

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO, BRATISLAVA  
KATEDRA INFORMATIKY



Diplomová práca

# System na podporu rozvrhovania na FMFI UK

Jaroslav Košťál

## ZADANIE DIPLOMOVEJ PRÁCE

Študent(ka) ..... JAROSLAV KOŠTÁL .....

vypracuje za účelom obhajoby v rámci záverečného konania vo vysokoškolskom magisterskom štúdiu na UK FMFI diplomovú prácu s (predbežným) názvom:

..... Systém na podporu rozvoľovania .....

pod vedením pracovníka RNDr. Márie Pastorovej z pracoviska ..... KI .....

Cieľ diplomovej práce a ďalšie poznámky:

..... ANALÝZA PRÍSTUPOV .....

..... NÁVRH SYSTÉMU .....

..... IMPLEMENTÁCIA .....

podpis študenta ..... [signature] ..... podpis ved. dipl. práce ..... [signature] .....

Diplomová práca je tematicky priradená k špecializácii .....

..... Programové systémy .....

V Bratislave dňa ..... podpis gestora špec.: ..... [signature] .....

Poznámka: Vyplňujte v 3 exemplároch: 1 pre študenta, 1 pre vedúceho práce, 1 pre študijné oddelenie (odovzdá študent pri zápise do 4. ročníka).

Týmto prehlasujem, že som diplomovú prácu vypracoval samostatne s použitím uvedenej literatúry, elektronických zdrojov a s odbornou pomocou diplomového vedúceho.

Máj 2007

.....  
Jaroslav Košťál

Touto cestou by som sa chcel poďakovať svojej diplomovej vedúcej RNDr. Márii Pastorovej za odborné konzultácie a cenné pripomienky a rady pri tvorbe tejto práce.

Vďaka patrí aj Ing. Jánovi Petrikovi za poskytnutie exportovaných dát zo systému Študent.

Ďalej by som rád poďakoval svojej rodine a priateľom za ich podporu počas môjho štúdia.

## Abstrakt

Práca sa zaoberá problémom rozvrhovania v prostredí Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. V prvej časti autor analyzuje súčasnú situáciu rozvrhovania na fakulte a jej reálne potreby vzhľadom na špecifické podmienky, ktoré ďalej zohľadňuje v návrhu systému, ktorému sa venuje druhá časť práce. V poslednej kapitole je okrem implementácie základných častí systému na podporu rozvrhovania popísané aj rozhranie rozvrhového portálu, ktorý je súčasťou projektu.

**Kľúčové slová:** podpora rozvrhovania, rozvrhový portál

# Obsah

Úvod.....	8
<b>1 Analýza prístupov.....</b>	<b>9</b>
1.1 Problém rozvrhovania všeobecne.....	9
1.1.1 Algoritmický prístup.....	9
1.2 Rozvrhovanie na FMFI UK.....	10
1.2.1 Organizácia štúdia.....	10
1.2.2 Súčasný spôsob rozvrhovania.....	11
1.3 Analýza potrieb.....	12
1.3.1 Skupinový vs. individualizovaný prístup.....	12
<b>2 Návrh systému.....</b>	<b>14</b>
2.1 Členenie projektu.....	14
2.1.1 Systém na podporu rozvrhovania.....	14
2.1.2 Rozvrhový portál.....	14
2.2 Databázový model.....	14
2.2.1 Surrogate vs. natural key.....	14
2.2.2 Základné entity.....	16
2.2.3 Doplnkové entity.....	23
2.3 Role.....	26
2.3.1 Guest (návštevník/host').....	26
2.3.2 User (autentifikovaný používateľ).....	27
2.3.3 Student (študent).....	27
2.3.4 Teacher (učiteľ).....	27
2.3.5 Scheduler (rozvrhár).....	27
2.3.6 Admin (správca).....	27
2.4 Priority.....	29
<b>3 Implementácia.....</b>	<b>31</b>
3.1 Výber prostriedkov.....	31
3.1.1 Databázový systém (PostgreSQL).....	31
3.1.2 Web server (Apache).....	31
3.1.3 Programovací jazyk (PHP5).....	32
3.1.4 Aplikačný framework (Zend Framework).....	32
3.1.5 Podporné knižnice.....	34

3.2 Nifty Data Objects – ORM framework.....	35
3.2.1 Abstraktné vrstvy.....	37
3.2.2 NDOConditions.....	42
3.2.3 Overloading.....	42
3.2.4 Výkon.....	43
3.3 Adresárová štruktúra kódu.....	44
3.4 Import dát.....	45
3.4.1 Zdroj a formát dát.....	46
3.4.2 Mapovanie.....	50
3.4.3 Aktualizácia.....	51
3.4.4 Budúcnosť.....	54
3.5 Rozvrhový portál.....	55
3.5.1 Autentifikácia.....	55
3.5.2 Zoznamy.....	59
3.5.3 Rozvrhy.....	61
3.5.4 Nastavenia.....	62
3.6 Systém na podporu rozvrhovania.....	63
3.7 Testovacia prevádzka.....	66
<b>Záver.....</b>	<b>67</b>
<b>Prílohy.....</b>	<b>68</b>
Licencie.....	68

## Úvod

Jedným z kľúčových problémov organizácie štúdia každej vzdelávacej inštitúcie realizujúcej výučbu dennou formou je bezpochyby rozvrhovanie. Snahou školy je zosúladiť požiadavky vyplývajúce zo študijného plánu s prostriedkami a obmedzeniami, ktoré vychádzajú z jej reálnych dispozícií. Jedná sa predovšetkým o obmedzené zdroje akými sú miestnosti a čas učiteľov a študentov, ktoré je treba rozvrhnúť bez vzájomných kolízií, prípadne s minimalizáciou ich výskytu. Pritom sa usiluje o stav vyhovujúci maximálnemu množstvu zainteresovaných študentov a vyučujúcich bez zbytočných nesúvislostí či nasádzaní do neatraktívnych časov.

Možnosťami akým spôsobom možno riešiť túto neľahkú úlohu sa zaoberá prvá kapitola. Pozerá sa na konkrétnu situáciu na našej fakulte a špecifikuje požiadavky na systém použiteľný v jej prostredí.

Druhá kapitola predstavuje komplexný návrh celého systému formou definovania jednotlivých entít, rolí, priorít, vzájomných vzťahov a interakcií.

V tretej kapitole možno nájsť popis implementácie od výberu použitých prostriedkov cez vysvetlenie fungovania niektorých knižníc, ktoré bolo nutné naprogramovať, až po popis rozhrania samotného systému.



# 1 Analýza prístupov

## 1.1 Problém rozvrhovania všeobecne

Problém rozvrhovania vo všeobecnosti znamená priradiť ku každému predmetu v študijnom pláne miesto a čas vyučovania. Navyše predpokladáme, že ku každému predmetu je pridelený učiteľ a študenti (prípadne študijná skupina). Pri nasádzaní treba brať do úvahy mnohé ohraničujúce podmienky (constraints), ktoré treba pri vytváraní rozvrhu rešpektovať. Pritom rozlišujeme dva základné typy podmienok:

**nutné podmienky** – musia byť splnené za každých okolností. Sem môžeme zaradiť podmienky ako:

- jeden učiteľ nemôže naraz vyučovať viacero predmetov
- jeden študent/študijná skupina nemôže mať naraz dva rôzne predmety
- v jednej miestnosti nemôže v jednom čase prebiehať výuka viac ako jedného predmetu

**doplňujúce podmienky** – nemusia byť nutne splnené, ale ich splnením vytvárame predpoklad, že vzniknutý rozvrh bude kvalitnejší resp. akceptovateľnejší. Takýmito podmienkami môžu byť:

- zohľadňovanie časových preferencií podľa všeobecnej atraktivity časov prípadne osobných nastavení
- vytváranie čo naj súvislejších rozvrhov bez nevhodných medzier

### 1.1.1 Algoritmický prístup

Problém rozvrhovania (Timetable Problem - TTP) možno označiť za špeciálny prípad problému splňovania ohraničení (Finite Constraint Satisfaction Problem – FCSP), ktorý je pomerne dobre formalizovateľný a existuje množstvo rôznych algoritmických prístupov, ktoré dokážu

uspokojivo riešiť jednoduchý model s jasne definovanými ohraničujúcimi podmienkami. Skôr ako detailnejšie popíšem konkrétnu situáciu na našej fakulte, spomeniem aspoň niektoré algoritmy, ktoré dokážu riešiť tento problém (viď [6], [7]):

**Generate & Test**

**Backtracking**

**Propagating Constraints**

**Forward Checking**

**Backtracking s heuristikou**

**Simulated Annealing**

**Genetic Algorithm**

**Multiagentové systémy**

## **1.2 Rozvrhovanie na FMFI UK**

Hoci uvedené algoritmy dokážu ponúknuť uspokojivé riešenie jednoduchého rozvrhovacieho problému s obmedzeným počtom vstupov, na podmienky našej fakulty sú veľmi ťažko aplikovateľné z niekoľkých príčin:

- veľké množstvo komplikovaných ohraničujúcich podmienok vyplývajúcich so systémom štúdia
- mnohé podmienky sú len ťažko formalizovateľné a priveľmi by zvyšovali výpočtovú zložitosť automatizovaného riešenia
- veľké množstvo vstupných dát (predmetov, učiteľov, študentov)
- nedostatok miestností, ktoré by umožňovali širší manévrovací priestor pri nasádzaní
- individualizovaný prístup ku študentom (viď kapitolu 1.3.1)

### **1.2.1 Organizácia štúdia**

Výučba predmetov prebieha v predpísanom rozsahu a forme, ktoré určuje študijný plán, pričom jeden predmet môže byť v pláne viackrát v rôznych

blokoch. Pre potreby práce budem nazývať každý takýto výskyt kurzom. Každý kurz môže mať viacero rozsahov rôznej formy (napr. 2 hodiny prednášok a 1 hodina cvičení), ktoré sa navyše ešte môžu deliť na menšie časti (spravidla cvičenia), ktoré sa nasádzajú do rozvrhu zvlášť, avšak študent, ktorý má daný predmet zapísaný, navštevuje iba jedno cvičenie. Študenti sú, pre potreby rozvrhovania, rozdelení do študijných skupín, podľa ktorých sú zadeľovaní do jednotlivých častí rozsahu (cvičení). Každý rozsah (prípadne jeho časť) má priradeného svojho vyučujúceho.

### 1.2.2 Súčasný spôsob rozvrhovania

V súčasnosti sa rozvrh nasádza na našej fakulte ručne s pomocou zastaraného softvéru, ktorý slúži na generovanie výstupov. Tento program je však nevyhovujúci po stránke funkčnosti, kapacity (napr. obmedzený počet predmetov, miestností...), používateľského rozhrania (nekomfortné ovládanie DOS-like aplikácie) ako aj samotných výstupov (textový formát).

Samotnému procesu nasádzania rozvrhu predchádza predpríprava, počas ktorej sa jednotlivé rozsahy kurzov rozdelia do kategórií a určí sa presný harmonogram nasádzania.

Celý proces nasádzania trvá väčšinou okolo piatich dní. Postup pri nasádzaní je ovplyvnený rozdelením jednotlivých predmetov do rôznych kategórií, ktoré sa spravidla nasádzajú v iný deň. Najprv sa nasádzajú špeciálne predmety s vysokou prioritou, potom veľké prednášky do veľkých posluchární, nasledujú ostatné prednášky a napokon sa nasádzajú cvičenia a výberové predmety.

Pri nasádzaní sa stretnú všetci rozvrhári v jednej miestnosti, pričom každý má na malých lístočkoch svoje predmety, ktoré má nasadiť a tieto postupne pripevňuje na päť veľkých tabúl reprezentujúcich jednotlivé dni v týždni. Každá tabuľa obsahuje na horizontálnej osi jednotlivé časy (14 stĺpcov od 8:10 po 19:00) a na vertikálnej osi miestnosti. Takýto spôsob však zabezpečuje iba bezkolíznosť miestností a kontrola učiteľov a študijných skupín sa musí vykonávať dodatočne na zvláštnych papieroch. Každý nasadený predmet sa napokon musí zadať do spomínaného

programu, ktorý nakoniec vygeneruje textové rozvrhy.

### ***1.3 Analýza potrieb***

Požiadavkami na „Systém na podporu rozvrhovania“ sú:

- zabezpečovanie priebežnej kontroly integrity dát (kontrola kolízií nutných ohraničujúcich podmienok) pri nasádzaní rozvrhu
- zobrazovanie aktuálneho prehľadu nasadenia s ponukou najvhodnejšieho času na nasadenie vybraného kurzu.
- pohodlné zobrazovanie a úprava údajov o jednotlivých entitách systému
- zobrazovanie hotových rozvrhov študentov, učiteľov, miestností či kurzov vo vhodnej forme
- prístup študentov a učiteľov k rozvrhom a ďalším informáciám cez webové rozhranie s možnosťou autentifikácie
- zabezpečenie importu dát z informačného systému školy s možnosťou ich aktualizácie

#### **1.3.1 Skupinový vs. individualizovaný prístup**

Doterajší spôsob nasádzania rozvrhu pre študijné skupiny sa s novým kreditovým štúdiom, ktorý dáva študentom značnú voľnosť v tom, ktoré predmety si môžu či musia zapísať, stáva nie len neefektívnym ale dokonca ťažko realizovateľným. Dnes už takmer neexistujú disjunktné skupiny študentov, ktoré by bolo možné nasádzať nezávisle. Naopak študenti si vytvárajú svoj študijný plán do značnej miery individuálne, a preto sa aj rozvrhy jednotlivých študentov značne líšia. Z tohto dôvodu vidím nutnosť pristupovať k rozvrhovaniu priamo na základe údajov zo zápisu bez ohľadu na študijné skupiny.

Takýto prístup však ešte viac zvyšuje zložitosť systému a znemožňuje použitie jednoduchých algoritmických riešení vhodných skôr pre generovanie jednoduchých stredoškolských rozvrhov.

Nevyhnutnou podmienkou aby bolo možné aplikovať popísaný prístup sú

informácie zo zápisu, ktoré musia byť k dispozícii ešte pred samotným začatím nasádzania rozvrhu. V súčasnosti prebieha nasádzanie rozvrhu v zimnom semestri paralelne so zápsmi študentov, čo by znemožňovalo použitie individualizovaného systému na podporu rozvrhovania. V blízkej budúcnosti sa však uvažuje buď o elektronickom zápise či aspoň predzápise študentov na kurzy v časovom predstihu, čo by celý problém vyriešilo.

## 2 Návrh systému

### 2.1 Členenie projektu

Používateľské rozhranie systému je rozčlenené do dvoch hlavných modulov:

#### 2.1.1 Systém na podporu rozvrhovania

Prostredie na samotnú podporu rozvrhovania a nasádzanie predmetov je prístupné len pre rozvrhárov a je čiastočne oddelené od zvyšku systému. Ponúka predovšetkým nasledovné funkcie:

- nasádzanie predmetov do rozvrhu s pridelovaním učiteľov a vizualizáciou nasadeného rozvrhu
- farebné vyznačenie najvhodnejších časov na nasadenie zvoleného predmetu so zvoleným učiteľom
- kontrola kolízií a integrity dát

#### 2.1.2 Rozvrhový portál

Portál je k dispozícii pre verejnosť a je určený predovšetkým študentom a učiteľom, ktorí si môžu bez námahy prezerať svoj individuálny rozvrh, ktorý systém generuje podľa aktuálnych dát v databáze.

Prostredníctvom portálu je možný aj výpis a podrobnosti o jednotlivých entitách. Rozvrhári majú navyše možnosť tieto dáta upravovať, čo je dôležité hlavne vo fáze predprípravy pred samotným rozvrhovaním.

