

**Fakulta matematiky, fyziky a informatiky Univerzity Komenského v Bratislave**

Katedra informatiky

Informatika

# **Podpora ubytovacieho procesu**

(diplomová práca)

FILIP LIKAVČAN

**Vedúci diplomovej práce:** Mgr. Martin Králik

Bratislava, 2008

Čestne vyhlasujem, že som túto diplomovú prácu vypracoval  
samostatne s použitím citovaných zdrojov.

.....

Ďakujem môjmu vedúcemu diplomovej práce Mgr. Martinovi Králikovi za poskytnutú podporu a hlavne trpezlivosť. Ďalej chcem poďakovať Judite Sobotovej a Stanovi Maroszovi za podporu pri písaní tejto práce.

## **Bibliografická identifikácia**

Autor:	Filip Likavčan
Názov diplomovej práce:	Podpora ubytovacieho procesu
Škola:	Univerzita Komenského
Fakulta:	Fakulta matematiky, fyziky a informatiky
Katedra:	Katedra informatiky
Vedúci diplomovej práce:	Mgr. Martin Králik
Rozsah práce:	72 strán

Bratislava, máj 2008

## **Abstrakt**

Práca sa zaoberá analýzou ubytovacieho procesu Univerzity Komenského. Analýza popisuje ako funguje ubytovací proces a identifikuje v ňom problematické miesta. Na základe analýzy je v ďalšej časti práce navrhnutý a implementovaný systém na podporu ubytovacieho procesu. Systém umožňuje flexibilne konfigurovať kľúčové parametre ubytovacieho procesu na ľubovoľnej fakulte.

## **Kľúčové slová**

ubytovací proces, webová aplikácia, automatizácia

# Predhovor

Ubytovací proces, ktorý z pohľadu študenta začína podaním žiadosti a končí ubytovaním, predstavuje z pohľadu univerzity komplexnú technickú, manažérsku, právnu a morálnu problematiku.

Na proces prideľovania ubytovania a rozdeľovania internátnych miest sú kladené požiadavky z rôznych strán: zo strany žiadateľov o ubytovanie je to najmä požiadavka na transparentnosť systému a prideľovacích kritérií, zo strany internátu, univerzity a fakúlt ide najmä o požiadavky na efektivitu, čiže jednoduchú manipuláciu a minimálny počet zdĺhavých manuálnych krokov.

V tejto diplomovej práci si kladieme otázku, ako vyzerá ubytovací proces na Univerzite Komenského, a ako je možné prostriedkami informačných technológií prispieť k efektívnejšej a transparentnej realizácii jednotlivých jeho operácií.

V práci sme navrhli, implementovali a počas dvoch rokov testovacej prevádzky dôkladne vyladili systém na podporu ubytovacieho procesu. Systém umožňuje automatizovať mnohé kroky v procese, ktoré boli doteraz vykonávané manuálne. Zároveň je systém flexibilný a umožňuje konfiguráciu kľúčových parametrov tak, aby bol použiteľný pre všetky fakulty UK, prípadne iných univerzít.

# Obsah

<b>1 Úvod.....</b>	<b>8</b>
1.1 Využitie softvéru v ubytovacom procese na FMFI UK .....	9
1.1.1 Softvér na podporu činnosti ubytovacej komisie .....	9
1.1.2 Podávanie žiadostí cez internet .....	9
1.1.3 Prvý ucelený softvér na podporu ubytovacieho procesu na FMFI UK .....	10
<b>2 Popis ubytovacieho procesu na UK .....</b>	<b>11</b>
2.1 Pojmy.....	11
2.2 Rola: Žiadateľ o ubytovanie .....	14
2.2.1 Operácia: Podanie žiadosti.....	14
2.3 Rola: Fakulta.....	14
2.3.1 Operácia: Určenie predpokladaného počtu žiadateľov.....	14
2.3.2 Operácia: Vyhodnotenie žiadosti .....	14
2.3.3 Operácia: Pridelenie miest úspešným žiadateľom.....	14
2.4 Rola: Univerzita .....	15
2.4.1 Operácia: Pridelenie počtu miest pre fakultu .....	15
2.5 Rola: Internát .....	15
2.5.1 Operácia: Určenie počtu dostupných miest na internáte.....	15
2.5.2 Operácia: Pridelenie miest pre fakultu.....	15
2.5.3 Operácia: Výmena miest medzi fakultami.....	16
2.5.4 Operácia: Posúdenie a kontrola pridelených miest.....	16
2.5.5 Operácia: Poskytnutie ubytovania .....	16
2.6 Prenos a spracovanie dát .....	16
2.7 Problémy.....	17
2.7.1 Ochrana osobných údajov .....	17
2.7.2 Integrita a konzistencia údajov .....	18
2.7.3 Komunikácia .....	19
2.7.4 Archivácia údajov .....	19
<b>3 Funkčná špecifikácia systému na podporu ubytovacieho procesu.....</b>	<b>20</b>
3.1 Definícia pojmov.....	20
3.1.1 Technický administrátor systému .....	20
3.1.2 Administrátor systému .....	20
3.1.3 Oprávnená osoba .....	21
3.1.4 Žiadosť .....	21
3.1.5 Žiadateľ .....	21
3.1.6 Bodový zisk žiadosti.....	22
3.1.7 Záznam k žiadosti .....	22
3.1.8 Modifikátor .....	23
3.1.9 Celkový bodový zisk žiadosti .....	25
3.2 Používateľská časť .....	25
3.2.1 Činnosť: Registrácia žiadateľa .....	25
3.2.2 Činnosť: Podanie a upravovanie žiadosti.....	26
3.2.3 Činnosť: Výber miest.....	26
3.2.4 Činnosť: Výber kamarátov.....	27
3.2.5 Informačný portál .....	27
3.3 Administračná časť .....	28
3.3.1 Prístupové práva .....	28
3.3.2 Činnosť: Import študijnej histórie .....	28

3.3.3	Činnosť: Automatické generovanie záznamov k žiadosti.....	28
3.3.4	Činnosť: Správa administrátorov.....	28
3.3.5	Činnosť: Správa informačných článkov .....	29
3.3.6	Činnosť: Pridelovanie miest fakultám.....	29
3.3.7	Činnosť: Pridelovanie miest úspešným žiadateľom.....	29
3.3.8	Činnosť: Správa žiadostí.....	31
3.3.9	Činnosť: Stanovenie úspešných žiadostí.....	31
3.3.10	Činnosť: Nastavenie používateľskej časti.....	32
3.3.11	Činnosť: Generovanie výstupných zoznamov .....	32
<b>4</b>	<b>Technická špecifikácia systému na podporu ubytovacieho procesu .....</b>	<b>33</b>
4.1	Charakterizácia údajov .....	33
4.2	Platforma .....	33
4.3	Zabezpečenie .....	34
<b>5</b>	<b>Návrh systému na podporu ubytovacieho procesu .....</b>	<b>35</b>
5.1	Štruktúra databázy.....	35
5.2	Implementácia.....	45
5.2.1	Šablónovací systém XTemplate .....	46
5.2.2	Generátor excelovských dokumentov Spreadsheet_Excel_Writer .....	47
5.3	Konfigurácia .....	49
5.3.1	Konfigurácia typov záznamov a typov modifikátorov .....	49
5.3.2	Konfigurácia automatického generovania záznamov .....	50
5.3.3	Konfigurácia automatického generovania modifikátorov .....	51
5.3.4	Konfigurácia výpočtu bodového zisku žiadosti .....	53
5.3.5	Import histórie žiadateľov.....	54
5.4	Import údajov do základných číselníkov.....	55
5.5	Nasadenie aplikácie.....	55
5.6	Google Apps .....	56
<b>6</b>	<b>Používateľský manuál.....</b>	<b>58</b>
6.1	Používateľská časť .....	58
6.1.1	Registrácia žiadateľa.....	58
6.1.2	Podanie a upravovanie žiadosti .....	59
6.1.3	Výber miest .....	60
6.1.4	Výber kamarátov .....	62
6.2	Administračná časť .....	62
6.2.1	Správa administrátorov .....	62
6.2.2	Informačné správy .....	63
6.2.3	Import študijnej histórie .....	63
6.2.4	Vygenerovanie záznamov .....	64
6.2.5	Pridelovanie miest fakultám .....	64
6.2.6	Pridelovanie miest žiadateľom.....	65
6.2.7	Správa žiadostí .....	67
6.2.8	Stanovenie čiar .....	68
6.2.9	Stav žiadostí .....	69
6.2.10	Nastavenia .....	70
6.2.11	Generovanie výstupných zoznamov .....	70
<b>7</b>	<b>Záver .....</b>	<b>71</b>
<b>8</b>	<b>Literatúra a zdroje.....</b>	<b>72</b>

# 1 Úvod

Pre každého študenta, ktorý nepochádza z miesta, v ktorom sídli fakulta, na ktorej študuje, je ubytovanie na internáte dôležitý aspekt jeho života. Proces, ktorý z pohľadu študenta začína podaním žiadosti a končí ubytovaním, predstavuje z pohľadu univerzity komplexnú technickú, manažérsku, právnu a morálnu problematiku. V súčasnosti je veľká časť tohto procesu realizovaná bez komplexnejšej podpory informačnými technológiami, pričom posledný veľký technologický skok predstavovalo rozšírenie e-mailovej komunikácie.

Absencia technologicky vyspelého riešenia na podporu jednotlivých operácií procesu má za následok improvizáciu aktérov v rámci ich možností. Množstvo údajov prúdi len na papieroch, kompetencie sú niekedy nejasné, zlyháva komunikácia, a z toho vzniká celá škála omylov a nedorozumení. Vo všeobecnosti je možné povedať, že celý proces stojí a padá na kvalite ľudského faktora a že ochrana proti jeho zlyhaniu, či už neúmyselnému alebo plánovanému, je minimálna.

V tejto práci sme si stanovili cieľ vytvoriť systém, ktorý bude uľahčovať a zefektívňovať vykonávanie jednotlivých operácií procesu a automatizáciou znižovať riziko omylov. Požiadavkou je aj flexibilita, aby systém vedel podporiť zmeny v ubytovacom procese.



# 1.1 Využitie softvéru v ubytovacom procese na FMFI UK

## 1.1.1 Softvér na podporu činnosti ubytovacej komisie

História softvérov na podporu práce ubytovacej komisie Fakulty matematiky, fyziky a informatiky Univerzity Komenského (ďalej len FMFI UK) sa začína programom vytvoreným v roku 2001 Petrom Rafajom. Jeho desktopová aplikácia napísaná v jazyku Delphi a pracujúca nad databázou MySQL bola použitá niekoľkokrát počas jeho pôsobenia v ubytovacej komisii a ešte asi dva roky po jeho odchode. Aplikácia umožňovala pridelovať izby žiadateľom a generovať výstupy v podobe zoznamov. Aplikácia neumožňovala prácu cez sieť, ani nijak nevyužívala internet na prenos a publikovanie údajov. V dobe, keď bola vytvorená, znamenala veľký prínos pre prácu ubytovacej komisie. Výrazným problémom pri neskoršom používaní bola nedostatočná dokumentácia ako zdrojového kódu, tak aj používateľského rozhrania. Bolo problematické zapracovať do nej zmeny vyplývajúce zo zmien v ubytovacom procese, ako aj nové funkčné požiadavky, ako sú napríklad zmenené pravidlá pre vyhodnocovanie žiadostí.

Z týchto dôvodov sa neskôr aplikácia prestala používať.

## 1.1.2 Podávanie žiadostí cez internet

Ďalšia etapa vo využívaní softvéru v ubytovacom procese začala, keď sa členom komisie stal autor práce. V roku 2005 sa prvýkrát na FMFI UK zaviedlo podávanie žiadosti o ubytovanie cez internet, čím sa poskytla softvérová podpora pre ubytovací proces nielen ubytovacej komisii, ale aj žiadateľom. Elektronické podávanie žiadosti umožnilo zjednodušiť prácu komise, ktorá dovtedy všetky žiadosti pracne prepisovalo do počítača.

Výrazne sa uľahčilo podávanie žiadosti samotným žiadateľom. Pozitívna odozva bola logickým dôsledkom toho, že táto forma podávania žiadostí riešila hneď niekoľko problémov, ktoré sa vyskytovali pri papierových žiadostiach:

- podávanie papierových žiadostí bolo nepohodlné pretože bolo viazané na jedno konkrétne podacie miesto a obmedzené na časový interval, počas ktorého bol k dispozícii člen ubytovacej komisie

- pri prepise údajov z papierových žiadostí bolo väčšie riziko chybovosti
- prepis údajov z papierových žiadostí bol časovo náročný
- zrakovo postihnutí študenti potrebovali pri podávaní papierových žiadostí asistenciu

### **1.1.3 Prvý ucelený softvér na podporu ubytovacieho procesu na FMFI UK**

Z pohľadu žiadateľov znamenal rok 2005 výrazný pozitívny posun. Z pohľadu ubytovacej komisie išlo tiež o posun, ale po skúsenosti s programom Petra Rafaja len o dočasný. Obslužné skripty boli totiž určené len pre technicky vyspelého používateľa a neexistovala k nim dokumentácia. Hrozilo teda, že po odchode autora práce z ubytovacej komisie bude znovu potrebné začať „na zelenej lúke“.

Toto riziko si autor práce uvedomoval, a preto v roku 2006, po dvoch rokoch činnosti v ubytovacej komisii, vytvoril prvú ucelenú verziu softvéru na podporu ubytovacieho procesu na FMFI UK.

Vytvorenie nového softvéru autor práce nechápal len ako mechanické vytvorenie nástrojov, ktoré budú riešiť aktuálne potreby. Dôležité bolo analyzovať ako informačné technológie môžu pomôcť k prípadnej zmene a zlepšeniu samotného ubytovacieho procesu. Jedným z praktických dôsledkov tohto uvažovania bol návrh nových pravidiel pre ubytovanie na FMFI UK, ktoré autor práce vypracoval, a ktoré neskôr schválilo vedenie fakulty [1]. Bez využitia informačných technológií by napríklad bolo náročné kontrolovať, či sa daný študent hlási do prvého ročníka druhý krát, alebo či študuje dlhšie ako je odporúčaná dĺžka štúdia.

Pri vytváraní softvéru bola snaha umožniť čo najväčšiu flexibilitu a zároveň jednoduchosť používania. Tento softvér bol východiskovým bodom a zároveň umožnil otestovanie niektorých myšlienok pri návrhu systému, ktorý podporuje ubytovací proces na celej univerzite.

## 2 Popis ubytovacieho procesu na UK

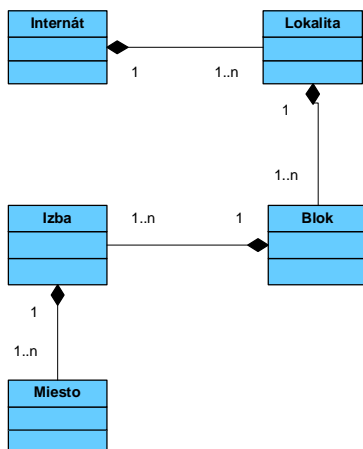
V tejto kapitole popíšeme ubytovací proces tak, ako v súčasnosti prebieha na Univerzite Komenského. Z tejto analýzy potom v ďalších kapitolách vychádzame pri špecifikácii požiadaviek, návrhu a implementácii systému.

### 2.1 Pojmy

#### *Miesto*

Miesto je lôžko na internátnej izbe. Internát je právny subjekt, ktorý spravuje jednu alebo viac lokalít. Lokalita označuje logickú podčasť internátu, ktorá má istú mieru samostatnosti. Lokalita sa v praxi obyčajne vymedzuje budovou (alebo sústavou budov). Lokalita je členená na bloky, pričom v praxi zväčša každá budova lokality tvorí jeden blok. Izba je vždy priradená bloku a môže mať jedno alebo viac miest.

Aj keď miesto je logicky najmenšia časť štruktúry, v celom procese sa s ním pracuje len ako s údajom o počte miest danej izby. Schematické zobrazenie štruktúry internátu je na Obr. 1.



Obr. 1: Schematické zobrazenie štruktúry internátu

### *Žiadosť o ubytovanie*

Žiadosť o ubytovanie je prejavom záujmu bývať na internáte nasledujúci akademický rok. Obsahuje údaje o tom, kto žiada o bývanie, či má záujem o zachovanie izby, na ktorej býva aktuálny akademický rok, prípadne môže uviesť inú izbu. Súčasťou žiadosti je aj čestné vyhlásenie, že žiadosť o ubytovanie nebola podaná aj na inej fakulte.

### *Oprávnená osoba*

Oprávnená osoba je osoba, ktorá môže podať žiadosť o ubytovanie na internáte. Oprávnenými osobami sú všetci študenti a doktorandi fakulty.

