



SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
Fakulta informatiky a informačných technológií

---

# Bug Hunters

## System pre podporu tvorby rozvrhov

(Tvorba softvérového systému v tíme)

---

Autori (tím 19):	Vladimír Krajčovič, Dušan Palo, Peter Líška, Matej Krchniak, Samuel Števaňák, Ondrej Buch, Peter Mikuška
Študijný odbor:	Softvérové inžinierstvo, Informačné systémy
Ročník:	1. ing.
Semester:	1.
Akademický rok:	2009/2010
Kontakt:	<a href="mailto:tp0910_tim19@googlegroups.com">tp0910_tim19@googlegroups.com</a>

# Obsah

---

<b>1</b>	<b>ZADANIE</b> .....	<b>1</b>
<b>2</b>	<b>ÚVOD</b> .....	<b>2</b>
2.1	ÚČEL A ROZSAH DOKUMENTU .....	2
2.2	SKRATKY A POUŽITÁ NOTÁCIA .....	2
2.3	POUŽITÁ LITERATÚRA .....	3
<b>3</b>	<b>ANALÝZA PROBLÉMU TVORBY ROZVRHOV</b> .....	<b>4</b>
3.1	VSTUPNÉ ÚDAJE PRE PROCES TVORBY ROZVRHU.....	4
3.2	ZBER ÚDAJOV PRE SEMESTRÁLNY ROZVRH.....	5
3.3	TVORBA SEMESTRÁLNYCH ROZVRHOV .....	8
3.4	PREPOJENIE S AIS .....	8
3.5	HRÁČI V SYSTÉME .....	9
<b>4</b>	<b>ŠPECIFIKÁCIA POŽIADAVIEK</b> .....	<b>10</b>
4.1	NEFUNKCIONÁLNE POŽIADAVKY .....	10
4.2	FUNKCIONÁLNE POŽIADAVKY .....	10
<b>5</b>	<b>ANALÝZA EXISTUJÚCEHO RIEŠENIA</b> .....	<b>12</b>
5.1	KONTROLA POŽIADAVIEK NA SYSTÉM .....	12
5.2	ARCHITEKTÚRA EXISTUJÚCEHO SYSTÉMU .....	13
5.3	ÚPRAVY FYZICKÉHO DÁTOVÉHO MODELU .....	14
5.4	EVIDENCIA A OPRAVA CHÝB EXISTUJÚCEHO SYSTÉMU .....	15
<b>6</b>	<b>PROTOTYP SYSTÉMU PRE PODPORU TVORBY ROZVRHOV</b> .....	<b>19</b>
<b>7</b>	<b>NÁVRH ROZŠÍRENÍ A ÚPRAV EXISTUJÚCEHO SYSTÉMU</b> .....	<b>20</b>
7.1	PRIPOMIENKOVANIE TESTOVACEJ VERZIE POUŽÍVATEĽMI .....	20
7.2	NÁVRH ĎALŠEJ PRÁCE NA PROJEKTE .....	20
<b>8</b>	<b>PRÍLOHY</b> .....	<b>22</b>
	<b>PRÍLOHA A</b> <b>DIAGRAM TRIED SYSTÉMU VYTVORENÉHO V ŠKOLSKOM ROKU 2008/2009</b>	
	<b>PRÍLOHA B</b> <b>MODEL ÚDAJOV SYSTÉMU ZO ŠKOLSKÉHO ROKU 2008/2009</b>	
	<b>PRÍLOHA C</b> <b>MODEL ÚDAJOV SYSTÉMU PO ÚPRAVÁCH V ŠKOLSKOM ROKU 2009/2010</b>	
	<b>PRÍLOHA D</b> <b>POUŽÍVATEĽSKÁ PRÍRUČKA</b>	
	<b>PRÍLOHA E</b> <b>INŠTALAČNÁ PRÍRUČKA</b>	
	<b>PRÍLOHA F</b> <b>OBSAH ELEKTRONICKÉHO MÉDIA</b>	

# 1 Zadanie

---

Napriek nasadeniu systému AIS, zostáva stále otvorená problematika vytvárania semestrálnych a skúškových rozvrhov. V súčasnosti je rozpracovaný špecializovaný systém pre tvorbu rozvrhov, ktorý už teraz umožňuje:

- evidenciu údajov potrebných pre vytvorenie rozvrhov,
- samotné ("ručné") vytvorenie/importovanie/exportovanie rozvrhu oprávneným používateľom,
- formuláciu požiadaviek na rozvrhy od pedagógov, ako aj ich zverejňovanie pre rôzne komunity používateľov (so základnou podporou riešenia "konfliktných" požiadaviek),
- sprístupnenie rozvrhu pre ostatných používateľov cez internet.

Vzhľadom na doterajší dlhodobější vývoj tohto systému a zmeny v prostredí fakulty, treba kriticky a tvorivo prehodnotiť všetky používateľské požiadavky, navrhnúť nutné zmeny systému a následne ich implementovať. Zmeny by mali odzrkadľovať najmä:

- širšie údajové prepojenie na akademický informačný systém FIIT a STU - AIS,
- komplexné bezpečnostné podmienky prevádzkovania celého systému na FIIT (aj za FEI - pre zdieľané prostriedky) a kompletizáciu príslušnej dokumentácie,
- zdokonalenie správy používateľov,
- experimentálna prevádzka pri príprave rozvrhu na LS 2009/2010,
- vyhodnotenie experimentálnej prevádzky,
- rozšírenie podsystemu formulovania meniacich sa požiadaviek na rozvrhy od pedagógov a študentov, ako aj ich riadeného sprístupňovania pre rôzne komunity používateľov (detailný ako aj prehľadový formát),
- kvantitatívnu analýzu požiadaviek z hľadiska kapacitných možností FIIT/FEI,
- efektívnejšiu podporu riešenia konfliktov požiadaviek,
- analýzu momentálneho stavu rozvrhu s identifikovaním vznikajúcich "konfliktov" a vyvolanie "riadeného riešenia konfliktu".

V zimnom semestri je hlavnou úlohou analyzovať momentálny stav systému, pričom je potrebné sa sústrediť na funkčnosť, korekciu existujúcich chýb a doplnenie nevyhnutne potrebných požiadaviek na systém. Výsledkom práce v zimnom semestri má byť spustenie experimentálnej prevádzky systému. Táto testovacia verzia by už mala slúžiť na pripomienkovanie rozvrhov v letnom semestri a taktiež na získanie spätnej väzby, aby bolo možné identifikovať a následne opraviť všetky nedostatky v letnom semestri. Vďaka testovacej verzii sa zistí čo používateľom prekáža, prípadne čo by očakávali ako rozšírenie.

## 2 Úvod

---

Plánovanie a organizovanie času je v dnešnom zrýchlenom svete veľmi dôležité. V niektorých prípadoch dokonca nevyhnutné. Takýmto prípadom je aj vyučovanie a študovanie v prostredí na vysokej škole. Väčšinou sú v reálnom prostredí kladené najrôznejšie požiadavky na rozvrhy a plány. Od priestorových obmedzení, časových obmedzení až po rôzne požiadavky na vybavenie a pomôcky potrebné pri vyučovaní. Takisto treba naplno využívať potenciál prístupných zdrojov (miestností, prístrojov...).

Tvorba rozvrhu je proces, pri ktorom treba zorganizovať čas pomerne veľkého množstva ľudí – študentov, pedagógov, cvičiacich... Zaužívané postupy papierovej evidencie požiadaviek a podobne sa stávajú málo flexibilné a málo efektívne, najmä pri tak veľkom množstve študentov. Je potrebné zefektívniť proces tvorby rozvrhov od zberu požiadaviek, cez samotnú tvorbu rozvrhu až po overenie rozvrhu. Pre tento účel bol navrhnutý a implementovaný elektronický systém na podporu tvorby rozvrhov, ktorý sa primárne zameriava na zber požiadaviek na rozvrh. Na tomto systéme pracovali predchádzajúce tímy a podarilo sa im tento projekt dostať do fázy, kedy môžeme hlavnú vývojovú fázu považovať za ukončenú. Hlavnou úlohou nášho tímu je preto tento systém otestovať a vykonať ešte potrebné zmeny na to, aby bol tento systém prevádzkyschopný.

### 2.1 Účel a rozsah dokumentu

Tento dokument obsahuje analýzu procesu tvorby rozvrhov na vysokej škole a prehľad základných požiadaviek na systém pre podporu tvorby rozvrhov. Ďalej sa v dokumente nachádza analýza prevzatého systému od minuloročného tímu. Do tejto analýzy spadá zhodnotenie stavu systému (známe a novo objavené chyby v systéme), prehľad dátového modelu a architektúry celého systému a model implementovaných tried.