### 2.2 Databázový model

#### 2.2.1 Surrogate vs. natural key

Pred samotným návrhom databázového modelu bolo nutné rozhodnúť sa medzi dvoma prístupmi k voľbe primárnych kľúčov:

**natural key** – prirodzený kľúč logicky súvisiaci so záznamom

**surrogate key** – umelý (automaticky generovaný) kľúč

Neexistuje všeobecný konsenzus, ktoré riešenie je lepšie, avšak ja som zvolil druhý typ, nakoľko ponúka nasledovné výhody:

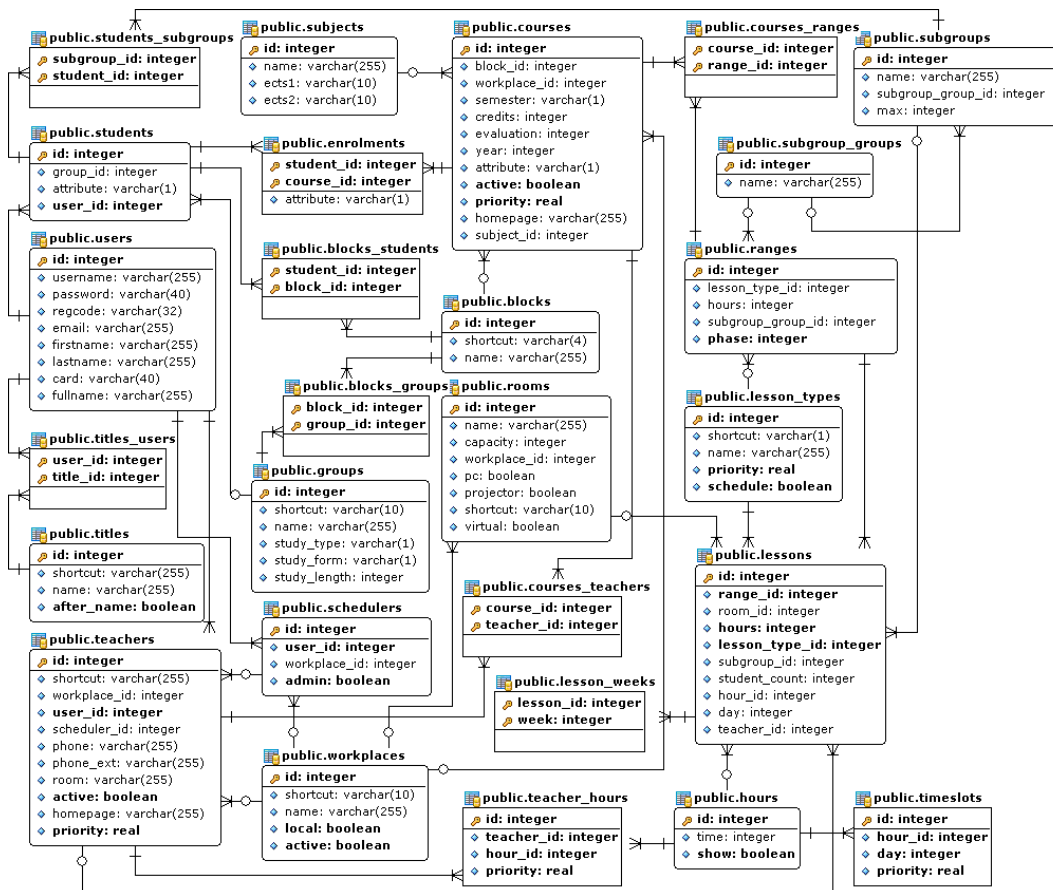
- **nemennosť** – zmenou prirodzeného kľúča (napr. skratky názvu) nedochádza k zmene primárneho kľúča a teda zostávajú zachované vzťahy s ostatnými entitami aj bez kaskádovania aktualizácií (CASCADE UPDATE)
- **kompatibilita** – mnohé abstraktné databázové vrstvy, predovšetkým ORM systémy využívajú surrogate key s autoinkrementáciou
- **výkon** – jednoduchý dátový typ (spravidla integer) sa lepšie indexuje ako prirodzené kľúče, ktoré môžu byť zložené z viacerých atribútov či zložitých dátových typov. Vo väčšine prípadov však optimalizácia na systémovej úrovni eliminuje tento efekt.
- **jednotnosť** – primárne kľúče v celom modeli sú rovnakého typu a spravidla aj názvu, čo uľahčuje manipuláciu so záznamami

Všetky entity v našom modeli majú primárny atribút *id* typu SERIAL<sup>1</sup>. To sa však netýka relácií slúžiacich ako spájacia tabuľka pre vzťahy typu N:M, ktoré majú kompozitný primárny kľúč zložený z cudzích kľúčov asociovaných relácií.

---

<sup>1</sup> SERIAL je dátový typ INTEGER s asociovanou sekvenciou využívanou na autoinkrementáciu primárneho kľúča

### 2.2.2 Základné entity



Obrázok 1: Entitno-relačný databázový model

Jednotlivé entity sú reprezentované tabuľkou v relačnej databáze. Názvom príslušnej tabuľky je meno entity v plurále.

#### 2.2.2.1 User (používateľ systému)

Používateľ systému je základnou entitou slúžiacou na riadenie prístupu k systému ako aj na udržiavanie základných informácií o každom používateľovi.

Entita obsahuje nasledovné atribúty:

- *username* – prihlasovacie meno. Používateľ si ho volí sám pri registrácii a následne ho využíva pri každom prihlásení.
- *password* – heslo zakódované hashovacou funkciou (SHA1), ktoré si takisto zvolí používateľ pri registrácii, slúži na autentifikáciu



používateľa

- *regcode* – registračný kód na registráciu prostredníctvom emailu
- *card* – zreteženie posledných štvorčíslí čísla študenta/učiteľa a čísla ISIC/ITIC karty slúži na registráciu na základe príslušného preukazu
- *email* – email používateľa využívaný predovšetkým na odoslanie registračného kódu
- *firstname* – krstné meno používateľa
- *lastname* – priezvisko používateľa

Každý používateľ môže mať navyše niekoľko titulov, čo reprezentované N:M vzťahom k entite *Title*.

Entity *Student*, *Teacher* a *Scheduler* sú tiež používateľmi systému a obsahujú povinný cudzí kľúč *user\_id*, ktorý určuje o ktorého používateľa sa jedná. Z tohto vyplýva, že jeden používateľ môže byť asociovaný s viacerými reálnymi používateľskými entitami, ktoré zdieľajú rovnaké prihlasovacie ako aj osobné údaje.

Táto entita zodpovedá abstraktnej roli *user* (viď kapitola 2.3.2 na strane 27).

### 2.2.2.2 *Student* (*student*)

Študent je používateľom, ktorý patrí práve do jednej študijnej skupiny (*Group*) a má pridelených niekoľko vlastných študijných blokov (*Block*). Všetky bloky študenta sú zjednotením jeho vlastných blokov a blokov jeho skupiny.

Kurzy reprezentované entitou *Course*, ktoré majú študenti zapísané sú pripojené cez join-table *enrolments*, ktorá navyše obsahuje atribút obsahujúci informáciu o tom, či má študent daný kurz zapísaný povinne, voliteľne alebo alternatívne, prípadne či má predmet zapísaný druhýkrát.

Atribúty:

- *user\_id* – povinná referencia na entitu *User*
- *group\_id* – referencia na študijnú skupinu (*Group*)

### 2.2.2.3 *Teacher* (učiteľ)

Entita reprezentujúca učiteľa obsahuje údaje špecifické pre jeho rolu ako pracovisko, rozvrhár, kurzy, priorita a ďalšie doplňujúce informácie.

Ide o tieto atribúty:

- *user\_id* – povinná referencia na entitu *User*
- *shortcut* – štandardná skratka učiteľa používaná v študijnom programe
- *scheduler\_id* – referencia na svojho rozvrhára (*Scheduler*)
- *workplace\_id* – referencia na pracovisko (*Workplace*)
- *room* – miestnosť (kancelária) na fakulte, kde učiteľ pôsobí
- *homepage* – domovská stránka učiteľa (túto si môže učiteľ sám zadať vo svojich nastaveniach)
- *phone*, *phone\_ext* – telefónne číslo, klapka
- *active* – príznak, či je učiteľ v systéme aktívny a bude sa zobrazovať v ponukách učiteľov
- *priority* – priorita učiteľa na účely rozvrhovania

Učiteľ je tiež pridružený ku kurzom (*Course*) podľa študijného plánu. Skutočne nasadené hodiny (*Lesson*) majú určeného svojho učiteľa zvlášť.

Každý učiteľ si môže v systéme nastaviť priority jednotlivých časových úsekov reprezentovaných entitami *TeacherHour* (viď kapitola 2.2.3.5). Čas, v ktorom učiteľ nemá definovanú prioritu sa pokladá za voľný (resp. s maximálnou prioritou).

### 2.2.2.4 *Scheduler* (rozvrhár)

Rozvrhár rozširuje entitu *User* o referenciu na pracovisko v jeho kompetencii a príznak, či sa jedná o správcu systému prostredníctvom nasledovných atribútov:

- *user\_id* – povinná referencia na entitu *User*
- *workplace\_id* – referencia na pracovisko (*Workplace*)

- *admin* – príznak, či sa jedná o správcu systému

### 2.2.2.5 *Course* (kurz)

Táto entita reprezentuje kurzy a informácie o nich podľa študijného programu na aktuálny rok.

Atribúty:

- *subject\_id* – predmet (referencia na *Subject*)
- *block\_id* – študijný blok (referencia na *Block*)
- *workplace\_id* – pracovisko (referencia na *Workplace*)
- *semester* – semester, v ktorom sa má kurz nasadiť ('Z' pre zimu, 'L' pre leto)
- *credits* – počet kreditov kurzu (jeden predmet môže mať v praxi rôzny počet kreditov v rôznych blokoch, preto je táto informácia pri kurze a nie pri predmete)
- *evaluation* – percentuálny podiel hodnotenia pripadajúci na vyučovaciu časť semestra
- *year* – odporúčaný rok štúdia
- *attribute* – charakter kurzu z hľadiska povinnosti jeho absolvovania
  - 'P' – povinný kurz
  - 'A' – alternatívny kurz
  - 'V' – voliteľný kurz
- *homepage* – domovská stránka kurzu (spravidla nastavuje asociovaný učiteľ)
- *active* – príznak, či je kurz aktívny alebo suspendovaný
- *priority* – priorita zohľadňovaná pri navrhovaní nasádzania

Kurz je asociovaný s učiteľmi (*Teacher*), ktorí by mali podľa študijného plánu zabezpečovať jeho výučbu ako aj so študentami, ktorí majú predmet zapísaný (viď. *Student* v kapitole 2.2.2.2).

Rozsah výučby určený študijným plánom vyjadruje asociácia s entitami

*Range.*

#### 2.2.2.6 **Range** (rozsah)

Táto entita vyjadruje rozsah kurzu, ktorý sa má nasadiť a jeho typ (napr. 2-hodinová prednáška). pričom jeden rozsah môže byť pridelený k jednému alebo viacerým kurzom (*Course*).

Atribúty:

- *lesson\_type\_id* – referencia na typ vyučovacej hodiny (*LessonType*)
- *hours* – počet vyučovacích hodín (po sebe nasledujúcich časových úsekoch), počas ktorých bude prebiehať výučba kurzu
- *subgroup\_group\_id* – referencia na *SubgroupGroup*

#### 2.2.2.7 **Block** (študijný blok kurzov)

Kurzy sú zoskupené do študijných blokov reprezentované entitou *Block*.

Atribúty:

- *shortcut* – skratka bloku
- *name* – celý názov študijného bloku

#### 2.2.2.8 **Group** (študijná skupina)

Každý študent (*Student*) patrí do jednej študijnej skupiny, ktorá je asociovaná s niekoľkými študijnými blokmi (*Block*).

Atribúty:

- *shortcut* – skratka študijnej skupiny
- *name* – celý názov študijnej skupiny
- *study\_type* – typ štúdia
- *study\_form* – forma štúdia
- *study\_lenght* – dĺžka štúdia

Posledné tri atribúty sú iba informatívne a sú prebraté zo systému Študent.

### 2.2.2.9 *Subgroup* (podskupina študentov)

Táto entita umožňuje nasadiť jeden rozsah kurzu (*Range*) pre rôznych študentov zvlášť. Môžu byť určení konkrétni študenti patriaci do tejto podskupiny (napr. na cudzí jazyk sú študenti spravidla rozdelení podľa stupňa pokročilosti) ako aj maximálny počet študentov v podskupine, ale tieto údaje sú nepovinné. V takom prípade sa príslušnosť študenta do podskupiny určuje náhodne pri nasádzaní tak, aby mu bolo umožnené navštevovať aspoň jednu z nasadených podskupín.

Podskupiny pre jeden rozsah sú zoskupené do entity *SubgroupGroup*, ktorá je následne asociovaná s príslušným rozsahom.

Atribúty:

- *name* – informatívny názov podskupiny slúžiaci predovšetkým na jej rozlíšenie v ponukách
- *subgroup\_group\_id* – referencia na agregátor *SubgroupGroup*
- *max* – maximálny počet študentov

### 2.2.2.10 *Room* (miestnosť)

Entita miestnosti je kľúčovou pri nasádzaní rozvrhu a okrem virtuálnych miestností je podmienka bezkolíznosti miestnosti v čase jednou z mála zo striktných ohraničujúcich podmienok. V relácii s nasadenou hodinou (*Lesson*) sa nesmie vyskytovať miestnosť dvakrát v rovnakom čase.

Atribúty:

- *shortcut* – skratka predmetu pre výber v ponukách
- *name* – celý názov miestnosti
- *workplace\_id* – referencia na pracovisko (*Workplace*), ktorému daná miestnosť patrí (resp. má právo do nej nasádzať). Nullová hodnota značí spoločnú miestnosť
- *virtual* – príznak, či sa jedná o virtuálnu miestnosť, teda miestnosť, ktorá reálne neexistuje, ale používa sa v špeciálnych prípadoch, kedy buď nie je známe kde bude výuka prebiehať alebo je určenie miesta v kompetencií niekoho iného – vtedy je virtuálnou

miestnosťou daná inštitúcia (napr. katedra). Kontrola kolízií virtuálnych miestností sa nevykonáva.

- *pc, projector* – príznaky, či miestnosť disponuje počítačmi (ich počet je určený kapacitou) resp. dataprojektorom
- *capacity* – maximálny počet študentov, ktorý je možné do miestnosti nasadiť

### 2.2.2.11 *Workplace* (pracovisko)

Pracoviskami sú najčastejšie katedry, ktoré sú priradené k učiteľom (*Teacher*), rozvrhárom (*Scheduler*), kurzom (*Course*) a miestnostiam (*Room*).

Existujú však aj iné pracoviská ako napr. iné fakulty či inštitúcie mimo fakulty, ktoré sa štandardne neponúkajú pri výbere pracoviska a majú nastavený príznak *local* na false. Tieto sa využívajú v prípade ak učiteľ učí alebo študent navštevuje kurz mimo našej fakulty.

Atribúty:

- *shortcut* – skratka pracoviska
- *name* – celý názov pracoviska
- *local* – príznak, či sa jedná o pracovisko na našej fakulte
- *active* – príznak, či je pracovisko aktívne, teda je možné ho vyberať v ponukách, nakoľko import obsahuje množstvo nepoužívaných pracovísk

### 2.2.2.12 *Lesson* (vyučovacia hodina/rozvrh)

Táto entita predstavuje jednotlivé nasadené hodiny. Obsahuje všetky potrebné informácie, na základe ktorých sa následne generujú všetky rozvrhy. Jej napĺňaním dochádza k samotnému nasádzaniu rozvrhu.

Entita spája rozsahy predmetov (*Range*) s časom (*Hour, day*), miestom (*Room*) a učiteľom (*Teacher*). V prípade, že sa študenti delia na podskupiny (najčastejšie na cvičeniach), určuje aj podskupinu (*Subgroup*).

Hoci každý rozsah má určený typ hodiny (*LessonType*) aj počet

vyučovacích hodín, entita *Lesson* obsahuje tieto atribúty takisto, nakoľko rozsah môže byť reálne nasadený s iným typom (napr. 'Kurz' ako 'Prednáška') aj počtom hodín (napr. 2-hodinová telesná výchova ako 8-hodinový blok), hoci vo väčšine prípadov bude hodnota atribútov rovnaká.

Väčšina predmetov sa v rozvrhu nasádza na celý semester, avšak existujú prípady, kedy je nutné nasadiť predmet iba na niektoré týždne semestra (napr. párne). Na to slúži tabuľka *lesson\_weeks*, ktorá reprezentuje vzťah nasadenia medzi hodinou a celočíselným týždňom semestra (1 až počet týždňov semestra definovaný v nastaveniach). Neexistencia záznamu v tejto tabuľke pre danú hodinu, znamená nasadenie na celý semester.

Atribúty:

- *range\_id* – referencia na rozsah (*Range*), ktoré v sebe skrýva informáciu ako o predmete, tak aj o študentoch, ktorí ho majú zapísaný.
- *day* – deň, do ktorého je hodina nasadená
- *hour\_id* – referencia na čas (*Hour*), kedy sa hodina začína
- *room\_id* – referencia na miestnosť (*Room*), kde bude prebiehať výučba
- *teacher\_id* – referencia na učiteľa (*Teacher*), ktorý bude hodinu vyučovať
- *lesson\_type\_id* – referencia na typ hodiny
- *hours* – počet vyučovacích hodín (po sebe nasledujúcich časových úsekov), počas ktorých bude prebiehať výučba predmetu.
- *subgroup\_id* – referencia na podskupinu študentov (*Subgroup*), v prípade, že je hodina rozdelená do skupín

### 2.2.3 Doplnkové entity

#### 2.2.3.1 *Subject* (predmet)

Tabuľka predmetov, je vlastne iba číselníkom názvov a kódov predmetov. Tento je potrebný nakoľko v študijnom pláne sa bežne vyskytujú rôzne

kurzy (*Course*) toho istého predmetu.

