Univerzita Komenského v Bratislave

Fakulta matematiky, fyziky a informatiky

## Andrea Spišáková

Návrh systému

Kurzy

## Patrícia Marmanová

## Dana Škorvánková

## Michal Brcko November 2017

Obsah

1. [Úvod 1](#_Toc344210410)

1.1 [Cieľ](#_Toc344210411) 1

1.2 Rozsah 1

1.3 Definície, akronymy a skratky 1

[1.4](#_Toc344210414) Prehľad zvyšku dokumentu 2

1. [Referencie .](#_Toc344210415)2
2. [Popis dekompozície 2](#_Toc344210427)

3.1 Dekompozícia modulov 2

3.1.1 Kontroléry 2

[3.2](#_Toc344210429) Dekompozícia dát 3

1. [Popis závislostí 3](#_Toc344210432)
2. Popis rozhraní 3  
    5.1 Funkčné rozhrania 3  
    5.1.1 Metódy podľa kontrolérov 4  
    5.1.2 Triedny diagram 6

5.2 Procesné rozhrania 7

1. Detailný dizajn 8

# Úvod

## Cieľ

Tento dokument popisuje návrh systému webovej aplikácie Kurzy, ktorá bude slúžiť Centru informačných technológií na správu školení a používateľom na prihlasovanie na jednotlivé školenia. Cieľom vývoja opísanej aplikácie je nahradiť a rozšíriť aktuálne existujúci univerzitný softvér.

## Rozsah

Aplikácia bude slúžiť pre správu školení organizovaných univerzitnými pracoviskami. Systém má za úlohu umožniť používateľom vytvárať jednotlivé kurzy a prihlasovať sa na ne. Implementovaných bude 5 rôznych používateľských rolí, ktoré popisujú práva používateľov, od základného návštevníka, až po administrátora. Po prihlásení do systému môžu všetci používatelia filtrovať a prezerať kurzy. V prípade, že je naplnená kapacita kurzu, systém poskytuje používateľovi možnosť hlásiť sa na kurz ako náhradník, resp. prejaviť o kurz záujem. To znamená, že používateľ bude notifikovaný v prípade, že by sa miesto uvoľnilo, alebo bola navýšená maximálna kapacita kurzu.

## Definície, akronymy a skratky

CIT Centrum informačných technológií,

súčasť univerzita sa delí na menšie celky (katedry, fakulty),

prerekvizita povinnosť, ktorá musí byť splnená pred absolvovaním

kurzu,

id identifikácia konkrétneho používateľa,

PHP PHP: Hypertext Preprocessor (programovací jazyk

používaný najmä pre vývoj webových aplikácií),

CSS3 Cascading Style Sheet (technológia na jednotné štýlovanie

obsahu HTML dokumentu),

HTML5 Hyper Text Markup Language (značkový jazyk určený na

tvorbu webových dokumentov a iných informácií

zobraziteľných vo webovom prehliadači),

MySQL My Structured Query Language (relačný databázový

server),

Symfony framework pre jazyk PHP,

framework softvérová štruktúra, ktorá slúži ako podpora pri

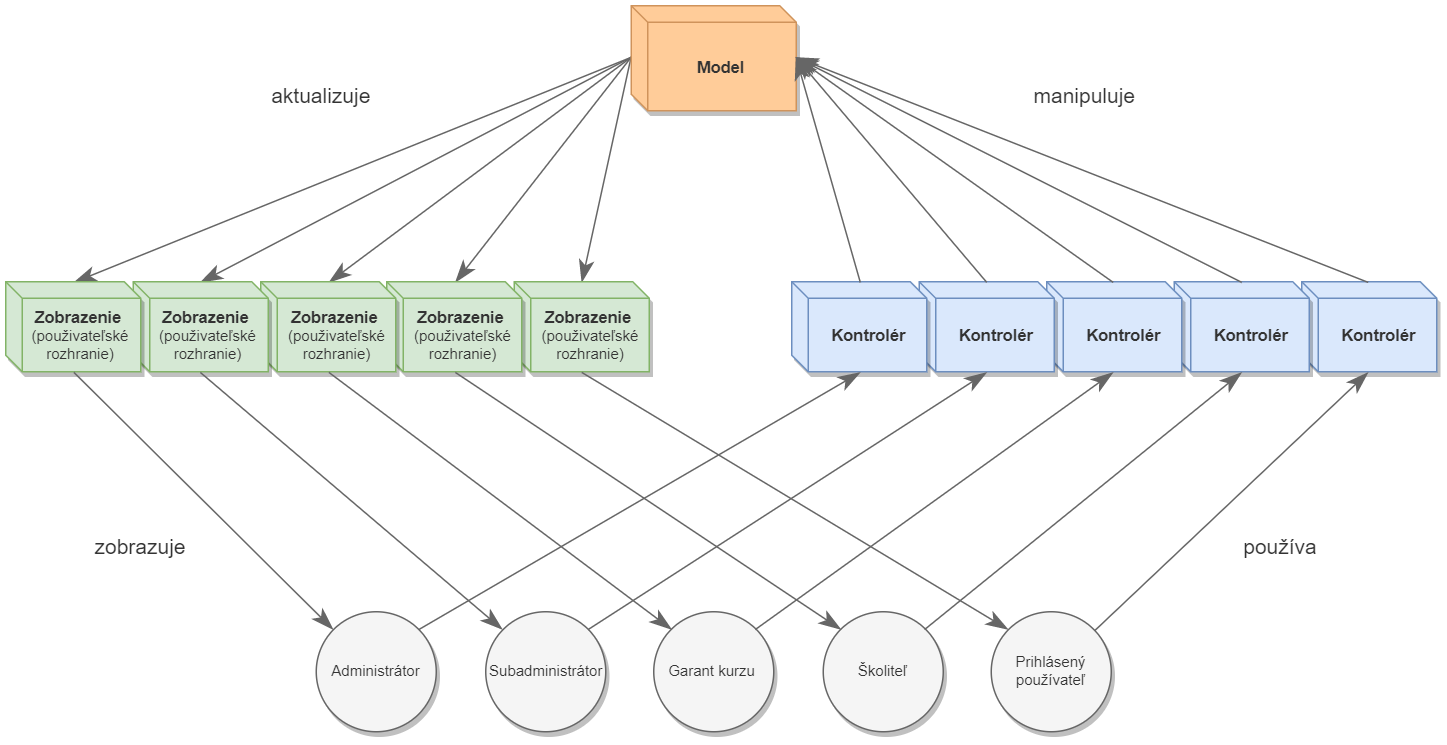
programovaní.

