVL*(SEM\_ZPUS\_PREDM.POCET\_HODIN, 0)) **as** "hodiny"  
**FROM** SEM\_UCI  
**LEFT JOIN** SEM\_ROZVRH **ON** SEM\_ROZVRH.UCI\_ID = SEM\_UCI.ID  
**LEFT JOIN** SEM\_ZPUS\_PREDM **ON** SEM\_ROZVRH.ZPUSOB\_ZAKONCENI\_PREDMETU\_ID = SEM\_ZPUS\_PREDM.ID  
**JOIN** SEM\_ROLE **ON** SEM\_ROLE.ID = SEM\_UCI.ROLE\_ID  
**JOIN** SEM\_PREDM\_PLAN **ON** SEM\_UCI.PREDM\_PLAN\_ID = SEM\_PREDM\_PLAN.ID  
**JOIN** SEM\_PREDMET **ON** SEM\_PREDM\_PLAN.PREDMET\_ID = SEM\_PREDMET.ID  
**WHERE** SCHVALENO = 1  
**GROUP BY** SEM\_UCI.UCITEL\_ID, SEM\_PREDMET.NAZEV, SEM\_ROLE.NAZEV  
**ORDER BY** "predmet", "role";

*Zdrojový kód 10: Pohled pro výpis rozvrhových aktivit.*

## Funkce

**CREATE OR REPLACE FUNCTION** *SEM\_BOOL\_TO\_STRING*(p\_bool **NUMBER**)  
**RETURN VARCHAR2 DETERMINISTIC IS  
begin  
 IF** p\_bool > 0 **THEN  
 RETURN 'Ano'**;  
 **ELSE  
 RETURN 'Ne'**;  
 **END IF**;  
**end**;

*Zdrojový kód 11: Funkce pro převod logické hodnoty na text.*

**CREATE OR REPLACE FUNCTION** *SEM\_DATUM\_NA\_DEN*(p\_datum **DATE**)  
**RETURN NUMBER DETERMINISTIC IS  
BEGIN  
 RETURN** *TRUNC*(p\_datum) - *TRUNC* (p\_datum, **'IW'**);  
**END**;

*Zdrojový kód 12: Funkce pro převod data na pořadí dne v týdnu.*

**CREATE OR REPLACE FUNCTION** *sem\_je\_mistnost\_volna* (p\_mistnost\_id **NUMBER**, p\_den\_v\_tydnu **NUMBER**, p\_od **NUMBER**, p\_do **NUMBER**, krome **NUMBER**)  
 **RETURN NUMBER AS** v\_pocet **NUMBER**(1);  
 **BEGIN  
 SELECT** *count*(*\**) **AS** pocet  
 **INTO** v\_pocet  
 **FROM** sem\_rozvrh  
 **JOIN** sem\_zpus\_predm  
 **ON** sem\_zpus\_predm.id = sem\_rozvrh.zpusob\_zakonceni\_predmetu\_id  
 **WHERE** mistnost\_id = p\_mistnost\_id  
 **AND** den\_v\_tydnu = p\_den\_v\_tydnu  
 **AND** (sem\_rozvrh.id <> krome **OR** krome **IS NULL**)  
 **AND** schvaleno = 1  
 **AND** (  
 (zacatek <= p\_od **AND** (zacatek + pocet\_hodin) > p\_od)  
 **OR** (zacatek < p\_do **AND** (zacatek + pocet\_hodin) >= p\_do)  
 **OR** (zacatek >= p\_od **AND** (zacatek + pocet\_hodin) <= p\_do)  
 );  
  
 **IF** v\_pocet > 0 **THEN  
 RETURN** 0;  
 **ELSE  
 RETURN** 1;  
 **END IF**;  
 **END**;

*Zdrojový kód 13: Funkce pro zjištění toho, zda je místnost v daný časový interval volná..*

**CREATE OR REPLACE FUNCTION** *sem\_skupina\_zaneprazdnena*(p\_predm\_plan\_id **NUMBER**, p\_den\_v\_tydnu **NUMBER**, p\_od **NUMBER**, p\_do **NUMBER**, krome **NUMBER**)  
 **RETURN NUMBER AS** v\_plan\_id **NUMBER**;  
 v\_pocet **NUMBER**(1);  
 **BEGIN  
 SELECT** sem\_predm\_plan.id **INTO** v\_plan\_id  
 **FROM** sem\_zpus\_predm  
 **JOIN** sem\_predm\_plan **ON** sem\_predm\_plan.id = sem\_zpus\_predm.predm\_plan\_id  
 **WHERE** sem\_zpus\_predm.id = p\_predm\_plan\_id;  
  