### *Žiadateľ*

Žiadateľ o ubytovanie je oprávnená osoba, ktorá využila svoje právo a podala žiadosť o ubytovanie.

### *Poradovník žiadateľov*

Žiadosti všetkých žiadateľov z fakulty sa zoradia do poradovníka. Spôsob vytvárania poradovníka je individuálny pre každú fakultu. Na väčšine fakúlt sa žiadosti hodnotia podľa vzorca, do ktorého vstupuje študijný priemer žiadateľa a vzdialenosť bydliska žiadateľa od miesta štúdia. Jednotlivé fakulty majú rôzne vzorce. Okrem ohodnotenia žiadosti bodmi sa niektoré žiadosti zvyhodňujú alebo znevýhodňujú podľa ďalších kritérií. Napríklad doktorandi majú zo zákona nárok na ubytovanie a preto sú ich žiadosti zvyhodnené.

### *Úspešný žiadateľ*

Úspešný žiadateľ je žiadateľ, ktorý získal právo na pridelenie miesta. Úspešný žiadatelia sa určujú na základe poradovníka, v ktorom sa spraví tzv. čiara, pričom každý žiadateľ nad čiarou je úspešný.

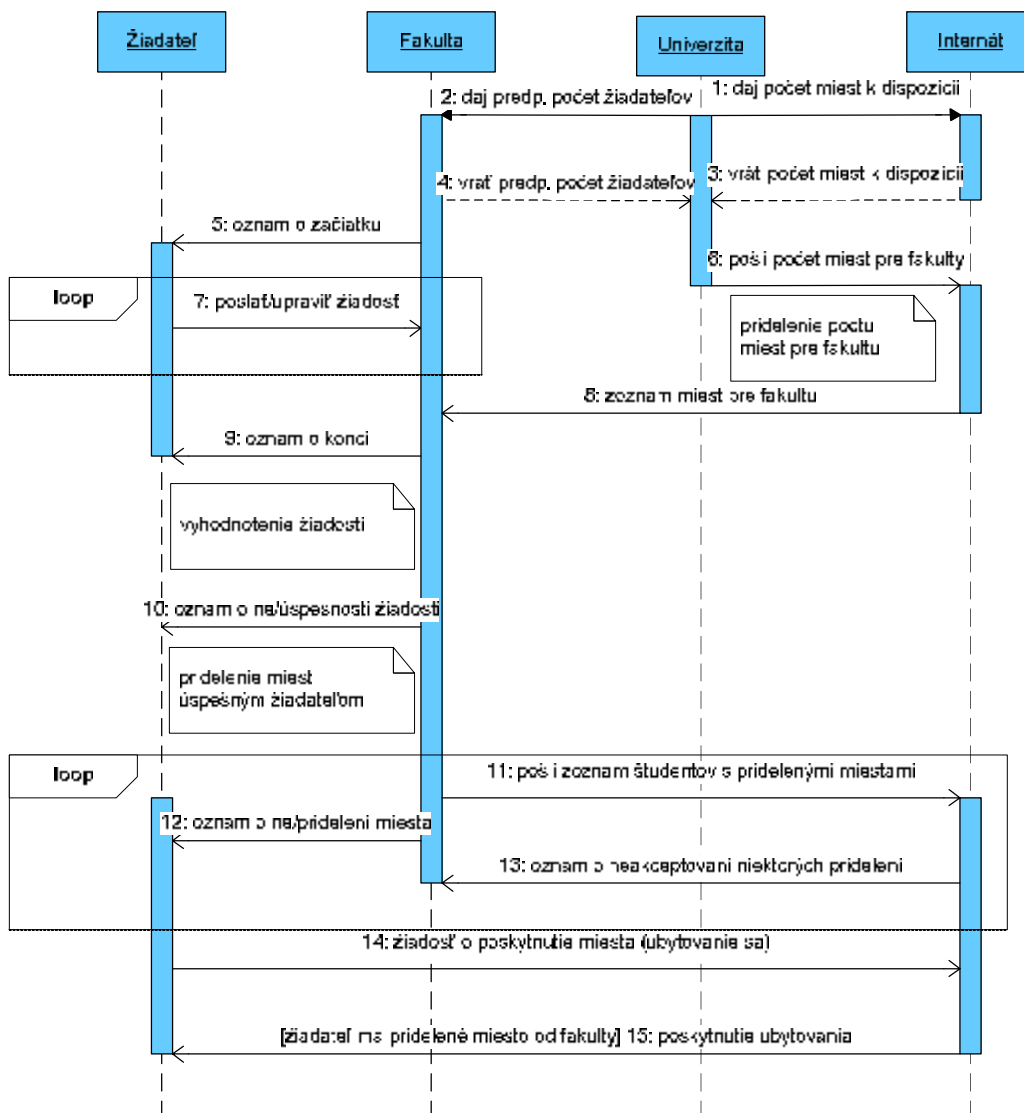
### *Zoznam miest pridelených fakulte*

Pre každú fakultu existuje zoznam izieb, na ktorých má pridelené aspoň jedno miesto. Ak z niektorej izby nemá pridelené všetky miesta, tak v zozname je táto skutočnosť

poznačená číslom, ktoré vyjadruje počet pridelených miest na danej izbe. Celkový počet miest na izbe sa určuje buď jednoduchým algoritmom (napr.: „dve miesta na izbe s párnym číslom, tri miesta na izbe s nepárnym číslom“) alebo súčasťou zoznamu sú aj explicitne uvedené počty miest.

### Zoznam miest pridelených úspešným žiadateľom

Zoznam miest obsahuje základné osobné údaje úspešného žiadateľa (meno, priezvisko, adresa) a údaj o izbe, na ktorej má pridelené miesto. Údaj o izbe sa skladá z lokality, bloku a čísla izby.



Obr. 2: Časová následnosť operácií v ubytovacom procese

## **2.2 Rola: Žiadateľ o ubytovanie**

Túto rolu realizuje oprávnená osoba a osoba žiadateľ o ubytovanie. Rola žiadateľ preto môže vykonávať operáciu „podanie žiadosti“.

### **2.2.1 Operácia: Podanie žiadosti**

Žiadateľ o ubytovanie si podá žiadosť o ubytovanie na príslušnej fakulte. Neexistuje kontrola, či žiadateľ nežiada o ubytovanie na viacerých fakultách. Výstupom je žiadosť o ubytovanie.

## **2.3 Rola: Fakulta**

Túto rolu v ubytovacom procese realizujú osoby členov ubytovacej komisie fakulty.

### **2.3.1 Operácia: Určenie predpokladaného počtu žiadateľov**

Fakulta určí predpokladaný počet žiadateľov na základe záujmu v aktuálnom akademickom roku, pričom zohľadní pomer počtu študentov v aktuálnom roku k počtu študentov v nasledujúcom roku. Výstupom je počet predpokladaných žiadateľov.

### **2.3.2 Operácia: Vyhodnotenie žiadosti**

Vstupom tejto operácie je zoznam žiadostí podaných na danej fakulte a počet miest, ktoré fakulta môže prideliť svojim žiadateľom. Fakulta vytvorí poradovník žiadostí a na základe počtu miest stanoví čiaru. Výstupom je zoznam úspešných žiadateľov.

### **2.3.3 Operácia: Pridelenie miest úspešným žiadateľom**

Vstupom tejto operácie je zoznam úspešných žiadateľov a zoznam miest pridelených fakulte. Fakulta pridelí úspešným žiadateľom konkrétne miesto na internáte. V praxi sa uplatňuje zásada stabilizácie, čo znamená, že žiadatelia dostávajú prioritne miesta, na ktorých sú ubytovaní aktuálny rok. Ak úspešný žiadateľ žiada o iné miesto, prípade dané

miesto fakulta nezískala na nasledujúci rok, tak dostane pridelené primerané miesto. Primerané znamená, že sa zohľadňuje prostredie, v ktorom býval a to sa snaží zachovať. Výstupom tejto operácie je zoznam úspešných žiadateľov s prideleným miestom.

## **2.4 Rola: Univerzita**

Túto rolu v ubytovacom procese realizujú osoby členov ubytovacej komisie univerzity.

### **2.4.1 Operácia: Pridelenie počtu miest pre fakultu**

Vstupom tejto operácie sú údaje o predpokladanom počte žiadateľov na fakultách univerzity a počet miest, ktoré internáty dali k dispozícii na ubytovanie. Na základe týchto údajov univerzita určí jednotlivým fakultám počty miest, na ktoré môžu prideliť úspešných žiadateľov.

Podoperáciou tejto operácie je rozdelenie prideleného počtu miest v rámci jednotlivých internátov. Toto rozdelenie robí univerzita, prípadne po dohode s jednotlivými fakultami. Konečným výstupom je počet miest, ktoré môžu fakulty prideliť na jednotlivých internátoch.

## **2.5 Rola: Internát**

Túto rolu v ubytovacom procese realizujú osoby zamestnancov ubytovacieho oddelenia internátu.

### **2.5.1 Operácia: Určenie počtu dostupných miest na internáte**

Internát určí koľko miest z celkovej kapacity je k dispozícii na ubytovanie úspešných žiadateľov. Tento počet nemusí byť zhodný s celkovou kapacitou internátu. Počet miest k dispozícii znižujú hlavne neobývatel'né miesta (poškodené) a miesta, ktoré slúžia na iný účel (napr. sú prenajímané ako hotelové). Výstupom je počet miest, ktoré sú k dispozícii na ubytovanie pre úspešných žiadateľov.

### **2.5.2 Operácia: Pridelenie miest pre fakultu**

Vstupom je počet miest, ktoré môžu fakulty pridelit' na internáte svojim úspešným žiadateľom. Výstupom je pridelenie konkrétnych miest pre jednotlivé fakulty.

### **2.5.3 Operácia: Výmena miest medzi fakultami**

Vstupom je požiadavka na výmenu miest, ktoré majú dve rôzne fakulty pridelené a výstupom je výmena pridelených miest medzi fakultami.

### **2.5.4 Operácia: Posúdenie a kontrola pridelených miest**

Vstupom je zoznam miest pridelených úspešným žiadateľom . Internát skontroluje, či nevznikli nejaké chyby (napr.: viac fakúlt ubytovalo žiadateľov na jedno miesto), prípadne či daný žiadateľ vyhovuje pravidlám internátu. Internát môže odmietnuť pridelenie miesta nejakému žiadateľovi, ak v minulosti porušil pravidlá internátu.

V prípade nájdených chýb, alebo nájdení nevyhovujúcich žiadateľov internát informuje o týchto skutočnostiach fakultu, ktorá urobí nápravu. To môže zahŕňať aj prehodnotenie do toho momentu neúspešných žiadostí a pridelenie ubytovania ďalším žiadateľom. Výstupom je zoznam chýb, ktoré treba opraviť.

### **2.5.5 Operácia: Poskytnutie ubytovania**

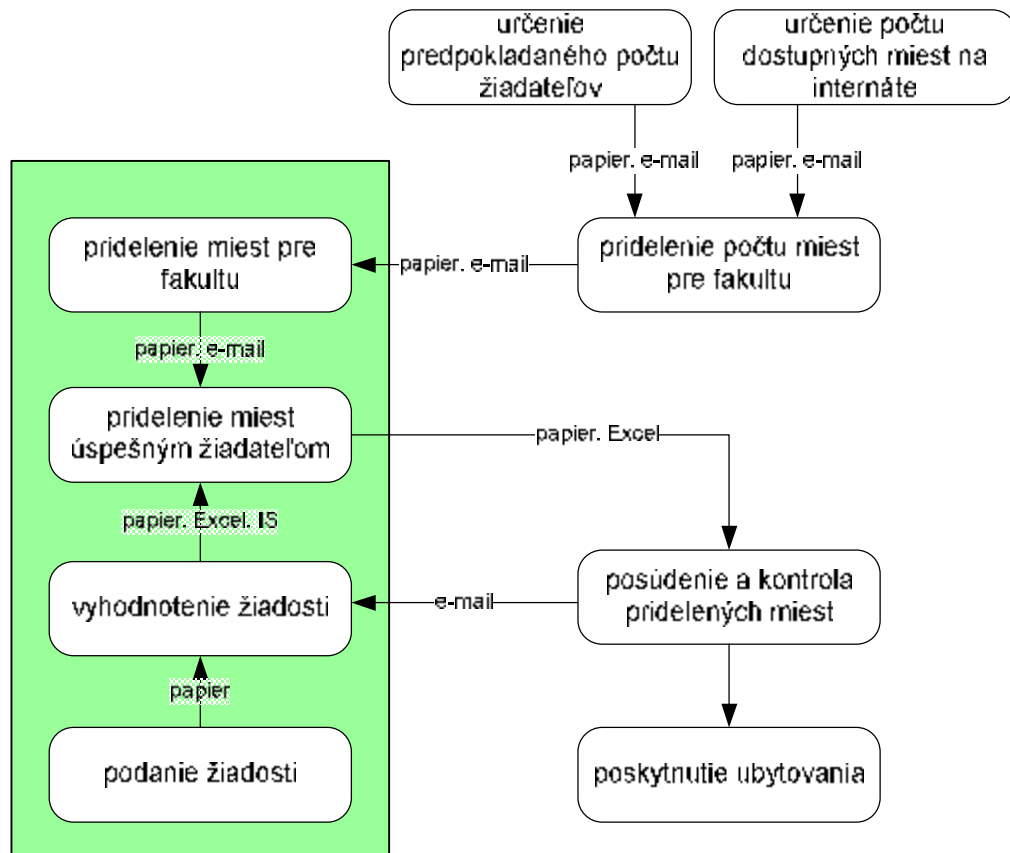
Úspešný žiadateľ s prideleným miestom požiada internát o poskytnutie ubytovania. Toto sa deje väčšinou na začiatku akademického roka.

## **2.6 Prenos a spracovanie dát**

Nasledujúci diagram znázorňuje akým spôsobom sú dáta v rámci ubytovacieho procesu prenášané a spracované. Ako je vidieť, proces je silne decentralizovaný a platformovo heterogénny. Nie je definovaný jednotný formát na prenos dát a na ich spracovanie sú používané rôzne prostriedky (podľa možností a schopností konkrétnych osôb realizujúcich jednotlivé roly).

Na operácie na Obr. 3 zvýraznené v obdĺžniku je zameraná táto práca.





Obr. 3: Tok dát medzi jednotlivými operáciami. V rámečku vľavo sú označené časti, na ktoré je zameraná táto práca.

## 2.7 Problémy

Vo všeobecnosti je možné konštatovať, že súčasný proces veľmi málo využíva moderné prostriedky informačných technológií a je príliš závislý na kvalite ľudského faktora. Proces síce má stanovené kompetencie a zodpovednosti pre jednotlivé roly, ale často neexistuje striktná kontrola ich vykonávania. Nezanedbateľným faktorom je aj komunikácia, ktorá prebieha ad hoc, bez presne stanovených pravidiel.

### 2.7.1 Ochrana osobných údajov

Ubytovací proces pracuje s osobnými a citlivými údajmi žiadateľov. Okrem základných identifikačných údajov sú k dispozícii aj citlivé informácie o prospechu žiadateľov a o ich študijnej minulosti. Ďalšími citlivými údajmi sú údaje o sociálnych či zdravotných podmienkach niektorých žiadateľov. Keďže proces je decentralizovaný, teda viackrát dochádza k presunu údajov medzi rôznymi systémami, zvyšuje sa riziko úniku

týchto údajov. Toto riziko ešte umocňuje fakt, že použité technológie (papier, e-mail) nie je možné dostatočne technicky zabezpečiť proti zneužitiu.

Legislatívne je ochrana osobných údajov v SR upravená najmä zákonom č. 428/2002 Z.z. o ochrane osobných údajov. Celý proces by bolo potrebné zosúladiť s týmto zákonom. V práci túto problematiku neriešime, ale na tomto mieste na ňu chceme poukázať.

## **2.7.2 Integrita a konzistencia údajov**

Dôsledkom slabého zabezpečenia procesu je aj slabá ochrana proti narušeniu integrity a konzistencie údajov. Podobne ako pri bezpečnostných rizikách aj tu je problematickým bodom technológia pri spracovaní a posielaní údajov. V ďalšom texte uvedieme niektoré incidenty z praxe.

*Na trojmiestnu izbu sú pridelení viac ako traja ľudia*

Prvým zdrojom tejto chyby môže byť internát, ktorý omylom povolí dvom fakultám prideliť na izbe viac miest ako je kapacita izby. Manuálne rozdeľovanie miest medzi fakulty zvyšuje pravdepodobnosť výskytu tejto chyby. Ak aj internát rozdelí miesta medzi fakulty správne, obdobná situácia môže nastať v prípade, že fakulty si neskôr vymenia miesta medzi sebou. V praxi sa občas stáva, že jedna fakulta na výmenu zabudne a pridelí na miesto, ktoré jej už v skutočnosti nepatrí, svojho žiadateľa. Jednotiacim momentom obidvoch situácií je absencia centrálného registra pridelených miest, ktorý by technickými prostriedkami vylučoval duplicity a iné chyby. Chyba sa v lepšom prípade odhalí v momente, keď internát kontroluje miesta pridelené žiadateľom a v horšom prípade, keď sa dvaja žiadatelia pokúsia ubytovať na rovnaké miesto.