## Prehľad zvyšku dokumentu

# Referencie

# Popis dekompozície

Aplikácia implementuje tzv. *Model-view-controller* softwarovú architektúru, ktorá rozdeľuje dátový model aplikácie, používateľské rozhranie a riadiacu logiku do troch nezávislých komponentov tak, že modifikácia niektorého z nich má len minimálny vplyv na ostatné.



*Obrázok 1 Mode-view-controller softwarová* architektúra

## Dekompozícia modulov

Aplikácia bude rozdelená na niekoľko modulov. Dve základné časti budú administrácia kurzov a prihlasovanie na kurzy.

### 3.1.1 Kontroléry

V závislosti od používateľských rolí bude systém implementovať 5 rôznych kontrolérov (ktoré budú sprostredkúvať prenos externých udalostí zvonka do vnútra systému a naopak).

1. **Prihlásený používateľ**

Koncový používateľ, ktorý si môže prezerať a filtrovať všetky kurzy. Môže prejaviť záujem o absolvovanie niektorého kurzu v budúcnosti. Prihlasuje sa na vypísané kurzy, alebo sa odhlasuje z kurzov, na ktoré sa zapísal. Môže kontaktovať školiteľa prostredníctvom e-mailu. Nemá možnosť vidieť žiadne informácie o ostatných koncových používateľoch a ani nemôže meniť informácie o kurzoch, ktoré už absolvoval.

1. **Školiteľ**

Používateľ, ktorý je zodpovedný za termín kurzu, ktorý mu bol pridelený garantom kurzu a bude ho viesť. Môže prihlasovať a odhlasovať používateľov na kurz, alebo ich kontaktovať. Tiež môže tlačiť automaticky generované dokumenty (zoznam prihlásených na kurz a certifikáty o absolvovaní kurzu) vo formáte pdf.

1. **Garant kurzu**

Používateľ, ktorý je zodpovedný za jemu pridelený kurz, prípadne viac kurzov, subadministrátorom. Jeho úlohou je vypisovať termíny školení a prideľovať im jednotlivých školiteľov (má možnosť priradiť ako školiteľa aj samého seba) a vypisovať základné informácie o kurze – anotácia, požiadavky, prerekvizity.

1. **Subadministrátor**

Používateľ, ktorého právomoci sa pohybujú len v rámci jednej súčasti a nemôže zasahovať do správy kurzov a používateľov, ktorí patria do inej súčasti. Jeho úlohou je určovať garantov kurzov, vytvárať nové kurzy a upravovať už existujúce kurzy.

1. **Administrátor**

Hierarchicky najvyššia používateľská rola, ktorá má na starosti chod celého systému. Má všetky právomoci na správu používateľov a kurzov. Jeho úlohou je určovať subadministrátorov, dohliadať na aktuálnosť a správnosť informácií a pomáhať riešiť prípadné problémy.

# Popis rozhraní

## 5.1 Funkčné rozhrania

V tejto sekcii dokumentu je popísaná funkcionalita jednotlivých kontrolérov (ich metódy a vstupné a výstupné parametre týchto metód). Obsahuje tiež triedny diagram detailne zobrazujúci atribúty, metódy a vzťahy medzi jednotlivými triedami.