Tiež sú spomenuté zmeny, ktoré sme vykonali v systéme od jeho prevzatia (úprava dátového modelu, oprava najpodstatnejších chýb a pod.). V poslednej časti sa zaoberáme návrhom procesov a činností súvisiacich s nasadením a udržiavaním systému v prevádzke.

### 2.2 Skratky a použitá notácia

V Tab. 1 je uvedený zoznam skratiek používaných v tomto dokumente a ich význam.

Tab. 1 Použité skratky a ich význam

Skratka	Význam
AIS	Akademický informačný systém
LDAP	Lightweight Directory Access Protocol
APE	Administrátor prípravnej etapy

Skratka	Význam
STU	Slovenská technická univerzita
FIIT	Fakulta informatiky a informačných technológií
FEI	Fakulta elektroniky a informatiky
HTTP	Hypertext Transfer Protocol
PHP	Hypertext Preprocessor
HTML	Hypertext markup language
SQL	Standard Query Language
MVC	Model View Controller
CSS	Cascading Style Sheets

### 2.2.1 Notácia v modeloch

V diagramoch tried je použitá notácia UML 2.0.

Tab. 2 Notácia použitá v diagramoch tried

Diagram	Význam
<pre> classDiagram     class AutoLoadable     class Model     Model -- &gt; AutoLoadable </pre>	Dedenie – trieda Model je rozšírením triedy AutoLoadable
<pre> classDiagram     class RequestHandler     class Session     RequestHandler --&gt; Session : - session </pre>	Asociácia – trieda RequestHandler môže volať metódy triedy Session (Session alebo referencia na Session je súčasťou triedy RequestHandler)

### 2.3 Použitá literatúra

- [1] Tím BackSpace, Systém pre podporu tvorby rozvrhov, 2008/2009
- [2] Tím Fénix, Systém pre podporu tvorby rozvrhov, 2007/2008

## 3 Analýza problému tvorby rozvrhov

---

Proces tvorby rozvrhov môžeme rozdeliť do niekoľkých krokov. V prvom kroku je potrebné zozbierať všetky potrebné údaje pre vytvorenie rozvrhu. Následne je potrebné analyzovať jednotlivé zozbierané údaje a postupne ich zapracovávať do rozvrhu. Po zapracovaní všetkých údajov do rozvrhu vzniká základný prototyp rozvrhu. Konečným krokom tvorby je modifikovanie zostaveného rozvrhu do finálnej verzie a následne zverejnenie vytvoreného rozvrhu. Tu je potrebné analyzovať možnosti zverejňovania rozvrhov. Po nasadení rozvrhu do praxe je potrebné riešiť korekcie v rozvrhu, ktoré často krát vznikajú počas semestra. Po zapracovaní korekcií do semestrálneho rozvrhu je potrebné novú verziu rozvrhu znovu zverejniť.

### 3.1 Vstupné údaje pre proces tvorby rozvrhu

Pred odštartovaním procesu tvorby rozvrhu je nutné poznať všetky údaje, ktoré sa pri tvorbe využívajú. Údaje môžeme rozdeliť do týchto množín: množina miestností a ich kapacitné ohraničenie, vybavenie učebnými pomôckami (výpočtová a komunikačná technika, videoprojektory a pomôcky v laboratóriách, ...), množina otváraných predmetov a ich charakteristiky (maximálny počet študentov, počet dostupných pedagógov spolu s ich časovým harmonogramom, potrebné vybavenie, rozsah prednášok a rozsah cvičení, ...), množina študentov zapísaných na jednotlivé predmety a časový harmonogram výučby (štátne sviatky, dekan'ské voľno, študentská vedecká konferencia,...).

#### 3.1.1 Množina predmetov a ich charakteristiky

Prvú skupinu tvoria všetky otvárané predmety a ich charakteristiky. Tieto informácie o predmete poskytujú garanti jednotlivých predmetov alebo nimi poverené osoby. Základné charakteristiky predmetov sú meno predmetu a jeho skratka, nominálny ročník, odbor, kapacitné ohraničenie predmetu (maximálny počet študentov na predmete, minimálny počet študentov potrebných pre otvorenie predmetu ...), vybavenie učební, prípadne požiadavka na špecializované učebne (laboratóriá, učebne s výpočtovou technikou, telocvičňa ,...)). Okrem základných charakteristík predmetu možno zadefinovať rôzne prípady špeciálnych požiadaviek, ako napríklad:

- V niektorých týždňoch sa budú konať spoločné cvičenia vo veľkých miestnostiach
- Prednášky budú rozdelené do dvoch dní (napríklad 2 + 2 namiesto 4 súvislých hodín prednášok)
- Prednášky budú odprednášané za menší počet týždňov – v týchto týždňoch sa bude prednášať viac hodín, ako by sa malo, aby zostal celkový počet hodín zachovaný

- Rozdelenie prednášky na jednotlivé týždne a pridelenie prednášajúcich na každý týždeň v prípade, že prednášať chodia napríklad aj externí prednášatelia.
- Cvičenia sa nemusia konať každý týždeň, ale môžu byť napríklad každý druhý týždeň s dvakrát väčším počtom hodín – celkový počet hodín cvičení za semester musí zostať zachovaný
- Písomka (priebežný test) - je potrebné rezervovať miestnosť a dozor.

### **3.1.2 Množina pedagógov**

Túto skupinu tvoria všetci pedagógovia, ktorí sa podieľajú na vyučovaní daných predmetov (môže ísť o garanta, prednášajúceho, vedúceho cvičení alebo cvičiaceho pedagóga). Ich počet je známy po registrácii predmetov študentmi, kedy jednotliví garanti vyčlenia pedagógov na výučbu daného predmetu. K charakteristikám tejto množiny pedagógov patrí akceptovanie časového rozvrhu pedagóga (napríklad práca mimo školy) a tiež sledovanie vyťaženia (aby nemal veľký počet vyučovacích hodín do týždňa). Pri tejto skupine musíme brať do úvahy aj absencie jednotlivých pedagógov. Táto situácia sa nedá predvídať ale treba ju počas semestra korektne zapracovať do rozvrhu a tak ju vyriešiť.

### **3.1.3 Množina študentov**

Po registrácii predmetov jednotlivými študentmi získame prehľad o počte študentov na daný predmet. Musíme tiež brať do úvahy aj časové hľadisko študentov (aby nemal vyučujúce hodiny v preťažujúcom objeme resp. neprimerane „rozsekané“ od rána do večera), ďalej tiež špeciálne požiadavky študentov, ktoré väčšinou zadávajú zástupcovia študentov na základe sťažností alebo pripomienok.

### **3.1.4 Časový plán výučby**

Časový harmonogram štúdia zostavuje študijné oddelenie fakulty. Je potrebné aby sa do rozvrhu zapracovali aj rôzne udalosti, ktoré môžu vzniknúť počas semestra. Tieto udalosti vznikajú napríklad pri dekanskom voľne alebo študentskej vedeckej konferencii, kde je potrebná ich korekcia buď v procese tvorby rozvrhu alebo počas semestra.

## **3.2 Zber údajov pre semestrálny rozvrh**

Pri začatí procesu tvorby rozvrhu je potrebné aby rozvrhár mal prístup ku všetkým množinám údajov, z ktorých vychádza pri zostavovaní rozvrhu. Naša fakulta používa AIS, ktorý okrem iného umožňuje aj predregistráciu predmetov jednotlivými študentmi, ktorí si chcú v nasledujúcom akademickom roku zapísať tieto predmety. Spolu s týmito elektronickými predregistráciami je potrebné k povinne voliteľným predmetom zadať aj ich alternatívy (napr. v prípade prekročenia počtu študentov na daný predmet). Na základe predregistrácie je možné zistiť približný záujem o jednotlivé predmety, čiže je možné určiť, ktoré predmety budú nasledujúci akademický rok otvorené. Predregistrácia

slúži iba informatívne, nakoľko nie každý študent, ktorý urobil predregistráciu môže pokračovať v štúdiu. Predregistrácia sa v danom systéme nerieši nakoľko je v určitom rozsahu možná v AISe.

Ďalším krokom po predregistrácii je zápis študentov do jednotlivých ročníkov. Až vtedy je možné určiť presné počty študentov na predmety, nakoľko je zápis záväzný.