Atribúty:

- *name* – názov predmetu
- *shortcut* – skratka predmetu
- *ects1* – ECTS kód predmetu v dobiehajúcom štúdiu
- *ects2* – ECTS kód predmetu v novom štúdiu

Dva ECTS kódy sú zapríčinené prekryvaním dvoch typov štúdia na fakulte, ktoré je len dočasné. Vytváranie ďalšej entity pre ECTS kódy by bolo preto zbytočné nakoľko majú v našom systéme len informačný význam a následná zmena na jednotné ECTS kódy, bude v tomto prípade jednoduchšia.

### 2.2.3.2 *LessonType* (typ vyučovacej hodiny)

Typ vyučovacej hodiny (*Lesson*) je určený touto entitou, ktorá okrem plného názvu a skratky obsahuje aj prioritu.

Atribúty:

- *name* – názov typu (napr. Prednáška, Cvičenie, Seminár...)
- *shortcut* – skratka (napr. P, C, S...)
- *priority* – priorita typu (napr. prednáška má vyššiu prioritu ako cvičenie)
- *schedule* – príznak, či sa daný typ predmetu nasádza do rozvrhu (napr. prax sa nenasádza)

### 2.2.3.3 *Hour* (hodina/čas)

Tabuľka hodín (časov), ktorá rozdeľuje deň do jednotlivých časových úsekov (timeslotov)

Atribúty:

- *time* – čas začiatku časového úseku vyjadrený v minútach od polnoci (napr. 8:10 je reprezentované číslom  $490 = 8 \cdot 60 + 10$ )
- *show* – príznak, či sa daný čas zobrazuje v rozvrhu a pri nasádzaní



(napr. čas 7:20 sa štandardne nepoužíva, ale musí byť možnosť ho v prípade potreby použiť)

#### 2.2.3.4 *Timeslot* (časový úsek)

Každý časový úsek (tzv. timeslot) jednoznačne určený dňom a hodinou (*Hour*) môže mať svoju prioritu, ktorá sa zohľadňuje pri návrhu najvhodnejšieho nasadenia.

Atribúty:

- *day* – deň (celočíselné hodnoty 1 až 5 reprezentujúce pondelok až piatok)
- *hour\_id* – hodina (referencia na entitu *Hour*)
- *priority* – priorita časového úseku (napr. piatok 19:00 bude mať nižšiu hodnotu ako utorok 9:00)

#### 2.2.3.5 *TeacherHour* (časové preferencie učiteľa)

Mnoho učiteľov má rôzne povinnosti priamo nesúvisiace so samotnou výukou, pre to je nutné v systéme zachytiť informácie o čase kedy je a kedy nie je možné učiteľovi nasadiť predmet do rozvrhu. Na to slúži entita *TeacherHour*, ktorá zohľadňuje aj rôzne preferencie učiteľa jednotlivých časových úsekov formou priority.

Atribúty:

- *day* – deň
- *hour\_id* – hodina (referencia na *Hour*)
- *teacher\_id* – referencia na učiteľa (*Teacher*), ktorému daný časový úsek prideluje prioritu
- *priority* – priorita časového úseku

#### 2.2.3.6 *SubgroupGroup* (skupina podskupín)

V prípade, že sa rozsah nejakého kurzu rozdeľuje na podskupiny študentov (*Subgroup*), ktoré sa potom samostatne nasádzajú do rozvrhu, tieto podskupiny sú združené do *SubgroupGroup* a priradené k danému rozsahu.

Jediným atribútom okrem primárneho kľúča je:

- *name* – informatívny názov pre skupinu podskupín

Jedna skupina podskupín môže byť priradená k rôznym rozsahom. V prípade určenia konkrétnych študentov podskupín to zabezpečuje rovnaké členenie študentov na rozsahy. Ak študenti nie sú určení, môžeme využívať skupiny podskupín bez kontroly rovnakého rozdelenia študentov. Toto nám umožňuje vytvoriť virtuálne univerzálne skupiny podskupín. Napríklad pre potreby všetkých cvičení bez konkrétneho rozdelenia študentov, ktorých chceme rozčleniť do štyroch skupín môžeme vytvoriť „univerzálnu 4-skupinu podskupín“ zloženú zo štyroch podskupín, ktorú použijeme pre všetky takéto cvičenia.

### 2.2.3.7 *Title (akademický titul)*

Každý používateľ môže mať priradené získané akademické tituly formou tejto informatívnej entity obsahujúcej nasledovné atribúty:

- *shortcut* – skratka titulu (napr. Mgr.)
- *name* – celý názov titulu (napr. magister)
- *after\_name* – príznak, či sa píše titul za menom alebo pred menom

## 2.3 *Role*

Prístup do systému je riešený formou autentifikácie používateľov, ktorí získajú prístup k jednotlivým častiam systému na základe práv definovaných pomocou rolí.

Každý autentifikovaný používateľ môže vystupovať vo viacerých roliach.

Návrh uvažuje s nasledovnými piatimi rolami, ktorých presné práva sú zadané v tabuľke 1:

### 2.3.1 *Guest (návštevník/host)*

Neautentifikovaný používateľ má prístup k verejne dostupným informáciám ako sú zoznamy, rozvrhy a detaily jednotlivých predmetov, miestností a učiteľov.

Jedine návštevník má prístup k prihlasovaciemu a registračnému formuláru.

### **2.3.2 User** (*autentifikovaný používateľ*)

Jedná sa o akúsi abstraktnú rolu, ktorá slúži iba na zastrešenie všetkých autentifikovaných používateľov, pričom každý takýto používateľ môže mať mať dokonca viacero reálnych rolí (napr. učiteľ a zároveň rozvrhár)

### **2.3.3 Student** (*študent*)

Študenti majú oproti návštevníkom možnosť prezeráť si svoj vlastný individuálny rozvrh ako aj filtrované zoznamy podľa zapísaných predmetov.

### **2.3.4 Teacher** (*učiteľ*)

Učitelia môžu, okrem funkcií ponúkané študentom, upravovať doplnkové informácie o predmetoch, ktoré vyučujú, prezeráť si študentov, ktorí majú zapísaný jeho predmet a upravovať svoje časové preferencie pre potreby rozvrhovania.

### **2.3.5 Scheduler** (*rozvrhár*)

Úlohou rozvrhára je predovšetkým správa pridelených učiteľov a nasádzanie predmetov nimi vyučovaných. Každý rozvrhár je asociovaný s pracoviskom, ktoré môže mať vlastné miestnosti, ktoré takisto spravuje rozvrhár. V obmedzenej miere môže upravovať aj predmety spadajúce do jeho kompetencie.

### **2.3.6 Admin** (*správca*)

Správcom je rozvrhár, ktorý má definované maximálne práva. Môže modifikovať všetky zoznamy a zasahovať ľubovoľným spôsobom do procesu nasádzania (samozrejme pri dodržaní integrity dát). Jeho právomocou je pridávanie či mazanie ostatných používateľov a zodpovedá aj za import a správnosť dát.

V prostredí našej fakulty by mal byť správcom hlavný rozvrhár.

	guest	student	teacher	scheduler	admin
Prihlásenie + Registrácia	O	X	X	X	X
Nastavenia	X	O	O	O	O
Základné údaje	X	O	O	O	O
Časy	X	X	O	X	X
Prostredie	X	O	O	O	O
Systém	X	X	X	X	O
Rozvrh	O	O	O	O	O
Vlastný	X	O	O	X	X
Miestnosti	O	O	O	O	O
Učiteľa	O	O	O	O	O
Predmetu	O	O	O	O	O
Zoznam					
Predmetov					
výpis všetkých	O	O	O	O	X
výpis vlastných	X	O	O	O	X
editácia vlastných (čiastočná)	X	X	O	O	X
výpis + editácia všetkých	X	X	X	X	O
Miestností					
výpis všetkých	O	O	O	O	X
výpis vlastných	X	X	X	O	X
editácia mojich	X	X	X	O	X
výpis + editácia všetkých	X	X	X	X	O
Pracovnísk					
výpis všetkých	O	O	O	O	X
výpis vlastných	X	X	X	O	X
výpis + editácia všetkých	X	X	X	X	O
Učiteľov					
výpis všetkých	O	O	O	O	X
editácia všetkých	X	X	X	X	X
výpis + editácia všetkých	X	X	X	X	O
výpis vlastných	X	O	X	O	X
editácia vlastných (čiastočná)	X	X	X	O	X
Študentov					
výpis vlastných	X	X	O	X	X
výpis + editácia všetkých	X	X	X	X	O
Import	X	X	X	X	O
Nasádzanie	X	X	X	O	O
Odhlásiť	X	O	O	O	O

Tabuľka 1: ACL

## 2.4 Priority

Keďže úlohou systému má byť návrh vhodného času pre nasadenie predmetu, je nutné istým spôsobom rozlišovať, ktorý čas je vhodnejší ako iný. Takisto pri zobrazovaní nasadeného rozvrhu, ktorý nemusí byť bezkolízny, je potrebné jednotlivé nasadené hodiny porovnávať.

Na tento účel slúži systém priorít, ktorý vytvára isté metriky, pomocou ktorých môžeme porovnávať väčšinu entít v požadovanom kontexte. V systéme existujú dva druhy priorít:

- **definované priority** – sú buď pevne dané alebo perzistentne uložené v databázi
- **odvodené priority** – priority vypočítané podľa definovaného kľúča z definovaných priorít

V dátovom modeli sú definované priority pre nasledovné entity:

- *Enrolment* – priorita zápisu, určeného študentom a kurzom, ktorý má zapísaný, závisí predovšetkým od atribútu tohto záznamu (tj. či má študent zapísaný kurz povinne, alternatívne, voliteľne či druhý krát) ale aj od dodržania odporúčaného roku štúdia a či patrí kurz do jeho bloku.
- *Course* – každý kurz môže mať určenú prioritu, ale aj prioritu zohľadňujúcu počet a štruktúru študentov, ktorí majú daný kurz zapísaný
- *Teacher* – priority napr. vzhľadom na funkciu učiteľa (vedúci katedry a pod.)
- *Timeslot* – priority jednotlivých časových úsekov
- *TeacherHour* – učiteľom nastavené jeho časové preferencie
- *LessonType* – definované priority pre daný typ hodiny (napr. prednáška má vyššiu prioritu ako cvičenie)

Následne sa odvodzujú priority niektorých ďalších entít:

*Range* – odvodzuje sa na základe priority asociovaných kurzov, typu a dĺžky rozsahu (ide vlastne o alternatívu rozdelenia

- 
- *Lesson* – odvodzuje sa na základe priority rozsahu, učiteľa, typu a dĺžky

## 3 Implementácia

### 3.1 Výber prostriedkov

Pred samotnou implementáciou bolo nutné rozhodnúť aké prostriedky a na akej platforme budem používať a ktoré časti systému je vhodné riešiť formou externých podporných knižníc a čo bude naopak nutné vyvíjať vo vlastnej réžii.

Všetky použité softvérové produkty sú licencované open-source licenciami, ktorých presné znenie je možné nájsť podľa odkazov, uvedených v tabuľke 2 na strane 68.

#### 3.1.1 Databázový systém (PostgreSQL)

Voľba RDBMS bola do značnej miery determinovaná skutočnosťou, že tento systém je na našej fakulte využívaný. Tento produkt je navyše veľmi robustný a vyhovuje všetkým požiadavkám, ktoré si projekt vyžiada.

Vzhľadom na použitie abstraktných vrstiev medzi RDBMS a aplikačnou logikou, nie je závislosť na konkrétnom databázovom systéme veľká, no napriek tomu by jeho zmena bola zrejme sprevádzaná drobnými úpravami funkcií, využívajúcich špecifické vlastnosti Postgresu, ktoré nie sú pokryté abstraktnými vrstvami.

#### 3.1.2 Web server (Apache)

Podobne ako v prípade databázového systému aj web server Apache je na fakulte už používaný a nemal by byť problém s nasadením systému do tohto prostredia.

Pre správnu funkciu aj bezpečnosť je vyžadovaná podpora modulu `mod_rewrite` a povolenie vlastných `.htaccess` riadiacich súborov (`AllowOverwrite`)

Nevyhnutnou podmienkou z hľadiska bezpečnosti je podpora zabezpečeného protokolu `HTTPS`.

### 3.1.3 Programovací jazyk (PHP5)

Ponuka server-side skriptovacích jazykov je pomerne široká, avšak žiadna z alternatív (PHP, ASP, JSP, Ruby...) nie je výrazne dominantná, takže výber bol ovplyvnený predovšetkým jednoduchosťou inštalácie a dostupnosťou v prostredí našej fakulty.

Treba však zdôrazniť, že potrebná je minimálne verzia 5. Nielen preto, že to vyžaduje väčšina nových podporných knižníc, ale predovšetkým kvôli novému objektovému modelu a SPL (Standard PHP Library), ktoré sú nevyhnutné na správnu funkciu systému.

### 3.1.4 Aplikačný framework (Zend Framework)

Voľba správneho aplikačného frameworku je veľmi dôležitá, nakoľko do značnej miery ovplyvňuje akým spôsobom bude celý projekt fungovať.

Existuje niekoľko desiatok produktov, preto je nevyhnutné jasne špecifikovať aké vlastnosti by mal taký framework spĺňať. Pre potreby nášho projektu sú dôležité najmä nasledujúce charakteristiky:

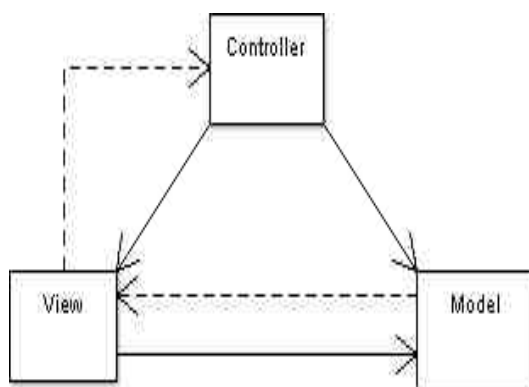
- **MVC model** – framework by mal fungovať na základe návrhového vzoru Model-View-Controller (viď kapitolu 3.1.4.1)
- **PHP5** – mnohé frameworky hoci podporujú aj PHP5, v snahe zachovať spätnú kompatibilitu so staršou verziou 4, nevyužívajú mnohé konštrukty a vlastnosti nového objektového modelu a SPL.
- **modulárnosť a flexibilita** – modulárnosť frameworku by mala spočívať nie len v možnosti rozširovať jeho funkcionality, ale predovšetkým v možnosti programátora vybrať si čo potrebuje a nenútiť ho používať príliš robustné riešenia, ktoré nie sú adekvátne rozsahu jeho projektu.
- **autentifikačný a autorizačný modul** – framework by mal obsahovať modul na autentifikáciu používateľov systému a ponúkať mechanizmy na riadenie prístupu na základe oprávnení.
- **ORM** – jednoduchou a transparentnou formou by mala byť riešená perzistencia dátových objektov v databáze



### 3.1.4.1 MVC Pattern

Návrhový vzor Model-View-Controller spočíva v oddelení prezentačnej a aplikačnej logiky. Celá aplikácia sa člení na 3 časti:

- **Model** – obsahuje aplikačnú logiku a prístup k dátam
- **View** – je pohľad na model a plní prezentačnú funkciu zobrazovania dát z modelu vhodnou formou pre klienta.
- **Controller** – v prostredí web aplikácií typicky spracováva HTTP požiadavky a na základe URL a parametrov vyberá činnosť, ktorá sa má vykonať. Následne načíta potrebné modely a dáta z nich posiela príslušnému pohľadu na zobrazenie.



Obrázok 2: Model-View-Controller pattern

Výhody MVC:

- viacero pohľadov na jeden model (napr. rozvrh v HTML, PDF, TXT, XML, CSV...)
- jednoduchšia úprava prezentačnej vrstvy
- zvýšenie prehľadnosti kódu a zjednodušenie údržby

### 3.1.4.2 Zend Framework

Zend Framework spĺňa všetky vytýčené požiadavky okrem poslednej. Hoci modul *Zend\_Db* poskytuje prístup k databáze, úroveň abstrakcie, ktorú ponúka je nedostatočná. Keďže sú ale jednotlivé moduly frameworku

minimálne vzájomne previazané, nič nebráni použitiu iného riešenia na prístup k databáze, výberom ktorého sa zaoberám v nasledujúcej kapitole.