 **SELECT** *count*(*\**) **AS** pocet  
 **INTO** v\_pocet  
 **FROM** sem\_rozvrh  
 **JOIN** sem\_zpus\_predm **ON** sem\_zpus\_predm.id = sem\_rozvrh.zpusob\_zakonceni\_predmetu\_id  
 **JOIN** sem\_predm\_plan **ON** sem\_predm\_plan.id = sem\_zpus\_predm.predm\_plan\_id  
 **WHERE** sem\_predm\_plan.id = v\_plan\_id  
 **AND** den\_v\_tydnu = p\_den\_v\_tydnu  
 **AND** (sem\_rozvrh.id <> krome **OR** krome **IS NULL**)  
 **AND** schvaleno = 1  
 **AND** (  
 (zacatek <= p\_od **AND** (zacatek + pocet\_hodin) > p\_od)  
 **OR** (zacatek < p\_do **AND** (zacatek + pocet\_hodin) >= p\_do)  
 **OR** (zacatek >= p\_od **AND** (zacatek + pocet\_hodin) <= p\_do)  
 );  
  
 **IF** v\_pocet > 0 **THEN  
 RETURN** 0;  
 **ELSE  
 RETURN** 1;  
 **END IF**;  
 **END**;

*Zdrojový kód 14: Funkce pro zjištění toho, zda je studijní skupina v daný čas zaneprázdněná.*

## Uložené procedury

Uveďte jaké uložené procedury Váš projekt používá včetně SQL všech použitých stored procedures.

## Spouště

**create or replace TRIGGER** SEM\_SMAZ\_OBRAZEK  
**AFTER UPDATE of** obrazek\_id **OR DELETE ON** sem\_ucitel  
**for each row  
BEGIN  
 DELETE FROM** SEM\_OBRAZEK **WHERE ID** = :**old**.obrazek\_id;  
**END**;

*Zdrojový kód ???: Spoušť, která po vymazání učitele nebo editaci jeho obrázku smaže starý obrázek.*

**CREATE OR REPLACE TRIGGER** sem\_nastav\_den\_v\_tydnu  
**BEFORE INSERT OR UPDATE ON** sem\_rozvrh  
**FOR EACH ROW  
BEGIN  
 IF** :**NEW**.presne\_datum **IS NOT NULL THEN** :**NEW**.den\_v\_tydnu := SEM\_DATUM\_NA\_DEN(:**NEW**.presne\_datum);  
 **END IF**;  
**END**;

*Zdrojový kód ???: Spoušť, která po editaci nebo vložení rozvrhové aktivity automaticky nastaví den v týdnu na základě zadaného data.*

## Indexy

V projektu nejsou využity žádné neklíčové indexy.

## Sekvence

V projektu jsou využity pouze sekvence pro generování unikátních ID v tabulce.

**CREATE SEQUENCE** TABULKA  
**START WITH** 1  
**INCREMENT BY** 1;

*Zdrojový kód ???: Struktura sekvence použitá pro AUTO\_INCREMENT ID ve všech tabulkách.*

## Systémový katalog

Vypište min. 10 vlastností Vámi vybraných databázových objektů.

# Aplikace

## Použité prostředí

V rámci projektu je vytvořena webová aplikace pro správu rozvrhů a úvazků na vysoké škole. Serverová část je napsána v *PHP 7* za využití frameworku *Nette*. Namísto *HTML* je použit šablonovací systém *Latte*, jenž je součástí *Nette*.

Pro komunikaci s databází Oracle bylo nutné stáhnout zdrojové kódy *PHP* a zkompilovat vlastní verzi tohoto jazyka s přidaným ovladačem *PDO\_OCI*. Na straně databáze je vedle *SQL* používáno i *PL/SQL*. Jedná se o nástavbu nad SQL přidávající konstrukce pro procedurální programování.

Stylování je zprostředkováno v *CSS 3* za využití jmenné konvence *BEM*. Pro pohodlnější import dat a dodatečnou validaci některých formulářů byl použit jazyk *JavaScript*.