Zber údajov nezávisí len od študentov ale do zberu údajov zasahujú aj ďalšie typy používateľov:

- garant
- prednášajúci
- vedúci cvičení
- cvičiaci
- pracovníci ŠO

Jeden zamestnanec môže patriť do rôznych skupín používateľov zároveň. Napríklad môže byť garantom predmetu, prednášajúcim a cvičiacim pedagógom v rámci toho istého predmetu. Môže vzniknúť situácia kde je na daný predmet prihlásený veľký počet študentov, kedy sa daný predmet spravidla vyučuje vo viacerých inštanciách, kde jednotlivé prednášky prednášajú iní ľudia.

Základné požiadavky na jednotlivé predmety zadávajú garanti alebo nimi poverené osoby (prednášajúci, cvičiaci,...). Čiže ide o definovanie hlavných vlastností a parametrov predmetu ako sú:

- číslo a názov predmetu
- skratka predmetu
- semester, v ktorom sa predmet vyučuje (LS alebo ZS)
- študijný program
- spôsob ukončenia predmetu (zs, z, kz)
- počet hodín prednášok - ak má semester 12 týždňov a každý týždeň by mala byť trojhodinová prednáška, tak počet hodín prednášok je 36.
- počet hodín cvičení - rovnaký výpočet ako pri počte hodín prednášok.
- počet prednášok - v prípade ak chce garant viac prednášok ako je štandardných 12, potom môže napríklad 12 trojhodinových prednášok nahradiť 18-timi dvojhodinovými prednáškami.



- počet cvičení - ak by mala byť každý týždeň jedna hodina cvičenia, tak garant môže zadať 6 cvičení, ktoré sa môžu vyučovať napr. každý druhý týždeň alebo druhú polovicu semestra.
- počet študentov predmetu - najskôr dajú z predregistrácie, potom podľa zápisu počtu ľudí, ktorý si predmet zapísali.
- kapacita cvičenia - maximálny možný počet študentov, ktorí sa môžu zúčastniť jedného termínu cvičenia. V tomto prípade sa jedná hlavne o cvičenia, ktoré sa vyučujú v počítačových učebniach alebo laboratóriách, kde je obmedzený počet miest.

Po definovaní základných vlastností predmetu je možné zdefinovať jednotlivé požiadavky na:

- miestnosti konania prednášky

prednášajúci pedagóg zadá tento typ požiadavky kde určí počet študentov zapísaných na daný predmet, vyberie vhodnú miestnosť s dostatočnou kapacitou

- miestnosti konania cvičení

cvičiaci predmetu má možnosť zdefinovať počet študentov v jednej skupine a následne vybrať kapacitne vhodnú miestnosť a špecifikovať typ miestnosti (napr. softvérové štúdio, laboratórium, PC učebňa,...)

- vybavenie

prednášajúci pedagóg má možnosť si vybrať vybavenie na daný predmet (dataprojektor, notebook, stoličky navyše)

- spôsob výučby

prednášajúci a cvičiaci pedagógovia môžu definovať časový harmonogram výučby predmetu v danom semestri, to znamená, že predmet môže mať pravidelný aj nepravidelný harmonogram výučby. Pravidelný spočíva v tom, že prednášky a cvičenia sú v priebehu semestra v rovnakom čase a v rovnakej dĺžke. Pri nepravidelnom harmonograme ako napríklad keď potrebuje prednášajúci odprednášať učivo čím skôr, tak sa prednášky konajú na začiatku semestra intenzívnejšie ako ku koncu semestra (napr. pri externom prednášajúcom pedagógovi zo vzdialenejších miest). Inou možnosťou je konanie prednášok a cvičení v každom druhom prípadne treťom týždni semestra.

### 3.3 Tvorba semestrálnych rozvrhov

Po zozbieraní všetkých potrebných množín údajov sa môže pristúpiť k samotnému procesu tvorby rozvrhu.

#### 3.3.1 Postup tvorby

Najprv sa do rozvrhu pridávajú prednášky, tak aby spĺňali požiadavky na daný predmet (kapacita, vybavenie miestností,...) a tiež aby neboli v kolízií s inými predmetmi ako napríklad v jednej miestnosti dve úplne rozdielne prednášky v ten istý čas. Následne sa stanoví rozdelenie študentov do krúžkov. V prípade prvákov sa študenti do krúžkov zaraďujú direktívne algoritmicke, najčastejšie abecedne v definovaných n-členných „krúžkoch“, v prípade vyšších ročníkov sa jednotliví študenti sami registrujú na dané termíny cvičení. Následne sa jednotlivým cvičeniam pridelia pedagógovia tak, aby sme čo najlepšie spĺňali požiadavky na ich časový plán. Jednotliví pedagógovia sú v zozname s pridelenou prioritou, ktorá nám určuje ktorého pedagóga musíme uprednostniť pri tvorbe rozvrhu ako napríklad externých študentov ich časový plán kedy môžu sa zúčastňovať na výučbe.

Tiež musíme riešiť situáciu keď môže nastať kolízia jednotlivých predmetov pre daného študenta alebo pedagóga, kedy nemôžu mať súčasne viac vyučovacích hodín. Výnimku tvoria predmety, ktoré nemajú spoločných študentov alebo pedagógov kedy nenastáva kolízia v rozvrhu. Tiež pre nižšie ročníky je dobre rozdeliť jednotlivé vyučovacie hodiny rovnomerne na celý týždeň. V prípade vyšších ročníkov je vhodné rozložiť ich do niekoľkých dní (dvoch až troch). Je vhodné zabezpečiť optimálnosť rozvrhu pre každého, čo nemožno vždy dosiahnuť, preto treba aspoň minimalizovať možné odchýlky od optimálneho rozvrhu (minimalizovať počet voľných hodín v rozvrhu).

Po ukončení iterácie tvorby nám vznikne prototyp rozvrhu. Tvorca rozvrhu musí neustále kontrolovať, či daná iterácia rozvrhu nie je v rozpore s požiadavkami. V prípade, že je v rozpore, musí sa rozhodnúť ako rozpor vyriešiť. Pri riešení takýchto situácií musí uprednostňovať požiadavky s vyššou prioritou (napr. neprítomnosť pedagóga). Ak sa nedá situácia vyriešiť, je potrebné zmeniť rozvrh alebo jeho časť.

Opakovaním procesu tvorby predchádzajúcej iterácie vzniká ucelenejší pohľad na rozvrh kde sa zapracúvajú všetky požiadavky a výsledný rozvrh sa viac a viac podobá na finálny časový plán výučby.

### 3.4 Prepojenie s AIS

Prepojením s AIS sa zaoberali už tri predošlé tímy. Všetky dospeli k záveru, že keďže sa jedná o komerčný produkt, nie je možný prístup k zdrojovému kódu a databáze. Preto nie je možné priamo prepojiť systémy navzájom. Autori kvôli bezpečnosti systému neumožňujú prístup priamo k databáze. AIS má možnosť exportu niektorých množín údajov z databáz ako napríklad množinu predmetov, pedagógov, študentov,... Tiež umožňuje import údajov pre zverejňovanie už vytvorených rozvrhov. Prepojenie AIS a systému je nepriamo zabezpečené cez vopred dohodnutý súborový formát. Kde minuloročné tímy narazili na nedostatočnú spoluprácu s autormi AIS z ich strany.

Momentálne sú k dispozícii určité exporty z databázy AIS vo forme excelovských súborov.

Ak to zhrnieme, tak niektoré časti AIS sa stále prispôbujú potrebám STU. Požiadaviek na tvorcov AIS je však veľa, a tak sa zatiaľ k zmenám v systéme na tvorbu rozvrhov nijak nepokročilo. Komunikácia s tvorcami AIS je veľmi zdĺhavá a komplikovaná, a preto sa aj jednoduché problémy riešia veľmi dlho. Preto je potrebné riešiť samostatný systém na tvorbu rozvrhov s vlastnou databázou zamestnancov, miestností, vybavenia...

### 3.5 Hráči v systéme

Do systému bude mať prístup niekoľko rôznych typov používateľov. Jednotliví používatelia budú mať sprístupnenú len určitú funkčnosť systému. Je to dôležité z pohľadu zadávania požiadaviek a tvorby rozvrhu.

Používatelia systému:

**Rozvrhár** – používateľ (používatelia) systému, ktorý sa priamo podieľa na vytváraní rozvrhu na základe zozbieraných údajov (zoznam predmetov, miestností, požiadaviek, ...). Tomuto typu používateľa systém sprístupňuje rôznu funkčnosť, ktorá umožňuje zisťovať kolízie. Ďalej systém umožňuje prehľadné zobrazenie všetkých druhov požiadaviek. Ďalšou funkčnosťou systému je vytváranie viacerých verzií rozvrhu.