Implementácia MVC patternu v podaní tohto frameworku je neinvazívna a ponúka všetky jeho výhody. Použitý je návrhový vzor *Front Controller*, ktorý zabezpečuje prístup ku všetkým častiam systému cez jeden vstupný bod, čo umožňuje efektívne riadenie prístupu k jednotlivým zdrojom.

Súčasná verzia frameworku 0.9.3 Beta, s ktorou náš projekt pracuje, má stabilné API a finálna verzia by mala byť k dispozícii v priebehu jedného mesiaca.

Využívané moduly:

- *Zend\_Controller*, *Zend\_View* – MVC pattern
- *Zend\_Auth* – autentifikačný modul
- *Zend\_Acl* – autorizačný modul (kontrola prístupu)
- *Zend\_Config* – prístup ku konfiguračným súborom
- *Zend\_PDF* – generovanie PDF súborov
- *Zend\_Loader*, *Zend\_Registry*, *Zend\_JSON* – pomocné moduly

### 3.1.5 Podporné knižnice

#### 3.1.5.1 DBAL

Na prístup do databázy je použitá knižnica *Creole* (<http://creole.phpdb.org>). Táto vytvára databázovú abstraktnú vrstvu, čím umožňuje pohodlnejší prístup k databáze.

Hlavnými výhodami knižnice sú:

- PHP5 – čisté využívanie nového objektového modelu PHP5 a predovšetkým SPL (iterátory)
- metadata extraction – získavanie metadát (schémy databázy) priamo z databázy, čím sa akékoľvek zmeny schémy v databáze automaticky prejaví aj v aplikácii.
- datatype abstraction – transparentný preklad dátových typov medzi

PHP a RDBMS (napr. PostgreSQL boolean reprezentovaný 't'/'f' sa transparentne prekladá na natívne true/false a naopak)

Creole pokrýva najnižšie vrstvy abstrakcie a zabezpečuje prístup k databáze pre vyššie vrstvy (viď obrázok 3 v kapitole 3.2.1).

### **3.1.5.2 ORM**

Pre Object-Relation Mapping som sa rozhodol využiť vlastné riešenie, ktoré je podrobnejšie popísané v kapitole 3.2.

### **3.1.5.3 PDF**

Hoci existuje viacero open-source riešení na generovanie PDF súborov, napokon som sa rozhodol použiť priamo modul *Zend\_PDF* obsiahnutý v *Zend Frameworku*. Riešenia ponúkajúce priamy prevod HTML do PDF nevracali uspokojivé výsledky, preto bude nutné vyriešiť PDF výstupy vo vlastnej rézii formou view scriptu.

### **3.1.5.4 Javascript**

Client-side scripting je do značnej miery ovplyvnený samotným klientom - teda internetovým prehliadačom. Realita je bohužiaľ taká, že každý prehliadač (resp. renderovacie jadro) implementuje javascript s istými odlišnosťami. Práve na abstrakciu nad týmito odlišnosťami slúžia rôzne javascriptové knižnice.

Ja som sa rozhodol použiť veľmi obľúbenú knižnicu *prototype*, ktorú využívam predovšetkým na zjednodušenie asynchrónnych HTTP požiadaviek (AJAX) ako aj manipuláciu s DOM (Document Object Model).

Na zlepšenie užívateľského prostredia a zvýšenie interaktivity ďalej využívam knižnice *script.aculo.us*, *yui-ext*, *Nifty Corners Cube*.

## **3.2 Nifty Data Objects – ORM framework**

Medzi relačnými tabuľkami jednotlivých entít v databáze a ich objektovými reprezentáciami existuje tzv. *dátová priepasť*. Na prekonanie tejto priepasti slúžia práve ORM systémy, ktoré zabezpečujú mapovanie

medzi objektami a relačnými tabuľkami (Object-Relation Mapping). Táto problematika je pomerne široká a systémy implementujúce takýto prístup k dátam bývajú často veľmi zložité a robustné.

Ja som sa rozhodol vytvoriť vlastnú implementáciu ORM, ktorá by spĺňala nasledovné charakteristiky:

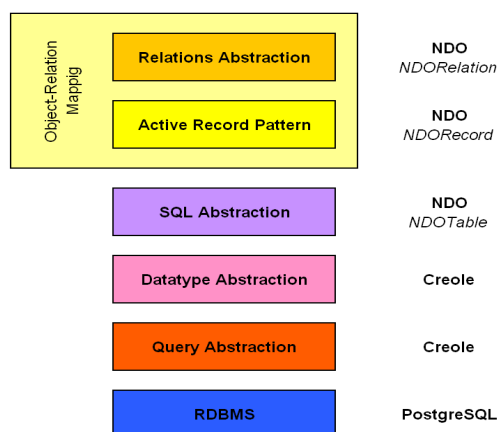
- čo najjednoduchšie používanie (rozhranie poľa)
- transparentné objekto-orientované vzťahy všetkých typov (1:N, N:1, N:M)
- automatické zistenie schémy z databázy
- bezpečné generovanie primárnych kľúčov v konkurentnom prostredí

Existujúce riešenia, ktoré som mal možnosť testovať, trpeli niektorými z nasledujúcich vlastností, pre ktoré som sa rozhodol použiť vlastné riešenie:

- neexistencia priamej podpory vzťahov typu N:M (napr. *propel*)
- nutnosť definovať schému databázy v programe a aktualizovať ju pri každej úprave
- reštriktívne požiadavky na databázovú schému (pomenovanie názvov tabuliek, atribútov či kľúčov)
- neštandardné riešenie vzťahov ako zdvojené spájacie tabuľky a pod. (napr. *ezpdo*)
- nevhodné či zložité rozhranie API

Na samotný prístup k databáze sa používa knižnica *Creole*, ktorá zabezpečuje nadväzovanie spojenia s databázou, vykonávanie SQL dotazov a transparentný preklad dátových typov. Výsledky SELECTov vracia formou SPL iterátorov, ktoré sa následne konvertujú na objekty prípadne asociatívne pole.

### 3.2.1 Abstraktné vrstvy



Obrázok 3: Database Abstraction Layers

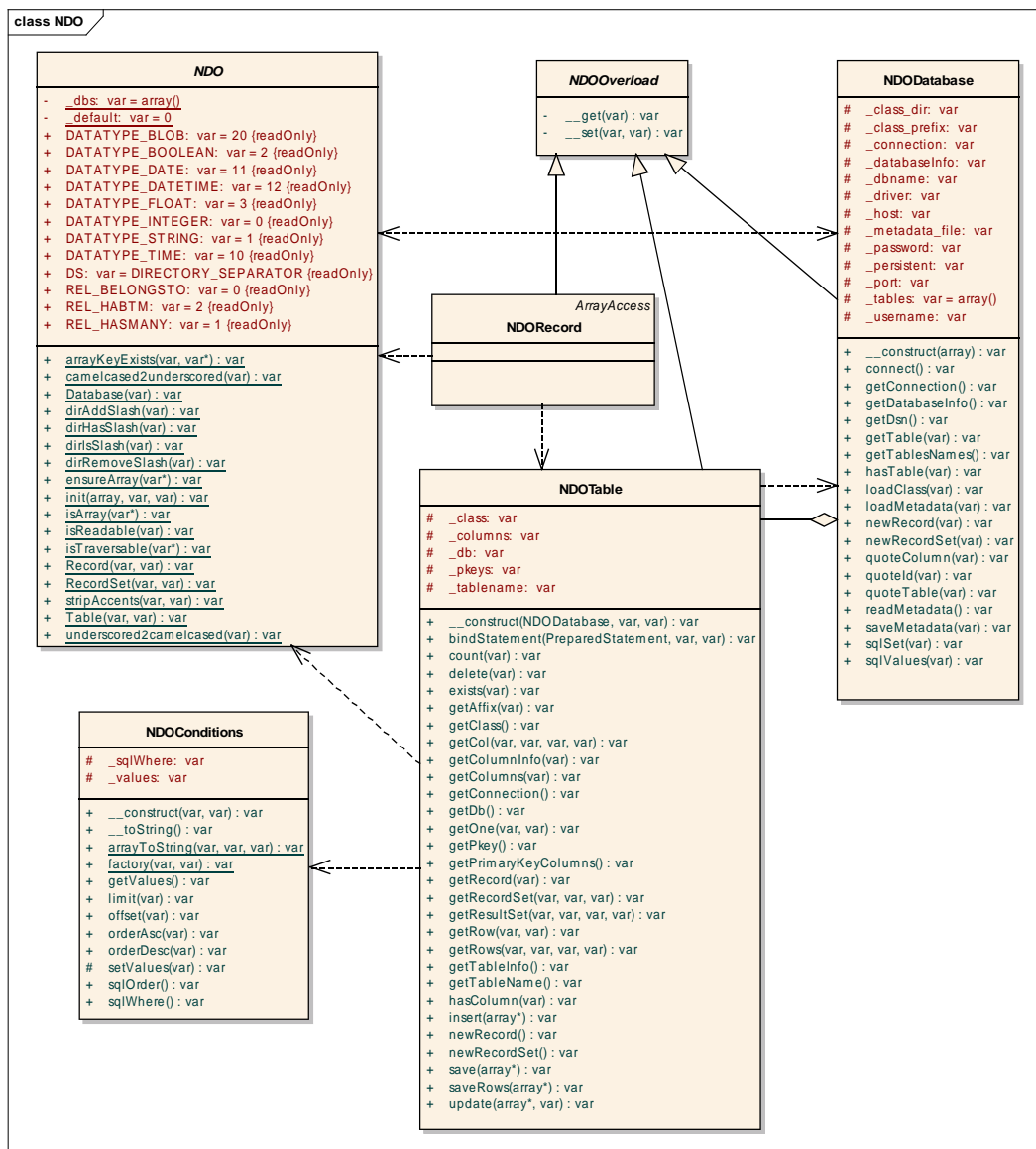
#### 3.2.1.1 SQL Abstraction

Základnou vrstvou, ktorú NDO framework vytvára je abstrakcia najpoužívanejších SQL dotazov formou triedy *NDOTable*, ktorá reprezentuje tabuľku v relačnej databáze.

Úlohy vrstvy sú nasledovné:

- získavanie údajov z relačnej tabuľky na základe definovaných podmienok (pomocou objektu *NDOConditions* viď kapitolu 3.2.2) s transparentnou konverziou datatypov vo forme natívneho skaláru, asociatívneho poľa alebo objektu(ov) vyššej abstraktnej vrstvy (*NDORecord*, *NDORecordSet*)
- vkladanie nových záznamov určených asociatívnym poľom a získavanie automaticky generovaného surrogate primárneho kľúča novovloženého záznamu pre ďalšie použitie
- úprava existujúcich záznamov na základe asociatívneho poľa a definovaných podmienok (*NDOConditions*)

Hoci sa dá trieda používať priamo, jej hlavnou úlohou je slúžiť vyššej vrstve a poskytovať jednoduché získavanie dát z databázy ako aj pohodlné vkladanie a aktualizovanie záznamov.



### 3.2.1.2 Active Record Pattern

#### NDORecord

Trieda *NDORecord* je implemetáciou návrhového vzoru *Active Record* a predstavuje rodičovskú triedu všetkých perzistentných objektov. Jednotlivé perzistentné objekty, ktoré tvoria *Model MVC* patternu, sú

implementáciami rozhrania *ArrayAccess*, čo umožňuje prístup k jednotlivým atribútom formou rozhrania poľa.

Ukážka znázorňuje prístup k perzistentnému objektu ako k poľu:

```
<?php
$teacher = NDO::Record('Teacher');
/* vytvorí inštanciu perzistentného objektu Teacher reprezentujúceho
   učiteľa */

$teacher->find(245);
/* nájde učiteľa s primárnym kľúčom 245 */

$teacher['phone'] = '02/12345678';
/* nastaví atribút 'phone' */

$teacher->save();
/* uloží vykonané zmeny */

$teacher->reset();
/* vyčistí inštanciu objektu */

$teacher['shortcut'] = 'Pastorová M.';
/* nastaví atribút 'shortcut' novému objektu */

$teacher->find();
/* vyhľadá prvého učiteľa, ktorého skratka je 'Pastorová M.' */

$homepage = $teacher['homepage'];
/* uloží domovskú stránku učiteľa do dočasnej premennej $homepage */

$teacher->reset();
/* vyčistí inštanciu objektu */

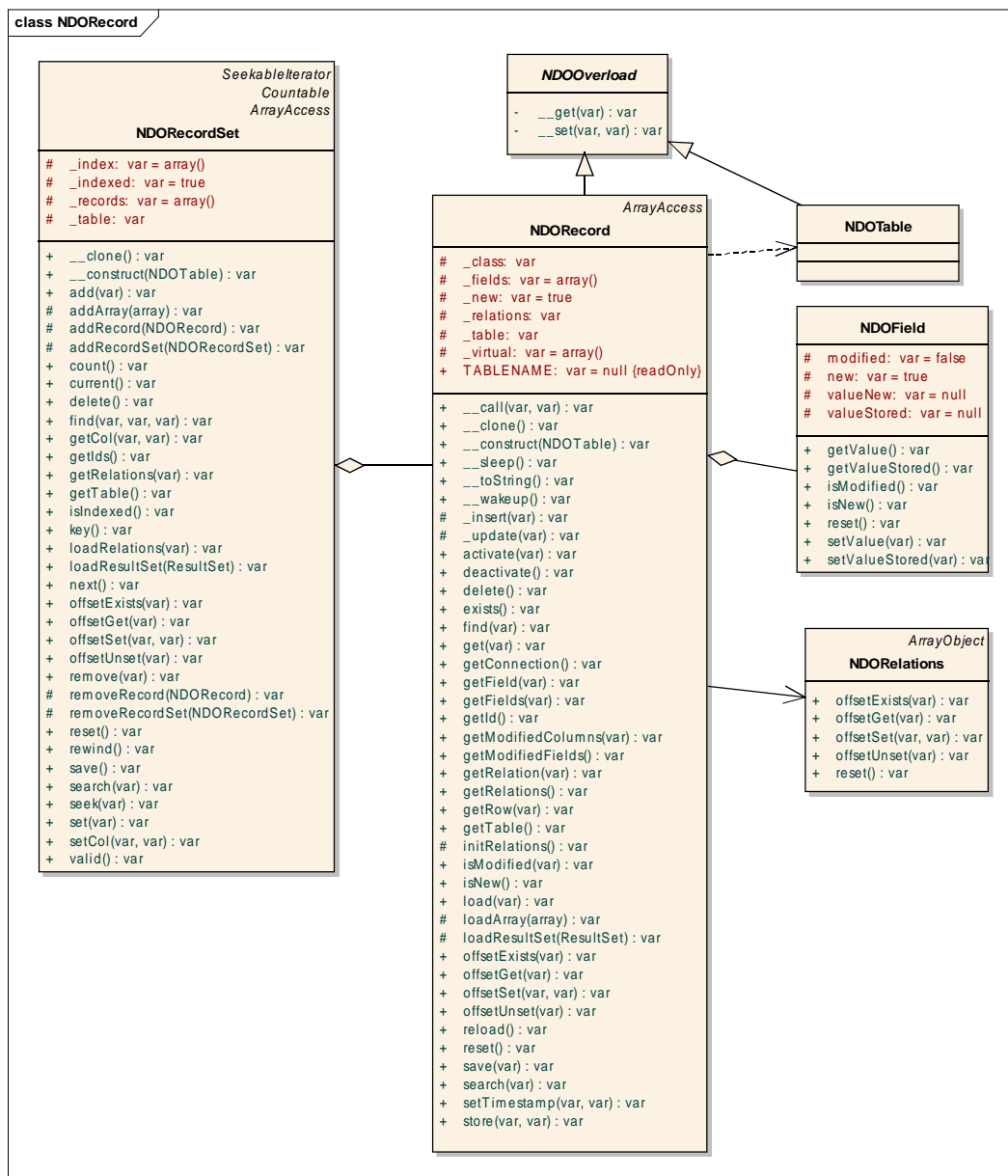
$teacher['homepage'] = $homepage;
/* nastaví atribút 'homepage' novému záznamu */

$teacher['shortcut'] = 'Mrkvička J.';
/* nastaví skratku učiteľa */

$teacher->save();
/* vloží nový záznam do databázy */

print $teacher['id'];
/* vypíše vygenerovaný kľúč nového záznamu */

?>
```