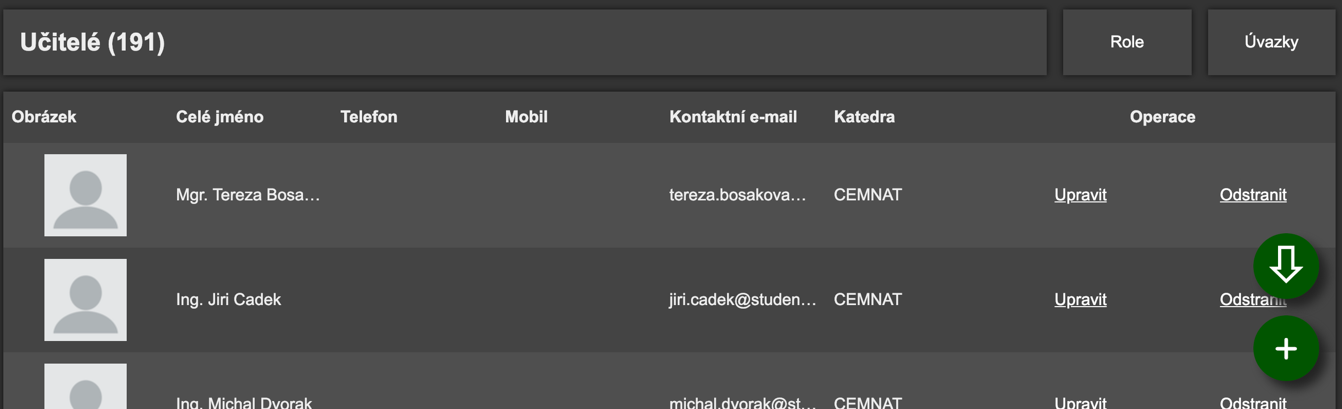
Vývoj celé aplikace probíhal v prostředí *PHP Storm 2018.2* na serveru *MAMP*, který je obdobou *LAMP* a *WAMP* pro operační systém macOS.

## Řízení uživatelských účtů

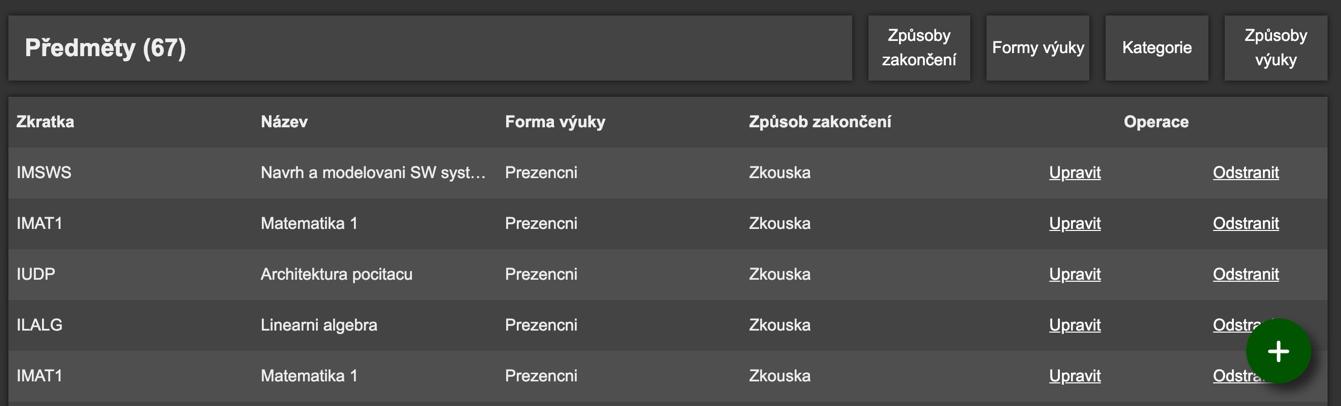
V aplikaci jsou celkem 3 úrovně oprávnění, z nichž některé lze kombinovat – učitel může být zároveň administrátor). Uživatel by neměl v uživatelském rozhraní vidět prvky (např. tlačítko pro smazání), pro jejichž ovládání nemá oprávnění. V případě, že se toto zabezpečení pokusí obejít, další kontrola oprávnění probíhá na serveru. Kontrola probíhá vždy v metodách *render\** (např. renderEdit) nebo *action\** (např. actionDelete) v *presenteru*.