**Administrátor** – spravuje systém, databázu a používateľov.

**Administrátor prípravnej etapy** – tento typ používateľov má za úlohu napĺňanie a aktualizáciu systému dôležitými údajmi. Tieto údaje tvoria: množina miestností, predmetov, pedagógov, časový harmonogram. Medzi týchto používateľov sa začleňujú aj pracovníci ŠO, ktorí môžu napĺňať daný systém údajmi.

**Garant** – používateľ zadáva do systému požiadavky na predmet (čas prednášok alebo cvičení, priestory a vybavenie). Garant si môže prezerať vytvorené rozvrhy.

**Pedagóg** – používateľ systému, môže vkladať do systému požiadavky na prednášku a na svoj osobný časový plán. Ďalej má možnosť prezerať si rozvrh.

**Študent** – každý študent si môže pozrieť vytvorený rozvrh. Študent nijako nezasahuje do tvorby rozvrhu, ani do systému nevkladá žiadne údaje.

Je potrebné aby používatelia systému, ktorí majú prístup ku funkčnosť systému na vkladanie a modifikovanie údajov, mali do systému prístup pomocou autentifikácie. V prípade študentov nie je potrebné prihlasovanie sa do systému. Títo budú mať len možnosť prezerať si vytvorené rozvrhy.

## 4 Špecifikácia požiadaviek

---

V tejto kapitole sa zaoberáme špecifikáciou požiadaviek na systém. Opíšeme špecifikáciu funkcionálnych a nefunkcionálnych požiadaviek.

### 4.1 Nefunkcionálne požiadavky

Systém bude stavaný na tieto základné nefunkcionálne požiadavky:

- Počet používateľov bude rádovo v desiatkach (počas vytvárania rozvrhu) až stovkách (prezeranie publikovaného rozvrhu).
- Použijeme aplikačný server Apache s nainštalovaným modulom PHP.
- Použijeme databázový server PostgreSQL.
- Systém bude platformovo nezávislý ale v škole bude prevádzkovaný na operačnom systéme LINUX.

### 4.2 Funkcionálne požiadavky

Funkcionálne požiadavky systému sa budú odvíjať od príslušnej roly používateľa v systéme. Jednotliví používatelia budú mať sprístupnenú len im patriacu funkcionálnu. Ako už bolo opisované vyššie, reálni používatelia nemusia byť rozdelení len do týchto skupín. Môžu patriť zároveň do viacerých z nich, napríklad: používateľ je zároveň garantom predmetu a prednášajúcim, takže bude môcť využívať funkcie dostupné oboj skupinám.

#### 4.2.1 Garant Predmetu

- Keďže garant predmetu nemusí byť zároveň prednášajúcim, nemusí určovať miestnosť a ani potrebné vybavenie, pretože to záleží od požiadaviek prednášajúcich a vedúceho cvičení.
- Garant predmetu musí mať oprávnenie na zadanie názvu predmetu, ročníka, študijného programu a časového rozsahu predmetu v počte hodín prednášok a cvičení v jednom týždni
- Musí mať možnosť určiť prednášajúceho (prípadne prednášajúcich) a vedúceho cvičení.
- Musí mať možnosť meniť požiadavky (pokým sa nezačne samotný proces tvorby rozvrhu).

#### 4.2.2 Prednášajúci

- Prednášajúci musí mať možnosť určiť reálny priebeh prednášok, to znamená, že musí mať možnosť zadať týždenný počet hodín prednášok, počet prednášok za týždeň, počet hodín jednej prednášky, počet študentov na prednáške, preferovanú miestnosť a potrebné vybavenie ktoré bude mať k dispozícii počas prednášky.

- Musí mať možnosť určiť tieto údaje pre každý týždeň zvlášť (podľa potreby).
- Musí mať možnosť podať špeciálne požiadavky (neštandardné).
- Musí mať možnosť zadať preferovaný čas výučby pre každý týždeň.
- Musí mať možnosť meniť požiadavky (pokým sa nezačne samotný proces tvorby rozvrhu).

#### **4.2.3 Vedúci cvičení**

- Vedúci cvičení musí mať možnosť určiť reálny priebeh cvičení, to znamená, že musí mať možnosť zadať týždenný počet hodín cvičení, počet cvičení za týždeň, počet hodín jedného cvičenia, počet študentov a preferované miestnosti pre jednotlivé skupiny na cvičeniach.
- Musí mať možnosť určiť tieto údaje pre každý týždeň zvlášť (podľa potreby).
- Musí mať možnosť zadať preferovaný čas výučby pre každý týždeň.
- Musí mať možnosť meniť požiadavky (pokým sa nezačne samotný proces tvorby rozvrhu).

#### **4.2.4 Rozvrhár**

- Musí mať možnosť zobrazenia prehľadu požiadaviek k jednotlivým predmetom.
- Musí mať možnosť prehľadného zobrazenia časových priorít pedagógov ku jednotlivým predmetom.
- Musí mať možnosť prezerať konfliktnú tabuľku. Pod konfliktom sa myslia situácie, keď má predmet rovnakých prednášajúcich alebo rovnakých študentov ako iný predmet.

#### **4.2.5 Administrátor**

- Spravovanie používateľov: pridávanie, modifikovanie a odoberanie používateľov, nastavovanie prístupových oprávnení.

#### **4.2.6 Autentifikácia**

Používatelia sa budú môcť autentifikovať do systému pomocou prihlasovacích údajov zhodnými s prihlasovacími údajmi do AIS. Toto je zabezpečené školským LDAP serverom, kde sú uložené prihlasovacie údaje všetkých študentov a zamestnancov.

## 5 Analýza existujúceho riešenia

---

V tejto kapitole je opísaná analýza systému, ktorý bol vyvíjaný po minulé roky. Konkrétne sa tu nachádza zhrnutie, do akej miery boli splnené požiadavky na systém, ktoré sú dôležité na prototypové nasadenie a otestovanie systému. Ďalej sa tu nachádza diagram tried, ktorý bol vypracovaný kvôli lepšej prehľadnosti prepojenia existujúceho systému. Bolo potrebné analyzovať existujúce chyby v systéme, aby bolo možné podstatné nedostatky odstrániť a následne spustiť testovaciu verziu systému.

### 5.1 Kontrola požiadaviek na systém

Hlavná požiadavka na systém je, aby mohol byť nasadený na skúšobnú prevádzku v letnom semestri. Systém, ktorý sme prebrali po tíme BackSpace bol navrhnutý vo veľkej miere správne, ale museli sme spraviť revíziu a opravu dátového modelu. Síce je systém navrhnutý správne, ale obsahuje ešte stále veľmi veľa chýb. Od minulého roka ostalo neopravených a odhalených 17 chýb. Takže sme zvolili postup neustáleho testovania systému a odhaľovania nových chýb. Naša pozornosť sa zameriava na odstránenie tých chýb, kvôli ktorým nefunguje správne zber požiadaviek, nakoľko je toto primárny cieľ. Počas testovania sme odhalili už okolo 40 chýb, z ktorých minimálne  $\frac{3}{4}$  je potrebné odstrániť, aby sa mohol systém nasadiť do testovacej prevádzky.

Jeden z najväčších nedostatkov systému bola chýbajúca validácia vstupných údajov od užívateľa. Táto validácia chýbala skoro v každej interakcii systému s užívateľom. A ak bola implementovaná, bola implementovaná neprehľadne a nedala sa použiť pre iný model. Nám sa podarilo tento podstatný nedostatok vyriešiť generickým riešením. Samotná validácia prebieha v metóde modelu „bind“, v ktorej sa kontroluje minimálna, maximálna dĺžka povinnosť poľa a prebiehajú tu aj všetky ostatné validácie. Dôležitým faktorom je, že validácia eliminuje aj prípad, že by sa nejaký užívateľ snažil o tzv. Code Injection. Týmto sa stáva systém podstatne bezpečnejší.

Ďalšia požiadavka na systém bola, aby sa ukladali časové pečiatky ak používateľ raz odošle svoje požiadavky a tým bol zabezpečený pre rozvrhára jasný dôkaz o tom odkiaľ tieto požiadavky získal a že si ich sám nevymyslel. Týmto bude umožnené aj to, že používateľ bude môcť povedať, že chce použiť rovnaké požiadavky na predmet také isté ako napr. minulý rok. Táto požiadavka sa musí riešiť na väčšine objektov takže toto budeme riešiť na dátovej úrovni. Budeme musieť vytvoriť tabuľku, ktorá bude reprezentovať verzie alebo časové pečiatky.