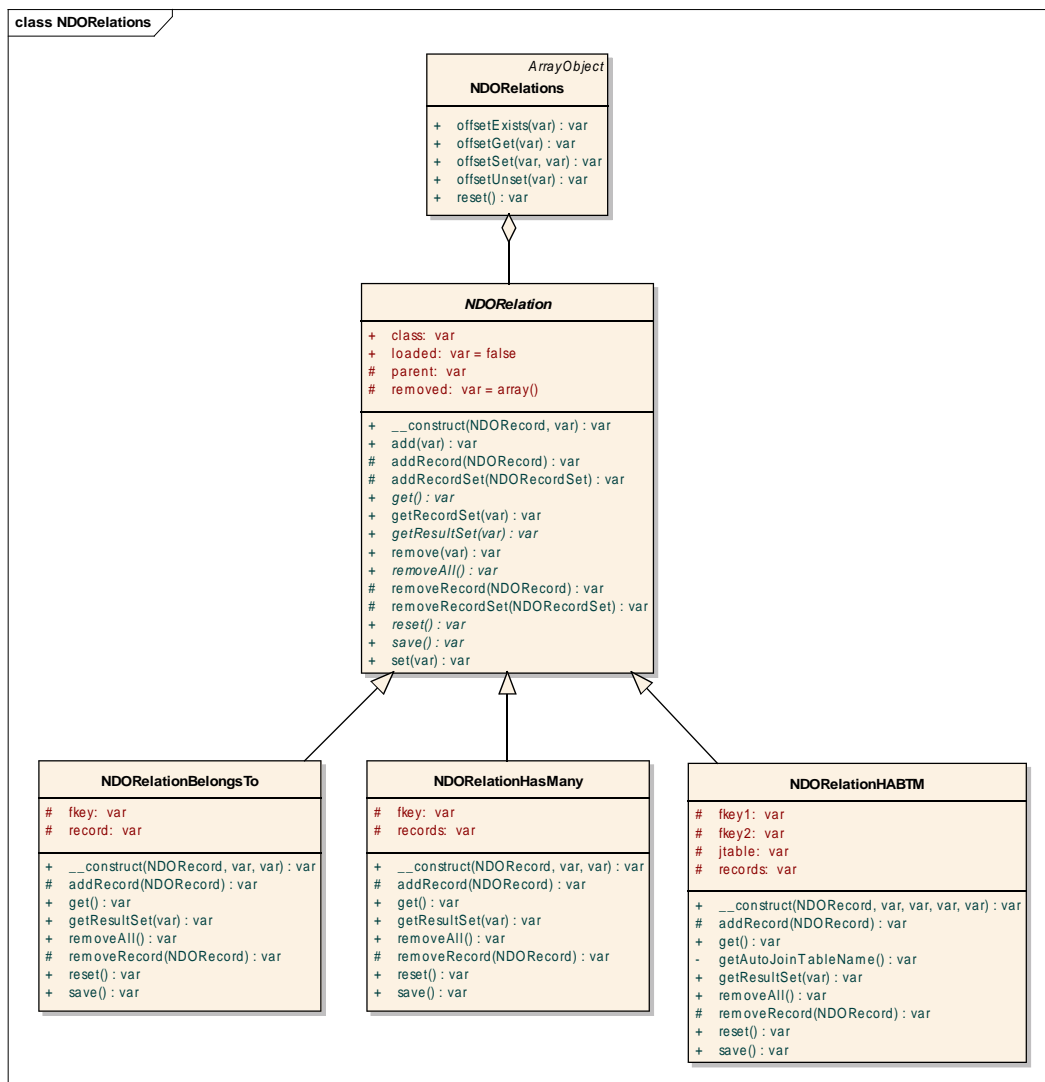
Obrázok 5: NDORecord - Active Record Pattern

## NDORecordSet

Táto trieda predstavuje agregátor perzistentných objektov (*NDORecord*) rovnakého typu (reprezentujúcich rovnakú entitu). Ak má príslušná entita jednoduchý primárny kľúč, tento je použitý ako kľúč pre priamy prístup k objektu formou rozhrania poľa. Prostredníctvom tejto triedy je možné vykonávať niektoré akcie jedným príkazom na viacerých objektoch. *NDORecordSet* sa využíva predovšetkým ako návratová hodnota metód vracajúcich kolekcie objektov (napr. vyhľadávanie, SELECTy).



### 3.2.1.3 Relation Abstraction



Obrázok 6: NDORelation - Relation Abstraction

Perzistentný objekt, rovnako ako relačná tabuľka, je vo vzťahu s ďalšími objektami. Na transparentnú manipuláciu s asociovanými objektami slúži trieda *NDORelation*, ktorá je generalizáciou nasledovných troch konkrétnych typov vzťahov:

- **NDORelationBelongsTo** – vzťah N:1 (belongs to)
- **NDORelationHasMany** – vzťah 1:N (has many)
- **NDORelationHABTM** – vzťah N:M (has and belongs to many)

Prvý typ vzťahu vracia pri prístupe k objektu formou poľa jeden asociovaný objekt (*NDORecord*) zvyšné dva vracajú všetky asociované objekty (*NDORecordSet*).

Ukážka prístupu k asociovaným objektom:

```
<?php
$teacher = NDO::Record('Teacher');
/* vytvorí inštanciu perzistentného objektu Teacher */
$teacher->find(array('shortcut'=>'Pastorová M.'));
/* vyhledá prvého učiteľa, ktorého skratka je 'Pastorová M.' */
$workplace = $teacher['Workplace'];
/* do premennej $workplace uloží perzistentný objekt (potomok triedy
  NDORecord) reprezentujúci pracovisko (entitu Workplace) asociovaný s
  daným učiteľom vo vzťahu N:1 */
$lessons = $teacher['Lessons'];
/* do premennej $lessons načíta všetky asociované hodiny vo forme objektu
  triedy NDORecordSet (ide o vzťah N:1) */
$courses = $teacher['Courses'];
/* do premennej $courses načíta všetky asociované kurzy vo forme objektu
  triedy NDORecordSet (ide o vzťah N:M) */
foreach ($courses as $course) print $course['Subject']['name'];
/* vypíše názvy predmetov asociovaných so všetkými kurzami daného učiteľa
  */
?>
```

### 3.2.2 NDOConditions

*NDOConditions* je objekt, ktorý definuje podmienky určujúce čo a akým spôsobom sa má z databázy získať či v databáze upraviť. Jeho úlohou je generovať SQL klauzuly ako *WHERE*, *ORDER BY*, *LIMIT*, *OFFSET*. Výhodou samostatnej triedy na definovanie podmienok, je jej jednoduchá rozšíriteľnosť o ďalšie formy podmienok. Používa sa ako vstupný parameter pre metódy na získavanie záznamov a úpravu existujúcich záznamov.

### 3.2.3 Overloading

Triedy *NDODatabase*, *NDOTable* a *NDORecord* sú potomkami triedy *NDOOverload*, ktorá zabezpečuje overloading neexistujúcich atribútov na getter/setter metódy<sup>2</sup> a sprístupňuje privátne atribúty s prefixom '\_' na čítanie<sup>3</sup>.

<sup>2</sup> napr. `$object->value` volá `$object->getValue()`

<sup>3</sup> napr. `$object->value` vráti `$object->_value`, ale neumožní zápis

### 3.2.4 Výkon

Použitie akéhokoľvek frameworku spravidla vyžaduje určitú réžiu, čo sa nevyhnutne odzrkadľuje na výkone. Pri databázových abstraktných vrstvách (a ORM systémoch zvlášť) je táto otázka veľmi citlivá. Na zmiernenie niektorých spomaľujúcich efektov použitých knižníc som použil nasledovné opatrenia:

#### 3.2.4.1 *Metadata cache*

Získavanie metadát (schémy databázy) priamo pomocou dotazov na databázu je síce výhodné pri zmene schémy, ale v prípade bezstavovej web aplikácie fungujúcej systémom request-response si to môže vyžiadať zbytočné náklady na získavanie rovnakých údajov. Preto je celá schéma databázy uložená v dočasnom súbore, ktorý sa vytvára automaticky, ak ešte neexistuje. V prípade zmeny schémy, potom stačí tento súbor vymazať a pri najbližšom prístupe k modelu sa vytvorí aktuálny.

#### 3.2.4.2 *Lazy initialization*

Všetky atribúty, pre získanie ktorých je nutný dotaz na databázu, sú inicializované až v momente ich použitia. Tým sa zabráni zbytočnej réžii spojenej so získavaním údajov, ktoré sa v danej request-response iterácii vôbec nepoužívajú.

#### 3.2.4.3 *LoadRelations*

V praxi sa často vyskytujú prípady, kedy je nutné pristupovať k asociovaným objektom množiny objektov. Napríklad máme množinu študentov a potrebujeme pristupovať k predmetom ktoré majú zapísané. Ak by sme postupne prechádzali každého študenta a zisťovali jeho predmety, museli by sme vykonať veľa dotazov na databázu. Ak by sme ďalej chceli pristupovať k nejakým asociovaným objektom týchto predmetov, počet dotazov bude ďalej rásť. Na ošetrenie tejto situácie slúži metóda *LoadRelations*, ktorá pre množinu objektov jediným dotazom získa všetky asociované objekty požadovaného typu a správne ich priradí k jednotlivým objektom. V našom príklade teda zavoláme túto metódu na množine študentov s požiadavkou o predmety. Získame tak jedným

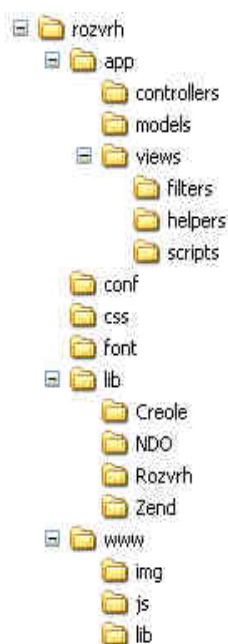
dotazom všetky predmety, ktoré majú zapísané študenti z danej množiny a tieto budú navyše správnym spôsobom priradené k jednotlivým študentom. Na takto získanej množine predmetov môžeme opäť volať metódu *LoadRelations* a získať tak asociované objekty pre tieto predmety.

#### 3.2.4.4 *Stored procedures*

Hoci NDO ponúka určitú mieru SQL abstrakcie, na zložitejšie dotazy je stále nutné použiť SQL dotazy. Najefektívnejším spôsobom ako riešiť priame dotazy je použitie „uložených procedúr“ priamo na strane databázy. Takéto riešenie je nielen najvýkonnejšie, ale umožňuje aj parametrizované procedúry, ktorých vhodným použitím sa dajú zložité dotazy zjednodušiť a sprehľadniť.

### 3.3 *Adresárová štruktúra kódu*

Zdrojový kód web aplikácie je členený do adresárov podľa obrázku 7.



Obrázok 7: Štruktúra adresárov

Význam jednotlivých adresárov je nasledovný:

- *app* – aplikačné súbory členené podľa MVC
  - *controllers* – action controllers, ktoré vykonávajú požadované

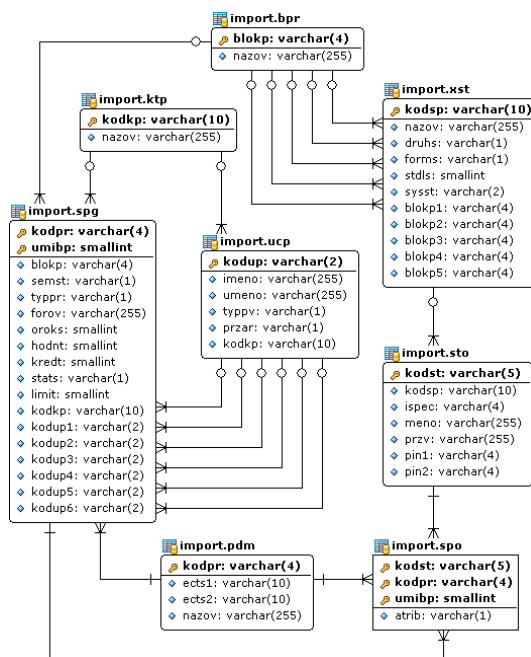
- akcie
  - *models* – dátové objekty (rozšírenia triedy *NDORRecord*)
  - *views* – pohľady zabezpečujúce renderovanie výstupov
    - *filters* – triedy na filtrovanie dát do formátu vhodného na zobrazenie
    - *helpers* – pomocné zobrazovacie funkcie využívané v šablónach
    - *scripts* – skripty slúžiace ako šablóny
- *css* – zdrojové súbory pre generovanie kaskádových štýlov
- *font* – TrueType fonty využívané pri generovaní PDF súborov
- *lib* – PHP5 knižnice
  - *Creole* – DBAL pre prístup k databáze
  - *NDO* – Nifty Data Objects (vlastná ORM knižnica)
  - *Rozvrh* – vlastné knižnice a rozšírenia Zend Frameworku
  - *Zend* – Zend Framework Library
- *www* – obsahuje súbory prístupné cez web (index.php)
  - *img* – obrázky a ikony používané na web stránke
  - *js* – vlastné javascriptové knižnice
  - *lib* – knižnice, ktoré majú byť prístupné cez webové rozhranie (javascript, CSS)

### 3.4 Import dát

Na to aby bolo možné systém v skutočnosti využívať bolo nutné od začiatku myslieť na dáta, ktoré sa používajú v súčasnom informačnom systéme fakulty. Bez podpory importu dát by bolo totiž veľmi problematické naplnenie databázy, ale predovšetkým jej udržiavanie v aktuálnom stave.

### 3.4.1 Zdroj a formát dát

Ako zdroj dát pre potreby projektu slúžia CSV súbory exportované z fakultného informačného systému *Študent*. Jedná sa o osem súborov, ktorých štruktúra a vzájomné prepojenia sú znázornené na obrázku 8.



Obrázok 8: Export zo systému Študent

#### 3.4.1.1 KTP

Zoznam pracovísk.

Atribúty:

- *kodkp* – skratka pracoviska (primárny kľúč)
- *nazov* – názov pracoviska

#### 3.4.1.2 PDM

Zoznam názvov a ECTS kódov predmetu.

Atribúty:

- *kodup* – interný kód predmetu (primárny kľúč)
- *ects1, ects2* – ECTS kódy predmetu
- *nazov* – celý názov predmetu

### 3.4.1.3 UCP

Tabuľka učiteľov.

Atribúty:

- *kodpr* – interný kód učiteľa (primárny kľúč)
- *imeno* – skratka mena učiteľa
- *umeno* – celé meno učiteľa
- *typpv* – typ pracovného vzťahu (D/E/?)
- *przar* – pracovné zaradenie (A/U/V)
- *kodkp* – referencia na pracovisko (*KTP*)

atribúty *typpv* a *przar* v našom systéme nevyužívame.

### 3.4.1.4 SPG

Kurzy ponúkané študijným programom.

Atribúty:

- *kodpr* – referencia na predmet (*PDM*)
- *umibp* – doplnok *kodpr* na primárny kľúč resp. rozlíšenie jednotlivých kurzov s rovnakým predmetom
- *blokp* – referencia na blok (*BPR*)
- *semst* – semester ('Z'/'L')
- *typpr* – kurz je v študijnom pláne ako povinný (P), voliteľný (V) alebo alternatívny (A)
- *forov* – forma a rozsah výučby (čiarkou oddelené rozsahy textového formátu, kde prvý znak reprezentuje typ/formu výučby nasledovaný počtom vyučovacích hodín, za ktorým ešte môže nasledovať poznámka)
- *oroks* – odporúčaný rok štúdia
- *hodnt* – percentuálny podiel hodnotenia predmetu pripadajúci na výučbovú časť semestra

- *kred*t – počet kreditov
- *stats* – status predmetu ('s' – suspendovaný)
- *limit* – nevyužívaný limit na počet študentov
- *kodkp* – referencia na pracovisko (*KTP*)
- *kodup1..kodup6* – referencie na učiteľov (*UCP*)

#### 3.4.1.5 BPR

Číselník študijných blokov

- *blokp* – 4-znaková skratka študijného bloku (prvé dva znaky značia odbor a posledné dva znaky definujú špecializáciu)
- *nazov* – celý názov študijného bloku

#### 3.4.1.6 XST

Tabuľka študijných skupín

- *kodsp* – skratka študijnej skupiny
- *nazov* – celý názov študijnej skupiny
- *druhs* – druh štúdia ('B' - bakalárske, 'M' - magisterské)
- *forms* – forma štúdia ('d' - denné)
- *stdls* – štandardná dĺžka štúdia v rokoch
- *sysst* – systém štúdia ('ka' – dobiehajúce štúdium, 'kp' – nové štúdium)
- *blokp1..blokp5* – referencie na študijné bloky (*BPR*). Tieto však môžu byť uvedené aj neúplne - iba odborom a špecializáciu má určený každý študent zväšť v atribúte *ispec* (napr. MA - určený iba odbor matematika)

#### 3.4.1.7 STO

Tabuľka študentov

- *kodst* – interný kód študenta
- *kodsp* – referencia na študijnú skupinu (*XST*)



- *ispec* – najviac 6 konkatenovaných 2-znakových špecializácií určujúcich spolu s informáciou o odbore (zo študijnej skupiny) študijné bloky predmetov do ktorých študent patrí
- *meno* – krstné meno študenta
- *przv* – priezvisko študenta
- *pin1* – posledné štvorčísle čísla študenta na ISIC karte
- *pin2* – posledné štvorčísle čísla ISIC karty

#### 3.4.1.8 SPO

Tabuľka študentských zápisov.