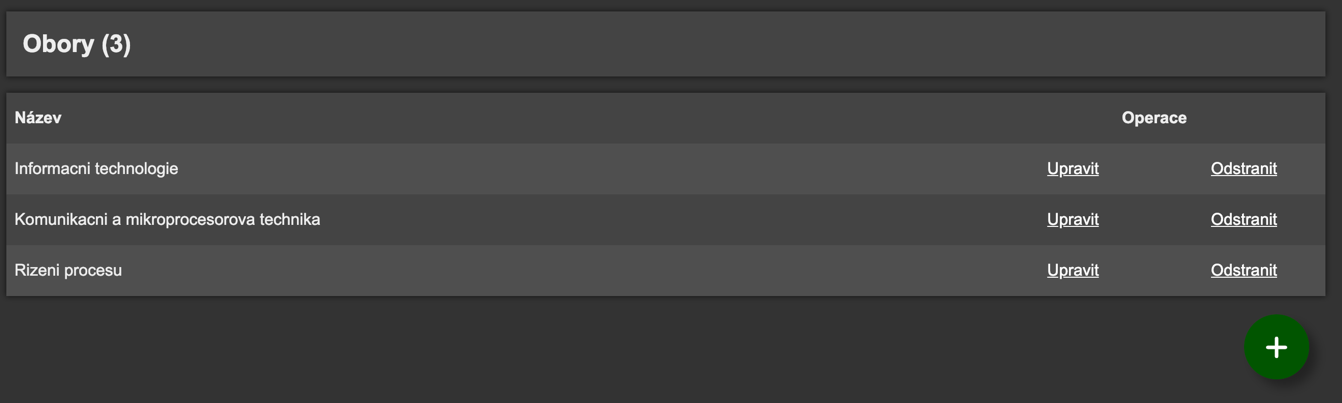
* **Nepřihlášený**: Může prohlížet všechna data.
* **Učitel**: Může to, co nepřihlášený. Navíc může vytvářet vlastní rozvrhové akce. Ty ovšem musí schválit administrátor. Rozvrhové akce musí projít validací. Může také upravovat svůj účet.
* **Administrátor**: Může to, co učitel. Navíc může upravovat všechny ostatní uživatele i jejich rozvrhové akce, které zároveň i schvaluje. U rozvrhových akcí se kromě obsazenosti místnost neprovádí žádná validace. Může určovat, jací uživatele budou mít administrátorská oprávnění.