Pripomienkovanie je ďalšia požiadavka na systém. Ide o to, že ak sa používateľovi nebude niečo na systéme páčiť, aby mal možnosť vyjadriť svoju nespokojnosť a oznámiť svoju pripomienku vývojárom systému. Systém pre podporu rozvrhov sa už vyvíja viac ako 4 roky. To znamená, že vývojári sú každý rok iní. Od toho sa odvíja aj náš návrh na

riešenie tejto požiadavky. Vytvoríme štandardný formulár, do ktorého bude môcť používateľ napísať svoje pripomienky a celý tento záznam sa uloží do databázy. Vzhľadom na neustálu zmenu vývojárov bude vhodné ak si bude tieto pripomienky môcť prezerať administrátor systému.

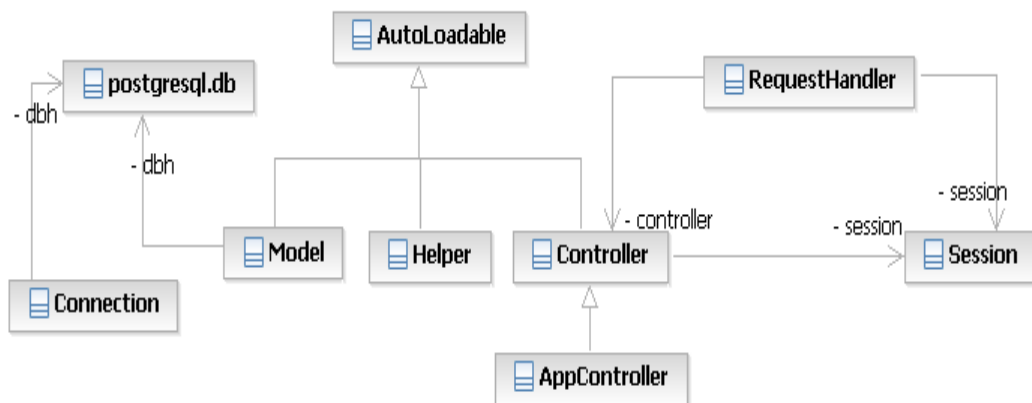
Implementácia on-line pomocníka je jedna z najnovších požiadaviek. Ide hlavne o to, aby aj počítačovo menej zdatní používatelia sa dokázali v systéme rýchlo zorientovať. Zatiaľ by sme chceli riešiť on-line pomocníka štýlom, že bude na každej stránke, na ktorej bude tento pomocník dostupný odkaz na dokument typu PDF, ktorý bude popisovať funkčnosť aktuálnej stránky. V budúcnosti by sme chceli implementovať on-line pomocníka priamo na stránke a nepoužívať žiadne externé dokumenty.

## 5.2 Architektúra existujúceho systému

Prevzatý systém je navrhnutý podľa architektúry klient-server. Klientskú časť systému tvorí webový prehliadač používateľa. Serverovú časť tvorí databázový systém PostgreSQL s databázovým serverom, HTTP server s PHP prostredím a samotný aplikácia, ktorá je implementovaná v jazyku PHP. Databázový server a HTTP server môžu byť nainštalované na jednom fyzickom počítači (prevádzkový server).

### 5.2.1 Model implementovaných tried

Triedy sú implementované a organizované podľa vzoru „model-view-controller“. Jadro systému tvoria triedy zobrazené na Obr. 1. Na diagrame sú znázornené základné vzťahy medzi týmito triedami.



Obr. 1 Diagram tried jadra systému

Aplikačne špecifické triedy je možné vďaka tejto architektúre pridávať do systému špecializáciou tried Model, View a Controller a ApplicationController. Napríklad špecializáciou triedy „Model“ sú vytvorené jednotlivé triedy reprezentujúce základné entity v systéme. Inštancie týchto tried poskytujú rozhranie na prácu s príslušnými databázovými entitami

(napr. trieda „Equipment“ zdedená od triedy „Model“ predstavuje model vybavenia, ktorých entity sa nachádzajú v databázovej tabuľke „vybavenie“).

### 5.3 Úpravy fyzického dátového modelu

Z dôvodu úpravy chýb bolo potrebné vykonať menšie zásahy do dátového modelu. Jednalo sa hlavne a podrobnejšie nadefinovanie referenčných obmedzení. Pre porovnanie dátový model zo školského roku 2008/2009 sa nachádza v prílohe B a nami vytvorený dátový model je v prílohe C.

#### 5.3.1 Zmena č.1.

Pri oprave chyby č.5 – „Pri pridávaní a úprave používateľa nemá byť medzi skupinami "Všetci"" sme sa rozhodli, že skupina „Všetci“ dostane id 0. Všetky ostatné skupiny budú mať id nezáporné, nenulové a tak môžeme zo zobrazení vylúčiť skupiny, ktorých používateľ je členom implicitne (majú záporné alebo nulové id).

Z dôvodu zmeny id sme upravili nasledovné referenčné obmedzenia:

- v tabuľke **menu**, stĺpcu **group\_id** bolo zmenené obmedzenie na „ON UPDATE CASCADE ON DELETE CASCADE“
- v tabuľke **clenstvo**, stĺpcu **id\_skupina** bolo zmenené obmedzenie na „ON UPDATE CASCADE ON DELETE CASCADE“
- v tabuľke **vyucuje\_predmet**, stĺpcu **id\_pedagog\_typ** bolo zmenené obmedzenie na „ON UPDATE CASCADE ON DELETE CASCADE“

#### 5.3.2 Zmena č.2.

Pri oprave chyby #45 – „Bug so stolickami pri pridávaní požiadaviek na prednasky“ bolo tiež potrebné zmeniť id stoličiek v tabuľke **vybavenie**. Z toho dôvodu boli taktiež doplnené referenčné obmedzenia:

- v tabuľke **poziadavka\_vybavenie**, stĺpcu **id\_vybavenie** bolo zmenené obmedzenie na „ON UPDATE CASCADE ON DELETE CASCADE“
- v tabuľke **vybavenie\_miestnost**, stĺpcu **id\_vybavenie** bolo zmenené obmedzenie na „ON UPDATE CASCADE ON DELETE CASCADE“

#### 5.3.3 Zmena č.3.

Pre prehľadnosť sme tabuľku **poziadavka** doplnili o nový stĺpec **cas\_pridania**, ktorý obsahuje čas pridania požiadavky. Rozvrhár má teraz takto možnosť vidieť nielen obsah požiadaviek, ale aj kto požiadavky zadal a kedy. Zároveň sme zadávanie požiadaviek rozšírili tak, aby sa uchovávali všetky zadané požiadavky. Rozvrhár tak má možnosť prezrieť si históriu všetkých zadaných požiadaviek daného pedagóga na daný predmet.



## 5.4 Evidencia a oprava chýb existujúceho systému

V tejto časti sú opísané chyby systému, ktoré boli nájdené ale nevyriešené v minulom roku, ďalej chyby ktoré sme dodatočne našli my a tiež zoznam chýb, ktoré sa nám už podarilo opraviť. Z časového hľadiska sme nestihali opraviť všetky vyskytujúce sa chyby, takže sme určili ich prioritu a následne opravili najnevyhnutnejšie.

### 5.4.1 Zaznamenané chyby

Niektoré chyby, ktoré nestihli byť opravené, boli nájdené tímom BackSpace v prechádzajúcom období. Ich zoznam je uvedený v Tab. 3 **Chyba! Nenašiel sa žiaden zdroj odkazov..**

Tab. 3 Chyby zaznamenané v školskom roku 2008/2009

Číslo	Opis chyby alebo nedostatku
1	Ošetriť nezadané heslo pri prihlasovaní
2	Chyba vo výbere údajov z DB, našlo sa 0
3	Pri zadávaní predmetu môžem vybrať len jeden študijný program
4	Pri úprave používateľa (administrátor) sa duplikujú jeho práva
5	Pri pridávaní a úprave používateľa nemá byť medzi skupinami "Všetci"
6	Ošetriť zadanie neplatného mena/hesla
7	Pridanie používateľa - môžem viac rázi pridať rovnakú osobu
8	Pridávanie miestností - v prípade nevyplnenia alebo chybného vyplnenia položiek dať o tom informáciu
9	Pridávanie miestností - kontrolovať kladnosť kapacity, lebo ide zadávať aj záporné
10	Pridávanie predmetu - je možné pridať predmet bez zadania mena a kódu
11	Pridávanie vybavenia - možno pridať vybavenie bez zadania typu(mena)
12	Pridávanie vybavenia (asi aj inde) – nekontroluje sa maximálna dĺžka zadávaných parametrov
13	Správa používateľov - ak ma používateľ pridanú skupinu a ta sa odstráni tak sa to neuloží
14	Prezeranie požiadaviek na predmet
15	Pridanie požiadavky na predmet
16	Pridanie používateľa ktorý neexistuje v AISe => úspešné

### 5.4.2 Novoobjavené chyby a nedostatky

Počas analýzy a testovania existujúceho systému sme našli niekoľko doposiaľ neevidovaných chýb a nedostatkov. Keďže sme niektoré chyby opravovali, je možné že niektoré ďalšie vznikli aj našim zásahom. Uvádzame zoznam takýchto chýb v Tab. 4.