Atribúty

- *kodst* – referencia na študenta (*STO*)
- *kodpr*, *umibp* – referencia na kurz (*SPG*), *kodpr* je zároveň referenciou na predmet (*PDM*)
- *attribute* – informačný znak o tom, či má študent daný kurz zapísaný povinne (P), voliteľne (V) alebo alternatívne (A), prípadne či má predmet zapísaný druhýkrát (2).

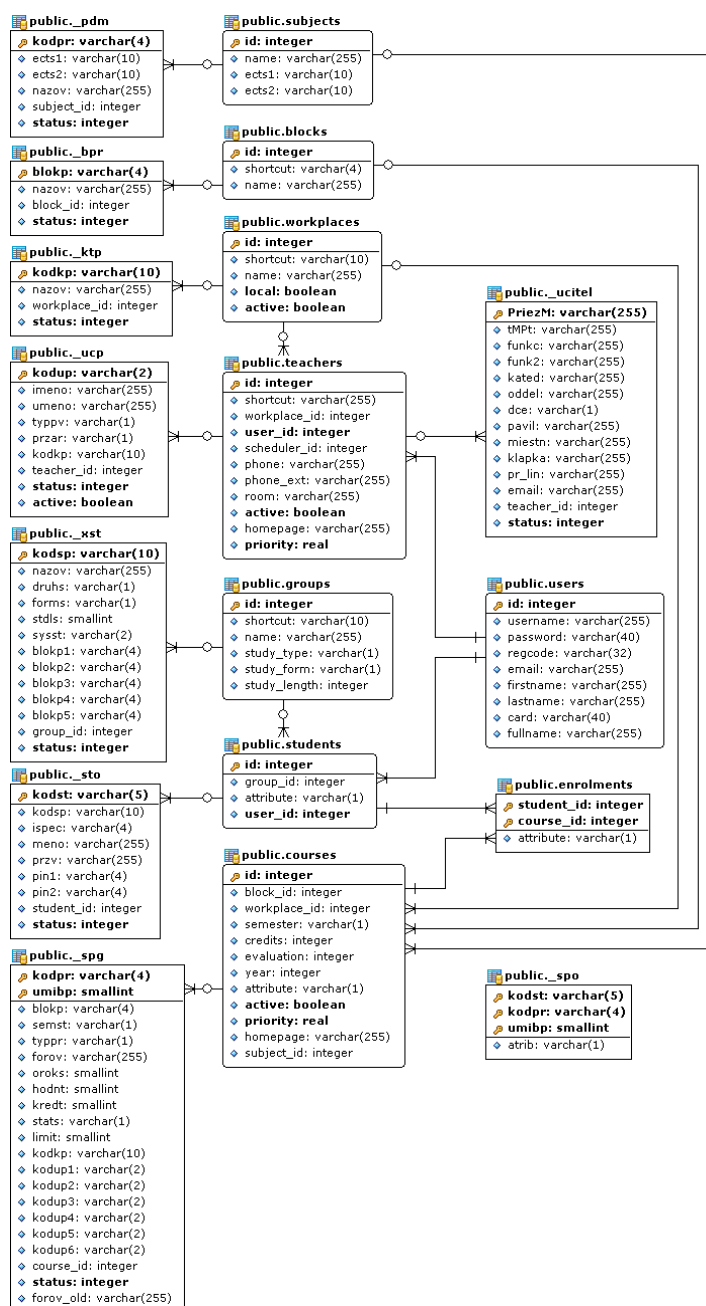
#### 3.4.1.9 Ucitel

Súbor *ucitel.csv* obsahuje podrobnejšie informácie o zamestnancoch fakulty poskytnutý zo systému *Personalistika a mzdy (PaM)*:

- *PriezM* – skratka mena v tvare „Priezvisko K.“ (napr Košťál J.) kompatibilná s atribútom *imeno* tabuľky *UCP*
- *tMPt* – celé meno v tvare „tituly Meno Priezvisko, tituly“ (napr. Mgr. Jaroslav Košťál, PhD.)
- *funkc*, *funkc2* – funkcie učiteľa
- *kated* – skratka pracoviska
- *pavil*, *miestn* – pavilón a miestnosť, kde učiteľ pôsobí
- *klapka*, *pr\_linka* – telefonický kontakt (klapka a priama linka)
- *email* – email (potrebný na zaslanie registračného kódu)

### 3.4.2 Mapovanie

Jednotlivé exportované tabuľky sú mapované na entity nášho systému podľa obrázku . Do pôvodných tabuliek sú pridané referencie na zodpovedajúce záznamy v našom systéme a atribút *status* informujúci o stave importu pre potreby aktualizácie. Niektoré tabuľky majú doplňujúce atribúty využívané pri synchronizácii.



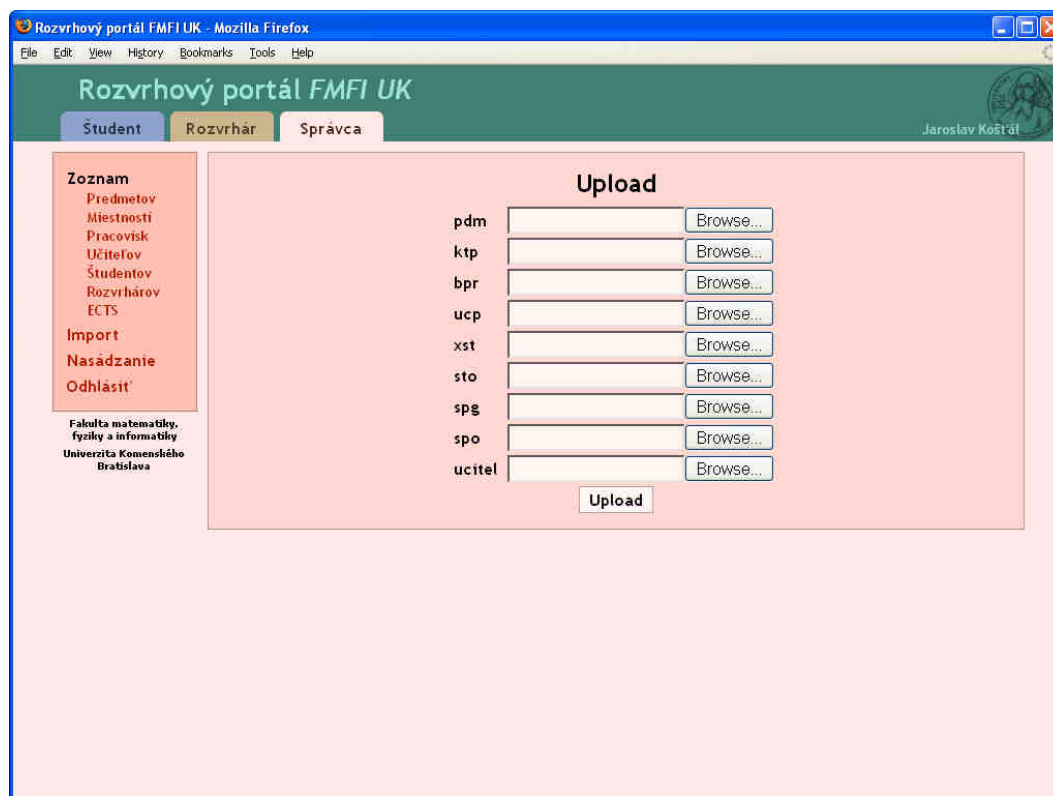
Obrázok 9: Mapovanie importovaných dát na vlastný model

### 3.4.3 Aktualizácia

Import je nutné vykonať každý rok, kvôli aktualizácii študijného programu ako aj študentských zápisov, prípadne zmien v ďalších entitách. Je možné ho však v prípade potreby vykonať aj v priebehu roka (napr. opravný zápis)

Samotný import prebieha v dvoch krokoch:

#### 3.4.3.1 Upload

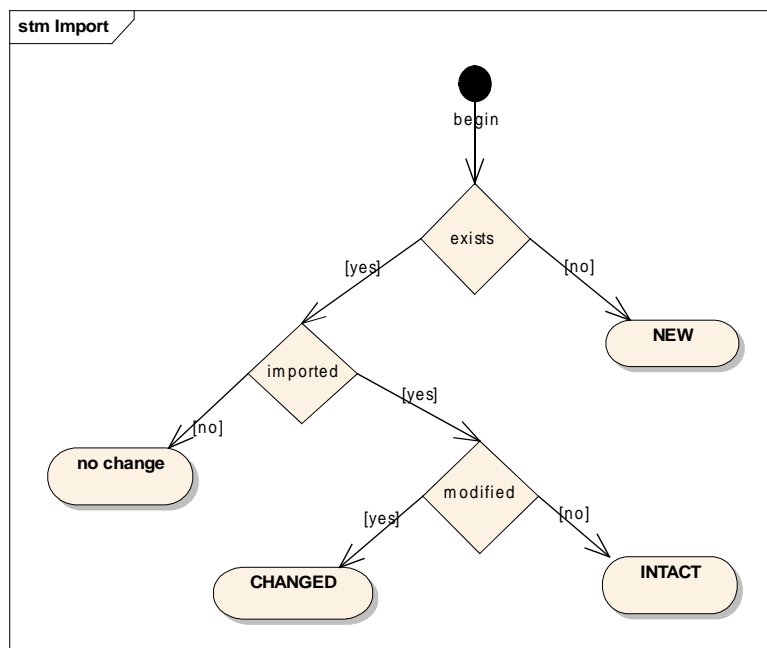


Obrázok 10: Import (upload súborov)

Upload súborov môže vykonať správca prostredníctvom portálu. Počas neho sa iba načítajú príslušné tabuľky a upraví sa *status* jednotlivých záznamov (viď rozhodovací diagram na obrázku 11):

- *NEW* – nový doposiaľ neexistujúci záznam
- *CHANGED* – zmenený záznam (ak sa zmenil ľubovoľný atribút)
- *INTACT* – záznam nebol zmenený

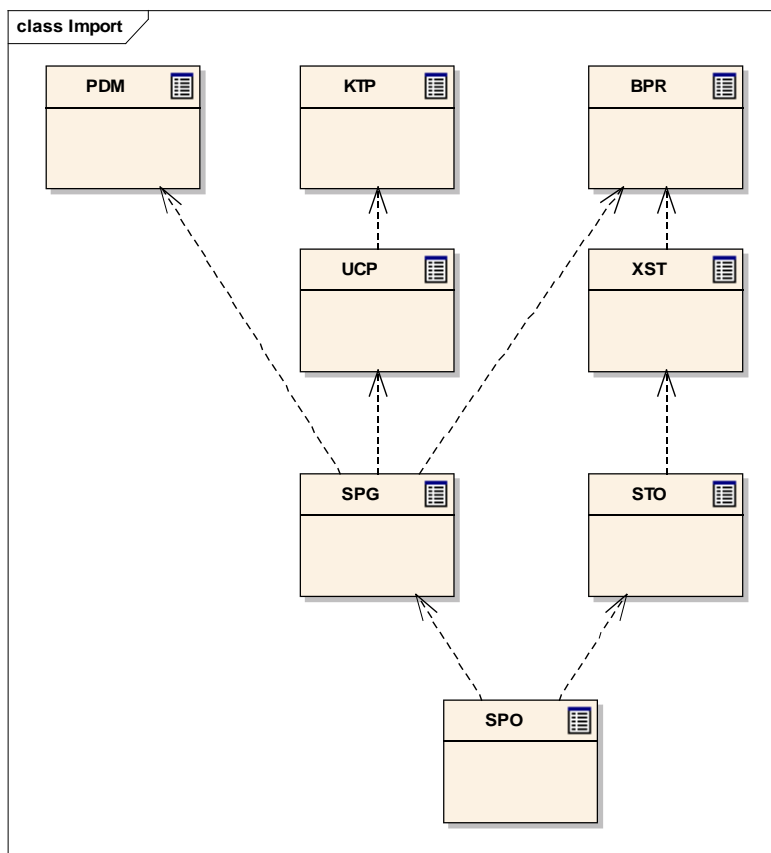
Nahrať sa môže ľubovoľná podmnožina súborov v ľubovoľnom poradí.



Obrázok 11: Rozhodovací diagram

### 3.4.3.2 Synchronizácia

Synchronizácia načítaných tabuliek s tabuľkami v systéme musia prebiehať v danom poradí podľa závislostí znázornených na obrázku 12.



Obrázok 12: Závislosti pri importe

Pri synchronizácii sa prihliada na *status* záznamov a podľa toho sa vykonávajú ďalšie činnosti:

- *NEW* – pridá sa nový záznam a nastaví *status* na *IMPORTED*
- *CHANGED* – synchronizuje sa a nastaví *status* na *IMPORTED*
- *INTACT* – nerobí sa nič
- *IMPORTED* – nerobí sa nič (prípadne sa môže vymazať záznam v našom systéme, keďže záznam nebol po uploade upravený – toto by ale vyžadovalo vykonať upload príslušnej tabuľky bezprostredne pred synchronizáciou)

Synchronizácia každej tabuľky prebieha vlastným spôsobom. Za zmienku stojí pomocný atribút *forov\_old* tabuľky *SPG*, ktorý uchováva pôvodný atribút *forov* po predchádzajúcej synchronizácii a umožňuje tak zamedziť strate prepojení rozsahov v prípade, že bol aktualizovaný kurz bez zmeny rozsahov.

### 3.4.4 Budúcnosť

V súčasnosti pracuje projekt s údajmi poskytnutými zo systému *Študent*, doplnených o informácie o učiteľoch zo systému *PaM*. Tieto údaje však často obsahujú rôzne nekonzistentnosti a redundantné dáta vyplývajúce zo spôsobu ich fungovania a vzájomnej (ne)komunikácie jednotlivých subsystémov.

Hoci v minulosti umožnili súčasné informačné systémy výrazne uľahčiť školskú agendu, na dnešné pomery sú už značne nevyhovujúce, a preto dochádza k ich postupnému náhrádzaniu novými systémami, ktoré by bolo v ďalšej fáze vývoja projektu vhodné použiť na import či priamu výmenu údajov. Jedná sa predovšetkým o nasledovné systémy:

- **CDO** (*Centrálne databáza osôb*) – systém by poskytoval integrované údaje o všetkých osobách na škole (viď [8], [9]). Pre náš projekt by boli dôležité informácie o učiteľoch a študentoch, ktoré budú centralizované a bez neuhov súčasného systému.
- **Študent 2** – v blízkej budúcnosti má dôjsť k nahradeniu jednotlivých inštancií systému *Študent* na fakultách univerzity novým celouniverzitným systémom, ktorý by mal poskytovať kvalitnejšie údaje o predmetoch, študijných plánoch a zápisoch študentov.

Za predpokladu, že sa výraznejšie nezmení samotný systém štúdia, nemalo by byť veľmi problematické adaptovať náš projekt na nový informačný systém školy.

## 3.5 Rozvrhový portál



Obrázok 13: Rozvrhový portál

### 3.5.1 Autentifikácia

Neprihlásení používateľia portálu majú iba obmedzené možnosti jeho využitia, na sprístupnenie ďalších služieb systému musí byť používateľ autentifikovaný aby bolo možné overiť či má príslušné oprávnenia (autorizácia) na jednotlivé úkony.

#### 3.5.1.1 Registrácia

Prvým krokom k prístupu na chránené časti systému je registrácia, pri ktorej používateľ získa (resp. si určí) prihlasovacie údaje.

Tú je možné vykonať dvoma spôsobmi:

- **registračným kódom** – emailom je používateľovi zaslaný registračný kód, ktorý slúži na overenie identity
- **ISIC kartou** – na základe čísla študenta a čísla ISIC karty je vykonaná identifikácia študenta.

The screenshot shows a web browser window with the title "Rozvrhový portál FMFI UK". The page has a navigation bar with buttons for "Študent", "Učiteľ", "Rozvrhár", and "Správca", and a "Neprihlásený" status indicator. On the left, there is a sidebar with "Prihlásiť" and "Zoznam" (Predmetov, Učiteľov, Miestnosti) and the faculty name "Fakulta matematiky, fyziky a informatiky, Univerzita Komenského Bratislava". The main content area is titled "Registrácia" and contains the following form fields:

- Registračný kód:
- Zadajte nové prihlasovacie údaje:
- Prihlasovacie meno:  (a-zA-Z0-9)
- Heslo:  (min. 8 znakov bez diakritiky)
- Registrovat' button

A red note at the bottom states: "\* Študenti môžu využiť registráciu na základe ISIC karty".

