



## ZADÁNÍ DIPLOMOVÉ PRÁCE

<b>Název:</b>	Portál pro podporu studia a klasifikace student
<b>Student:</b>	Bc. Zden k Balák
<b>Vedoucí:</b>	Ing. Pavel Kordík, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Webové a softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Moderní vysoké školy mají informa ní systém, který umož ňuje hodnocení semestrálních úloh student , klasifikaci test a podporuje pedagogické innosti tak, aby data bylo možno dále zpracovat. Nastudujte podobné systémy, zmapujte pot eby student , pedagog a vedení FIT VUT v této oblasti. Po dohod s vedoucím práce navrh ňte a implementujte portál, který podpo í studijní agendu a umož ňí získávat dále zpracovatelná data o klasifikaci student . Portál pe liv ňte a zdokumentujte.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdí k, CSc.  
d kan

V Praze dne 17. února 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
SOFTWAREVÉ INŽENÝRSTVÍ



Diplomová práce

## **Portál pro podporu studia a klasifikace studentů**

*Bc. Zdeněk Balák*

Vedoucí práce: Ing. Pavel Kordík, Ph.D.

9. května 2017



---

## Poděkování

Rád bych touto cestou poděkoval Ing. Pavlovi Kordíkovi, Ph.D za podporu a rady při tvorbě této práce, dále bych chtěl poděkovat Bc. Martinovi Královi a Tomášovi Martínkovi za podporu aplikace grafickým designem, Bc. Václavovi Blažejovi a Bc. Štěpánovi Plachému za technickou podporu, Ing. Alešovi Fišerovi, Ing. Stanislavovi Kuznetsovovi a Mgr. Michaele Býčkové za zpětnou vazbu k textu práce



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

Praha dne 9. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Zdeněk Balák. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Balák, Zdeněk. *Portál pro podporu studia a klasifikace studentů*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Tato práce se zabývá návrhem systému pro řízení výuky, konkrétněji systému pro použití na FIT ČVUT. Práce popisuje tvorbu systému od analýzy, přes návrh až po realizaci. Využito bylo standardních webových technologií a metod pro návrh softwarového díla. Výsledkem práce je fungující prototyp aplikace, která již může sloužit ke správě výuky. Aplikace byla navržena s ohledem na možnost budoucího rozšiřování. Využit se dá praktická část pro další rozvoj systému pro správu obsahu nebo přímo text práce jako inspirace pro tvorbu webové aplikace

**Klíčová slova** System pro řízení učení, Webová aplikace, java, REST, Spring Boot, AngularJS

---

# Abstract

This thesis is concerned with learning management system, more specifically system to be used in FIT CTU. It describes creation from analysis through design to implementation. It uses software engineering methods as well as web development technologies. The result is functioning prototype that can be used to support courses. Practical part of the thesis can be used to further extend and develop the learning management system. The text of the thesis can be used as inspiration for web application development.

**Keywords** Learning Management System, Web application, java, REST, Spring Boot, AngularJS

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Rešerše</b>	<b>3</b>
1.1 Současný stav: Edux . . . . .	3
1.2 Sběr požadavků . . . . .	4
1.3 Rešerše existujících řešení . . . . .	7
<b>2 Rozvaha</b>	<b>13</b>
2.1 Vize . . . . .	13
2.2 Cíle této práce - praktická část . . . . .	13
2.3 Popis produktu . . . . .	14
2.4 Formální požadavky . . . . .	14
2.5 Use Cases . . . . .	14
2.6 Návrh uživatelského rozhraní . . . . .	16
<b>3 Realizace</b>	<b>25</b>
3.1 Použité technologie . . . . .	25
3.2 Návrh architektury . . . . .	32
3.3 Návrh datového modelu . . . . .	33
3.4 Popis REST API . . . . .	35
<b>4 Uživatelská příručka</b>	<b>37</b>
4.1 Seznam pojmů . . . . .	37
4.2 Návod pro vyučující . . . . .	38
4.3 Návod pro studenty . . . . .	47
<b>5 Testování</b>	<b>51</b>
5.1 Heuristická analýza . . . . .	51
5.2 Testování s uživateli . . . . .	55

<b>Závěr</b>	<b>59</b>
<b>Literatura</b>	<b>61</b>
<b>A Seznam použitých zkratk</b>	<b>63</b>
<b>B Dotazník v rámci řešerše</b>	<b>65</b>
<b>C Definice REST API</b>	<b>71</b>
<b>D JSON specifikace objektů</b>	<b>77</b>
<b>E Obsah CD</b>	<b>87</b>