*Internát eviduje iné miesto ako žiadateľovi pridelila fakulta, prípade internát neeviduje žiadne pridelené miesto pre žiadateľa*

Problém v tomto prípade spôsobuje absencia centrálného registra žiadateľov, a teda neexistencia jednoznačného identifikátora pre žiadateľa. Internát pre svoje potreby využíva svoj interný informačný systém, do ktorého importuje zoznamy žiadateľov s pridelenými miestami. Keďže neexistuje identifikátor žiadateľov import musí využívať nejakú heuristiku na identifikáciu žiadateľov zo zoznamu voči internému zoznamu osôb. Problém nastáva, keď je zvolená heuristika nesprávna. Konkrétne ak sa zvolí ako identifikácia meno a priezvisko, tak žiadatelia s rovnakým menom a priezviskom budú identifikovaní ako

jedna osoba. Iný spôsob je ručné importovanie zoznamu žiadateľov do interného informačného systému. V tomto prípade je zvýšené riziko zadania chybných údajov a zlyhania ľudského faktora.

#### *Strata zoznamu (alebo jeho časti) s pridelenými miestami*

Zoznam s pridelenými miestami je vytváraný manuálne na papieri alebo v Exceli. V prípade, že sa na fakulte vymenia ľudia v ubytovacej komisii, hrozí že nenájdu čiastočne rozpracovaný zoznam po svojich predchodcoch. Ubytovacie komisie často pracujú tímovo a údaje si vymieňajú osobne alebo e-mailom. Tu zase hrozí strata údajov pri zlyhaní komunikačného kanála. Problém je dôsledkom absencie centrálného registra miest pridelených žiadateľom. Aj v tomto prípade hrá významnú úlohu ne/zlyhanie ľudského faktora.

### **2.7.3 Komunikácia**

Ako už bolo spomenuté v úvode, komunikácia prebieha ad hoc – neexistuje norma, ktorá by definovala spôsob akým komunikácia prebieha. V praxi potom dochádza k tomu, že jednotliví aktéri napríklad používajú osobné e-mailové adresy. V prípade, že aktér prestane vykonávať danú rolu, strácajú sa informácie, ktoré prijal alebo odoslal. Ak napríklad komunikoval so žiadateľmi, tak títo ho ešte nejakú dobu oslovujú s otázkami ohľadom ubytovania.

Iný typ komunikačných problémov sú problémy v komunikácii so žiadateľmi. Ak sa vyskytne nejaký problém je potrebné ich kontaktovať. Súčasný systém síce zisťuje kontaktné údaje, ale napríklad nevyžaduje povinne platnú e-mailovú adresu alebo telefónne číslo, čo sú najrýchlejšie prostriedky na komunikáciu.

### **2.7.4 Archivácia údajov**

V procese sa pracuje s archívnymi údajmi, ale podobne ako pri ich spracovaní, aj tu sa často používajú zastarané prostriedky. Napríklad: ak pri podávaní žiadostí žiadateľ požiadala o rovnaké miesto aké má aktuálny rok, tak buď uvedie, o ktoré miesto ide, alebo ho ubytovacia komisia musí prácne zisťovať – v lepšom prípade z elektronického dokumentu, v horšom z papierového. Problém s archiváciou nastáva aj pri komunikácii – neprebieha archivácia dôležitej e-mailovej komunikácie medzi jednotlivými aktérmi procesu.

## **3 Funkčná špecifikácia systému na podporu ubytovacieho procesu**

Cieľom tejto kapitoly je komplexne popísať požiadavky na systém podporujúci vyššie špecifikovanú časť ubytovacieho procesu (Obr. 3) tak, aby mohla byť navrhnutá realizácia týchto požiadaviek a ich implementácia s použitím ľubovoľných technológií.

Systému rozdeľujeme na používateľskú a administračnú časť. Používateľská časť bude slúžiť role žiadateľ na vykonanie operácie „podanie žiadosti“. V tejto časti sa tiež budú zverejňovať informačné správy s informáciami určenými pre žiadateľa. Administračná časť slúži roliam fakulta, univerzita a internát na realizáciu ich operácií ubytovacieho procesu (Obr. 3).

### **3.1 Definícia pojmov**

#### **3.1.1 Technický administrátor systému**

Technický administrátor systému je osoba, ktorá má prístup k technologickým častiam systému (databáza, server, ai.). Zmyslom jeho existencie je zabezpečiť chod systému a realizovať jeho technické nastavenia. Technické nastavenia sú také nastavenia, ktoré je možné vykonať len priamym prístupom k danému prostriedku (napríklad zápis priamo do databázy, vytvorenie exportného skriptu, a pod.). Technický administrátor teda musí ovládať technológie, pomocou ktorých bude systém implementovaný a ktoré slúžia na jeho beh.

#### **3.1.2 Administrátor systému**

Administrátor systému je osoba, ktorá môže vykonávať povolené činnosti v administračnej časti systému.

Údaje administrátora sa skladajú z prihlasovacieho mena a hesla.

### **3.1.3 Oprávnená osoba**

Oprávnená osoba je identifikovaná identifikačným údajom. Pre študentov a doktorandov bude týmto identifikačným údajom číslo študentskej ISIC karty, pre novoprijatých študentov (ktorí ešte nemajú ISIC kartu) to bude rodné číslo (vrátane čísla za lomítkom).

### **3.1.4 Žiadosť**

Žiadosť bude obsahovať tieto údaje:

- interný identifikátor žiadateľa
- rodné číslo
- číslo ISIC karty
- meno, priezvisko, pohlavie
- ulicu, obec, psč
- e-mailová adresa
- fakulta, na ktorej žiadateľ žiada o ubytovanie
- typ štúdia
- študijný odbor
- rok štúdia v nasledujúcom akademickom roku
- či chce žiadateľ bývať na rovnakom mieste ako aktuálny akademický rok (len pre súčasných študentov)
- či je fajčiar
- vzdialenosť bydliska od miesta fakulty
- textová poznámka
- študijný priemer na fakulte v posudzovanom období
- priemery z troch profilových predmetov na strednej škole
- priemery z prvých troch ročníkov strednej školy
- stav žiadosti („úspešná“, „neúspešná“, „neposúdená“)

### **3.1.5 Žiadateľ**

Žiadateľom je oprávnená osoba, ktorá sa zaregistrovala a uložila svoju žiadosť.

### 3.1.6 Bodový zisk žiadosti

Každá žiadosť pri vyhodnocovaní získa číselnú hodnotu, ktoré nazveme „bodový zisk žiadosti“. Bodový zisk žiadosti sa počíta z niektorých údajov žiadosti podľa stanoveného vzorca. Obyčajne sa do úvahy berie údaj o študijnom prospechu a vzdialenosti bydliska od miesta, v ktorom sídli fakulta. Technický administrátor systému môže definovať ľubovoľný počet vzorcov, ktoré priradí fakultám. Takto bude možné namodelovať vzorce podľa špecifických potrieb jednotlivých fakúlt.

### 3.1.7 Záznam k žiadosti

Ku každej žiadosti bude možné definovať doplňujúce údaje nazvané „záznamy“. Záznam vyjadruje o danej žiadosti nejakú relevantnú skutočnosť, ktorá ovplyvňuje umiestnenie žiadosti v poradovníku.

Záznam obsahuje údaj o jeho type, krátky komentár a modifikátor. Modifikátor je popísaný v ďalšom texte (3.1.8 Modifikátor).

Typ záznamu ma takéto atribúty:

- názov
- pozícia v poradovníku
- priorita
- radenie v rámci rovnakej pozície v poradovníku
- spôsob zoradenia

Názov typu záznamu je slovné označenie typu záznamu zobrazované v systéme pri práci s nim.

Atribút „pozícia v poradovníku“ určuje základnú pozíciu žiadosti v poradovníku. Tento atribút môže nadobúdať štyri hodnoty: „začiatok“, „stred“, „koniec“ a „mimo zoznamu“.

Jedna žiadosť môže mať priradených viac typov záznamov. Do úvahy sa však berie len záznam, ktorého typ má najväčšiu prioritu. Atribút „priorita“ slúži na určenie priority typu záznamu. Môže nadobúdať kladné celočíselné hodnoty. Najväčšiu prioritu určuje najväčšia číselná hodnota.

Ak má typ záznamu pozíciu v poradovníku „začiatok“ alebo „koniec“, tak atribút „radenie v rámci rovnakej pozície v poradovníku“ určuje zoradenie žiadostí v rámci definovanej pozície v poradovníku. Skupiny žiadostí sa podľa tohto atribútu zoradia vzostupne.

Atribút „spôsob zoradenia“ môže nadobúdať dve hodnoty: „podľa celkového bodového zisku“ a „abecedne“. Čo je celkový bodový zisk žiadosti bude popísané ďalej v texte (3.1.9 Celkový bodový zisk žiadosti). Zoradením podľa abecedy sa myslí zoradenie podľa priezviska žiadateľa. Ak má typ záznamu pozíciu v poradovníku „začiatok“ alebo „koniec“, tak atribút „spôsob zoradenia“ určuje akým spôsobom sa zoradujú jednotlivé žiadosti s rovnakou pozíciou v poradovníku a rovnakou prioritou. Všetky žiadosti s pozíciou „stred“ sa zoradujú podľa celkového bodového zisku, bez ohľadu na ich prioritu.

Ak žiadosť nemá priradený žiadny záznam, tak má implicitne pozíciu v strede.

Záznamy budú umožňovať uprednostňovať a znevýhodňovať žiadosti podľa stanovených kritérií. Spôsobom akým sú záznamy navrhnuté umožňuje stanoviť ľubovoľné kritériá a zoradovanie žiadostí podľa nich.

Vo všeobecnosti teda kritériá (záznamy) definujú štyri základné situácie:

1. žiadosť je uprednostnená (zaradená na začiatok poradovníka)
2. žiadosť je znevýhodnená (zaradená na koniec poradovníka)
3. žiadosť je posudzovaná iba podľa bodového zisku (bude sa nachádzať medzi uprednostnenými a znevýhodnenými žiadosťami)
4. žiadosť je zrušená (vyradená z poradovníka)

### **3.1.8 Modifikátor**

Ku záznamu môže byť pridaný jeden modifikátor. Modifikátor obsahuje údaje o jeho type a početnosti.

Typ modifikátora má takéto atribúty:

- názov
- základná hodnota
- spôsob výpočtu
- maximálna hodnota
- typ

Názov typu modifikátora je slovné označenie modifikátora zobrazované v systéme pri práci s nim. Atribút „základná hodnota“ môže nadobúdať ako hodnotu celé nezáporné číslo. Atribút „spôsob výpočtu“ môže nadobúdať dve hodnoty: „exponenciálny“, „násobiaci“. Atribút „maximálna hodnota“ môže nadobúdať ako hodnotu celé nezáporné číslo. Atribút „typ“ môže nadobúdať dve hodnoty: „pozitívny“, „negatívny“ a určuje, či modifikátor žiadosť zvyhodňuje (je pozitívny) alebo znevýhodňuje (je negatívny).

Ak k záznamu pridáme modifikátor, musíme určiť jeho početnosť. Na základe základnej hodnoty a spôsobu výpočtu typu modifikátora a početnosti modifikátora sa vypočíta hodnota modifikátora pre daný záznam nasledovne:

- ak je spôsob výpočtu násobiaci, tak sa základná hodnota typu modifikátora vynásobí početnosťou modifikátora
- ak je spôsob výpočtu exponenciálny, tak sa základná hodnota typu modifikátora umocní na početnosť modifikátora

Ak hodnota modifikátora (HM) je väčšia ako maximálna hodnota typu modifikátora, tak sa hodnota nastaví na túto maximálnu hodnotu.

Bodový zisk záznamu N sa stanoví nasledovne:

$$N = ZT \cdot \left( \frac{HM}{100} \right)$$

pričom platí:

- $ZT = 1$ , ak je typ modifikátora „negatívny“ a
- $ZT = -1$ , ak je typ modifikátora „pozitívny“.

Príklad 1: Fakulta má pravidlo, ktoré určuje, že študenti, ktorí už v minulosti začali študovať na fakulte a štúdium neukončili budú zaradení na koniec poradovníka, v ktorom budú zoradení podľa celkového bodového zisku. V systéme zadefinujeme takýto typ záznamu:

Názov	Pozícia v poradovníku	Priorita	Spôsob zoradenia
znevýhodnená žiadosť	koniec	1000	podľa celkového bodového zisku



V systéme ďalej zadefinujeme takýto typ modifikátora:

Názov	Základná hodnota	Spôsob výpočtu	Maximálna hodnota	Typ
viacnásobné štúdium	50	násobiaci	100	negatívny

Študent Ján Novák už 3 krát študoval na fakulte, tak mu k žiadosti pridáme záznam typu „znevýhodnená žiadosť“. K záznamu priradíme modifikátor „viacnásobné štúdium“ s početnosťou 3. Bodový zisk záznamu jeho žiadosti má teda hodnotu:

$$\frac{\min(100, 3 \cdot 50)}{100} = 1$$

Jeho žiadosť bude zaradená na koniec poradovníka, kde budú všetky žiadosti so záznamom „znevýhodnená žiadosť“ zoradené podľa celkového bodového zisku.

Príklad 2: Fakulta má pravidlo, podľa ktorého dekan môže uprednostniť žiadosť žiadateľa, ktorý preukáže nutnosť pridelenia miesta zo zdravotných dôvodov. Takémuto žiadateľovi priradíme k žiadosti záznam tohto typu:

Názov	Pozícia v poradovníku	Priorita	Spôsob zoradenia
dekanská výnimka	začiatok	100	podľa celkového bodového zisku

### 3.1.9 Celkový bodový zisk žiadosti

Celkový bodový zisk žiadosti je súčtom bodového zisku žiadosti a bodového zisku záznamu danej žiadosti.

## 3.2 Používateľská časť

Všetky úkony (registrácia, podanie žiadosti, špecifikovanie miest a kamarátov) musia mať vysvetľujúci textový popis.

### 3.2.1 Činnosť: Registrácia žiadateľa

Podávať žiadosť môžu len oprávnené osoby. Oprávnená osoba sa musí najprv zaregistrovať, pričom pri registrácii preukáže svoju identitu zadaním správneho

identifikačného údaju. Pri registrácii si zvolí prihlasovacie meno a heslo a e-mailovú adresu. E-mailová adresa sa musí overiť, aby sa zaručilo, že v prípade potreby bude registrovaná osoba na nej dostupná. E-mailová adresa sa overí tak, že na ňu systém odošle e-mail s overovacou linkou, na ktorú musí žiadateľ kliknúť. Až po overení registrácie bude možné prihlásiť sa a podať žiadosť.

### **3.2.2 Činnosť: Podanie a upravovanie žiadosti**

Žiadateľ bude mať možnosť jednoducho vytvoriť žiadosť o ubytovanie. Okrem základných údajov bude mať možnosť napísať ľubovoľnú textovú poznámku, v ktorej vyšpecifikuje prípadné ďalšie požiadavky voči ubytovacej komisii.

Žiadateľ bude môcť upravovať žiadosť (vrátane pridružených údajov o miestach a kamarátoch) počas doby, ktorú stanoví ubytovacia komisia.

Pri prvom prihlásení a vytvorení novej žiadosti systém predvyplní všetky dostupné údaje z databázy oprávnených osôb. Žiadateľ môže meniť tieto údaje:

- meno, priezvisko, pohlavie
- ulicu, obec, psč
- typ štúdia
- fakulta (v prípade, že žiadateľ študuje na viacerých fakultách)
- študijný odbor
- rok štúdia v nasledujúcom akademickom roku
- či chce žiadateľ bývať na rovnakom mieste ako aktuálny akademický rok (len pre súčasných študentov)
- či je fajčiar
- vzdialenosť bydliska od miesta fakulty
- textová poznámka

Študijný odbor sa bude vyberať zo zoznamu všetkých študijných odborov zvolenej fakulty. Ak žiadateľ študuje len na jednej fakulte, tak fakulta bude pevne zvolená.

### **3.2.3 Činnosť: Výber miest**

Súčasťou žiadosti bude aj špecifikovanie primeraného počtu miest, na ktorých chce žiadateľ bývať spolu so stanovením ich poradia podľa priority. Systém musí umožniť zadávanie miesta strojovo čitateľným spôsobom, teda nie len textové políčko. Dôvodom je odstrániť nutnosť manuálne interpretovať údaj o mieste. Musí tiež byť možné vyjadriť záujem o miesto, ktoré ma žiadateľ pridelené aktuálny akademický rok (stačí vo forme „chce/nechce“ bývať na tom istom mieste).