### 5.1.1 Metódy podľa kontrolérov

### **Bežný používateľ (trieda User)**

|  |  |
| --- | --- |
| Názov metódy | Funkcionalita |
| enrollInCourse(Course course): boolean | prihlásenie na konkrétny termín kurzu   * vracia úspešnosť vykonanej akcie |
| disenrollFromCourse(Course course): boolean | odhlásenie z termínu kurzu   * vracia úspešnosť vykonanej akcie |
| listMyCourses(): ArrayList<Enrollment> | vylistovanie všetkých zapísaných a absolvovaných kurzov   * vracia pole objektov triedy Enrollment (obsahuje referenciu na kurz, používateľa a informáciu o absolvovaní resp. neabsolvovaní kurzu) |
| listAvailableCourses(): ArrayList<Course> | vylistovanie všetkých dostupných kurzov, na ktoré sa daný používateľ môže prihlásiť   * vracia pole objektov triedy Course |
| sendEmail(Supervisor s, String message): void | kontaktovanie školiteľa prostredníctvom e-mailu |

### **Školiteľ (trieda Supervisor)**

|  |  |
| --- | --- |
| Názov metódy | Funkcionalita |
| supervisedCourses(): ArrayList<Course> | zobrazenie všetkých kurzov, ktoré daný školiteľ vedie   * vracia pole objektov triedy Course |
| enrolledUsers(Course c): ArrayList<User> | zobrazenie všetkých používateľov prihlásených na kurz   * vracia pole objektov triedy User |
| printCertificate(User u, Course c): void | tlač certifikátu o absolvovaní kurzu používateľom |
| setAttendance(User u, Course c, boolean attended): void | potvrdenie účasti resp. neúčasti používateľa na kurze |
| enrollUser(User u, Course c): void | zápis používateľa na kurz (dá sa využiť napr. ak má používateľ splnené prerekvizity externe) |

### 

### **Garant kurzu (trieda Master)**

|  |  |
| --- | --- |
| Názov metódy | Funkcionalita |
| listCoursesByType(CourseType coursetype): ArrayList<Course> | zobrazenie všetkých termínov daného (typu) kurzu   * vracia pole objektov triedy Course |
| setSupervisor(Supervisor supervisor, Course course): void | priradenie školiteľa k termínu kurzu |
| setVisibility(CourseType coursetype, boolean visible): void | nastavenie viditeľnosti kurzu |
| createCourse(Timestamp t, String place, int capacity): Course | vytvorenie termínu kurzu (parametre – dátum, čas, miesto, kapacita)   * vracia vytvorený objekt triedy Course |
| deleteCourse(Course course): void | zrušenie termínu kurzu |

### 

### **Subadministrátor (trieda Subadmin)**

|  |  |
| --- | --- |
| Názov metódy | Funkcionalita |
| createCourseType(String name, String desc, String prereqs, boolean visible): CourseType | vytvorenie nového kurzu (parametre – názov, popis, prerekvizity, viditeľnosť)   * vracia objekt vytvoreného kurzu |
| editCourse(CourseType ctype, String name, String desc, String prereqs, boolean visible): void | upravenie existujúceho kurzu |
| deleteCourseType(CourseType ctype): void | zrušenie kurzu |
| setMaster(Master master, CourseType ctype): void | priradenie garanta ku kurzu |

### 

### **Administrátor (trieda Admin)**

|  |  |
| --- | --- |
| Názov metódy | Funkcionalita |
| setSubadmin(Department d, Subadmin subadmin): void | priradenie subadministrátora k súčasti |
| listCourses(Department d): ArrayList<Course> | zobrazenie termínov všetkých kurzov |

### 5.1.2 Triedny diagram

### Triedny diagram zobrazuje atribúty, metódy tried a vzťahy medzi triedami. Nadtrieda User pokrýva základnú funkcionalitu prihláseného používateľa. Táto trieda má štyri podtriedy (Admin, Subadmin, Master, Supervisor), ktoré od nej dedia atribúty a metódy a každá pridáva nejakú vlastnú špecifickú funkcionalitu v závislosti od právomocí konkrétnej používateľskej role.

### Trieda CourseType predstavuje typ kurzu, zatiaľ čo trieda Course predstavuje konkrétny termín nejakého typu kurzu. Medzi triedami Course a CourseType je vzťah agregácie (pričom CourseType je celok a Course je časť). Jeden typ kurzu môže mať ľubovoľne veľa termínov, ale jeden termín kurzu patrí vždy práve k jednému typu kurzu.

Trieda Department predstavuje súčasť univerzity. Súčasti sú hierarchicky usporiadané – každá súčasť má atribút rodič (zrejme koreňová súčasť – samotná univerzita – tam bude mať null), kde je referencia na nadradenú súčasť a atribút deti, kde sú referencie na podradené súčasti.

### Trieda Enrollment je pomocná trieda, ktorá obsahuje atribút s referenciou na používateľa a atribút s referenciou na termín kurzu. K týmto referenciám pridáva ešte vlastné atribúty graduated a attended, ktoré poskytujú informácie o tom, či používateľ kurz úspešne absolvoval a či sa zúčastnil.

Obrázok 2 Triedny diagram