---

## Seznam obrázků

2.1	Use Cases diagram - STUDY PORTAL . . . . .	15
2.2	Wireframe - Úvodní stránka a časová osa . . . . .	17
2.3	Wireframe - Správa předmětu . . . . .	18
2.4	Wireframe - Správa stránek předmětu . . . . .	19
2.5	Wireframe - Stránka pro tvorbu materiálů . . . . .	19
2.6	Wireframe - Stránka s aktualitami . . . . .	20
2.7	Wireframe - Události předmětu . . . . .	20
2.8	Wireframe - Hodnocení předmětu . . . . .	21
2.9	Wireframe - Hodnocení testu . . . . .	22
2.10	Wireframe - Individuální hodnocení studenta . . . . .	23
2.11	Wireframe - Správa notifikací . . . . .	23
3.1	Životní cyklus entity [7] . . . . .	27
3.2	Schéma architektury aplikace . . . . .	32
3.3	Návrh datového modelu . . . . .	34
4.1	Screenshot - Nastavení předmětu . . . . .	39
4.2	Screenshot - Správa stránek . . . . .	40
4.3	Screenshot - Tvorba stránky . . . . .	41
4.4	Screenshot - Zobrazení událostí . . . . .	42
4.5	Screenshot - Vytváření události . . . . .	42
4.6	Screenshot - Přidávání materiálů . . . . .	43
4.7	Screenshot - Přidávání záznamu na časovou osu . . . . .	43
4.8	Screenshot - Hodnocení předmětu . . . . .	44
4.9	Screenshot - Ukázka speciálního hodnocení . . . . .	45
4.10	Screenshot - Hromadné hodnocení . . . . .	46
4.11	Screenshot - Individuální hodnocení . . . . .	47
4.12	Screenshot - Ukázka časové osy . . . . .	48
4.13	Screenshot - Moje hodnocení předmětu . . . . .	49
4.14	Screenshot - Testy předmětu . . . . .	50



---

# Úvod

Systém pro řízení učení je virtuální prostředí pro sdílení studijních materiálů. Standardně podporuje tvorbu předmětů, materiálů a má různé funkce jako hodnocení a evidence výsledků. Výhodou je nastavení sdílení materiálů pro určitou skupinu. Na FIT ČVUT se v současné době využívá EDUX, který je prezentován jako dočasné řešení.

Osobní motivace zabývat se tímto tématem vyplývá z vlastní zkušenosti jakožto studenta. V průběhu studia mi chyběl přehled důležitých událostí a každý týden jsem hledal, co mě v nejbližších dnech čeká. To probíhalo tak, že jsem musel procházet profily všech zapsaných předmětů a kontrolovat, jestli se neblíží nějaký deadline.

Při tomto procházení profilů všech předmětů jsem velice často narážel na situace, kdy se v předmětu nacházely prázdné sekce, které byly pravděpodobně automaticky vygenerovány, i přes to, že pro předmět nebyly relevantní.

Na základě těchto zkušeností jsem se rozhodl zkusit navrhnout vlastní řešení, které by pro mě, jakožto pro studenta, bylo přívětivější. Zároveň jsem se snažil, aby řešení bylo pro učitele přinejmenším stejně uživatelsky přívětivé jako to současné.

## Práce je dělena do následujících kapitol

- **Rešerše** - rešerše shrnuje současný stav řešení na FIT ČVUT a zkoumá jakým způsobem problematiku řeší existující systémy pro řízení učení.
- **Analýza** - tato část popisuje ideální řešení a poté definuje cíle této práce. Shrnuje formální požadavky, případy užití a obsahuje návrh uživatelského rozhraní.
- **Realizace** - tato část popisuje hlavní použité technologie, navrhuje architekturu aplikace a popisuje datový model aplikace.

- **Uživatelská příručka** - Tato část má sloužit jako návod k využívání aplikace, jsou zde popsány všechny funkce.
- **Testování** - každá webová aplikace se musí podrobovat testování, při tvorbě aplikací dochází ke spoustě chyb, které se jiným způsobem zpravidla neodhalí. Demonstrovány jsou dva způsoby testování: S uživateli a Bez uživatelů.

Cílem práce je zmapovat potřeby FIT ČVUT, navrhnout a vytvořit prototyp aplikace, která by v budoucnu mohla sloužit jako systém pro řízení učení na univerzitách.



---

# Rešerše

## 1.1 Současný stav: Edux

Edux[1] lze označit jako tzv. Learning Management System (LMS), tj. systém pro správu výuky.

- Povinný pro všechny předměty na FIT.
- Vychází ze systému DokuWiki (PHP) s celou řadou interních modifikací.
- Princip: wiki s předdefinovanou strukturou a polostrukturovanými dokumenty.

### Funkce Eduxu [2]

- Dashboard - sada dlaždic na úvodní stránce eduxu
  - Seznam aktuálně vyučovaných předmětů studenta
  - Přehled novinek zvolených předmětů
  - On-line data z KOSu
  - Informace z IS
  - Personifikace uživatelem
- Správa studijních materiálů
  - příprava textů a prezentací přímo v systému (ve formě wiki stránek)
  - nahrávání externích dokumentů
  - tvorba webových slidů a webových skript
  - řízení přístupových práv k materiálům podle KOSu
  - materiály od studentů (např. výstupy práce) – osobní namespace
  - archivace materiálů po semestrech