Výber miesta bude prebiehať v štyroch krokoch:

1. výber lokality
2. výber bloku
3. výber izby

Žiadateľ môže v každom z prvých troch krokov uviesť aj možnosť „všetky“, čím dá najavo, že má napríklad záujem o konkrétnu lokalitu a blok, ale konkrétna izba v rámci bloku pre neho nie je podstatná. Ak žiadateľ v nejakom kroku uvedie možnosť „všetky“ nasledujúce kroky sa automaticky nastavlia na „všetky“ a prejde sa na štvrtý krok.

4. zadanie textovej poznámky

Ak žiadateľ nešpecifikuje konkrétnu izbu, tak v štvrtom kroku môže uviesť či má záujem o miesto s internetom a či má záujem o jednolôžkovú izbu. Pri uvedení konkrétneho miesta nemá zmysel určovať tieto atribúty, pretože miesto ich má pevne určené.

### **3.2.4 Činnosť: Výber kamarátov**

Systém musí umožniť špecifikovať s kým chce žiadateľ bývať. Dôvodom na túto požiadavku je zefektívnenie pridelovania miest a k tomu potrebnej komunikácie. Pokiaľ je totiž žiadateľ nespokojný s prideleným miestom, vzniká potreba na ďalšiu komunikáciu, čo zbytočne zaťažuje ubytovaciu komisiu. Kamarát sa bude pridávať zadaním jeho prihlasovacieho mena.

### **3.2.5 Informačný portál**

System bude plniť funkciu informačného portálu. Musí byť preto možné zverejňovať textové informácie v podobe článkov. Článok môže ale nemusí mať špecifikovaný dátum, čo vychádza z toho, že niektoré informácie majú obmedzenú platnosť, zatiaľ čo iné platia dlhodobo.

## **3.3 Administračná časť**

### **3.3.1 Prístupové práva**

Jednotlivé činnosti budú dostupné pre administrátorov, ktorí na ne dostanú prístupové právo. Okrem prístupového práva k činnosti existujú prístupové práva k fakultám a k internátom. Administrátor bude môcť mať definované prístupové právo k žiadnej alebo jednej fakulte a žiadnemu alebo jednému internátu.

Kombináciou prístupových práv k činnostiam a fakultám budú administrátori realizovať jednotlivé roly vyššie popísaného ubytovacieho procesu.

### **3.3.2 Činnosť: Import študijnej histórie**

Študenti aj doktorandi majú svoju študijnú históriu, pod čím sa chápu údaje o tom odkedy študujú, koľkokrát študovali, či v minulosti úspešne absolvovali nejakú fakultu, či boli zo štúdia vylúčení, a pod. System musí umožňovať naimportovať takéto údaje a následne s nimi pracovať. Údaje o študijnej histórii sa môžu využiť pri ohodnotení žiadosti, a to takým spôsobom, že z týchto údajov budú automaticky vygenerované záznamy (a modifikátory) k žiadosti.

Administrátor môže naimportovať len študijnú históriu žiadateľov z fakulty, ku ktorej má prístupové právo.

### **3.3.3 Činnosť: Automatické generovanie záznamov k žiadosti**

System musí umožňovať automatické vygenerovanie záznamov (a modifikátorov) k žiadosti z importovanej študijnej histórie.

Administrátor môže vygenerovať záznamy k žiadosti len žiadateľom z fakulty, ku ktorej má prístupové právo.

### **3.3.4 Činnosť: Správa administrátorov**

System musí umožniť pridávať a rušiť administrátorov, definovať im prístupové práva a meniť ich heslá.

### **3.3.5 Činnosť: Správa informačných článkov**

Informačné články budú slúžiť ako zdroj informácií pre žiadateľov. Zobrazovanie správ bude voliteľne buď trvalé, alebo len stanovené obdobie. Dôležitou informáciou pre žiadateľov sú poradovníky a zoznamy s pridelenými miestami. Ku každej správe preto bude možné priradiť poradovník súčasných a novoprijatých študentov, zoznam pridelených miest a zoznamy definované tagmi.

Administrátor môže spravovať len informačné články fakulty, ku ktorej má prístupové právo.

### **3.3.6 Činnosť: Pridelovanie miest fakultám**

System musí umožňovať efektívne a rýchlo určiť, ktoré miesta na internáte môže pridelať svojim žiadateľom daná fakulta. Miesta musí byť možné pridelať po jednom ale aj hromadne. V prípade, že na miesto je už priradený nejaký úspešný žiadateľ, systém nesmie povoliť zmeny priradenia fakulty.

Administrátor môže pridelať miesta fakultám len na internáte, ku ktorému má prístupové právo.

### **3.3.7 Činnosť: Pridelovanie miest úspešným žiadateľom**

Z pohľadu ubytovacej komisie je toto kľúčová časť celého systému, preto musí umožňovať efektívnu prácu so žiadosťami a miestami. System bude pracovať s rádovo stovkami žiadateľov, ktorým treba pridelit' miesta.

Pri pridelaní miest sa bude postupne prechádzať cez zoznam žiadateľov a nepriradeným žiadateľom sa budú pridelať voľné miesta. System musí umožňovať filtrovanie zoznamu všetkých žiadateľov podľa rôznych kritérií.

Príklady niektorých potrebných kritérií:

- pohlavie žiadateľa
- žiadateľ, ktorý ešte nemá pridelené miesto
- či je žiadateľ novoprijatý alebo už študuje

- či žiadateľ žiada o miesto, na ktorom aktuálne býva
- či žiadateľ žiada o konkrétne miesto

Technický správca systému bude môcť definovať zoznamy podľa požiadaviek ubytovacej komisie.

Okrem filtrovania zoznamu bude možné žiadosti zaraďovať do manuálne definovaných zoznamov. Manuálny zoznam sa vytvorí zadaním jeho názvu. Žiadosť sa do neho priradí manuálnou voľbou „priradiť“ a odoberie sa z neho voľbou „odobrať“. Manuálne zoznamy musí mať možnosť definovať osoba, ktorá prideliuje miesta. Tento proces sa nazýva tagovanie údajov. Naším cieľom je značkovací systém, ktorý dáva používateľom zmysel.

[2]

Tagovaním sa dosiahne flexibilita v kategorizácii žiadostí podľa rôznych minoritných kritérií, kvôli ktorým nie je efektívne žiadať technického správcu o vytvorenie filtra. Navyše niekedy vznikne potreba kategorizovať žiadosti podľa externých kritérií alebo podľa pozorovaní, ktoré nie sú strojovo zistiteľné (napríklad z textovej poznámky pri žiadosti).

Príklady niektorých manuálnych zoznamov z praxe:

- žiadatelia, ktorí v poznámke k žiadosti uviedli, že im prekáža hluk
- žiadatelia, ktorí chcú bývať s partnerom
- rôzne interné podzoznamy pre uľahčenie práce ubytovacej komisie

Pri postupnom prechádzaní žiadateľov s neprideleným miestom je dôležité zoradenie zoznamu úspešných žiadateľov. Systém musí umožniť zoradenie podľa priezviska alebo podľa poradovníka.

Pri prideliovaní miesta musia byť zobrazené všetky relevantné údaje viažuce sa k žiadosti. Konkrétne sa musia zobrazovať tieto údaje:

- základné údaje o žiadateľovi
  - meno a priezvisko, pohlavie, pridelené miesto, študijný odbor a ročník, poznámka k žiadosti
- požadované miesta aj s poznámkami k nim
- zoznam kamarátov, ktorých uviedol žiadateľ

- pre každého kamaráta zoznam základných údajov daného žiadateľa a zoznam jeho miest
- zoznam žiadateľov, ktorí žiadateľa uviedli ako kamaráta
  - pre každého žiadateľa zoznam jeho základných údajov a zoznam jeho miest
- zoznam žiadateľov, ktorí žiadajú o rovnaké miesta ako žiadateľ
- zoznam tagov (manuálnych zoznamov), ktoré má žiadosť priradené

Žiadateľa bude možné priradiť len na voľné miesto. Žiadateľovi s priradeným miestom nie je možné priradiť iné miesto, ale najprv mu musí byť odobrané pôvodné miesto.

Odoberania miest žiadateľom bude realizované v rámci tejto činnosti

Administrátor môže pridať miesta len žiadateľom z fakulty, ku ktorej má prístupové právo a dostupné sú len tie miesta, ktoré má priradené zvolená prístupná fakulta.

### **3.3.8 Činnosť: Správa žiadostí**

Systém musí umožniť vytvoriť a zmeniť žiadosť aj cez administratívne rozhranie. V praxi občas vznikne potreba, aby žiadosť do systému pridala ubytovacia komisia. Najčastejšou príčinou je, keď niekto podá papierovú žiadosť. Súčasťou upravovania žiadosti je aj priradovanie záznamov a modifikátorov.

Administrátor môže vytvárať a meniť len žiadosti študentov fakulty, ku ktorej má prístupové právo.

### **3.3.9 Činnosť: Stanovenie úspešných žiadostí**

V najjednoduchšom (a najčastejšom) prípade bude postup určenia úspešných žiadostí nasledujúci:

1. stanoví sa maximálny celkový bodový zisk žiadosti (tzv. čiara)
2. žiadosti sú postupne prechádzané od vrchu poradovníka smerom dolu a každá, ktorá má menší celkový bodový zisk ako je stanovené maximum je úspešná (bodový zisk žiadostí, ktoré sú zoradované podľa abecedy je stanovený na nula)

Uvedený algoritmus sa nebude aplikovať automaticky, ale bude musieť byť manuálne spustený. Systém okrem podpory tohto algoritmu musí umožňovať aj explicitne určiť, či je žiadosť úspešná, neúspešná alebo neposúdená. Nová žiadosť bude mať automaticky predvolený stav neposúdená.

Čiary sa budú určovať zadaním maximálneho bodového zisku úspešnej žiadosti.

Administrátor môže nastaviť len stav žiadostí, ktoré boli podané na fakulte, ku ktorej má prístupové právo.

### **3.3.10 Činnosť: Nastavenie používateľskej časti**

Systém musí umožňovať „vypnúť“ registrácie oboch typov, ako aj prihlasovanie sa do používateľskej časti.

Administrátor môže nastavovať používateľskú časť len pre fakultu, ku ktorej má prístupové právo.

### **3.3.11 Činnosť: Generovanie výstupných zoznamov**

Systém bude umožňovať vygenerovať excelovské dokumenty pre poradovníky oboch typov a pre zoznam pridelených miest.

Administrátor môže generovať výstupné zoznamy len pre fakultu, ku ktorej má prístupové právo.



## **4 Technická špecifikácia systému na podporu ubytovacieho procesu**

V tejto kapitole uvidíme technické požiadavky na implementáciu funkčných požiadaviek popísaných v predchádzajúcej kapitole.

### **4.1 Charakterizácia údajov**

System bude pracovať súhrnne s údajmi rádovo tisícov oprávnených osôb a žiadateľov, pričom jedna fakulta bude spravovať rádovo stovky oprávnených osôb a žiadateľov . Všetky zoznamy v používateľskom rozhraní musia byť preto pripravené na vizualizáciu rádovo stoviek položiek.

V systéme budú ďalej spravované rádovo tisíce miest, pričom pre jeden blok na internáte budú existovať rádovo desiatky až stovky miest.

Veľkosť údajov (aj pridružených) pre oprávnenú osobu, žiadateľa aj miesto je maximálne niekoľko kilobajtov, čiže pamäťové nároky údajov (bez riadiacích dát, napríklad indexových štruktúr databázy, ai.) v celom systéme budú rádovo v megabajtoch.

### **4.2 Platforma**

Keďže systém je určený pre rádovo tisíce ľudí, systém bude bežať ako webová aplikácia vo webovom prehliadači.

Klient bude implementovaný značkovacím jazykom HTML, kaskádovými štýlmi CSS a jazykom Javascript.

Server bude naprogramovaný v skriptovacom jazyku PHP 4 a ako úložisko dát bude slúžiť relačná databáza MySQL 4. Použitý skriptovací jazyk aj databáza sú dostupné pre

všetky relevantné operačné systémy, preto voľba operačného systému servera je na rozhodnutí technického administrátora.

System potrebuje k svojej činnosti funkčný e-mailový server na odosielanie e-mailov. Posielanie e-mailov bude realizované skriptovacím jazykom, a teda nebude viazané na konkrétny e-mailový server, čiže voľba e-mailového servera je opäť na rozhodnutí technického administrátora.

### **4.3 Zabezpečenie**

System pracuje s citlivými údajmi, preto musí byť kladený veľký dôraz na ich praktické zabezpečenie proti strate, úniku, neoprávnenému použitiu a zneužitiu, prístupu neoprávnených osôb k nim.

Komunikácia webového prehliadača so serverom musí prebiehať cez zabezpečené http spojenie (https). Http spojenie prenáša údaje od webového klienta na server bez akéhokoľvek kryptovania. HTTP protokol využíva nezabezpečený TCP protokol ako jeho transportný mechanizmus. Tento je nekryptovaný, a teda môže byť zachytený útočníkmi. HTTPS je v základe ten istý protokol aplikačnej vrstvy ako HTTP, ale je tunelovaný cez bezpečný transportný mechanizmus. [3]

System predpokladá zabezpečenie servera štandardnými prostriedkami. Toto má na starosti technický administrátor systému, prípadne administrátor servera.

System využíva priamo e-mailovú komunikáciu iba pri potvrdení registrácie a overovaní e-mailovej adresy registrovaného používateľa. V žiadnom e-maile nemôžu byť uvedené citlivé údaje ako napríklad heslo, osobné údaje žiadateľa (okrem mena a priezviska) a pod.

# 5 Návrh systému na podporu ubytovacieho procesu

V tejto kapitole popíšeme technické aspekty implementovaného systému, spôsob akým sa konfiguruje a tiež podrobne sformulujeme požiadavky na import údajov z externých systémov.

## 5.1 Štruktúra databázy

Databáza sa skladá z tridsiatich tabuliek, ktorých štruktúru a význam si popíšeme v tejto sekcii. Štruktúra databázy je zachytená aj na diagrame, ktorý je na konci tejto časti (Obr. 4).

### *tblAdmins*

Táto tabuľka obsahuje zoznam administrátorov. Administrátor je definovaný prihlasovacím menom a heslom. Stĺpec `password` obsahuje heslo zahašované jednosmernou funkciou `md5`.

Záznamy v tabuľke je možné pridávať, upravovať a mazať vo webovej aplikácii. Stĺpec `is_superuser` určuje, či je administrátor superadminom, čo znamená, že ho nie je možné zmazať priamo vo webovej aplikácii.

Štruktúra: `admin_id (int), login (varchar), password (varchar), is_superuser (tinyint)`

### *tblApplicationsListFilters*

Táto tabuľka obsahuje zoznam filtrov žiadostí. Filter je definovaný názvom, SQL požiadavkou a číslom, ktoré určuje jeho pozíciu v zozname. Stĺpec `sql_query` obsahuje

štandardnú SQL požiadavku, ktorá musí vrátiť zoznam identifikátorov používateľov z tabuľky `tblUsers`. SQL požiadavka môže obsahovať reťazec `%%FACULTY_ID%%`, ktorý aplikácia dokáže nahradiť za identifikátor fakulty, ku ktorej má povolený prístup prihlásený administrátor. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `filter_id (int), name (varchar), sql_query (text), order_no (int)`

#### *tblDormitories*

Táto tabuľka obsahuje zoznam internátov. Internát je definovaný menom a skratkou. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), name (varchar), abbreviation (varchar)`

#### *tblLocalities*

Táto tabuľka obsahuje zoznam lokalít internátov. Lokalita je definovaná menom, skratkou a identifikátorom internátu, ku ktorému patrí. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), name (varchar), abbreviation (varchar), dormitory_id (int)`

Väzby: `dormitory_id => tblDormitories.id`

#### *tblBlocks*

Táto tabuľka obsahuje zoznam blokov na lokalitách internátov. Lokalita je definovaná názvom a identifikátorom lokality, ku ktorej patrí. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), name (varchar), locality_id (int)`

Väzby: `locality_id => tblLocalities.id`

#### *tblRooms*

Táto tabuľka obsahuje zoznam izieb. Izba je definovaná číslom, kapacitou, typom, poschodím a identifikátorom bloku, na ktorom sa nachádza. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), number (int), capacity (int), type (enum 'bezne', 'internet'), floor (int), block_id (int)`

Väzby: `block_id => tblBlocks.id`

### *tblPlaces*

Táto tabuľka obsahuje zoznam miest. Miesto nemá okrem identifikátora žiadne stĺpce, ktoré by ho bližšie špecifikovali. K danej izbe môže existovať maximálne taký počet miest, aký je uvedený v stĺpci `capacity` záznamu danej izby. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `place_id (int), room_id (int)`

Väzby: `room_id => tblRooms.id`

### *tblFaculties*

Táto tabuľka obsahuje zoznam fakúlt. Fakulta je definovaná názvom a zaužívanou skratkou. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), name (varchar), abbreviation (varchar)`

### *tblGlobalMessages*

Táto tabuľka obsahuje zoznam informačných správ, ktoré sa zobrazujú pre používateľov na webe. Základné údaje správy sú nadpis, perex a text správy. Perex je označenie pre krátky text, ktorého účelom je uviesť a upútať pozornosť na nasledujúci text alebo naznačiť o čom text bude. Perex by mal mať dĺžku minimálne tri vety. [4]

Stĺpec `publish` môže nadobúdať hodnoty určuje, či sa správa zobrazuje len určité obdobie od času `publish_date_from` po čas `publish_date_till` alebo stále bez ohľadu na čas.