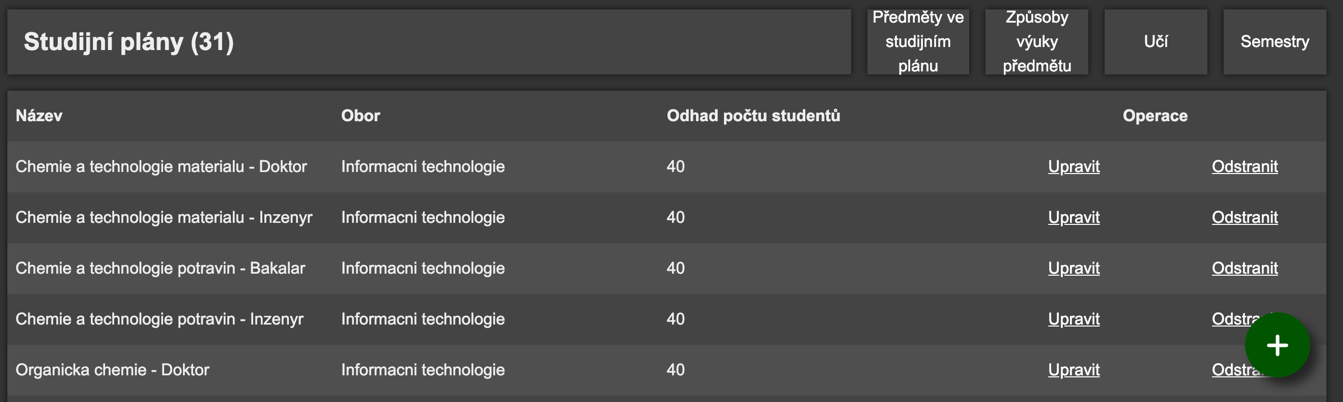
## Moduly

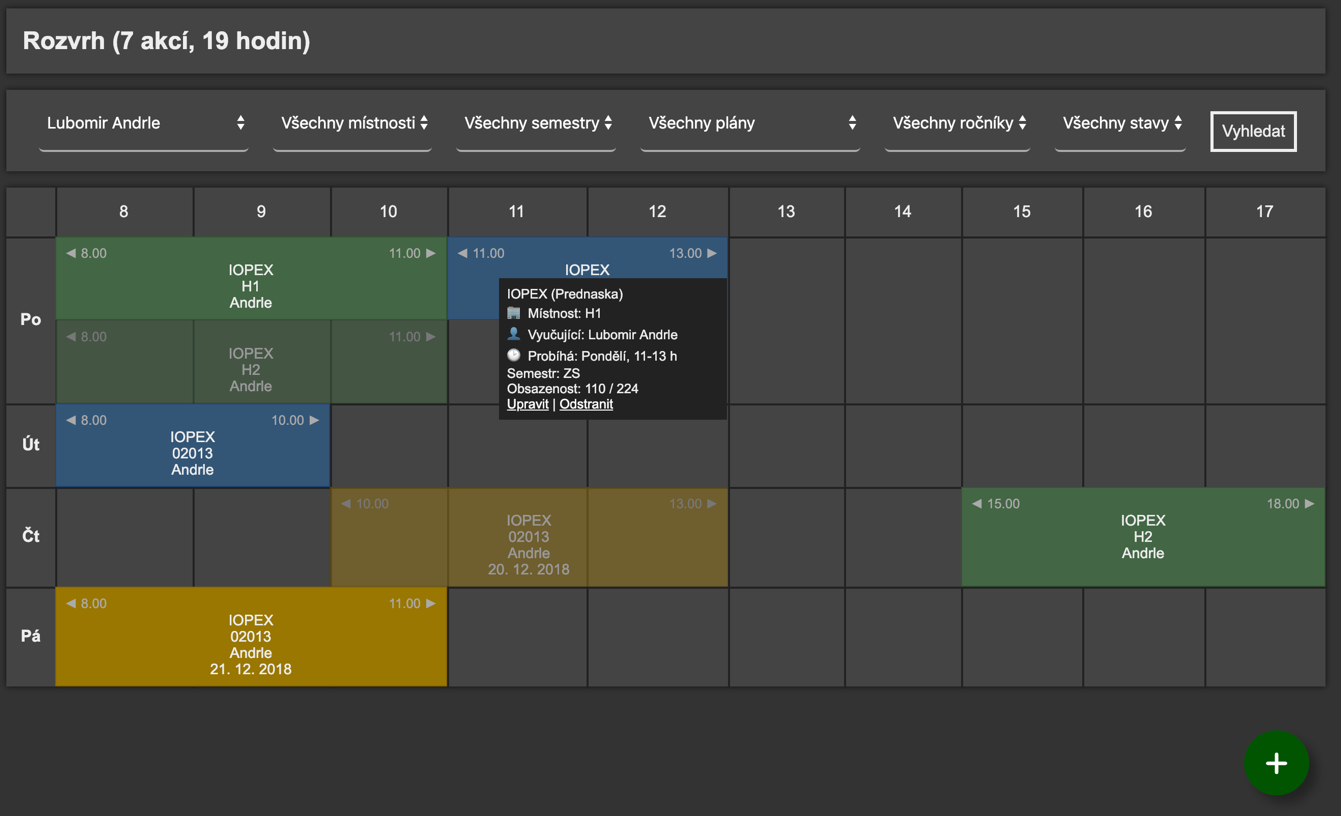
  
*Obrázek ???: Na kartě vyučujících je možné zobrazit si informace o vyučujících a vyexportovat jejich seznam ve formátu JSON. Administrátor navíc může libovolného učitele editovat, smazat nebo přidat.*

*  
Obrázek ???: Karta pracovišť zobrazuje seznam všech kateder. Ve stejné sekci je dostupný i seznam fakult a místností. Administrátor může všechny položky editovat, mazat nebo přidávat.*

*  
Obrázek ???: Karta předmětů zobrazuje předměty, jejich formu výuky a způsob zakončení. Ve stejné sekci jsou dostupné i karty Způsoby zakončení, Formy výuky, Kategorie a Způsoby výuky. Opět platí, že administrátor může všechny položky editovat, mazat nebo přidávat.*

*  
Obrázek ???: Na kartě oborů jsou viditelné všechny dostupné obory. Administrátor je může editovat, mazat nebo vytvářet.*

*  
Obrázek ???: Karta Studijní plány zobrazuje všechny studijní plány a odhadovaný počet studentů. Ve stejné sekci jsou k dispozici i karty Předměty ve Studijním plánu, Způsoby výuky, Učí a Semestry.*

*  
Obrázek ???: Karta rozvrhů. Jsou zde graficky zobrazeny všechny rozvrhové akce vyhovující filtru. Filtrovat lze podle vyučujícího, místnosti, semestru, plánu, ročníku nebo stavu. Různé druhy barevných akcí jsou barevně odlišené. Neschválené akce jsou poloprůhledné. Po najetí kurzorem myši nad rozvrhovou akci se zobrazí podrobnější informace. Učitel může přidávat vlastní rozvrhové akce, které musí být schválený administrátorem. Administrátor může editovat, přidávat nebo mazat libovolné akce.*