Tab. 4 Chyby a nedostatky systému nájdené v zimnom semestri školského roku 2009/2010

Číslo	Opis chyby alebo nedostatku
-------	-----------------------------

Číslo	Opis chyby alebo nedostatku
17	DB ignoruje 'NOT NULL' obmedzenia
18	Editácia používateľa - refactoring
19	Pri neautorizovanom prístupe chyba presmerovanie
20	Nezrovnalosť pri ukladaní požiadaviek pre cvičiaceho
21	Ape - pridanie miestnosti
22	Všetky moduly - kontrola zadaných údajov
23	All - podpora pripomienkovania v systéme
24	Pridanie používateľa - LDAP autocomplete problémy
25	Časové priority - chyba pri ukladaní alebo načítavaní
26	Equipment - code injection
27	Subject - code injection
28	Teacher - editácia požiadaviek
29	Scheduler - prezeranie požiadaviek na prednášky a cvičenia
30	Online help pre celý portál
31	Help - časové priority
32	Help - požiadavky pre cvičiaceho a pedagóga
33	Help - garant predmetu
34	Požiadavky na viac predmetov (nekorektne mazanie)
35	Scheduler - nedajú sa prepínať rozloženia pri požiadavkách na prednášky
36	Scheduler - nekorektne deaktivované položky pri prezeraní požiadaviek na cvičenia
37	Scheduler - zobrazenie časových priorít
38	Home - chybná správa ak nie som prihlásený
39	Menu - refactoring
40	Nekorektný timeout a následný logout ?
41	Prebytočný súbor postgresql.php
42	Duplicitné hodnoty v tabuľke vyucuje_predmet
43	Refactoring spoločného prehľadu požiadaviek u schedulera + pridanie dátumu do požiadavky
44	Pridávanie požiadaviek - označiť všetky týždne naraz
45	Chyba so stoličkami pri pridávaní požiadaviek na prednášky
46	Viaceré pohľady a rozloženia k nim
47	Požiadavky na prednášky
48	Požiadavky na cvičenia - chybné zobrazovanie miestností a iné.
49	Scheduler - prezeranie požiadaviek na prednášky a cvičenia - refactoring
50	Teacher - doplnenie zoznamu predmetov
51	Pract - pridávanie požiadaviek (chyba možnosť pridať komentár)
52	Administrator - pridať preberanie práv
53	Ape - priradenie výučby pedagógom
54	Scheduler - prezeranie požiadaviek (pridať nadpisy)

Číslo	Opis chyby alebo nedostatku
55	Garant - upraviť spôsob zobrazovania pridelovania predmetov
56	Požiadavky na prednasky - chybné ukladanie týždňov v rámci rozložení
57	logout - možno odhlásiť aj nikoho

### 5.4.3 Opravené chyby

Spustenie systému v testovacom režime je hlavná požiadavka v prvej časti tímového projektu a preto sme niektoré chyby alebo nedostatky už opravili. Zoznam opravených chýb a úprav je v Tab. 5 Tab. 5.

Tab. 5 Opravené chyby v zimnom semestri školského roku 2009/2010

Číslo	Opis chyby alebo nedostatku
1	Ošetriť nezadané heslo pri prihlasovaní
4	Pri úprave používateľa (administrátor) sa duplikujú jeho práva
5	Pri pridávaní a úprave používateľa nemá byť medzi skupinami "Všetci"
6	Ošetriť zadanie neplatného mena/hesla
7	Pridanie používateľa - môžem viac razí pridať rovnakú osobu
8	Pridávanie miestností - v prípade nevyplnenia alebo chybného vyplnenia položiek dať o tom informáciu
9	Pridávanie miestností - kontrolovať kladnosť kapacity, lebo ide zadávať aj záporné
10	Pridávanie predmetu - je možné pridať predmet bez zadania mena a kódu
11	Pridávanie vybavenia - možno pridať vybavenie bez zadania typu(mena)
12	Pridávanie vybavenia (asi aj inde) – nekontroluje sa maximálna dĺžka zadávaných parametrov
14	Prezeranie požiadaviek na predmet
15	Pridanie požiadavky na predmet
16	Pridanie používateľa ktorý neexistuje v AISE => úspešné
18	Editácia používateľa - refactoring
21	Ape - pridanie miestnosti
22	Všetky moduly - kontrola zadaných údajov
25	Časové priority - chyba pri ukladaní alebo načítavaní
26	Equipment - code injection
27	Subject - code injection
29	Scheduler - prezeranie požiadaviek na prednášky a cvičenia
36	Scheduler - nekorektne deaktivované položky pri prezeraní požiadaviek na cvičenia
37	Scheduler - zobrazenie časových priorít
39	Menu - refactoring
42	Duplicitné hodnoty v tabuľke vyucuje_predmet

Číslo	Opis chyby alebo nedostatku
43	Refactoring spoločného prehľadu požiadaviek u schedulera + pridanie dátumu do požiadavky
45	Chyba so stoličkami pri pridávaní požiadaviek na prednášky
46	Viacere pohľady a rozloženia k nim
51	Pract - pridávanie požiadaviek (chýba možnosť pridať komentár)
52	Administrátor - pridať preberanie práv
53	Ape - priradenie výučby pedagógom
54	Scheduler - prezeranie požiadaviek (pridať nadpisy)
55	Garant - upraviť spôsob zobrazovania pridelovania predmetov
56	Požiadavky na prednášky - chybné ukladanie týždňov v rámci rozložení
57	logout - možno odhlásiť aj nikoho

## **6 Prototyp systému pre podporu tvorby rozvrhov**

Cieľom vytvárania prototypu v zimnom semestri školského roku 2009/2010 bola testovacia prevádzka systému, ktorá mala pomôcť k získaniu spätnej väzby od používateľov vo forme pripomienok. Na základe tejto spätnej väzby sa bude odvíjať ďalšia práca na projekte. Testovacia prevádzka k dátumu tejto verzie dokumentácie trvá, takže zatiaľ nie sú určené žiadne závery. Momentálne evidujeme testovanie 3 používateľmi, objavenie jednej chyby v systéme (a jej následnej opravy) a zadanie 1 pripomienky. Po testovacej prevádzke bude nasledovať analýza zozbieraných pripomienok a následné vyhodnotenie, ktorými sa budeme ďalej primárne zaoberať.

Keďže bol systém vyvíjaný už v školskom roku 2008/2009, tak našou primárnou úlohou bolo analyzovať aktuálnu funkčnosť systému, identifikovať existujúce chyby a opraviť prípadne doplniť najakútnejšie chybné časti systému. Väčšina našej práce bola preto zameraná na hľadanie a odstraňovanie chýb v systéme, aby testovacia prevádzka prebehla čo najlepšie a teda aby sme mohli získať plnohodnotnú väzbu od používateľov.

Stručné zhrnutie funkcií vytvoreného prototypu k dátumu tejto verzie dokumentácie:

- Priradenie garantov k jednotlivým predmetom
- Evidencia miestností (báza dát zhodná so systémom AIS)
- Evidencia predmetov (báza dát zhodná so systémom AIS)
- Evidencia vybavenia potrebného k výučbe
- Zadanie požiadaviek na predmet garantom predmetu (skratka predmetu, výber prednášajúceho a vedúceho cvičení, rozsah prednášok a rozsah cvičení)
- Zadanie požiadaviek na predmet prednášajúcim predmetu (počet prednášok, časový rozsah prednášok, kapacita miestnosti, počet študentov, požadované vybavenie a iné špecifiká vyskytujúce sa pri zadávaní požiadaviek na prednášky)
- Zadanie požiadaviek na predmet cvičiacim predmetu (počet cvičení, časový rozsah cvičení, kapacita miestnosti, počet študentov v skupine, počet cvičení súčasne a iné špecifiká vyskytujúce sa pri zadávaní požiadaviek na cvičenia)
- Zobrazenie zadaných požiadaviek tvorcovi rozvrhu
- Zadávanie časových priorít prednášajúcimi a cvičiacimi
- Zobrazenie časových priorít prednášajúcich a cvičiacich tvorcovi rozvrhu
- Zadávanie pripomienok k systému používateľmi (dôležité pre ďalší rozvoj systému)
- Prezeranie zadných požiadaviek na systém administrátormi
- Základné funkcie systému ako prihlasovanie (kontrola cez systém ldap), odhlasovanie a správa používateľov
- On-line pomocník

## 7 Návrh rozšírení a úprav existujúceho systém

---

Táto kapitola slúži na opis činností, ktoré bude potrebné vykonať po úspešnom nasadení testovacej verzie systému a tiež je uvedený hrubý návrh rozšírení a úprav, ktoré plánujeme implementovať.