Stĺpec `message_tag` sa slúži pre špeciálne správy, ktoré sa používajú na popis niektorých obrazoviek webovej aplikácie. V aplikácii sa na takéto správy referuje práve cez tento textový tag.

Správa môže mať určené, aby sa pod jej textom zobrazoval jeden zo zoznamov (zoznam miest, poradovníky študentov, ai.). Zoznam, ktorý sa má zobrazit' je identifikovaný stĺpcom `list_tag`.

Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor pomocou webovej aplikácie.

Štruktúra: message\_id (int), subject (varchar), perex (varchar), body (text), url (varchar), creation\_date (datetime), publish\_date\_from (datetime), publish\_date\_till (datetime), publish (enum 'temporary', 'enum'), message\_tag (varchar), list\_tag (varchar), faculty\_id (int)

Väzby: faculty\_id => tblFaculties.id

### *tblModifierTypes*

Táto tabuľka obsahuje zoznam typov modifikátorov. Typ modifikátora je detailne popísaný v časti 3.1.8 Modifikátor.

Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: modifier\_type\_id (int), name (varchar), value (double), computation (enum 'exponencial', 'multiplication'), limit (double), type (enum 'positive', 'negative'), faculty\_id (int)

Väzby: faculty\_id => tblFaculties.id

### *tblProperties*

Táto tabuľka obsahuje register nastavení aplikácie. Nastavenie je definované textovým identifikátorom, hodnotou a určením, či je nastavenie platne globálne pre celú aplikáciu alebo len pre niektorú z fakúlt.

Nastavenia fakúlt do tabuľky pridáva, upravuje a vymazáva administrátor pomocou webovej aplikácie. Globálne nastavenia určuje technický administrátor.

Štruktúra: property\_tag (varchar), property\_value (varchar), property\_type (enum 'faculty, global'), entity\_id (int)

Väzby: entity\_id => tblFaculties.id

### *tblRecordTypes*

Táto tabuľka obsahuje zoznam typov záznamov. Typ záznamu je detailne popísaný v kapitole 3. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: record\_type\_id, record\_type\_tag (varchar), list\_positions (enum 'top', 'bottom', 'computed', 'removed'), group\_position (int), use\_points (enum 'yes', 'no'), prioritu\_no (int), faculty\_id (int)

Väzby: faculty\_id => tblFaculties.id

### *tblSeasons*

Táto tabuľka obsahuje zoznam období (akademických rokov). Obdobie je definované názvom. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: id (int), name (varchar)

#### *tblSections*

Táto tabuľka obsahuje zoznam sekcií administračného rozhrania. Sekcia predstavuje činnosť definovanú v kapitole 3. Sekcia je definovaná jej textovým identifikátorom, názvom a číslom, ktoré určuje zoradenie. Záznamy do tabuľky pridáva, upravuje a vymazáva programátor webovej aplikácie.

Štruktúra: section\_id (int), section\_tag (varchar), section\_name (varchar), order\_no (int)

#### *tblStudents*

Táto tabuľka obsahuje zoznam oprávnených osôb. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor pri importe.

Štruktúra: student\_id (varchar), title (varchar), name (varchar), surname (varchar), address (varchar), city (varchar), psc (varchar), type (int), year (int), section (int), distance (double), duration (double), sex (int), rating (double), ISIC (varchar), ISIC\_other (varchar), rating1 (double), rating2 (double), rating3 (double), rating\_profile1 (double), rating\_profile2 (double), rating\_profile3 (double), birth\_number (varchar), student\_type (enum 'new', 'present'), faculty\_id (int)

Väzby: type => tblStudyTypes.id, section => tblStudySections.section\_id , faculty\_id => tblFaculties.id

#### *tblStudentHistory*

Táto tabuľka obsahuje zoznam záznamov o študijnej histórii žiadateľa. Záznam je definovaný textovým identifikátorom udalosti (napr. „absolvoval“, „zanechal\_studium“, „vylucenie“, a pod.), začiatočným a koncovým rokom.

Štruktúra: student\_id (varchar), history\_action (varchar), start\_year (int), end\_year (int)

Väzby: student\_id => tblStudents.student\_id

#### *tblStudentsSections*

Táto tabuľka obsahuje zoznam študijných odborov. Študijný odbor je definovaný jeho kódom z univerzitného informačného systému, skratkou a názvom. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `section_id (int), code (varchar), abbreviation (varchar), name (varchar), faculty_id (int)`

Väzby: `faculty_id => tblFaculties.id`

#### *tblStudyTypes*

Táto tabuľka obsahuje zoznam typov štúdia. Typ štúdia je definovaný textovým identifikátorom a názvom. Záznamy do tabuľky pridáva, upravuje a vymazáva technický administrátor.

Štruktúra: `id (int), faculty_id (int), tag (varchar), name (varchar)`

Väzby: `faculty_id => tblFaculties.id`

#### *tblUserApplications*

Táto tabuľka obsahuje zoznam žiadostí. Záznamy do tabuľky pridáva a upravuje žiadateľ cez webovú aplikáciu.

Štruktúra: `user_id (int), student_id (varchar), title (varchar), name (varchar), surname (varchar), sex (enum 'male', 'female'), address (varchar), city (varchar), psc (varchar), type (int), year (int), section (int), smoking (enum 'yes', 'no'), distance (double), duration (double), message (text), sameroom (enum 'yes', 'no'), status (enum 'undecided', 'successful', 'unsuccessful'), faculty_id (int)`

Väzby: `faculty_id => tblFaculties.id, student_id => tblStudents.id, type => tblStudyTypes.id, section => tblStudySections.section_id`

#### *tblUsers*

Táto tabuľka obsahuje zoznam zaregistrovaných používateľov. Pri registrácii sa overuje e-mailová adresa, preto sa pre každého používateľa vygeneruje overovací reťazec (`verification_hash`), ktorý sa pošle e-mailom ako súčasť odkazu, na ktorý musí používateľ kliknúť. Záznamy do tabuľky pridáva a upravuje žiadateľ cez webovú aplikáciu.



Štruktúra: `user_id (int), login (varchar), password (varchar), e_mail_address (varchar), verified (enum 'yes', 'no'), verification_date (datetime), verification_hash (varchar)`

#### *tblCustomLists*

Táto tabuľka obsahuje zoznam manuálne definovaných zoznamov (tagov). Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: `list_id (int), name (varchar), faculty_id (int)`

Väzby: `faculty_id => tblFaculties.id`

#### *tblAdminsEntitiesRights*

Táto tabuľka viaže tabuľky `tblAdmins` a jednu z tabuliek `tblFaculties` alebo `tblDormitories`, a tak umožňuje zaznamenať, ku ktorej fakulte alebo internátu má daný administrátor prístup. Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: `admin_id (int), entity_type (enum 'faculty', 'dormitory'), entity_id (int)`

Väzby: `entity_id => tblFaculties.id, entity_id => tblDormitories.id, admin_id => tblAdmins.admin_id`

#### *tblAdminsRights*

Táto tabuľka viaže tabuľky `tblAdmins` a tabuľku `tblSections`, a tak umožňuje zaznamenať, ktorú činnosť môže daný administrátor vykonávať. Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: `admin_id (int), section_id (int)`

Väzby: `faculty_id => tblFaculties.id, section_id => tblSections.id`

#### *tblApplicationsInCustomLists*

Táto tabuľka viaže tabuľky `tblCustomLists` a tabuľku `tblUsers`, a tak umožňuje evidovať žiadosti používateľov, ktoré sú zaradené do manuálneho zoznamu (sú ním otagované). Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: `list_id (int), user_id (int)`

Väzby: user\_id => tblUsers.user\_id, list\_id => tblCustomLists.list\_id

#### *tblApplicationsList*

Táto tabuľka slúži na uloženie aktuálneho podzoznamu žiadateľov určeného buď manuálne definovaným zoznamom alebo zvoleným preddefinovaným filtrom. Ukladanie sa robí preto, aby sa pri každej akcii pri vykonávaní činnosti „Činnosť: Pridelovanie miest úspešným žiadateľom“ nemusel vykonávať komplikovanú SQL požiadavku definovanú filtrom.

Štruktúra: order\_no (int), user\_id (int), faculty\_id (int)

Väzby: faculty\_id => tblFaculties.id, user\_id => tblUsers.user\_id

#### *tblAssignedPlaces*

Táto tabuľka zaznamenáva pridelenie miest fakultám a žiadateľom. V tabuľke sú archivované pridelenia pre jednotlivé obdobia. Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: place\_id (int), season\_id (int), faculty\_id (int), user\_id (int), date (datetime), note (text)

Väzby: faculty\_id => tblFaculties.id, place\_id => tblPlaces.place\_id, season\_id => tblSeasons.id, user\_id => tblUsers.user\_id

#### *tblUserApplicationsRecords*

Táto tabuľka obsahuje záznamy k žiadosti používateľa. Záznam k žiadosti je detailne popísaný v časti 3.1.7 Záznam k žiadosti. Záznamy do tabuľky pridáva, upravuje a vymazáva administrátor cez webovú aplikáciu.

Štruktúra: record\_id (int), user\_id (int), record\_type (varchar), comment (text), modifier\_type (int), modifier\_count (int)

Väzby: user\_id => tblUsers.user\_id, record\_type => tblRecordsTypes.record\_type\_tag

#### *tblUsersFriends*

Táto tabuľka obsahuje zoznam kamarátov, ktorých si používateľ vybral. Stĺpec mutual označuje, či sa používatelia pridali navzájom ako kamaráti. Záznamy do tabuľky pridáva, upravuje a vymazáva žiadateľ cez webovú aplikáciu.

Štruktúra: friend\_id (int), user\_id (int), login (varchar), description (text), mutual (enum 'yes', 'no')

Väzby: user\_id => tblUsers.user\_id

#### *tblUsersPlaces*

Táto tabuľka obsahuje zoznam lokalít, blokov alebo izieb, ktoré používateľ uviedol, že na nich chce bývať. Stĺpec flag označuje, či používateľ mal vybranú konkrétnu izbu pridelenú aj aktuálny akademický rok. Stĺpec is\_oneplace označuje, či chce používateľ bývať na izbe s jednotkou. Stĺpec has\_internet označuje, či chce používateľ bývať na izbe s internetom. Stĺpce location, block a room majú buď hodnotu nula, čo znamená, že používateľ si nevybral konkrétnu lokalitu, blok alebo izbu, prípadne majú hodnotu identifikátora daných entít z tabuliek tblLocalities, tblBlocks alebo tblRooms.

Záznamy do tabuľky pridáva, upravuje a vymazáva žiadateľ cez webovú aplikáciu.

Štruktúra: place\_id (int), user\_id (int), description (text), location (int), block (int), room (int), has\_internet (enum 'yes', 'no'), is\_oneplace (enum 'yes', 'no'), order\_no (int), flag (varchar)

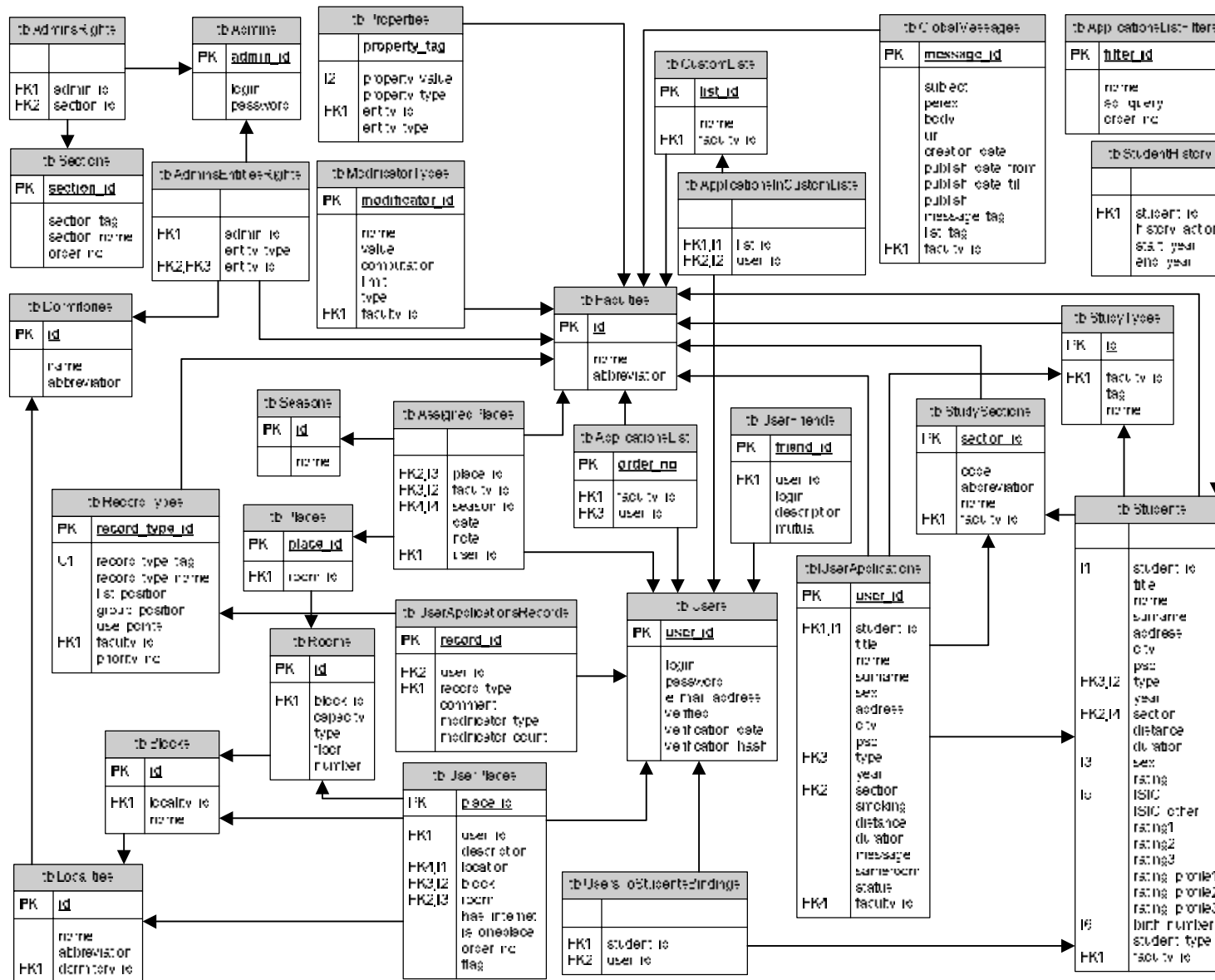
Väzby: user\_id => tblUsers.user\_id, location => tblLocalities.id, block => tblBlocks.id, room => tblRooms.id

#### *tblUsersToStudentsBindings*

Táto tabuľka obsahuje väzby medzi oprávnenými osobami z tabuľky tblStudents a registrovanými používateľmi z tabuľky tblUsers. Pri registrácii používateľa sa skontroluje, či sa jeho student\_id už nachádza v tejto tabuľke. Ak áno, tak sa mu prideli príslušné user\_id. Ak nie, tak sa do tejto tabuľky vloží nový záznam, a vytvorí sa nové user\_id.

Štruktúra: student\_id (varchar), user\_id (int)

Väzby: user\_id => tblUsers.user\_id, student\_id => tblStudents.student\_id



Obr. 4: Štruktúra databázy

## 5.2 Implementácia

Pri vývoji aplikácie sme veľký dôraz kládli na použiteľnosť. Všetky vizuálne prvky sú dosiahnuté použitím kaskádových štýlov. Štruktúra HTML kódu je navrhnutá tak, aby najdôležitejšie veci boli v kóde ako prvé. Tým sa dosiahlo, že aplikácia je pohodlne použiteľná aj na zariadeniach, ktoré majú obmedzenú schopnosť interpretácie farieb. Aplikácia tiež bola vyvíjaná s ohľadom na zrakovo postihnutých používateľov [5].