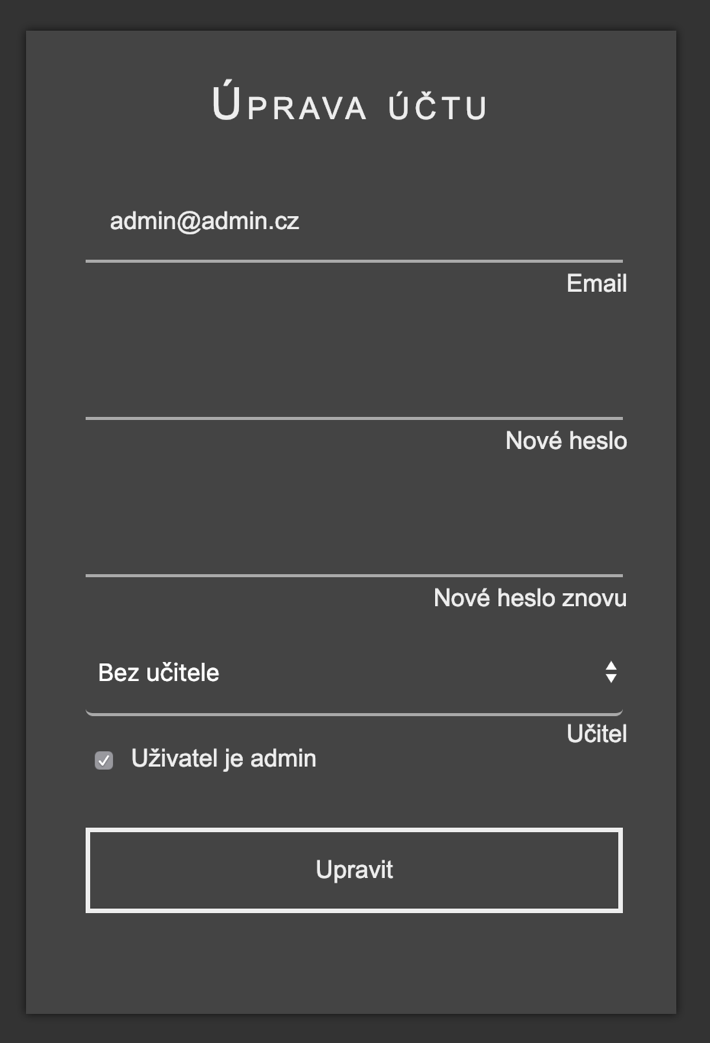
Obdobně jsou vyřešený i další karty. Platí, že pro každou tabulku v databázi existuje karta, která umožňuje zobrazení všech řádků. Administrátoři na každé kartě mohou navíc i libovolná data editovat, přidávat nebo mazat.

**Formuláře**

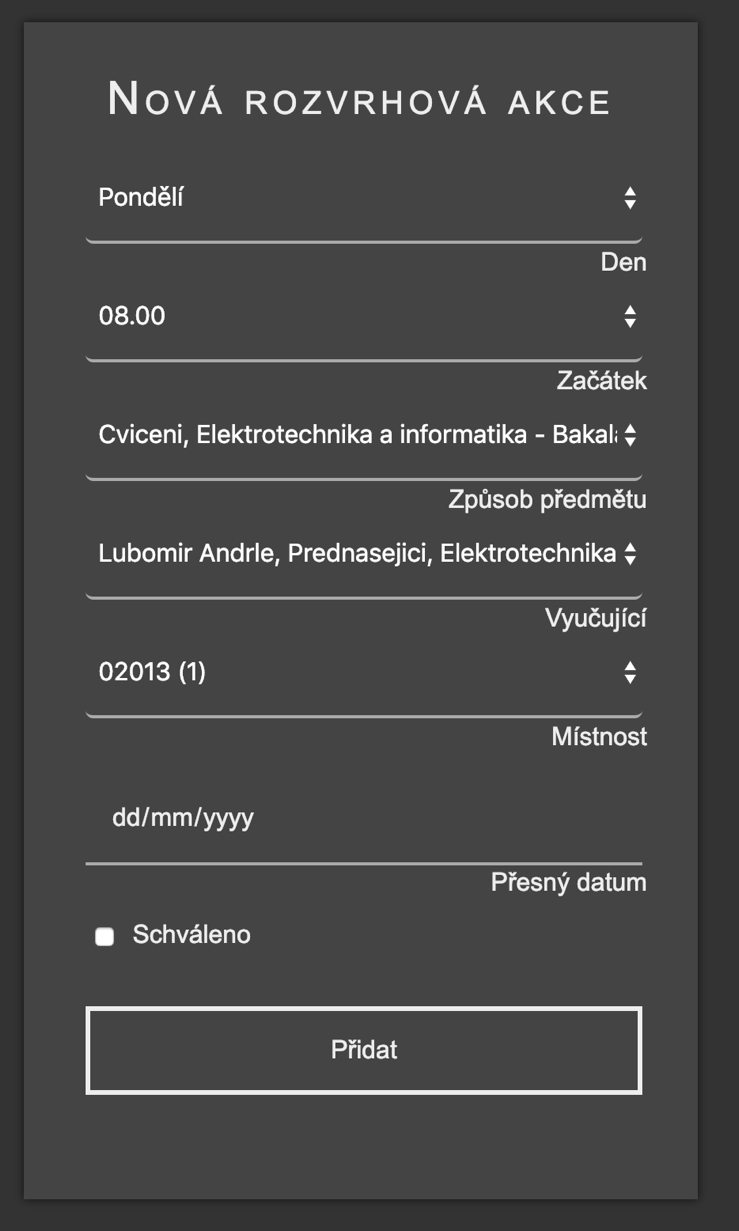
V aplikaci je několik formulářů. Nejjednodušším z nich je ten přihlašovací:

  
*Obrázek ???: Přihlašovací formulář*

Další formulář umožňuje upravit vlastní údaje. V případě, že se formulář vyplňuje administrátor, je zde možné i přiřadit učitelský účet nebo nastavit administrátorská oprávnění.

  
*Obrázek ???: Formulář pro editaci uživatele*

V aplikaci jsou pro každou databázovou tabulku formuláře pro přidání a pro úpravu jednoho záznamu. Liší se pouze v počtu a druhů vstupů.

  
*Obrázek ???: Formulář pro přidání rozvrhové akce.*

## Orientace ve zdrojovém kódu

Vstupním bodem aplikace je soubor */www/index.php*. Ten obsahuje import souboru *bootstrap.php*, který zařídí automatické načítání tříd, načtení konfiguračních souborů a samotné zobrazení požadované stránky*.* Aplikace má následující adresářovou strukturu:

* app *(zdrojové kódy vlastní serverové aplikace)*
  + config *(globální a lokální konfigurační soubory)*
  + constants *(konstanty)*
  + controls *(znovupoužitelné komponenty)*
    - templates *(latte šablony komponent)*
  + model *(modelové třídy a třídy pro práci s databází)*
  + presenters *(presentery)*
    - templates *(šablony presenterů)*
  + router *(směrovací)*
  + utils *(pomocné třídy)*
* database *(zdrojové kódy pro databázi)*
  + model *(diagram databáze pro Oracle Data Modeler)*
  + sql *(SQL skripty pro vytvoření tabulek, spouští, funkcí, procedur, pohledů a naplnění daty)*
* log *(ukládání záznamů chyb, které při běhu aplikace nastaly)*
* temp *(dočasné soubory)*
  + cache *(cache, zde si např. RobotLoader ukládá adresy načtených tříd – není tedy nutné třídy ručně importovat pomocí include nebo require)*
* vendor *(externí knihovny, je zde např. knihovna Nette a šablonovací systém Latte)*
* www *(statické soubory, které si prohlížeč stahuje)*
  + css *(kaskádové styly)*
  + image *(obrázky)*
  + js *(JavaScript)*

Níže je detailněji popsaný obsah adresáře *app*.

**Adresář config**

Nachází se zde konfigurační soubory *config.neon* a *config.local.neon*. Do lokální konfigurace je vhodné ukládat citlivá data jako jsou přihlašovací údaje do databáze. Tento soubor je zmíněn v souboru *.gitignore***.**

**Adresář constants**

Adresář s konstantami. Je zde umístěn *Trait Enum*, který umožňuje používat výčty, jakým je *Days.php* ve stejném adresáři.

**Adresář controls**

Pokud se v aplikaci nachází nějaký opakující se kód či kód, který by bylo vhodné umístit do samostatné struktury, je dobré vytvořit komponentu. Komponenta se skládá z *PHP* třídy a z *Latte* šablony. Každou komponentu lze libovolně parametrizovat. V *Latte* souboru vypadá vykreslení komponenty následovně:

{control *NAZEV\_KOMPONENTY, …PARAMETRY*}

*ScheduleControl.php* umožňuje grafické zobrazení rozvrhových akcí. Barevně odlišuje několik typů rozvrhových akcí a neschválené akce zobrazuje poloprůhledně. Po najetí kurzorem myši nad rozvrhovou akci se o ní zobrazí více informací.

V celé aplikaci je používán výpis databázových tabulek. K tomuto účelu slouží komponenta *TableControl.php, popř. její alternativa TableImageControl.php*. Komponenta v prvním parametru přijme pole objektů. Druhý parametr rozhoduje o tom, zda komponenta vykreslí tlačítka pro editaci a smazání každého záznamu. V dalších parametrech je seznam sloupců, které se mají zobrazit.