- Klasifikace
  - libovolná struktura klasifikace tvořena garantem předmětu
  - automatizace výpočtů a kontroly stanovených podmínek
  - podpora učitelských přehledů
- Harmonogram
  - tvorba harmonogramu pro jednotlivé typy výuky
  - upozornění na důležité termíny (odevzdání úkolu)
  - podpora svátků, volna a přesuny výuky

### Výhody Eduxu

- Povinné využití - i v případě, že se materiály k předmětu nachází jinde
- Tvorba materiálů - díky základu na doku wiki je tvorba materiálů velice jednoduchá a nabízí spoustu možností
- Struktura klasifikace může být vytvořena libovolně
- Možnost nahrávání a sdílení souborů

### Nevýhody Eduxu

- Složitý způsob úpravy menu předmětu. Někteří učitelé kvůli tomu ne-  
mažou předgenerovanou strukturu a vzniká spousta "slepých" odkazů
- Zadávání aktualit předmětu na statickou stránku do každého předmětu  
zvláště je nepraktické. Student musí zkontrolovat každý předmět zvláště,  
aby zjistil, jestli nějaká aktualita nepřibyla
- Student není upozorněn na změnu hodnocení
- Student nemá přehled důležitých deadlinů/datumů - musí projít všechny  
svoje předměty, aby zjistil, jestli ho v nejbližší době nečeká test/deadline
- Není responzivní

## 1.2 Sběr požadavků

### 1.2.1 Požadavky studentů

Pro zjištění zájmu o nové funkce mezi studenty byl vypracován dotazník B. Na tento dotazník odpovědělo 108 respondentů a zájem o funkcionality byl následující:

- Časová osa zobrazující deadliny domácích úkolů/semestrálních prací/-termíny testů - zájem cca 94%
- Aktuality předmětu formou notifikací v portálu - zájem cca 84%
- Aktuality předmětu e-mailem z portálu - zájem cca 73%
- Notifikace o změně hodnocení - zájem cca 84%
- Upozornění o změně hodnocení e-mailem - zájem cca 67%
- Nástěnka předmětu, kde mohou studenti komunikovat se všemi účastníky předmětu - zájem cca 61%

### Konstruktivní poznámky studentů

- *Nezapomenout na hard/soft deadliny.*
- *Mělo by se na to myslet také z pohledu kantorů. Aktuálně se dá všechno na eduxu dohledat, jen to dá práci. Bylo by hloupé, aby kvůli pohodlí studentů kantoři museli spravovat bambilión stránek (edux, marast, progtest, aktivita na FB, ..., a teď k tomu další aplikace), jen aby to studenti měli všechno pod nosem.*
- *Bylo by dobré na tohle použít export do kalendáře, stejně jako se dá vyexportovat klasický rozvrh - automaticky se tam přidávají i zapsané zkoušky. Kdyby tam přibylo upozornění na testy a deadliny, tak by to bylo ideální. Rozhodně nehledám další stránku / aplikaci na komunikaci s učiteli.*
- *E-mail skoro nepoužívám, jednotné notifikační centrum v aplikaci by se mi používalo líp.*
- *Možnost úpravy proti datům v KOSu a tak (např. chodím na jinou paralelku, než mám zapsanou).*

### 1.2.2 Požadavky učitelů

Druhý dotazník byl připraven pro učitele B, pro zjištění, jak stávající EDUX používají, jak sdílejí se studenty materiály apod. Výsledek dotazníku je následující:

#### Jakým způsobem poskytují studentům materiály

- Nahrávám soubory na EDUX - 74%
- Vytvářím materiály v EDUXu - 45,2%

## 1. REŠERŠE

---

- Odkazuji na vlastní stránku - 24,7%
- Nahrávám materiály na git - 13,7%
- Používám jiný systém (Moodle, Marast,..) - 8,2%

### Jakým způsobem získávají studenti v mém předmětu body

- Zkouška - 70%
- Testy - 66%
- Aktivita na cvičeních - 60%
- Domácí úkoly - 55%
- Semestrální práce - 55%
- Aktivita na přednáškách - 11%
- Docházka 11%

### Při hodnocení využívám

- Součet bodů a ty tvoří známku podle standardní stupnice - 71,2%
- Výsledný počet bodů se počítá složitějším vzorcem - 17,8%
- Zámka se zadává ručně, protože nezávisí jen na bodech - 9,6%
- Předmět je ukončen zápočtem - 1,4%

### Konstruktivní poznámky učitelů

- *Distribuci materiálů by šlo zjednodušit. EDUX je v tomto směru značně nepřístupný. Aktuality na EDUXu nikdo nečte - student nebude pravidelně proklikávat svých X předmětů.*
- *Možnost nastavení upozornění na klasifikaci na eduxu.*
- *Bylo by fajn mít možnost edux více dynamický - například zadefinovat na jednom místě deadline a z ostatních míst se na něj pouze odkazovat přes proměnnou. Takhle se stává, že to člověk někde zapomene v novém roce změnit.*
- *Používat edux co nejvíce to jde, udelat jeho aktualizovani pres git a textove soubory. Studenti by si mohli nastavit zasilani aktualit e-mailem.*