Ďalším dôležitým aspektom bol návrh formulárov. Nestarostlivo navrhnuté formuláre je jedna z oblastí, ktorá znepríjemňuje používateľom používanie aplikácie. Ak sú jednotlivé položky formulára rozhádzané a neorganizované, je ťažké na prvý pohľad usúdiť, ktorý popis patrí ktorej položke. [6]

Aplikácia je implementovaná v jazyku PHP 4 [7]. Adresárová štruktúra vyzerá nasledovne:

- `includes/`
  - `xtemplate/`
  - `Spreadsheet/`
- `config/`
  - `modifiers/`
  - `ratings/`
  - `records/`
- `out/`
  - `css/`
  - `images/`
  - `tpl/`

V adresári `includes` sú základné obslužné rutiny pre databázu, `session` a konfiguračné rozhranie a ďalej triedy a funkcie implementujúce samotnú aplikáciu. V podadresári `xtemplate` je knižnica šablónovacieho systému `Xtemplate` a v podadresári `Spreadsheet` je knižnica slúžiaca na generovanie excelovských dokumentov.

V adresári `config` a v jeho podadresároch sa nachádzajú konfiguračné nastavenia systému pre jednotlivé fakulty. Spôsob konfigurácie popíšeme v ďalšom texte.

V adresári `out` sa nachádzajú všetky zdroje, ktoré slúžia na vizualizáciu. Medzi ne patria súbory kaskádových štýlov (CSS), obrázky a HTML šablóny používané systémom Xtemplate.

### 5.2.1 Šablónovací systém XTemplate

Šablónovací systém XTemplate umožňuje oddeliť aplikačnú vrstvu od prezentačnej tým, že HTML dokumenty cez ktoré je webová aplikácia vizualizovaná sú umiestnené do samostatných súborov (šablón), ktoré sú v aplikačnom kóde obsluhované práve pomocou knižnice XTemplate. [8]

XTemplate rozdeľuje HTML šablónu do hierarchicky usporiadaných blokov. Druhá základná riadiaca štruktúra je premenná, ktorú môže použiť na ľubovoľnom mieste šablóny. Generovanie výslednej stránky pozostáva z pridelovania hodnôt jednotlivým premenným a generovania (aj viacnásobného) jednotlivých blokov.

Na krátkom príklade predstavíme ako sa pracuje s knižnicou XTemplate. Pri vytvorení inštancie objektu XTemplate zvolíme, ktorú šablónu chceme použiť.

```
$page = new XTemplate('out/tpl/example.tpl');
```

Šablóna je `example.tpl` definovaná takto:

```
<!-- BEGIN: main -->
  <h1>Nadpis</h1>
  <h2>{PODNADPIS}</h2>

  <!-- BEGIN: zoznam -->
    <p>Zoznam</p>
    <!-- BEGIN: polozka_zoznamu -->
      {POLOZKA.meno} {POLOZKA.cislo}<br />
    <!-- END: polozka_zoznamu -->
  <!-- END: zoznam -->
<!-- END: main -->
```

V aplikácii priradíme hodnoty premenným `PODNADPIS` a `POLOZKA`. Premenná `$polozka` je pole s textovými indexami, ku ktorým ako vidíme môžeme v šablóne pristupovať cez bodku (`POLOZKA.meno`).

Kód generujúci HTML kód z danej šablóny môže vyzeráť napríklad takto:

```
$page->assign('PODNADPIS', 'Ahoj svet!');

$polozky = array(
    array('meno' => 'Filip', 'cislo' => 13),
    array('meno' => 'Judita', 'cislo' => 28)
);

foreach ($polozky as $polozka)
{
    $page->assign('POLOZKA', $polozka);
    $page->parse('main.zoznam.polozka_zoznamu');
}

$page->parse('main.zoznam');
$page->parse('main');

$page->out('main');
```

Jednotlivé bloky generujeme v poradí od najhlbšie vnoreného po koreňový. Blok `main.zoznam.polozka_zoznamu` sme vygenerovali viackrát. Posledné volanie metódy `out` vráti na štandardný výstup vygenerovaný HTML kód:

```
<h1>Nadpis</h1>
<h2>Ahoj svet!</h2>

    <p>Zoznam</p>
        Filip 13<br />
        Judita 28<br />
```

## 5.2.2 Generátor excelovských dokumentov Spreadsheet\_Excel\_Writer

Na generovanie excelovských dokumentov, ktoré slúžia ako výstup pre rôzne zoznamy používame knižnicu `Spreadsheet_Excel_Writer` z balíka `PEAR`. [9]

Knižnica umožňuje vykonávať základné operácie s excelovskou tabuľkou: nastavenie šírky stĺpca, spojenie viacerých buniek do jednej, nastavenie formátovania bunky, nastavenie obsahu bunky, nastavenie otočenia strany, ai.

Na jednoduchom príklade ukážeme vygenerovanie excelovského súboru.

```
// Vytvorenie instancie dokumentu
$xmls =& new Spreadsheet_Excel_Writer();

// Nastanie mena suboru
$xmls->send('example.xls');

// Vytvorenie sheetu
$sheet =& $xmls->addWorksheet('Ahoj svet!');

// Nastavenie sirky stlpcov ([prvy stlpec], [posledny stlpec], [sirka])
$sheet->setColumn(0, 3, 4);

// Nastavenie otocenia strany
$sheet->setLandscape();

// Vytvorenie instancie formatovania
$formatovanie =& $xmls->addFormat();

// Nastavenie velkosti pisma
$formatovanie->setSize(14);

// Nastavenie zarovnavania
$formatovanie->setHAlign('center');

// Nastavenie "tucneho" textu
$formatovanie->setBold();

// Nastavenie farby okraja
$formatovanie->setBorderColor('silver');

// Nastavenie hrubky ciary okraja
$formatovanie->setBorder(1);

// Spojenie viacerych buniek do jednej ([prvy riadok], [prvy stlpec],
[posledny riadok], [posledny stlpec])
$sheet->mergeCells(0, 0, 0, 11);

// Zapisanie obsahu do bunky ([riadok], [stlpec], [obsah], [formatovaci
objekt])
```



```
$sheet->write(2, 1, 'Ahoj svet!', $formatovanie);  
  
// Vytvorenie dokumentu a jeho odoslanie na standardny vystup  
$xls->close();
```

## 5.3 Konfigurácia

Základná konfigurácia aplikácie pozostáva z nastavenia prístupových údajov pre pripojenie k databázovému serveru. Tieto nastavenia sú v súbore `_setup.php`, ktorý sa nachádza v koreňovom adresári aplikácie.

```
$db_host = "localhost";           // database server host  
$db_port = 3306;                 // database server port  
$db_name = "nazov_db";          // database name  
$db_login = "login";            // login name to database  
$db_password = "heslo";        // database account password
```

V súbore `_setup.php` je ešte možné nastaviť, či má aplikácia vyžadovať beh cez zabezpečené spojenie (https). V prípade, že server nepodporuje https je možné toto nastavenie vypnúť. Odporúčame však nechať toto nastavenie vždy zapnuté a zabezpečiť server s podporou https.

```
define('FORCE_HTTPS',           true);
```

### 5.3.1 Konfigurácia typov záznamov a typov modifikátorov

Záznamy a modifikátory sú podrobne popísané v kapitole Funkčná.

Pri pridávaní záznamu k žiadosti si administrátor vyberá zo zoznamu typov záznamov. Tento zoznam je uložený v databázovej tabuľke `tblRecordTypes`. Pred začatím vyhodnocovania žiadostí musí technický administrátor do tejto tabuľky vložiť záznamy, ktoré zodpovedajú dohodnutým typom záznamom. Rovnaká situácia nastáva pri definovaní typov modifikátorov, ktoré sú uložené v tabuľke `tblModifierTypes`.

V prípade, že fakulta nemá žiadne špeciálne kritéria na zostavovane poradovníka a vystačí si s jednoduchým vzorcom, nie je potrebné definovať žiadne typy záznamov ani typy modifikátorov.

V stručnosti pripomenieme, že aplikácia umožňuje okrem manuálneho pridávania záznamov a modifikátorov pre danú žiadosť aj automatické vygenerovanie záznamov a modifikátorov podľa stanovených kritérií.

Aplikácia podporuje podávanie a vybavovanie žiadostí pre viac fakúlt. Keďže každá fakulta môže mať definované iné kritéria pre vyhodnocovanie žiadostí bolo potrebné zamyslieť sa nad spôsobom ako umožniť jednoducho konfigurovať tieto kritéria.

V nasledujúcich troch častiach popíšeme zmysel a vytváranie konfiguračných tried pre jednotlivé fakulty. Triedy sú ukladané do samostatných súborov v jednom z troch podadresárov adresára `config`. V adresári `config` sa nachádza súbor `main.php`, ktorého štruktúra je takáto:

```
$config = array(
    [faculty_id] => array(
        'modifiers_include' => [path to the modifiers class],
        'modifiers_class' => [name of the modifiers class],
        'records_include' => [path to the records class],
        'records_class' => [name of the records class]
        'generic_records_description' => [description text],
        'rating_include' => [path to the rating class],
        'rating_class' => [name of the rating class]
    ),
    ...
);
```

### 5.3.2 Konfigurácia automatického generovania záznamov

Automatické generovanie záznamov sa konfiguruje zadaním triedy, ktorá implementuje metódu `generateGenericRecords`. Zadaná trieda musí rozširovať triedu `Records`. V metóde je dostupná vlastnosť `$this->actualUser`, ktorá obsahuje pole s údajmi žiadateľa. Záznam sa vytvorí zavolaním metódy `$this->saveRecord(record_type_tag, description)`.

Zadefinujme napríklad tieto typy záznamov (nedefinujeme všetky atribúty):

<code>record_type_id</code>	<code>record_type_tag</code>	<code>record_type_name</code>
1	doktorand	Doktorand
2	vseobecny	Všeobecný záznam

Ďalej si zadefinujeme triedu `FMFI_Records`, nasledovne:

```
class FMFI_Records extends Records
{
    function generateGenericRecords()
    {
        if ($this->actualUser['type'] == 3)
        {
            $this->saveRecord('doktorand', 'Doktorand');
        }
    }
}
```

Po spustení automatického generovania záznamov sa pre žiadateľov, ktorí majú typ štúdia s identifikátorom 3 (doktorand) vytvorí typ záznamu „doktorand“. Tento záznam ovplyvní umiestnenie žiadateľov v poradovníku podľa toho akým spôsobom je definovaný (viac v 3.1.7 Záznam k žiadosti). Takýmto spôsobom je možné vygenerovať ľubovoľné záznamy na základe ľubovoľného algoritmu.

Okrem údajov v poli `$this->actualUser` je možné využiť aj záznamy o študijnej histórii žiadateľa. Spôsob akým je možné to spraviť je popísaný v nasledujúcej časti.

Konfiguračné triedy ukladáme do samostatných súborov do adresára `config/records`. Menná konvencia názvov súborov je `[skratka fakulty]_records.class.php`. Menná konvencia názvov tried je `[skratka fakulty]_Recods`.

### 5.3.3 Konfigurácia automatického generovania modifikátorov

Automatické generovanie modifikátorov sa konfiguruje zadefinovaním triedy, ktorá implementuje metódu `generateGenericModifiers`. Zadefinovaná trieda musí rozširovať triedu `Modifiers`. V metóde je dostupná vlastnosť `$this->actualUser`, ktorá obsahuje pole s údajmi žiadateľa. Modifikátor sa vytvorí zavolaním metódy `$this->saveModifierRecord(record_type_tag, description, $modifier_type_id, $modifier_count)`.

Z popisu modifikátora vyplýva, že je vždy viazaný na záznam. Predpokladajme existenciu typov záznamov z časti 5.3.2 Konfigurácia automatického generovania záznamov. Zadefinujme tento typ modifikátora (nedefinujeme všetky atribúty):

modifier_type_id	name	value	computation	type
1	absolvoval	450	multiplication	negative

Modifikátor vyjadruje fakt, že ak daný žiadateľ už v minulosti x krát úspešne absolvoval štúdium na VŠ, k bodovému zisku jeho žiadosti sa pripočíta  $4,5 * x$  bodov.

Ďalej si zdefinujeme triedu `FMFI_Modifiers`, ktorá implementuje algoritmus popísaný v predchádzajúcom odstavci:

```
class FMFI_Modifiers extends Modifiers
{
    function generateGenericModifiers()
    {
        $student_id = $this->actualUser['student_id'];

        $history = new History($this->db);
        $historyRecords = $history->getHistory($student_id);

        $gc = 0;

        foreach ($historyRecords as $hr)
        {
            if ($hr['history_action'] == 'absolvent_mgr')
            {
                $gc++;
            }
        }

        $this->saveModifierRecord(
            'vseobecny',
            'Absolvoval ' . $gc . 'krat',
            1,
            $gc
        );
    }
}
```

Konfiguračné triedy ukladáme do samostatných súborov do adresára `config/modifiers`. Menná konvencia názvov súborov je [skratka

fakulty]\_modifiers.class.php. Menná konvencia názvov tried je [skratka fakulty]\_Modifiers.

Metóda `getHistory` triedy `History` vráti pre daného študenta pole záznamov, ktorých štruktúra je nasledovná:

- `history_action` – textový identifikátor udalosti (napr. „absolvoval“, „zanechal\_studium“, „vylucenie“, a pod.)
- `start_year`
- `end_year`

### 5.3.4 Konfigurácia výpočtu bodového zisku žiadosti

Bodový zisk žiadosti je popísaný v časti 3.1.6 Bodový zisk žiadosti. Vzorec na výpočet bodového zisku žiadosti sa definuje vytvorením triedy, ktorá rozširuje triedu `Rating` a implementuje metódu `getValue($application, $type)`, kde `$application` je pole so všetkými údajmi žiadosti a `$type` určuje, či ide o súčasného alebo novoprijatého študenta (možné hodnoty sú „new“ a „present“).

Nasledujúca trieda je ukážkou výpočtu bodového zisku žiadosti pre súčasných študentov na FMFI UK.

```
class FMFI_Rating extends Rating
{
    function getValue($application, $type)
    {
        $a = $application;

        switch ($type)
        {
            case 'present' :
                if ($a['type'] == 3 ||
                    $a['type'] == 6)
                {
                    $result = 0;
                }
                else
                {
```

```

        $d = $a['distance'];
        $result = $a['rating']
            + round(
                3 / ((0.0004
                    * ($d * $d))
                    + (0.6 * $d) + 3),
                3
            );
    }
    break;
}
}
}
}

```

Konfiguračné triedy ukladáme do samostatných súborov do adresára `config/ratings`. Menná konvencia názvov súborov je `[skratka fakulty]_rating.class.php`. Menná konvencia názvov tried je `[skratka fakulty]_Rating`.

### 5.3.5 Import histórie žiadateľov

Import študijnej histórie je popísaný v časti 3.3.2 Činnosť: Import študijnej histórie. Formát vstupných údajov pre import histórie vychádza z formátu , ktorý autor práce dostal z Univerzity Komenského. Tento formát je však možné použiť pre ľubovoľnú univerzitu alebo fakultu.

Vstupné údaje sú v textovom súbore, v ktorom každý záznam je na samostatnom riadku. Jeden záznam sa skladá z identifikátora študenta v externom informačnom systéme a z údaje o študijnej histórii. Riadok vyzerá takto: “[id študenta]”, “[údaj o histórii]”. Údaj o histórii sa interpretuje nasledovne:

"2001..." "2002..." "2003..." "2004"	Študent aktuálne študuje. Číslo udáva rok nástupu na štúdium, každá bodka ďalší rok štúdia.
"1996...2000A"	Úspešné ukončenie štúdia (A = absolvent Mgr. štúdia)
"1998.2000B 2001 2002A"	Úspešné ukončenie štúdia (B = absolvent Bc. štúdia)
"2004Z"	Neúspešný pokus o štúdium: zanechal v prvom roku
"2003 2004V"	Neúspešný pokus o štúdium: vylúčený po druhom roku
"2001X"	Presup na inú školu po prvom roku
"2000."	Po druhom roku sa už nezapísal, prípadne nenastúpil po prerušení

"2001Z 2002.."	V minulosti nedošťudoval a aktuálne opäť študuje
"2001V 2002.. "	V minulosti nedošťudoval a aktuálne opäť študuje

## 5.4 Import údajov do základných číselníkov

Základné číselníky sú databázové tabuľky, ktoré obsahujú údaje potrebné na podávanie, vyhodnocovanie a vybavovanie žiadostí a na pridelovanie miest pre jednotlivé fakulty.