Obrázok 14: Registrácia pomocou registračného kódu

The screenshot shows the same web browser window and navigation bar as in the previous image. The main content area is titled "Registrácia" and contains the following text and form fields:

Na ISIC karte študentov FMFI UK sú pod fotografiou dve čísla. Prvé je číslo študenta a druhé je číslo ISIC karty. Tieto dve čísla slúžia na registráciu do rozvrhového portálu. Napíšte ich do príslušných políček nižšie a kliknite na tlačítko "Registrovat'". Obrázok názorne ukazuje, o ktoré čísla sa jedná.

Below the text is an image showing three cards: an "International Student Identity Card" (ISIC) with a smiley face, a "Číslo študenta" card with the number "000 123 4567", and a "Číslo ISIC karty" card with the number "S 123 123 123".

The form fields are:

- Číslo študenta:  (10 miestne)
- Číslo ISIC karty:  (13 miestne)
- Zadajte nové prihlasovacie údaje:
- Prihlasovacie meno:  (bez diakritiky)
- Heslo:  (min. 8 miestne bez diakritiky)
- Registrovat' button

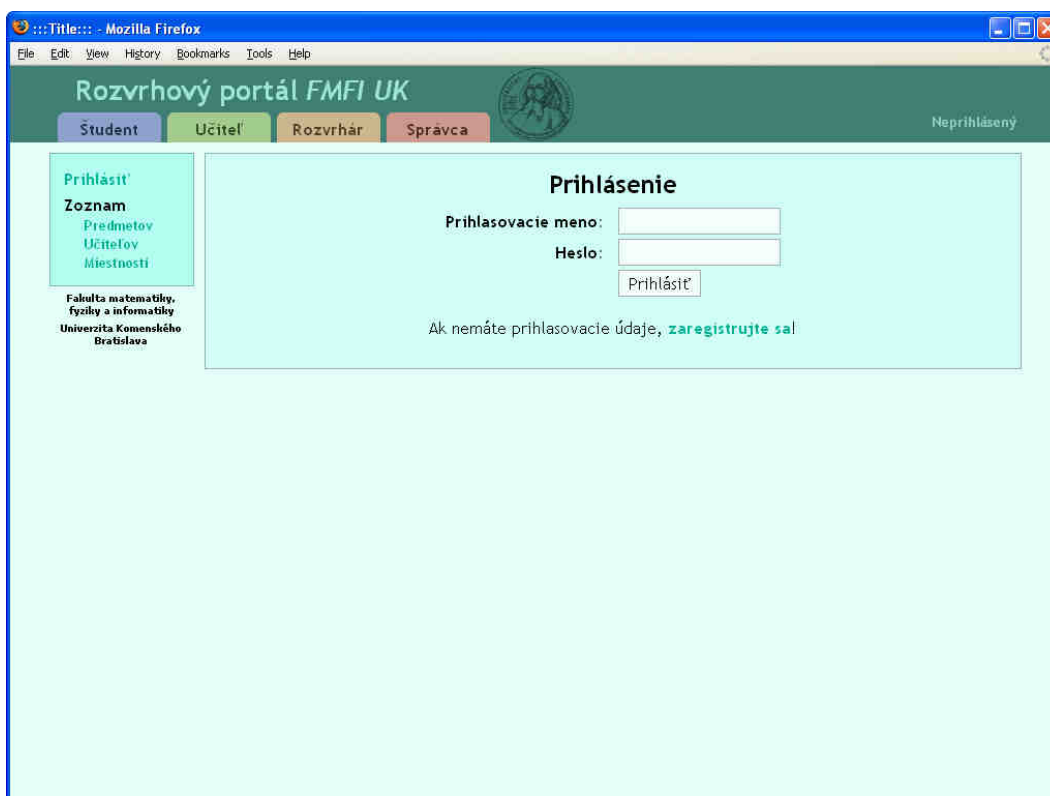
Obrázok 15: Registrácia pomocou ISIC karty



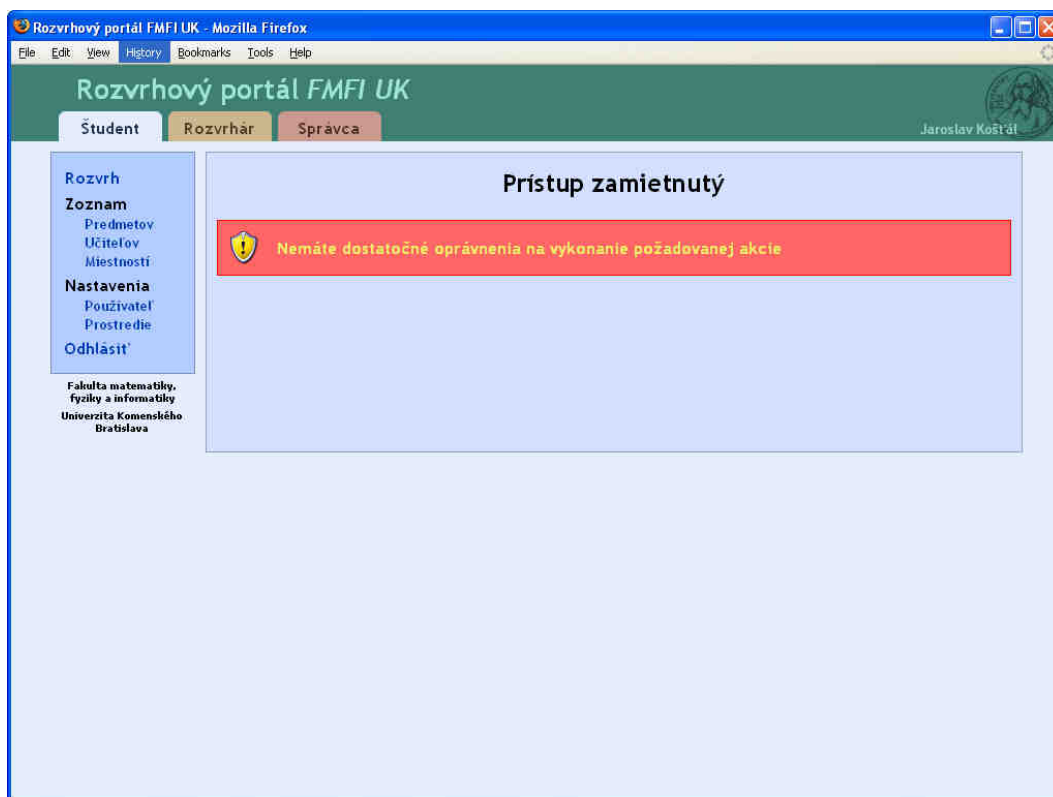
### 3.5.1.2 Prihlásenie

Registrovaný používateľ môže prísť k prihláseniu sa do systému. Používa už iba svoje prihlasovacie meno a heslo, ktoré si zvolil pri registrácii.

Po prihlásení získa prístup k jednotlivým roliam prostredníctvom záložiek v záhlaví stránky. Každá rola má inú farbu a po jej aktivovaní sa zmení farebné ladenie celého prostredia, pre lepšiu orientáciu v systéme.



Obrázok 16: Prihlásenie



Obrázok 17: Nepovolený prístup

V prípade pokusu o prístup na stránku, ku ktorej nemá daný používateľ dostatočné oprávnenia, zobrazí sa chybové hlásenie.

### 3.5.1.3 Odhlásenie

Na korektné odhlásenie zo systému slúži posledná položka menu. V prípade, že sa používateľ predpísaným spôsobom neodhlási, podstupuje vedome riziko možného zneužitia. Toto je znížené automatickým odhlásením používateľov, ktorí sú stanovenú dobu neaktívni.

## 3.5.2 Zoznamy

The screenshot shows the 'Rozvrhový portál FMFI UK' web application. The main content area displays a data grid of lecture rooms. The table has the following columns: Skratka, Názov, Kapacita, Pracovisko, PC, and Projektor. The rows list various lecture rooms with their respective abbreviations, names, capacities, and equipment. A search bar is located at the top left of the grid. The sidebar on the left contains navigation links for different user roles and administrative functions. The bottom of the grid shows pagination information: Limit 10, Strana 1, / 3 (Záznamy 1-10 z 25).

	Skratka	Názov	Kapacita	Pracovisko	PC	Projektor			
<input type="checkbox"/>	C	Poslucháreň C	100						
<input type="checkbox"/>	F1	Poslucháreň F1	200						
<input type="checkbox"/>	F2	Poslucháreň F2	200						
<input type="checkbox"/>	F108		49						
<input type="checkbox"/>	F109		49						
<input type="checkbox"/>	F232		12						
<input type="checkbox"/>	F247		20						
<input type="checkbox"/>	F328		40						
<input type="checkbox"/>	A	Poslucháreň A	300		1				
<input type="checkbox"/>	B	Poslucháreň B	200		1				

Obrázok 18: Datagrid (zoznam miestností)

Zoznamy slúžia na výpis záznamov jednotlivých entít. Umožňujú pohodlné listovanie, vyhľadávanie a triedenie. Ak má používateľ príslušné oprávnenia, môže tabuľku priamo upravovať a zmeny ukladať. Po kliknutí na ikonu zobrazenia, sa objaví detail záznamu s linkami na asociované entity.

The screenshot shows the 'Rozvrhový portál FMFI UK' web application in a Mozilla Firefox browser. The interface includes a navigation menu with 'Študent', 'Rozvrhár', and 'Správca' tabs. A sidebar on the left contains a 'Zoznam' menu with options like 'Predmetov', 'Miestnosti', 'Pracovisk', 'Učiteľov', 'Študentov', 'Rozvrhárov', 'ECTS', 'Import', 'Nasádzanie', and 'Odhlásiť'. Below this is the faculty name: 'Fakulta matematiky, fyziky a informatiky, Univerzita Komenského Bratislava'.

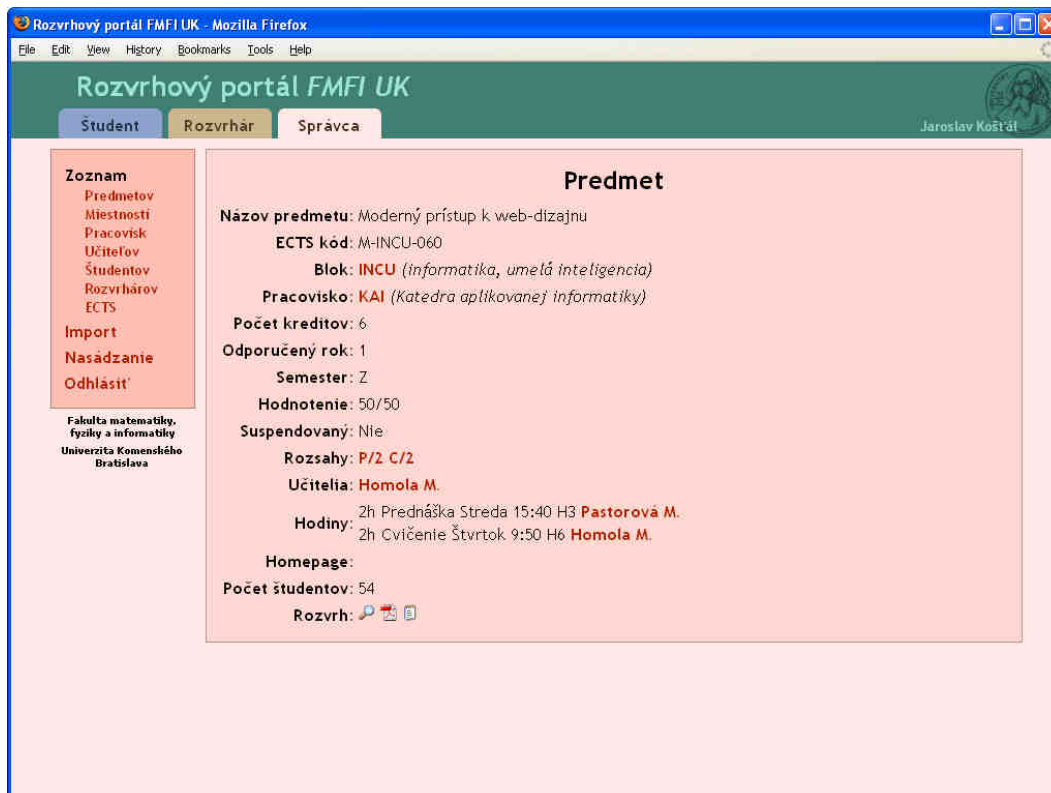
The main content area features a search bar labeled 'Hľadať:' and a table for editing lecture records. The table has columns for 'Skratka', 'Názov', 'Kapacita', 'Pracovisko', 'PC', and 'Projektor'. Each row includes a checkbox for selection and a set of icons for editing (magnifying glass, pencil, eraser, and refresh).

	Skratka	Názov	Kapacita	Pracovisko	PC	Projektor	
<input type="checkbox"/>	C	Poslucháreň C	100				[Icons]
<input type="checkbox"/>	F1	Poslucháreň F1	200				[Icons]
<input type="checkbox"/>	F2	Poslucháreň F2	200				[Icons]
<input type="checkbox"/>	F108		49				[Icons]
<input type="checkbox"/>	F109		49				[Icons]
<input type="checkbox"/>	F232		12				[Icons]
<input type="checkbox"/>	F247		20				[Icons]
<input type="checkbox"/>	F328		40				[Icons]
<input type="checkbox"/>	A	Poslucháreň A	300	FIFUK	<input type="checkbox"/>	<input checked="" type="checkbox"/>	[Icons]
<input type="checkbox"/>	B	Poslucháreň B	200			1	[Icons]

At the bottom of the table, there are navigation controls: 'Limit 10', 'Strana 1 / 3 (Záznamy 1-10 z 25)', and a 'Načítavam formulár...' button.

Obrázok 19: Úprava záznamu

Úpravu záznamu je možné vykonať buď priamo v tabuľke (čo vyžaduje javascript), alebo prostredníctvom formuláru na samostatnej stránke. Samostatný formulár spravidla umožňuje rozsiahlejšie úpravy aj asociovaných objektov.



Obrázok 20: Detaily predmetu

### 3.5.3 Rozvrhy

Všetky rozvrhy sú generované na základe aktuálnych dát v databáze. Entity, ku ktorým je možné zobraziť rozvrh sú nasledovné:

- Student – rozvrh študenta podľa reálne zapísaných kurzov
- Teacher – rozvrh učiteľa podľa predmetov, ktoré vyučuje
- Room – rovrh miestnosti
- Course – rozvrh kurzu

Keďže systém počíta s kolíznymi rozvrhmi, zobrazovanie kolidujúcich hodín sa riadi ich prioritami vypočítanými vzhľadom na entity, pre ktoré sa rozvrh renderuje. V grafickom zobrazení sa kolidujúca hodina s nižšou prioritou vypíše textovou formou pod rozvrhom.

	Pondelok	Utorok	Streda	Štvrtok	Piatok
8:10					
9:00					
9:50				C Plávanie Potocký	
10:40					
11:30		P XII Tvorba interaktívnych internetových aplikácií Pachterová M			
12:20					
13:10			S MZA1 Aproximácia funkcií a ortogonálne polynómy v numerickej matematike		
14:00					
14:50					
15:40					
16:30					
17:20					
18:10					
19:00					

Obrázok 21: Ukážka rozvrhu študenta

### 3.5.4 Nastavenia

Prihlásení používatelia môžu vykonávať rôzne nastavenia v závislosti od svojej role.

#### 3.5.4.1 Používateľské nastavenia

Každý používateľ má prístup k používateľským nastaveniam, kde si môže nastaviť niektoré základné údaje ako napríklad email.

Učitelia si tu môžu definovať svoje časové preferencie.