### Shrnutí požadavků

Z požadavků studentů, učitelů a z dosavadního fungování vyplývají následující požadavky:

- Možnost zadávat deadlines k událostem (a to i více než jeden) a ty zobrazovat integrovaně na jednom místě
- Vytvořit notifikační centrum v rámci aplikace, jednak upozorňovat na události (nové hodnocení) a za druhé přidat učitelům možnost zadávat aktuality, které se budou taktéž posílat formou notifikací
- Vytvořit API, které by mohlo předávat data dalším aplikacím (analýza výsledků studentů, export do kalendáře,..)
- Vytvořit systém bodování s možností výpočtů z různých hodnocení, ale i možností standardního hodnocení (součet bodů a známka podle standardní stupnice)
- Tvorba materiálů a možnost odkazovat na externí zdroje
- Jednoduchá správa předmětu (úprava menu, přidávání stránek)

## 1.3 Rešerše existujících řešení

Veškerá hodnocená konkurenční řešení patří do kategorie software Learning Management System (Systém pro řízení výuky).

### Moodle

Moodle je softwarový balíček pro tvorbu výukových systémů a elektronických kurzů na internetu. Používají ho nejen vysoké, střední ale i základní školy

### Funkcionalita

Moodle obsahuje online kurzy, a ty se skládají z jednotlivých modulů. Moduly umožňují přidávat:

- Studijní materiály – libovolný učební materiál ve formě textu, HTML, URL, přiloženého souboru libovolného formátu
- Přednášky – učební materiál formou přednášky
- Úkoly - libovolná instrukce žákům (např. prostudujte materiály, vypracujte, rozmyslete si, apod.)
- Testy a jejich automatizované vyhodnocení

## 1. REŠERŠE

---

- Ankety
- Diskusní fóra
- Slovníky - učitel i s eventuální pomocí studentů může vytvářet encyklopedii důležitých slov.

Moodle také umožňuje učitelům zobrazit hodnocení studentů a jejich statistiky

### **Uživatelské rozhraní**

Uživatelský zážitek není vždy úplně přívětivý, například při zapisování kurzů, kdy je nutné něco neustále potvrzovat. Jinak rozložení stránky odpovídá tradičnímu třísloupcovému layoutu a platí, že čím více se přidá obsahu do kurzu, tím se více natahuje stránka. Učitelský pohled navíc zahrnuje kombinaci studentského pohledu s pohledem vyučujícího, což může být matoucí a prodlužuje délku stránky.

### **Shrnutí**

#### **Výhody**

- Diskuzní fóra a ankety
- Posílání zpráv mezi uživateli
- Automatizované vyhodnocení testu

#### **Nevýhody**

- Je třeba vytvářet kurzy pomocí modulů
- Nejednotný obsah (každý kurz může obsahovat něco jiného)
- Těžší na administraci z pohledu učitele

### **JoomlaLMS**

JoomlaLMS je aplikace pro studijní účely postavená na populární správci obsahu Joomla CMS.

### **Funkcionalita**

Podobně jako u Moodle jsou jednotlivé části kurzů přidávány jako moduly v režimech. Systém obsahuje také tzv. komercializační balíček, tedy možnost prodávat přístup ke kurzům podobně jako nákup v elektronickém obchodě.

### **Uživatelské rozhraní**

Systém disponuje responzivním layoutem, který se dokáže přizpůsobit obrazovce zařízení, přes které uživatel přistupuje. Ovládání je jednoduché, ale vzhledem k možnostem může velikost obsahu systému narůst a změnit se v nepřehlednou aplikaci s velkou hromadou možností a uživatel se bude ztrácet.

### **Shrnutí**

#### **Výhody**

- Responzivní rozvržení rozhraní
- Možnost prodávat přístup ke kurzům
- Zobrazování posledních změn a novinek

#### **Nevýhody**

- Části kurzů jako moduly
- Nastavování režimu platby/zdarma při vytváření kurzu
- Velký počet možností může přerůst v nepřehlednou obrazovku

### **Eliademy**

Eliademy je online software pro podporu výuky

#### **Funkcionalita**

Systém je založen na softwaru Moodle. Podporuje import kurzů z Moodle (musí být ve formátu moodle). Systém poskytuje uživateli (učiteli) vytvářet online kurzy. Tyto kurzy může učitel využít nejen k výuce, ale i k prodeji jiným uživatelům. Dále může hodnotit, komunikovat se studenty, nahrávat videa, apod.

### **Uživatelské rozhraní**

Eliademy je plně responzivní. Po tvorbu předmětů využívá především wysiwyg editory, díky čemuž působí na uživatele velice jednoduše a přehledně.

### **Shrnutí**

#### **Výhody**

- Sdílení kurzů přes LinkedIn a sociální média
- Možnost prodeje kurzů

- Komunikace s třídou
- Deadline úkolů

### **Nevýhody**

- Předmět má velice volnou strukturu, je to prakticky sada článků
- Slabá česká lokalizace

### **FelSight**

FelSight je webová aplikace vytvořená na FEL. Není to sice learning management system, ale částečně se překrývá s požadavky, takže stojí za zmínění.