- tblFaculties
- tblDormitories
- tblLocalities
- tblBlocks
- tblRooms
- tblPlaces
- tblSeasons
  - pred spustením podávania žiadostí na nasledujúci akademický rok je potrebné pridať záznam o novom akademickom roku a v tabuľke tblProperties nastaviť správne id daného akademického roka (vlastnosť „active\_season\_id“)
- tblStudySections
- tblStudyTypes
- tblStudents

## 5.5 Nasadenie aplikácie

Ako vyplýva z technických požiadaviek, aplikácia ku svojej činnosti potrebuje webový server s podporou PHP 4 a databázový server MySQL 4.

Pri prvom nasadení je potrebné aplikáciu nainštalovať:

1. nahrať skripty na server
2. vytvoriť štruktúru databázy (spustením SQL skriptu v súbore structure.sql, ktorý sa nachádza v koreňovom adresári)

- po vytvorení databázovej štruktúry je potrebné z bezpečnostných dôvodov zmazať súbor `structure.sql` zo servera
- vo vytvorenej databázovej štruktúre existuje administrátor s prihlasovacím meno „admin“ a prihlasovacím heslom „admin“. Po prvom prihlásení je potrebné zmeniť heslo tomuto administrátorovi.

Ďalší postup sa opakuje vždy pred spustením podávania žiadostí na nasledujúci akademický rok.

1. naimportovanie údajov do základných číselníkov, prípade zmena údajov, ak napríklad v systéme pribudne nová fakulta, internát a pod. (pozri časť 5.4 Import údajov do základných číselníkov)
2. nakonfigurovanie aplikácie (pozri časť 5.3 Konfigurácia)
3. definovanie administrátorov a ich prístupových práv

Ak ide byť aplikácia použitá druhý a ďalší raz, technický administrátor musí najprv zmazať nepotrebné údaje z predchádzajúceho akademického roka.

Zmazať je potrebné záznamy z týchto tabuliek:

- `tblApplicationsInCustomLists`
- `tblApplicationsList`
- `tblCustomLists`
- `tblStudentsHistory`
- `tblStudents`
- `tblUserApplications`
- `tblUserApplicationsRecords`
- `tblUserFriends`
- `tblUserPlaces`
- `tblUsers`

## 5.6 Google Apps

Jeden z problémov, ktoré sme identifikovali pri súčasnej realizácii ubytovacieho procesu je problém s komunikáciou. Komunikáciou myslíme dynamickú komunikáciu



medzi jednotlivými osobami a statickú komunikáciu v podobe ukladania, vyhľadávania a používania dokumentov.

Na riešenie tohto problému navrhujeme použiť službu Google Apps od spoločnosti Google. Služba Google Apps umožňuje využívať viaceré aplikácie firmy Google na ľubovoľnej doméne. [10] Správca domény, na ktorej je služba nainštalovaná môže spravovať prístup používateľov k jej jednotlivým aplikáciám. Pre naše potreby sú dôležité aplikácie Gmail a Docs.

Gmail ako súčasť Google Apps poskytuje e-mailové rozhranie pre e-maily z danej domény a slúži teda ako výkonný e-mailový klient využívajúci vyhľadávaciu technológiu Google.

Docs je webový kancelársky balík zložený z textového a tabuľkového editora. Umožňuje jednoduché vytváranie a zdieľanie dokumentov a ich publikovanie na webe.

Pre potreby ubytovacieho procesu navrhujeme nainštalovať tento balík na vhodnej univerzitnej doméne. Každá osoba ubytovacieho procesu, ktorá vystupuje ako administrátor niektorej jeho časti, získa e-mailovú adresu, ktorej tvar bude definovaný konvenciou (napríklad „ubytovanie.[skratka fakulty]@uniba.sk” pre členov ubytovacích komisií). V prípade, že sa osoba prestane podieľať na administrácii niektorej časti ubytovacieho procesu, danú e-mailovú adresu bude môcť hneď začať používať iná osoba. Takto sa zaručí, že všetky informácie budú hneď dostupné a nebudú sa „strácať“. To isté platí aj pre vytvorené a zdieľané dokumenty.

## 6 Používateľský manuál

Cieľom tejto kapitoly je formou používateľského manuálu popísať implementovanú aplikáciu.

Aplikácia je členená na hornú časť, ľavý stĺpec a hlavný obsahový stĺpec. V hornej časti sa nachádza nadpis „Ubytovanie – [skratka fakulty]“ a logo aplikácie v pravom rohu.

### 6.1 Používateľská časť

Hlavná úvodná stránka webovej aplikácie zobrazuje zoznam všetkých fakúlt v systéme. Používateľ si jednu z nich vyberie a dostane sa na úvodnú stránku danej fakulty v rámci ubytovacieho systému.

Na stránke fakulty je v ľavom stĺpci formulár na prihlásenie a dva formuláre na registráciu súčasných a novoprijatých študentov. Administrátor môže určiť, ktoré formuláre sa zobrazujú, preto tam nemusia byť vždy zobrazené všetky tri. Na konci ľavého stĺpca je odkaz na hlavnú úvodnú stránku aplikácie.

V obsahovej časti stránky je zoznam článkov s ich krátkym popisom a pod zoznam sa zobrazuje úvodný text. Nadpisy článkov zobrazené v hornej časti zoznamu (nemajú uvedený dátum) obsahujú informácie, ktoré sú platné trvalo. Štandardne tu býva uverejnený postup ako sa zaregistrovať a podať žiadosť, či odpovede na najčastejšie otázky (FAQ). Pod týmto zoznamom sa nachádzajú články, ktorých platnosť je obmedzená (zoznamy, rôzne operatívne oznamy, a pod.).

#### 6.1.1 Registrácia žiadateľa

Ak chce študent alebo doktorand podať žiadosť musí sa najprv zaregistrovať. Pri registrácii je potrebné zadať prihlasovacie meno a heslo, e-mailovú adresu a buď rodné číslo alebo číslo ISIC karty študenta. Rodné číslo zadávajú novoprijatí študenti, ktorí ešte nemajú vystavené ISIC karty.

Pri registrácii sa overuje e-mailová adresa, aby ubytovacia komisia mohla žiadateľa v prípade potreby kontaktovať. Overenie prebieha tak, že na zadanú e-mailovú adresu systém odošle e-mail s linkou, na ktorú treba kliknúť. Až po overení e-mailovej adresy je možné prihlásiť sa a podať žiadosť.

Po prihlásení sa v ľavom stĺpci zobrazia štyri panely. V prvom paneli je zobrazené prihlasovacie meno používateľa a tlačidlo na odhlásenie. V druhom paneli sú základné údaje zo žiadosti používateľa. Tretí panel slúži na zobrazenie zoznamu miest, o ktoré žiadateľ prejavil záujem. Štvrtý panel slúži na zobrazenie zoznamu kamarátov, s ktorými chce žiadateľ bývať.

Prihláseny používateľ sa po odhlásení môže znovu prihlásiť a pokračovať v podávaní žiadosti a pridávaní kamarátov a miest. Administrátor určuje, či je k dispozícii prihlasovací formulár, a tým určuje obdobie počas, ktorého je možné meniť žiadosti a pridružené údaje.

### **6.1.2 Podanie a upravovanie žiadosti**

Po prvom prihlásení do aplikácie sa zobrazí výrazné upozornenie, že žiadosť ešte nebola podaná. Používateľ musí kliknúť na linku „Vyplniť žiadosť o ubytovanie“. Podanie žiadosti je rozdelené na tri kroky. V prvom kroku sa zobrazí formulár s predvyplnenými údajmi, ktoré žiadateľ len skontroluje a prípadne opraví. Po dokončení úprav prejde na ďalší krok kliknutím na tlačidlo „Uložiť a pokračovať“. Súčasný študent si môže vo formulári zvoliť, či chce bývať na tej istej izbe ako aktuálny akademický rok. Ak zvolí túto možnosť a dá uložiť žiadosť, zobrazí sa druhý krok, v ktorom musí zadať miesto, na ktorom býva aktuálny rok. Ak je tento údaj archivovaný v systéme, žiadateľ môže hneď prejsť na tretí krok, ktorým je oznam o úspešnom podaní žiadosti.



Obr. 4: Obrazovka po prvom prihlásení žiadateľa

Po vytvorení žiadosti sa v paneloch Miesta a Kamaráti zobrazia tlačidlá „Upraviť“. Po kliknutí na tlačidlo sa používateľ dostane do príslušného modulu, v ktorom môže upravovať vybrané miesta alebo kamarátov.

### 6.1.3 Výber miest

Po kliknutí na tlačidlo „Upraviť“ v paneli Miesta sa zobrazí zoznam vybraných miest a tlačidlo „Pridať miesto“. Žiadateľ si môže zvoliť maximálne päť miest, o ktoré má záujem. Miesta je možné v zozname posúvať smerom nadol a nahor, a tak určiť poradie ich priority. V zozname sa pre každé miesto zobrazuje jeho inteligentný popis a poznámka. Kliknutím na popis miesta alebo sa zobrazí formulár, v ktorom je možné ho zmeniť.

Po kliknutí na tlačidlo „Pridať miesto“ sa žiadateľovi pridá miesto s popisom „hociktoré miesto“ a zobrazí sa formulár, v ktorom je možné ho zmeniť. Zmena miesta spočíva v jeho postupnom špecifikovaní, ktoré prebieha v štyroch krokoch. Každá zmena sa hneď prejaví aj v zozname miest v paneli Miesta.

V prvom kroku si žiadateľ zvolí lokalitu, v ktorej chce mať miesto a klikne na tlačidlo „Krok 2 ->“. Ak si napríklad vyberie lokalitu „Výškové bloky“ v zoznam miest sa popis „hociktoré miesto“ zmení na „Výškové bloky, hociktoré miesto“.

V druhom kroku si žiadateľ zvolí blok v rámci zvolenej lokality a klikne na tlačidlo „Krok 3 ->“. Ak si napríklad vyberie blok A popis v zozname sa zmení na „VBA, hociktoré miesto“.

V treťom kroku si žiadateľ zvolí konkrétnu izbu v rámci zvoleného bloku a klikne na tlačidlo „Krok 4 ->“. Ak si vyberie napríklad izbu 104 popis v zozname sa zmení na „VBA 104“.

V každom z prvých troch krokov si používateľ môže zvoliť aj možnosť „všetky“. Tým dá najavo, že nemá záujem o konkrétnu izbu, ale len o konkrétnu lokalitu (ak pri voľbe bloku zvolí možnosť „všetky bloky“), prípadne len o konkrétny blok (ak pri voľbe izby zvolí možnosť „všetky izby“). Ak v niektorom kroku žiadateľ zvolí možnosť „všetky“ prejde hneď na štvrtý krok. Po kliknutí na tlačidlo „Pridať miesto“ sa teda do systému pridá miesto s lokalitou, blokom aj izbou nastavenou na možnosť „všetky“, čo vysvetľuje popis „hociktoré miesto“.

V štvrtom kroku môže žiadateľ napísať komentár k zvolenému miestu. Ak si nevybral konkrétnu izbu, môže zaškrtnúť, či si želá izbu s internetom alebo či má záujem o jednolôžkovú izbu. Každá izba má v systéme definované, či je s internetom a či je jednolôžková, čiže ak si vyberie konkrétnu izbu tieto voľby sa nastavujú automaticky a nedajú sa zmeniť. Inteligentný popis reflektuje aj tieto voľby. Ak má izba „VBA 104“ internet, tak jej popis bude „VBA 104 s internetom“. Ak je izba jednolôžková, tak jej popis bude „VBA 104, jednotka“. Ak si používateľ zvolí hociktoré miesto na VBA a vyjadří záujem o internet, tak popis miesta sa zmení na „VBA, hociktoré miesto s internetom“.

## Miesta

Môžete zadať až 5 miest, na ktorých by ste chceli bývať a zoradiť ich podľa priority.

Vyberte izbu	
1	<b>Internát:</b> Vyskové Bloky
2	<b>Blok:</b> A
3	<b>Izba:</b> všetky izby <input type="button" value="v"/> <input type="button" value="Krok 2"/> <input type="button" value="Krok 4 -&gt;"/>
4	<b>Ostatné údaje:</b> <input checked="" type="checkbox"/> chcem izbu s internetom <input type="checkbox"/> chcem jednolôžkovú izbu <b>Poznámka:</b> <input type="text"/>

Obr. 5: Špecifikovanie miesta

### 6.1.4 Výber kamarátov

Po kliknutí na tlačidlo „Upraviť“ v paneli Kamaráti sa zobrazí zoznam vybraných kamarátov a tlačidlo „Pridať kamaráta“. Kamarát je pri pridávaní identifikovaný prihlasovacím menom. Kliknutím na prihlasovacie meno kamaráta v zozname sa zobrazí formulár, v ktorom je možné napísať poznámku.

## 6.2 Administračná časť

Administračná časť aplikácie je dostupná na adrese [adresa aplikácie]/admin. V ľavom stĺpci administračnej časti sú dva panely. V prvom je zobrazené prihlasovacie meno aktuálne prihláseného administrátora a tlačidlo „Odhlásiť“. V druhom je zoznam činností, ku ktorým má administrátor oprávnenie. Nasledujúci text je členený podľa týchto činností.

### 6.2.1 Správa administrátorov

Po kliknutí na položku „Správa administrátorov“ sa zobrazí zoznam všetkých administrátorov v systéme a tlačidlo „Pridať administrátora“. Každému administrátorovi je možné zmeniť heslo a definovať prístupové práva. Ak administrátor v zozname nemá odkaz „Odstrániť“, znamená to, že je superadministrátorom, ktorého môže odstrániť iba technický administrátor priamym zápisom do databázy.

Obrazovka prístupových práv zobrazuje zoznam činností so zaškrávkami vedľa každej z nich. Právo k činnosti sa definuje zaškrtnutím jej zaškrávky. Na tejto obrazovke je možné určiť aj prístupové právo k fakulte alebo internátu, výberom zo zoznamu. Administrátor nemusí mať prístupové právo k žiadnej fakulte ani žiadnemu internátu. Ak činnosť vyžaduje, aby mal administrátor právo k nejakej fakulte a pokúša sa ju vykonávať administrátor bez práva k žiadnej fakulte, tak sa vypíše chybový oznam.

### **6.2.2 Informačné správy**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Informačné správy“ sa zobrazí zoznam všetkých správ pre danú fakultu a tlačidlo „Vytvoriť novú správu“. Na začiatku zoznamu je sedem správ, ktoré majú zelenú farbu pozadia a nedajú sa zmazať. Tieto správy sú špeciálne systémové správy, ktoré sa zobrazujú napríklad pri jednotlivých krokoch registrácie, nad formulárom na špecifikovanie miesta, a pod. Na obrazovke detailov správy sa zobrazuje formulár s jednotlivými atribútmi.

Ak má správa zadanú URL, tak po kliknutí na ňu v zozname v používateľskej časti aplikácie sa nezobrazí jej text, ale jej nadpis slúži ako odkaz na zadanú URL. Atribút „Publikovať“ určuje, či sa správa zobrazuje len počas nastaveného obdobia, alebo stále. Atribút „Perex“ slúži na krátke zhrnutie obsahu správy a v zozname sa zobrazuje pod nadpisom. Atribút „Zobrazíť zoznam“ určuje, či sa pod textom správy zobrazuje niektorý zo zoznamov. Na výber sú tri základné zoznamy: „poradovník súčasných študentov“, „poradovník novoprijatých študentov“ a „zoznam izieb“. Ak sú v systéme zadefinované nejaké tagy žiadostí, tak je možné zvoliť jeden z tagov a pod textom správy sa zobrazí zoznam žiadostí s daným tagom.

### **6.2.3 Import študijnej histórie**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Import študijnej histórie“ sa zobrazí formulár s textovým políčkom na zadanie posledného roku zaznamenaného v importovanej histórii a viacriadkové textové políčko na vloženie textu so záznamami o histórii študentov.

#### **6.2.4 Vygenerovanie záznamov**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Vygenerovanie záznamov“ sa zobrazí obrazovka s dvoma tlačidlami: „Vygenerovať pre súčasných študentov“ a „Vygenerovať pre novprijatých študentov“. Po kliknutí na jedno z tlačítek sa automaticky vygenerujú záznamy a modifikátory pre študentov danej fakulty.

#### **6.2.5 Pridelovanie miest fakultám**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakému internátu. Po kliknutí na položku „Pridelovanie miest fakultám“ sa zobrazí formulár, v ktorom je možné vybrať niektorú z lokalít daného internátu. Po výbere lokality sa zobrazí zoznam blokov danej lokality. Po výbere bloku sa zobrazí zvyšok formulára, v ktorom je možné zvoliť fakultu, ktorej ide administrátor pridelať miesta a zoznam izieb. V zozname izieb je pre každú izbu uvedené aká je jej celková a voľná kapacita. Ak je niektoré miesto na izbe priradené úspešnému žiadateľovi, tak nie je možné meniť fakultu, ktorá ho má priradené. V takom prípade je voľná kapacita menšia ako celková. Pre každú izbu sa tiež zobrazuje informácia o tom, ktorá fakulta má pridelené miesta na tejto izbe.