#### 3.5.4.2 Nastavenia prostredia

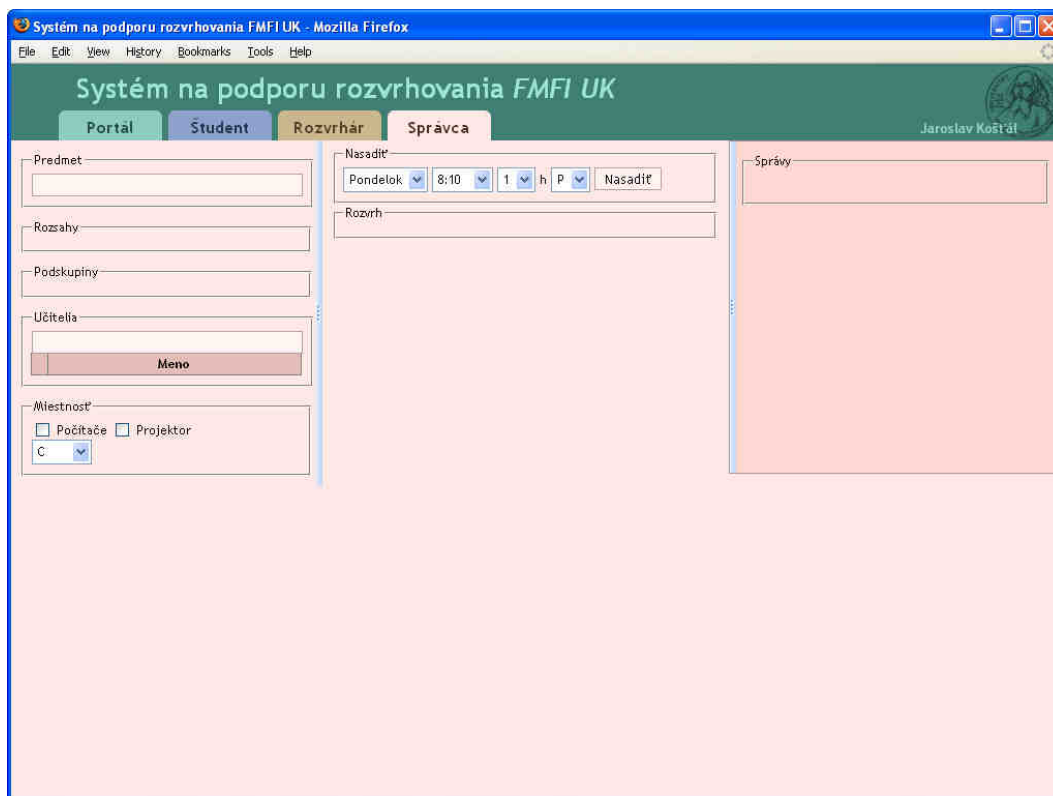
V nastaveniach prostredia je možné upraviť spôsob zobrazovania rozvrhu.

#### 3.5.4.3 Systémové nastavenia

Administrátor má prístup aj k systémovým nastaveniam, ktoré sú

uchované v tabuľke *options* a určujú základné charakteristiky systému akými sú napríklad aktuálny semester, počet týždňov semestra, fáza nasádzania a pod.

### 3.6 Systém na podporu rozvrhovania



Obrázok 22: Systém na podporu rozvrhovania

Používateľské rozhranie systému na podporu rozvrhovania sa člení na tri oblasti:

- v ľavej časti prebieha výber kurzu, jeho rozsahu, učiteľa, miestnosti a podskupiny, pokiaľ je pre daný rozsah určená
- v strednej časti sa zobrazuje čas, dĺžka a forma nasádzanej hodiny, pod ktorou sa zobrazuje rozvrh reflektujúci aktuálny stav nasadenia. Tento poskytuje informácie o najvhodnejšom čase pre nasadenie rozsahu a obsahuje pre každý časový úsek tri farebne odlišené polia reprezentujúce vhodnosť nasadenia pre:
  - študentov, ktorí majú zapísaný kurz prislúchajúci zvolenému

- rozsahu (zelená)
- učiteľa, ktorého sme pre daný rozsah zvolili (modrá)
- miestnosť (tyrkysová)
- v pravej časti sa zobrazujú informácie o chybách, kolíziách či úspešnom nasadení. Navyše sa tu zobrazujú podrobnosti o vybranom rozsahu a časovom úseku.

Samotné nasádzanie prebieha v niekoľkých krokoch:

- výber predmetu v poli s automatickým dopĺňaním názvov predmetov pre aktuálne prihláseného rozvrhára.
- výber rozsahu príslušných kurzov zodpovedajúcich vybranému predmetu
- výber podskupiny, v prípade, že sa daný rozsah delí na podskupiny
- výber učiteľa buď z učiteľov asociovaných podľa študijného plánu s príslušným kurzom, alebo doplnením učiteľa vlastného podľa rozvrhára
- výber časového úseku na základe zobrazeného multirozvrhu študentov, učiteľa a miestností (po kliknutí do rozvrhu sa nastaví príslušný čas)
- výber miestnosti s prípadným definovaním požiadaviek na jej vybavenie
- prípadné dodefinovanie inej dĺžky a typu vyučovacej hodiny, ak sa líši od rozsahu.



The figure shows three sequential screenshots of a software interface for selecting a subject and its scope. The interface is divided into several sections: 'Predmet' (Subject), 'Rozsahy' (Scopes), 'Podskupiny' (Subgroups), 'Učítelia' (Teachers), and 'Miestnosť' (Room).

**Screenshot 1 (Left):** The 'Predmet' field contains 'web'. The 'Rozsahy' section shows a list of scopes: 'Databázy pre WEB', 'Moderný prístup k web-dizajnu', and 'Tvorba webovských dokumentov'. The 'Učítelia' section has a table with one row: 'Meno'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a dropdown menu set to 'C'.

**Screenshot 2 (Middle):** The 'Predmet' field contains 'Moderný prístup k web-dizajnu'. The 'Rozsahy' section shows a loading indicator: 'Načítavam rozsahy...'. The 'Učítelia' section has a table with one row: 'Meno' and one entry: 'Homola M.'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a dropdown menu set to 'C'.

**Screenshot 3 (Right):** The 'Predmet' field contains 'Moderný prístup k web-dizajnu'. The 'Rozsahy' section shows a table with two rows:
 

	Typ	Blok
<input checked="" type="radio"/>	P/2	INCU
<input type="radio"/>	C/2	INCU

 The 'Učítelia' section shows a loading indicator: 'Načítavam učiteľov...'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a dropdown menu set to 'C'.

Obrázok 23: Nasádzanie (výber predmetu a jeho rozsahu)

The figure shows three sequential screenshots of a software interface for selecting a teacher and a room. The interface is divided into several sections: 'Predmet' (Subject), 'Rozsahy' (Scopes), 'Podskupiny' (Subgroups), 'Učítelia' (Teachers), and 'Miestnosť' (Room).

**Screenshot 1 (Left):** The 'Predmet' field contains 'Moderný prístup k web-dizajnu'. The 'Rozsahy' section shows a table with two rows:
 

	Typ	Blok
<input checked="" type="radio"/>	P/2	INCU
<input type="radio"/>	C/2	INCU

 The 'Učítelia' section has a table with one row: 'Meno' and two entries: 'Pastor K.' and 'Pastorová M.'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a dropdown menu set to 'C'.

**Screenshot 2 (Middle):** The 'Predmet' field contains 'Moderný prístup k web-dizajnu'. The 'Rozsahy' section shows a table with two rows:
 

	Typ	Blok
<input checked="" type="radio"/>	P/2	INCU
<input type="radio"/>	C/2	INCU

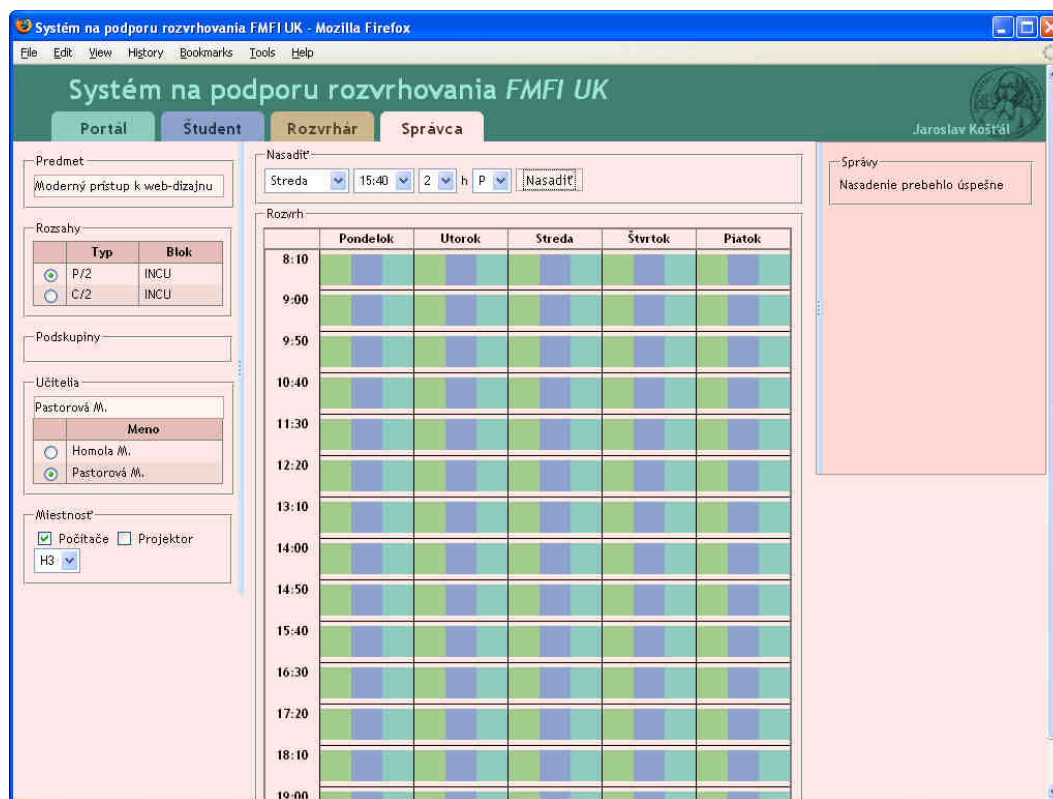
 The 'Učítelia' section has a table with one row: 'Meno' and two entries: 'Homola M.' and 'Pastorová M.'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a loading indicator: 'Načítavam miestnosti...'. The 'Počítače' checkbox is checked.

**Screenshot 3 (Right):** The 'Predmet' field contains 'Moderný prístup k web-dizajnu'. The 'Rozsahy' section shows a table with two rows:
 

	Typ	Blok
<input checked="" type="radio"/>	P/2	INCU
<input type="radio"/>	C/2	INCU

 The 'Učítelia' section has a table with one row: 'Meno' and two entries: 'Homola M.' and 'Pastorová M.'. The 'Miestnosť' section has checkboxes for 'Počítače' and 'Projektor', and a dropdown menu set to 'H3'. The 'Počítače' checkbox is checked.

Obrázok 24: Nasádzanie (výber učiteľa a miestnosti)



Obrázok 25: Úspešne nasadený predmet

Pri snahe nasadiť do rozvrhu kurz, ktorý by porušoval nutné ohraničujúce podmienky, je táto skutočnosť zobrazená v pravom oznamovacom poli.

### 3.7 Testovacia prevádzka

Keďže projekt sa stále vyvíja, jeho aktuálny stav ako aj text práce sa nachádza na stránke <http://rozvrh.afes.sk>. Dokončené časti systému je možné si na tejto stránke vyskúšať v testovacej prevádzke.

## Záver

V práci som analyzoval problém tvorby rozvrhu s dôrazom na špecifiká a potreby našej fakulty a navrhol som nový individualizovaný prístup k jeho riešeniu. Tento prístup som pretavil do návrhu celého rozvrhového systému spočívajúceho v rozvrhovom portáli a systéme na podporu rozvrhovania.

Následne som implementoval značnú časť návrhu s predpokladom nasadenia do skúšobnej prevádzky v septembri 2007, kedy bude možné implementáciu doladiť a prispôbiť dodatočným požiadavkám. V januári 2008 sa bude môcť systém využiť v reálnom prostredí na nasadenie letného rozvrhu paralelne so starým systémom. Ak sa systém osvedčí, nasledujúci semester už bude slúžiť v ostrej prevádzke.

Som presvedčený, že navrhnutý a čiastočne už aj implementovaný systém bude významnou mierou prispievať k zjednodušeniu a zefektívneniu procesu nasádzania na našej fakulte. Navyše rozvrhový portál umožní pohodlný prístup študentov aj učiteľov k hotovým individualizovaným rozvrhom bez potreby prácneho skladania z viacerých rozvrhov.

## Prílohy

### *Licencie*

PostgreSQL	BSD License	<a href="http://www.postgresql.org/about/licence">http://www.postgresql.org/about/licence</a>
PHP	The PHP License 3.01	<a href="http://php.net/license/3_01.txt">http://php.net/license/3_01.txt</a>
Apache httpd	Apache License 2.0	<a href="http://www.apache.org/licenses/LICENSE-2.0.txt">http://www.apache.org/licenses/LICENSE-2.0.txt</a>
Zend Framework	Zend Framework License 1.0	<a href="http://framework.zend.com/license">http://framework.zend.com/license</a>
Creole	GNU Lesser General Public License 2.1	<a href="http://www.gnu.org/licenses/lgpl.html">http://www.gnu.org/licenses/lgpl.html</a>
Prototype	MIT License	<a href="http://prototypejs.org/license">http://prototypejs.org/license</a>
Scriptaculous	MIT License	<a href="http://wiki.script.aculo.us/scriptaculous/show/License">http://wiki.script.aculo.us/scriptaculous/show/License</a>
Nifty Corners Cube	GNU General Public License	<a href="http://www.gnu.org/copyleft/gpl.html">http://www.gnu.org/copyleft/gpl.html</a>
Silk Icons	Creative Commons Attribution 2.5 License	<a href="http://creativecommons.org/licenses/by/2.5/">http://creativecommons.org/licenses/by/2.5/</a>

*Tabuľka 2: Licencie použitých softvérových produktov*

## Zoznam obrázkov

Obrázok 1: Entitno-relačný databázový model.....	16
Obrázok 2: Model-View-Controller pattern.....	33
Obrázok 3: Database Abstraction Layers.....	37
Obrázok 4: NDOTable – SQL Abstraction.....	38
Obrázok 5: NDORecord - Active Record Pattern.....	40
Obrázok 6: NDORelation - Relation Abstraction.....	41
Obrázok 7: Štruktúra adresárov.....	44
Obrázok 8: Export zo systému Študent.....	46
Obrázok 9: Mapovanie importovaných dát na vlastný model.....	50
Obrázok 10: Import (upload súborov).....	51
Obrázok 11: Rozhodovací diagram.....	52
Obrázok 12: Závislosti pri importe.....	53
Obrázok 13: Rozvrhový portál.....	55
Obrázok 14: Registrácia pomocou registračného kódu.....	56
Obrázok 15: Registrácia pomocou ISIC karty.....	56
Obrázok 16: Prihlásenie.....	57
Obrázok 17: Nepovolený prístup.....	58
Obrázok 18: Datagrid (zoznam miestností).....	59
Obrázok 19: Úprava záznamu.....	60
Obrázok 20: Detaily predmetu.....	61
Obrázok 21: Ukážka rozvrhu študenta.....	62
Obrázok 22: Systém na podporu rozvrhovania.....	63
Obrázok 23: Nasádzanie (výber predmetu a jeho rozsahu).....	65
Obrázok 24: Nasádzanie (výber učiteľa a miestnosti).....	65
Obrázok 25: Úspešne nasadený predmet.....	66

## Zoznam tabuliek

Tabuľka 1: ACL.....	28
Tabuľka 2: Licencie použitých softvérových produktov.....	68

## Zoznam použitej literatúry

- [1] BRÁZDA, J. *PHP 5 – začíname programovať*. Grada Publishing, Praha 2005. ISBN 80-247-1146-X
- [2] WATSON, R. T. *Data Management – Databases and Organizations*. John Wiley & Sons, Inc., 2002. ISBN 0-471-41845-5
- [3] AMBLER, S. W.; SADALAGE, P. J. *Refactoring Databases - Evolutionary Database Design*. Addison Wesley, 2006. ISBN 0-321-29353-3
- [4] SCHMULLER, J. *Myslíme v jazyku UML*. Grada Publishing, 2001. ISBN 80-247-0029-8
- [5] STEVENS, P.; POOLEY, R. *Using UML – Software Engineering with Objects and Components*. Addison-Wesley, 2000. ISBN 0-201-64860-1
- [6] RUŽAROVSKÝ, J. *Tvorba rozvrhov pomocou multiagentových systémov*. Diplomová práca MFF UK, Bratislava 1998
- [7] KUMAR, V. *Algorithms for Constraint-Satisfaction Problems: A Survey*. AI Magazine, 1992
- [8] TERKANIČ, J. *Zdieľanie informácií o osobách v informačnom systéme Univerzity Komenského*. Diplomová práca FMFI UK, Bratislava 2006
- [9] PÁLOS, G. *Komunikácia aplikácií v informačnom systéme Univerzity Komenského*. Diplomová práca FMFI UK, Bratislava 2006
- [10] <http://www.wikipedia.org>
- [11] <http://www.php.net>
- [12] <http://framework.zend.com>