### 7.1 Pripomienkovanie testovacej verzie používateľmi

K dátumu tejto verzie dokumentácie prebieha testovacia verzia systému a doposiaľ evidujeme 1 zadanú pripomienku. Očakávame nárast počtu pripomienok a podľa toho prispôbime naše zameranie v ďalšom semestri.

### 7.2 Návrh ďalšej práce na projekte

Predpokladáme, že po testovacej prevádzke budeme mať viacero pripomienok k funkčnosti systému a je možné, že vzniknú aj nové požiadavky. Túto spätnú väzbu analyzujeme, pomenujeme novoobjavené chyby a definujeme ďalšie potrebné rozšírenia a zmeny systému. Následne určíme prioritu a budeme postupne tieto zmeny a opravy zapracovávať od tých s najvyššou prioritou.

Máme naplánované niektoré rozšírenia, ktoré vypracujeme v nasledujúcom období po spustení testovacej verzie. Uvádžam ich zoznam:

- Pravdepodobne najdôležitejšie je pripomienkovanie rozvrhov, aby používatelia mohli dať vedieť ak im niečo prekáža.
- Dôležité rozšírenie je tiež vytváranie verzií požiadaviek, aby každá poslaná požiadavka bola jedinečná (ani zlé požiadavky sa nebudú dať zmazať, pretože v nich môže byť dôležitá informácia).
- Automatické preposielanie požiadaviek aj pripomienok k rozvrhom na email.
- Aktualizácia používateľskej príručky.
- Validácia kódu - zmeny v css a pohľadoch, nahodenie šablón do pohľadov (napríklad Smarty).
- PHP 5.3 kompatibilita - nahradenie vybraných funkcií za iné. Nakoľko veľká časť zmien sa týka db.php, spolu s tým sa dá doplniť funkcionality na QueryParams a Prepare a Execute.
- Evidovať požiadavky a predmety + všetky spojené dáta v rámci nejakého semestra akademického roka (aby nebolo potrebné vždy pred zberom mazať požiadavky).
- Pridať filtrovanie a usporadúvanie dát.
- Zmeniť štruktúru študentov, lebo teraz je veľké previazanie na iné tabuľky.
- Pridať unit-testy (Test Driven Development) prípadne testovanie správania (Behavioral Driven Development)

System budeme počas vývoja neustále testovať, aby sme objavili čo najväčší počet existujúcich chýb a následne určíme prioritu opráv a zmien, ktoré bude potrebné vykonať.

## **8 Prílohy**

---

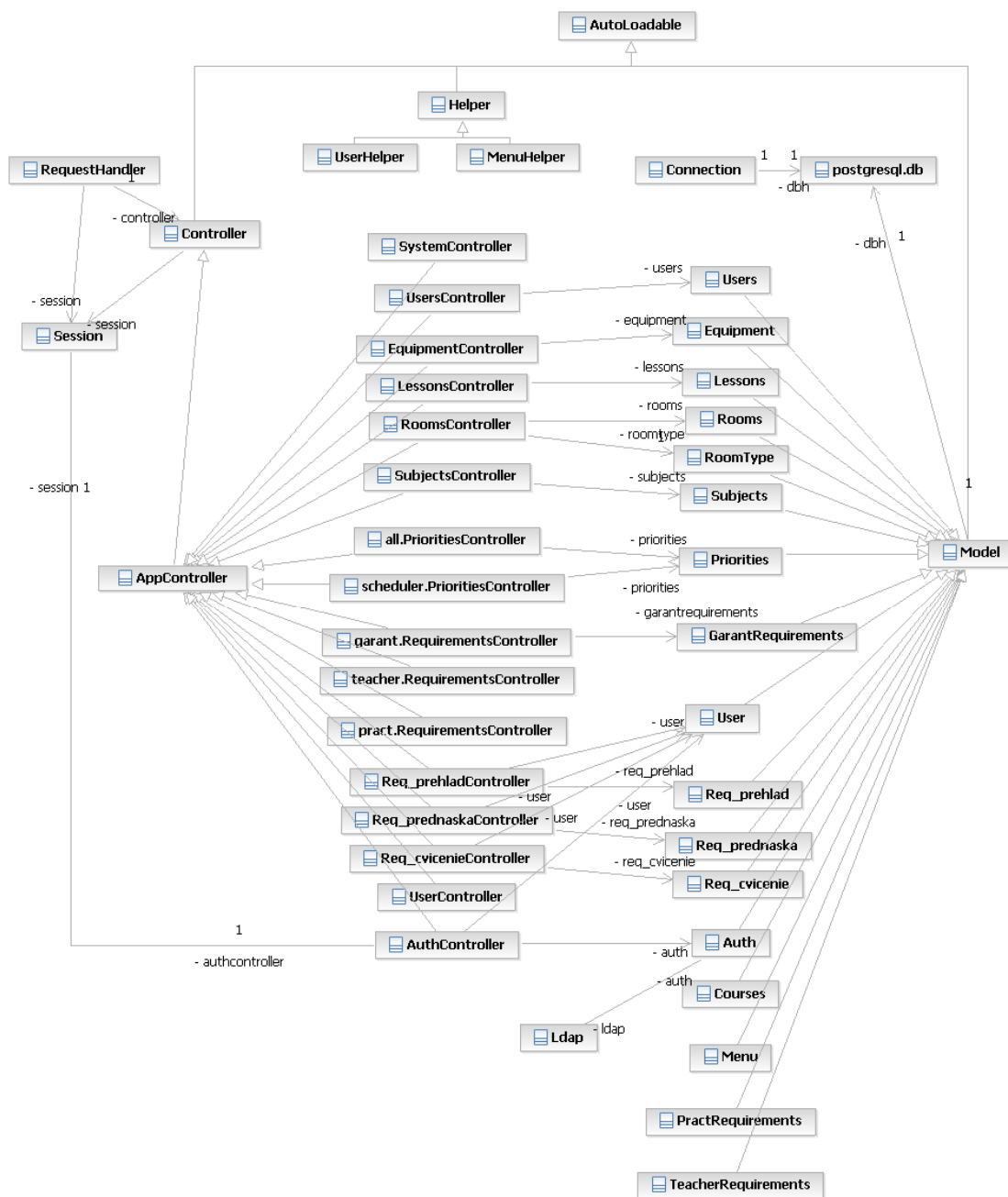
- A. Diagram tried systému vytvoreného v školskom roku 2008/2009
- B. Model údajov systému zo školského roku 2008/2009
- C. Model údajov systému po úpravách v školskom roku 2009/2010
- D. Používateľská príručka
- E. Inštalačná príručka
- F. Obsah elektronického média



## **PRÍLOHA A      Diagram tried systému vytvoreného v školskom roku 2008/2009**

---

V tejto prílohe sa nachádza obrázok diagramu tried systému, ktorý sme dostali po tíme BackSpace a vytvorili sme ho, aby sme lepšie spoznali systém a tiež aby sme získali väčší prehľad.