Po označení izby alebo viacerých izieb a kliknutí na tlačidlo „Miesta“ sa zobrazí druhý zoznam, ktorý obsahuje zoznam miest na označených izbách. Takto je možné priradiť fakulte len niektoré miesta na izbe.

Miesta na zvolených miestach, alebo priamo zvolené miesta sa priradia vybranej fakulte po kliknutí na tlačidlo „Nastaviť fakultu vybraným izbám (miestam)“.



### Prideľovanie miest fakultám

Tu je možné definovať a meniť priradenie miest fakultám.

Lokalita:	Vyskove Bloky
Blok:	A
Fakulta:	
Izby:	<ul style="list-style-type: none"><li>VB A 702 (2/2 - FMFI, FMFI)</li><li>VB A 705 (1/3 - FSEV)</li><li>VB A 717 (3/3 - FSEV, FSEV, FSEV)</li><li>VB A 718 (2/2 - FSEV, FSEV)</li><li>VB A 719 (3/3 - FSEV, FSEV, FSEV)</li><li>VB A 720 (2/2 - FSEV, FSEV)</li><li>VB A 721 (2/3 - FSEV, FSEV)</li><li>VB A 722 (2/2 - FSEV, FSEV)</li><li>VB A 723 (3/3 - FSEV, FSEV, FSEV)</li><li>VB A 724 (2/2 - FSEV, FSEV)</li><li>VB A 729 (2/3 - FSEV, FSEV)</li><li>VB A 812 (2/2 - FSEV, FSEV)</li><li>VB A 813 (1/3 - FSEV)</li><li>VB A 814 (2/2 - FSEV, FSEV)</li><li>VB A 815 (3/3 - FSEV, FSEV, FSEV)</li></ul>
Miesta:	<ul style="list-style-type: none"><li>VB A 717 (2007/2008 - FSEV)</li><li>VB A 717 (2007/2008 - FSEV)</li><li>VB A 717 (2007/2008 - FSEV)</li></ul>

Nastaviť fakultu vybraným izbám (miestam)

Obr. 6: Prideľovanie izieb a miest fakultám

### 6.2.6 Prideľovanie miest žiadateľom

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Prideľovanie miest žiadateľom“ sa zobrazí formulár, v ktorom je možné zvoliť jeden z preddefinovaných filtrov žiadostí, prípadne aj zoznam tagov žiadostí (Obr. 7). Žiadosti je možné filtrovať podľa pohlavia, podľa toho, či je žiadateľ súčasný alebo nový študent a či už má pridelené miesto. Štandardne sú žiadosti v zozname zoradené podľa priezviska žiadateľa. Zaškrtnutím zaškrtačadla „poradovník“ sa žiadosti zoradia podľa poradovníka, avšak spolu s týmto musí byť nastavené filtrovanie buď na súčasných študentov alebo na novoprijatých.

Pod filtrom sa nachádza odkaz „upraviť zoznamy“. Po kliknutí na tento odkaz sa zobrazí pod filtrom formulár na zedefinovanie nového tagu, prípadne zmazanie existujúceho.

### Prideľovanie miest žiadateľom

Tu je možné prideľovať miesta úspešným žiadateľom o ubytovanie. Žiadateľová žiadosť musí byť označená ako „úspešná“, aby mu na základe nej mohlo byť pridelené miesto.

Filter žiadostí:	Zoznam: pariky					
<input type="checkbox"/> muži	<input type="checkbox"/> ženy	<input type="checkbox"/> súčasní	<input type="checkbox"/> novoprijatí	<input type="checkbox"/> bez miesta	<input type="checkbox"/> poradovník	Nastaviť
<a href="#">upraviť zoznamy</a>						

Obr. 7: Filter žiadostí

Výberom jedného z filtrov (prípadne tagov) a kliknutím na tlačidlo „Nastaviť“ sa zobrazí formulár na priradovanie miest úspešným žiadateľom, prípadne zrušenie priradenia miesta (Obr. 8). Nad formulárom je nadpis „Žiadosti“ s počtom žiadosti, ktoré vyhovujú aktuálne nastavenému filtru.

Formulár sa skladá z dvoch zoznamov: zoznam žiadateľov a zoznam miest, tlačítok „Zobraziť“ a „Zrušiť priradenie“ a zaškrtávaná „zobraziť len neobsadené miesta“. Výberom žiadateľa zo zoznamu a kliknutím na tlačidlo „Zobraziť“ sa zobrazia detaily žiadateľa so všetkými pridruženými informáciami (Obr. 9) a ak je žiadateľ úspešný, tak vo formulári pribudne tlačidlo „Priradiť miesto“. Priradené miesto sa žiadateľovi odoberie výberom daného miesta zo zoznamu a kliknutím na tlačidlo „Zrušiť priradenie“.

**Priradenie miest žiadateľom**

Tu je možné priradiť miesta úspešným žiadateľom o ubytovanie. Žiadateľová žiadosť musí byť označená ako 'úspešná', aby mu na základe nej mohlo byť priradené miesto.

Filter žiadostí: všetky žiadosti

muži  ženy  súčasní  novoprijatí  bez miesta  poradovník

[upraviť zoznamy](#)

**Žiadosti (počet: 686)**

Žiadateľ: ---

Miesto: ---

zobraziť len neobsadené miesta

**Obr. 8: Formulár na priradenie miest úspešným žiadateľom**

V prvej časti detailov žiadateľa je tabuľka so základnými údajmi jeho žiadosti a odkazom na kompletné detaily jeho žiadosti, ktoré je možné upravovať. Vedľa tejto tabuľky je zoznam miest, ktoré žiadateľ uviedol vo svojej žiadosti. Pod zoznamom miest je zoznam tagov. Ak má žiadosť priradený niektorý tag, tak je pri ňom odkaz „odobrať“, inakšie je tam odkaz „pridať“. Kliknutím na názov tagu sa automaticky nastaví filter žiadostí na žiadosti označené týmto tagom.

V druhej časti detailov žiadateľa je obdobná tabuľka so základnými údajmi a požadovanými miestami pre každého z jeho kamarátov. Ak kamarát zároveň pridal aktuálne vybraného žiadateľa ako kamaráta, tak pri jeho mene je zobrazený text „obojsstranne“.

V tretej časti je zoznam žiadateľov, ktorí pridalí aktuálneho žiadateľa ako ich kamaráta.

V poslednej časti je zoznam ľudí (len meno, priezvisko a izba), ktorí v žiadosti vyjadrili záujem o tú istú konkrétnu izbu ako aktuálny žiadateľ.

ID:		<b>Miesta</b>
Meno a priezvisko:	Fád adokdj - <a href="#">detaily žiadosti</a>	Manzelaky, hociktoré miesto s internetom
Stav žiadosti	úspešná	Vyskove Bloky, hociktoré miesto s internetom
Pridelené miesto	VBA 1026	<b>Zoznamy</b>
Odbor a ročník:	počítačová grafika a geometria (dobiehajúce mgr.) 5	<a href="#">a_ma_mat_jednotku</a> <a href="#">pridať</a>
Pohlavie:	muž	<a href="#">a_ma_mat_sturak</a> <a href="#">odobrať</a>
Poznámka:		<a href="#">b_moze_mat_net</a> <a href="#">pridať</a>
		<a href="#">izby s balkonom</a> <a href="#">pridať</a>
		<a href="#">jednotky</a> <a href="#">pridať</a>
		<a href="#">jednotky s internetom</a> <a href="#">pridať</a>
		<a href="#">MIH</a> <a href="#">odobrať</a>
		<a href="#">panky</a> <a href="#">pridať</a>

Kamaráti		
ID:	<a href="#">_____ - detaily žiadosti</a>	<b>Miesta</b>
Meno a priezvisko:	ikos ru - obojstranne	Manzelaky, hociktoré miesto s internetom
Stav žiadosti	úspešná	Vyskove Bloky, hociktoré miesto s internetom
Pridelené miesto	VBA 1026	
Odbor a ročník:	matematika (dobiehajúce mgr.) 5	
Pohlavie:	muž	
Poznámka:		

chceme spolu byvať na dvojke.

Ľudia, ktorí zadali tohto človeka ako kamaráta		
ID:	<a href="#">_____ - detaily žiadosti</a>	<b>Miesta</b>
Meno a priezvisko:	ikos ru - obojstranne	Manzelaky, hociktoré miesto s internetom
Stav žiadosti	úspešná	Vyskove Bloky, hociktoré miesto s internetom
Pridelené miesto	VBA 1026	
Odbor a ročník:	matematika (dobiehajúce mgr.) 5	
Pohlavie:	muž	
Poznámka:		

Obr. 9: Detaily žiadateľa a pridružené informácie

Pri pokuse priradiť miesto žiadateľovi s už priradeným miestom sa zobrazí chybová správa. Rovnako sa zobrazí chyba ak sa na už obsadené miesto pokúšame pridať ďalšieho žiadateľa.

## 6.2.7 Správa žiadostí

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Správa žiadostí“ sa zobrazia dva zoznamy: zoznam žiadateľov a zoznam oprávnených osôb.

Po výbere jedného zo žiadateľov a kliknutí na tlačidlo „Zobraziť“ sa zobrazí obrazovka s detailami žiadosti a so zoznamom záznamov, ktoré sú priradené k žiadosti. Z tejto obrazovky je možné prejsť na úpravu detailov žiadosti (Obr. 10), kliknutím na tlačidlo „Upraviť žiadosť“ alebo je možné manuálne pridať záznam, prípadne modifikátor.

Po výbere jednej z oprávnených osôb a kliknutí na tlačidlo „Vytvoriť“ sa zobrazí formulár s detailami žiadosti a predvyplnenými údajmi. Takto je možné vytvoriť žiadosť napríklad vtedy, ak niekto zatelefonoje členovi ubytovacej komisie.

**Zobrazenie žiadosti**

Žiadateľ:

**Upraviť žiadosť**

ID študenta:	64962826	E-mailová adresa:	peka_kilaj@rudh.rk
Meno:	Denip	Priezvisko:	Tabib
Pohlavie:	muž	Fajčiar:	nie
Adresa:	Cofehirara 69	PSČ:	92041
Mesto:	Fuisizvud	Vzdialenosť:	70
Typ štúdia:	doktorand	Typ študenta:	súčasný
Odbor:	informatika (dobiehajúce mgr.)	Nasledujúci ročník:	5
Priemer z VŠ:	1.96		
Priemery zo SŠ:	Profilový predmet 1:	Profilový predmet 2:	Profilový predmet 3:
	1.trieda:	2.trieda:	3.trieda:
Rodné číslo:		ISIC:	123456789
Poznámka:			

**Obr. 10: Úprava detailov žiadosti**

## 6.2.8 Stanovenie čiar

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Stanovenie čiar“ sa zobrazí formulár s dvomi textovými

políčkami, pomocou ktorých je možné nastaviť maximálnu celkovú bodovu úspešnej žiadosti pre novoprijatých a súčasných študentov.

### 6.2.9 Stav žiadostí

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Stav žiadostí“ sa zobrazia štyri odkazy členené na dve časti: súčasní študenti a novoprijatí študenti. Obidve časti majú dva totožné odkazy: „Nastaviť neposúdené žiadosti nad čiarou na úspešné“ a „Zobraziť zoznam žiadateľov“.

Po kliknutí na prvý z odkazov sa nastaví všetky neposúdené žiadosti nad aktuálne stanovenou čiarou na úspešné.

Po kliknutí na druhý z odkazov sa zobrazí poradovník žiadateľov (Obr. 11), v ktorom je možné manuálne nastaviť stav žiadosti. Žiadosti so záznamom, ktorý ich posúva do hornej časti poradovníka (zvýhodnené) majú zelenú farbu pozadia. Žiadosti so záznamom, ktorý ich posúva do dolnej časti poradovníka (znevýhodnené) majú červenú farbu pozadia.

#### Stav žiadosti

Tu je možné nastaviť stav žiadosti na 'úspešná', 'neúspešná' alebo 'neposúdená'.

#	Meno	Priezvisko	Odbor	Body za žiadosť	M	Body
1.	Cpa	<a href="#">Acránztá</a>	učiteľstvo: matematika, deskriptívna geometria (dobiehajúce mgr.)	0	0	0
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
2.	Pániu	<a href="#">Akýzpadý</a>	doktorand	0.055	0	0.055
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
3.	Coma	<a href="#">Baciký</a>	manažérska matematika (dobiehajúce mgr.)	0	0	0
NEÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
4.	Hopib	<a href="#">Bali</a>	informatika (dobiehajúce mgr.)	0	0	0
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
5.	Moco	<a href="#">Bedirý</a>	doktorand	0.021	0	0.021
NEPOSÚDENÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
6.	Riem	<a href="#">Bibfk</a>	doktorand	0	0	0
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
7.	Ufidd	<a href="#">Bicfjuhá</a>	ekonomická a finančná matematika (dobiehajúce mgr.)	0	0	0
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						
8.	Ohcluh	<a href="#">Bajdéz-Dorúf</a>	doktorand	0.011	0	0.011
ÚSPEŠNÁ - <a href="#">úspešná</a>   <a href="#">neúspešná</a>   <a href="#">neposúdená</a>						

Obr. 11: Manuálne nastavovanie stavu žiadostí

### **6.2.10 Nastavenia**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Nastavenia“ je možné nastavovať pre danú fakultu, či sa zobrazuje prihlasovací panel, registračný panel pre novoprijatých študentov a registračný panel pre súčasných študentov. Tiež je možné nastaviť, či žiadatelia môžu upravovať svoje žiadosti, miesta a kamarátov.

### **6.2.11 Generovanie výstupných zoznamov**

Pre prístup k tejto činnosti musí mať administrátor povolený prístup k nejakej fakulte. Po kliknutí na položku „Generovanie výstupných zoznamov“ sa zobrazia tri odkazy: „Rozdelenie izieb“, „Poradovník súčasných študentov“ a „Poradovník novoprijatých študentov“. Kliknutím na niektorý z odkazov sa vygeneruje excelovský dokument s príslušným zoznamom.

## 7 Záver

Analýzou ubytovacieho procesu na Univerzite Komenského sa nám podarilo indentifikovať niekoľko závažných problémov. S motiváciou vnieť do procesu moderné riešenie pre spracovanie údajov, ich prenos a komunikáciu medzi aktérmi sme navrhli a implementovali systém, ktorý rieši nájdené problémy.

Systém prináša softvérový nástroj, v ktorom je možné modelovať dôležitú časť procesu (vyhodnocovanie úspešností žiadostí) s ohľadom na potreby jednotlivých fakúlt.

V tejto diplomovej práci sme zdokumentovali poznatky z viacročného pôsobenia autora práce v ubytovacej komisii FMFI UK. Počas tohto obdobia sme na FMFI UK zaviedli podávanie žiadostí cez internet. Od prvých jednoduchých formulárov sme postupne navrhli komplexný softvér na podporu ubytovacieho procesu, ktorý funguje v reálnej prevádzke už dva roky.

Funkčnú aplikáciu je možné otestovať na adrese: <http://www.doucko.sk/diplomovka/>

Aplikácia beží v ostrej prevádzke na adrese: <http://ubytovanie.fmph.uniba.sk/>

## 8 Literatúra a zdroje

1. *Pravidlá pridelovania ubytovania*. <http://www.fmph.uniba.sk/index.php?id=373>, 2006
2. Rashmi: *A cognitive analysis of tagging*. Článok:  
[http://www.rashmisinha.com/archives/05\\_09/tagging-cognitive.html](http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html), 2005
3. Stuttard, D., Pinto, M.: *The Web Application Hacker's Handbook*. Wiley Publishing, 2008
4. [http://en.wikipedia.org/wiki/Lead\\_paragraph](http://en.wikipedia.org/wiki/Lead_paragraph)
5. Pavlíček, R.: *Dokumentácia zásad prístupnosti webových stránok pre používateľov s ťažkým zrakovým postihnutím*. [http://www.blindfriendly.sk/download/bfw23\\_sk.doc](http://www.blindfriendly.sk/download/bfw23_sk.doc), 2005
6. Nielsen, J., Loranger, H.: *Prioritizing Web Usability*. New Riders, Berkeley, 2006
7. Kosek, J.: *PHP – Tvorba interaktívnych internetových aplikácií*. Grada Publishing, 1999
8. <http://sourceforge.net/projects/xtpl/>
9. [http://pear.php.net/package/Spreadsheet\\_Excel\\_Writer/](http://pear.php.net/package/Spreadsheet_Excel_Writer/)
10. <http://www.google.com/a/help/intl/en/edu/index.html>