### **Funkcionalita**

FelSight je integrován se systémem KOS a proto poskytuje také relevantní informace k danému studentovi jako jsou rozvrhy, informace z webu kateder a fakulty, harmonogramy, akce, informace o předmětech. Zatím je implementována pouze částečná podmnožina funkcionalit, ke kterým je dostupné aplikační rozhraní.

### **Uživatelské rozhraní**

Uživatelské rozhraní je intuitivní a vychází z obecné koncepce oblíbených webů a sociálních sítí. Dialogová okna však občas obsahují mnoho informací a je nutné dlouho scrollovat na požadovaný text. Kladně lze hodnotit také vyhledávání, které probíhá nad všemi objekty a výsledky vyhledávání jsou tak i kategorizovány.

### **Shrnutí**

#### **Výhody**

- Centrální bod nad ČVUT systémy
- Příjemné rozhraní
- Vyhledávání nad všemi objekty v systému

#### **Nevýhody**

- Aplikace je cílená pouze na studenty
- Umožňuje přístup k malé podmnožině informací
- Je úzce svázaná s ČVUT systémy



## **Sakai**

Sakai je webová aplikace pro online learning.

### **Funkcionalita**

Učitel přidává vlastní kurzy s popisem a jednotlivými částmi. Systém navíc podporuje matematickou notaci, takže se hodí pro technicky zaměřené vzdělávací instituce. Obsahuje také upozornění a inteligentní zpětnou vazbu, kde se informace o problému odesílá podle typu na správné oddělení.

### **Uživatelské rozhraní**

Rozhraní je responsivní, takže lze s aplikací pracovat na různých zařízeních. Například známkování probíhá hromadně, což velmi šetří čas a je možné takto ohodnotit velkou skupinu studentů v krátkém časovém okamžiku.

### **Shrnutí**

#### **Výhody**

- Responsivní design
- Hromadné známkování
- Inteligentní feedback

#### **Nevýhody**

- Vzrůstající nepřehlednost se složitostí kurzů
- Velké množství nastavení než systém začne pracovat podle představ



---

# Rozvaha

## 2.1 Vize

V ideálním případě má fakulta pro podporu výuky jeden systém, který je používán ve všech předmětech. Tento systém by měl mít tyto funkce:

- Systém zobrazuje uživateli obsah na základě jeho rolí
- Systém umožňuje tvorbu materiálů
- Systém podporuje verzování materiálů, minimálně na úrovni semestrů
- Systém umožňuje nahrávání materiálů/obrázků
- Systém podporuje zobrazení důležitých informací ze všech předmětů uživatele na jednom místě
- Systém podporuje správu klasifikace
- Systém podporuje zadávání a odevzdávání domácích úkolů
- Systém podporuje tvorbu testů/kvízů
- Systém automaticky vyhodnocuje řešení domácích úkolů/testů
- Systém komunikuje s uživatelem (formou upozornění)

## 2.2 Cíle této práce - praktická část

Tato práce se zaměřuje na tyto body:

- Systém zobrazuje uživateli obsah na základě jeho rolí
- Systém umožňuje tvorbu materiálů

## 2. ROZVAHA

---

- Systém podporuje zobrazení důležitých informací ze všech předmětů uživatele na jednom místě
- Systém podporuje správu klasifikace
- Systém podporuje zadávání domácích úkolů
- Systém komunikuje s uživatelem (formou upozornění)

### 2.3 Popis produktu

Systém pro řízení výuky "STUDY PORTAL" pomáhá studentům s organizací jejich studia a umožňuje učitelům snadno a efektivně organizovat výuku svých předmětů.

### 2.4 Formální požadavky

#### Funkční

- Portál umožňuje učiteli vytvářet materiály předmětu
- Portál umožňuje učiteli spravovat a zadávat hodnocení
- Portál zobrazuje studentovi jeho události na časové ose
- Portál zobrazuje studentovi studijní materiály
- Portál s uživatelem komunikuje formou notifikací