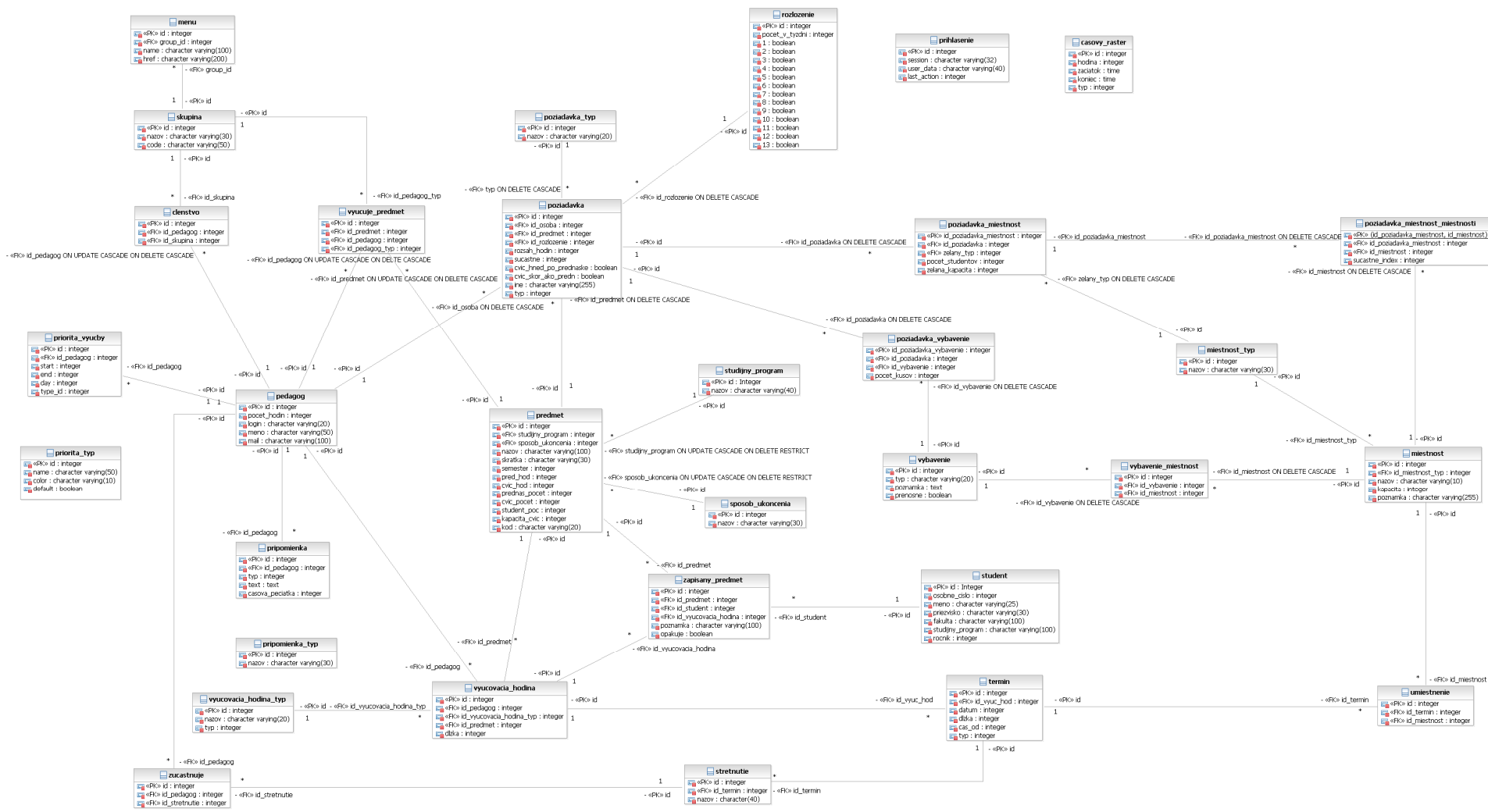


Obr. 2 Diagram tried systému na podporu tvorby rozvrhov

## **PRÍLOHA B      Model údajov systému zo školského roku 2008/2009**

---

Uvádzame model údajov, ktorý vytvoril tím BackSpace pred nami, aby sme vedeli čo a ako sme upravili od minulého roku. Navyše sme tento model prekreslili v nástroji Rational Software Architect.

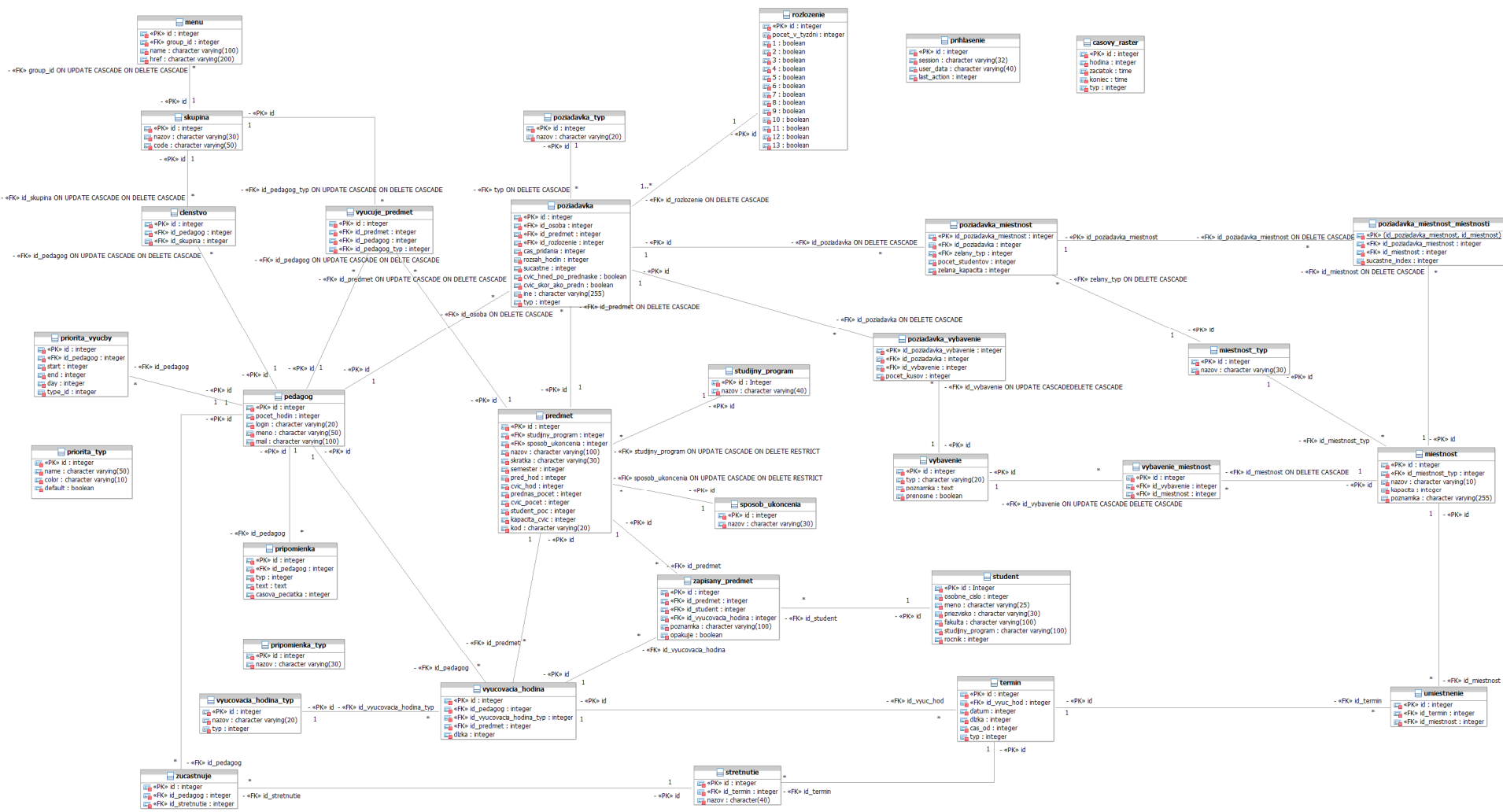


Obr. 3 Model údajov systému zo školského roku 2008/2009 nakreslený v RSA

**PRÍLOHA C      Model údajov systému po úpravách v školskom  
roku 2009/2010**

---

Táto príloha obsahuje model údajov po úpravách, ktoré sme spravili my v rámci opravovania chýb alebo rozširovania systému.



Obr. 4 Model údajov systému po úpravách v školskom roku 2009/2010

## **PRÍLOHA D      Používateľská príručka**

---

V tejto prílohe sa nachádza používateľská príručka k systému pre podporu tvorby rozvrhov.

## **PRÍLOHA E      Inštalčná príručka**

---

V tejto prílohe uvádzame inštalčnú príručku k systému pre podporu tvorby rozvrhov.



## **PRÍLOHA F      Obsah elektronického média**

---

<b>Adresár</b>	<b>Obsah</b>
\Src	Aktuálna verzia systému k dátumu 8.12.2009
\Doc	Dokumenty týkajúce sa aktuálneho systému k dátumu 8.12.2009
\docs, minulorocny tim	Dokumenty vypracované tímom Backspace v školskom roku 2008/2009
\Ponuky	Elektronická verzia ponúk tímu Bug Hunters na predmet Tímový projekt v školskom roku 2009/2010
\Riadenie	Elektronická verzia dokumentov riadenia tímu Bug Hunters
\Systemova_dokumentacia	Elektronická verzia tohto dokumentu a ďalších dokumentov týkajúcich sa projektu k dátumu 8.12.2009
\DB	Záloha databázy k dátumu 8.12.2009