Každá stránka může odkazovat na další stránky ze stejné kategorie. Toto menu, jež může být na každé stránce jiné, je reprezentováno komponentou *TabMenuControl.php*. Ta přijme pole odkazů a vykreslí je.

**Adresář model**

Nette je framework, který pracuje s architekturou *MVP*. Odlišuje tak aplikační logiku, vykreslování a zbytek aplikace. Modelové třídy pracují přímo s daty a mají na starost jejich získávání z databáze. Stěžejní je *BaseModel.php****,*** abstraktní třída, ze které dědí všechny ostatní modely. Tento předek totiž za využití *Dependency Injection*získá připojení k databázi a poskytuje ho všem svým potomkům.

Většina dalších modelů implementuje rozhraní *IDatabaseWrapper*. To umožňuje nad databází provádět CRUD operace za využití metod *getAll, getById, updatebyId, deleteById* a *insert*.

Tyto modely pak jen doplňují vlastní *SQL dotazy*. Platí, že pro každou tabulku v databázi je vytvořen vlastní model, např.: *UserModel.php, FacultyModel.php* nebo *RoomModel.php*.

**Adresář presenters**

Další částí architektury *MVP*je *presenter.* Ten propojuje model se šablonou (*template)*. *Presenter* na základě požadavku klienta získá od modelu potřebná data a vloží je do šablony. Každý presenter má alespoň jednu šablonu, která takto získaná data vykresluje. Většina presenterů má alespoň 3 metody pro komunikaci se šablonou a 3 šablony.

* renderDefault *(vkládá data do šablony default.latte, slouží pro výpis všech položek),*
* renderAdd *(kontroluje potřebná oprávnění pro vykreslení add.latte),*
* renderEdit *(kontroluje potřebná oprávnění a vkládá data do edit.latte).*

Všechny 3 šablony jsou umístěny ve vlastním podadresáři (*User*, *Faculty*, *Room*). Každý tento podadresář tedy obsahuje své vlastní 3 šablony, většinou pojmenované *default*, *add* a *edit*.

**Adresář router**

Obsahuje továrnu pro vytvoření směrovače. Ten obsahuje pole cest, přičemž každá se skládá z masky URL a příslušné akce. Pokud chceme, aby pod adresou */login* proběhlo vykreslení šablony presenteru *SignPresenter*.php *in.latte*, jež obsahuje přihlašovací formulář, je nutné vytvořit následující cestu:

**new** Route(**'login'**, **'Sign:in'**);

Bylo by však nepraktické takto vypisovat všechny možné adresy:

**new** Route(‘**teacher’**, **'Teacher:default'**);

**new** Route(‘**teacher/add’**, **'Teacher:add**);

**new** Route(‘**teacher/edit/13’**, **'Teacher:edit'**);

**new** Route(‘**room’**, **'Room:default'**);

**new** Route(‘**room/add’**, **'Room:add**);

**new** Route(‘**room/edit/997’**, **'Room:edit'**);

Většina adres má stejnou strukturu a v definici cesty je možné používat zástupné parametry. V případě, že budeme chtít mít adresu ve formě *presenter/akce/parametr*, definujeme si následující cestu:

**new** Route(‘**<presenter>/<action>[/<id>]**, **'Homepage:default'**);

První parametr v URL bude použit pro název *presenteru*, druhý pro název akce a třetí, nepovinný, pro parametr, jenž bude dostupný uvnitř *presenteru*. Pokud nebude zadán první nebo druhý parametr, proběhne vykreslení šablony *default.latte* v presenteru *HomepagePresenter.php*.

Je možné mít více cest. Hledá se vždy první cesta, která vyhovuje zadané adrese. Je tedy vhodné mít specifické cesty (např. pokud přihlášení nemá být pod adresou *sign/in* ale *login*) na začátku.

**Adresář utils**

Adresář určený pro pomocné třídy. Nachází se zde pouze třída *Time.php* pro snadnější manipulaci s časem.

Závěr

V závěru pochvalně i kriticky zhodnoťte realizovanou databázovou aplikaci a uveďte, jak by se aplikace mohla v budoucnu využívat.

**Přílohy**

1. Skripty SQL
2. Zdrojové kódy aplikace, grafika, apod. vše v jednom zabaleném ZIPU!
3. Samostatně spustitelná aplikace.
4. Ostatní (libovolné soubory doplňující Vaši tvorbu databázové aplikace či nutné nestandardní doplňky nutné pro spuštění Vaší aplikaci)