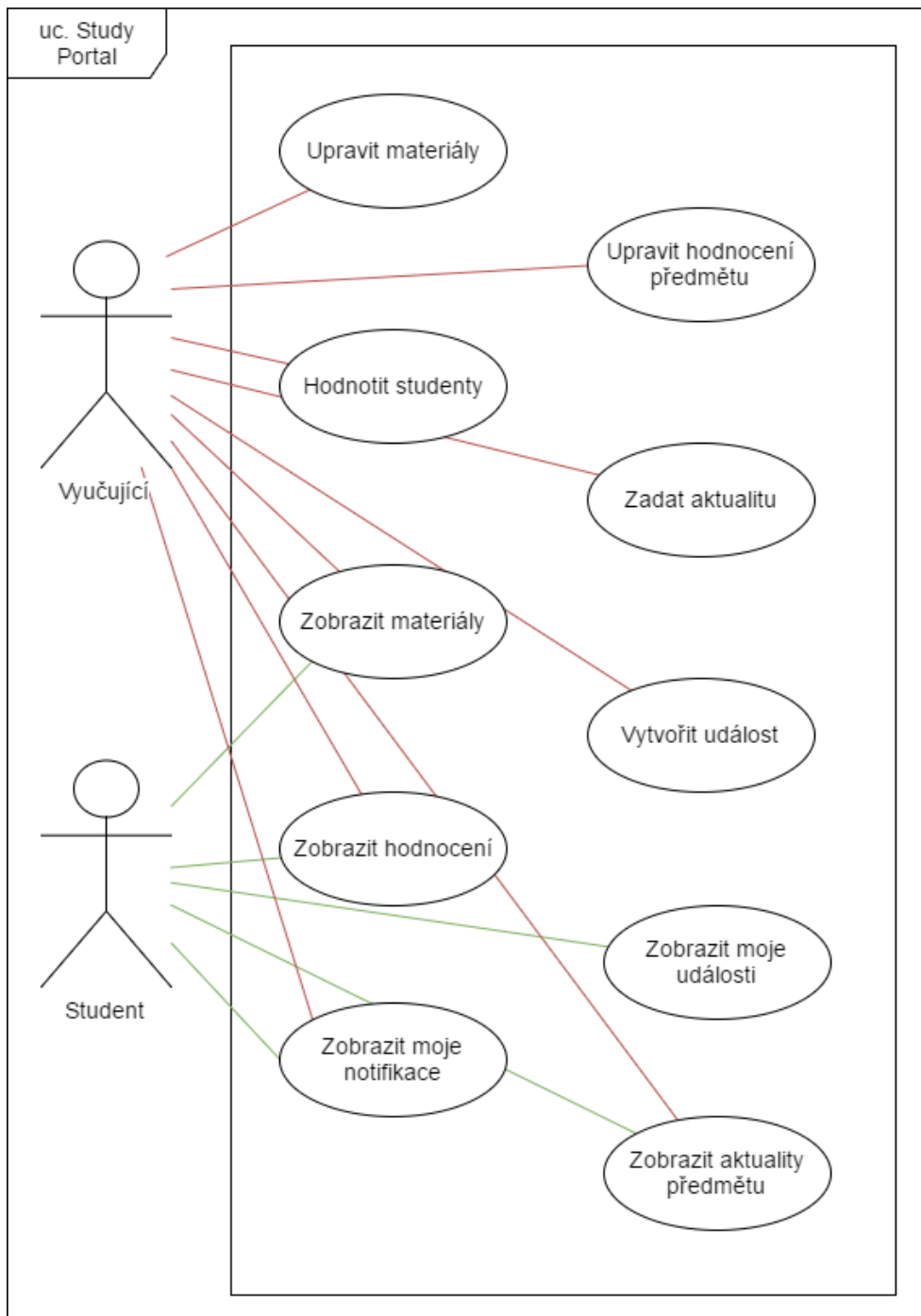
#### Nefunkční

- Implementace v jazyce Java
- Výsledkem bude samostatně spustitelný soubor ve formátu JAR
- Ověření uživatele pomocí autorizačního serveru ČVUT

### 2.5 Use Cases

Případ užití 2.1 popisuje funkce systému na základě interakcí jednotlivých rolí se systémem. Tento systém rozlišuje dvě základní role: Student, Učitel.

- Učitel zadává a upravuje studijní materiály
- Učitel definuje hodnocení předmětu
- Učitel zadává aktuality předmětu



Obrázek 2.1: Use Cases diagram - STUDY PORTAL

- Učitel zadává události předmětu
- Učitel hodnotí studenty
- Uživatel si zobrazí materiály předmětu
- Uživatel si zobrazí hodnocení předmětu
- Uživatel si zobrazí svoje notifikace
- Uživatel si zobrazí aktuality předmětu
- Student si zobrazí svoje události

### 2.6 Návrh uživatelského rozhraní

Uživatelské rozhraní je možné navrhovat více způsoby. Mezi tyto způsoby patří tvorba wireframe, mockup a prototypování.

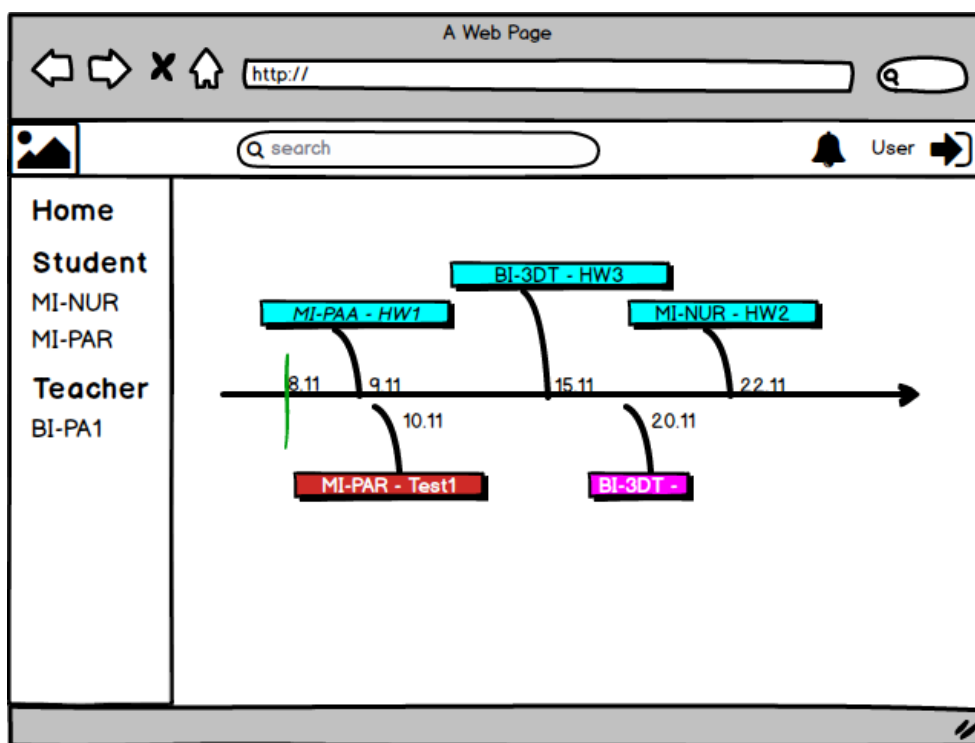
**Wireframe** Jde o jednoduchý náčrt funkčních prvků a základních rozložení aplikace. Může být vytvořen v softwaru k tomu určenému nebo jednoduše načrtnutý tužkou na papír. Wireframe nereflktuje vzhled výsledné aplikace, jde v něm jen o rozložení komponent a o základní funkce. Ve wireframe by neměla chybět žádná z klíčových komponent aplikace.

**Prototyp** Prototyp je simulace interakce uživatele s aplikací. Nemusí vypadat jako finální produkt, ale měl by se mu již alespoň trochu podobat (neměla by to být pouze směsice šedých boxů). Interakce s prototypem by měla odrážet to, jak bude vypadat interakce s finálním produktem. Prototypy se používají pro uživatelské testování.

**Mockup** Mockup je narozdíl od prototypu statická vizualizace aplikace. To znamená, že není interaktivní a tudíž se na ní nedají provádět uživatelské testy. Mockup může vypadat přesně jako výsledná aplikace a používá se pro získání zpětné vazby ke vzhledu, popřípadě pro prezentování sponzorům/zákazníkům před tím, než je aplikace hotová.

#### 2.6.1 Návrh aplikace

Návrh této práce byl proveden pomocí wireframes. Ty by měly nastínit, jaké stránky bude aplikace obsahovat a jaké budou mít funkcionality.



Obrázek 2.2: Wireframe - Úvodní stránka a časová osa

## Úvodní stránka 2.2

Aplikace má klasické rozložení webové aplikace podobné jako má například facebook <sup>1</sup> nebo linkedIn <sup>2</sup>. Vpravo nahoře je uživatel informován o svém přihlášení, najde tam uživatelské nastavení nebo notifikace. Uprostřed nahoře je vyhledávání, kde se budou dít vyhledávat například předměty. Vlevo se nachází navigace mezi jednotlivými předměty/stránkami aplikace. Na úvodní stránce se nachází časová osa uživatele. Pro studenta představuje přehled o důležitých událostech jeho studia (deadline, termíny testů,..).

<sup>1</sup><https://www.facebook.com/>

<sup>2</sup><https://www.linkedin.com/>

### Správa předmětu

- Přednášky
- Cvičení
- Prosemináře
- Laboratoře
- Testy
- Domácí úkoly
- Semestrální práce

---

**Uložit**

Obrázek 2.3: Wireframe - Správa předmětu

### **Správa předmětu 2.3**

Tato stránka bude přístupná jen vyučujícím předmětu, popřípadě podmnožině vyučujících. Nastavuje se zde jaké části předmět obsahuje.

### **Správa stránek předmětu 2.4**

Na této stránce může vyučující nastavovat stránky předmětu navíc. Stránkami předmětu se rozumí jakékoliv stránky, které nevyplývají z nastavení a podstaty předmětu (testy, hodnocení, přednášky,..). Tyto stránky zpravidla slouží pro tvorbu materiálů. Na tyto stránky se může odkazovat například u konkrétního testu nebo se mohou přidat přímo do postranního menu předmětu. Navíc se dá nastavit, aby se v postranním menu odkazovalo na stránku mimo aplikaci (například stránka facebooku předmětu).

### **Stránka pro tvorbu materiálů 2.5**

Stránky zadané ve správě stránek předmětu se poté mohou tvořit pomocí textového editoru. Pro prvotní verzi aplikace byl vybrán formát Mark down, především pro svoji jednoduchost a snadnost převodu na HTML. V budoucnu by mohla být tato stránka rozšířena o podporu více formátů. Student pak tuto stránku vidí jako statickou stránku s webovou reprezentací obsahu zadaného učitelem.



## Správa stránek

Název	Materiály
Adresa	materialy
Zobrazit v menu	Ano
<hr/>	
Název	Materiály k testu
Adresa	test-materials
Zobrazit v menu	Ne
<hr/>	
<input type="button" value="Přidat stránku"/>	

Obrázek 2.4: Wireframe - Správa stránek předmětu

## Vlastní stránka

<input type="button" value="H2"/> <input type="button" value="H3"/> <input type="button" value="H4"/> <input type="button" value="B"/> <input type="button" value="I"/> <input type="button" value="U"/> <input type="button" value="U"/> <input type="button" value="O"/> <input type="button" value="Zpět"/> <input type="button" value="Vpře"/> <input type="button" value="Odkaz"/>
Wysiwyg - input text...
<input type="button" value="Uložit"/>

Obrázek 2.5: Wireframe - Stránka pro tvorbu materiálů

## Aktuality

A paragraph of text with an [unassigned link](#).  
A second row of text with a [web link](#)

---

A paragraph of text with an [unassigned link](#).  
A second row of text with a [web link](#)

---

Zadat novou aktualitu

Obrázek 2.6: Wireframe - Stránka s aktualitami

## Testy

Zvolte paralelku ▼

<b>Název</b>	Test 1	<a href="#">Upravit</a>
<b>Popis</b>	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer malesuada. Duis ante orci, molestie vitae vehicula venenatis,	
<b>Odkazy</b>	www.materialy-k-testu.gov www.materialy-k-testu2.gov	
<b>Datum</b>	15.04.1993	
<b>Hodnocení</b>	Min: 5 - Max: 10	

---

Přidat test

Obrázek 2.7: Wireframe - Události předmětu

### Stránka s aktualitami 2.6

Na této stránce může učitel zadávat aktuality předmětu. Výhodou je, že o vytvoření nové aktuality budou všichni studenti předmětu informováni v rámci svých notifikací.

### Události předmětu 2.7

Událostí se rozumí jednotné označení pro přednášky, cvičení, prosemináře, laboratoře, testy, domácí úkoly a semestrální práce. Tato stránka vypadá

### Hodnocení předmětu

<b>Název</b>	Test 1	<a href="#">Upravit</a>
<b>Minimum</b>	5	
<b>Maximum</b>	10	
<b>Identifikátor</b>	test1	
<hr/>		
<b>Název</b>	Test 2	<a href="#">Upravit</a>
<b>Minimum</b>	5	
<b>Maximum</b>	10	
<b>Identifikátor</b>	test2	
<hr/>		
<b>Název</b>	Zápočet	<a href="#">Upravit</a>
<b>Minimum</b>	10	
<b>Vzorec</b>	test1 + test2	
<hr/>		

Přidat hodnocení

Obrázek 2.8: Wireframe - Hodnocení předmětu

pro všechny tyto události až na menší detaily stejně, takže bude popsána na stránce s testy. Událost má základní atributy: název, popis, paralelka a může být obohacena o libovolný počet materiálů. V opodstatněných případech může být k události přidán záznam na časovou osu, zpravidla u deadlinů a dat testů. Pokud je událost propojena s hodnocením, je zde zobrazeno i minimum a maximum bodů, které lze získat. Na stránce se zobrazují události podle paralelek, student zde defaultně vidí kombinaci všech svých paralelek. Vyučující má možnost přidávat události do paralelek, které vyučuje.

### Hodnocení předmětu 2.8

Existují dva druhy hodnocení:

- Standardní hodnocení - například testu nebo domácího úkolu, kde se definuje název, minimální počet dosažených bodů pro splnění, maximální možný počet dosažených bodů a identifikátor. Identifikátor slouží k podpoře výpočtů hodnocení
- Speciální hodnocení - mezi speciální hodnocení patří zápočet, klasifikovaný zápočet, zkouška nebo mezi-výpočet. U těchto hodnocení se definuje matematický vzorec, který jako proměnné využívá identifikátory standardních hodnocení.

## 2. ROZVAHA

---

**Pohled vyučujícího** Vyučující zde může vidět a zadávat jednotlivá hodnocení. Zároveň zde může přejít k hodnocení studentů. Hodnocení má dva formáty: Buď zadává všechna hodnocení pro jednoho studenta nebo může zadávat body z jednoho hodnocení všem studentům. To by mělo dopřát učitelům pohodlnější zadávání bodů (případ zadávání bodů z testu 1, kdy má opraveno 30 písemek oproti ústní zkoušce studenta, kdy chce vidět všechna jeho hodnocení v předmětu)

**Pohled studenta** Student má přehled o všech hodnocení a zároveň vidí své dosažené výsledky

### Hromadné hodnocení 2.9

**Hodnocení - Test 1**  
Minimum: 5 Maximum: 10

Zvolit paralelku ▼

Jméno studenta (username)	7
Jméno studenta (username)	5
Jméno studenta (username)	10
Jméno studenta (username)	2
Jméno studenta (username)	
Jméno studenta (username)	
Jméno studenta (username)	
Jméno studenta (username)	

Obrázek 2.9: Wireframe - Hodnocení testu

Tato stránka je přístupná pouze vyučujícím. Zadávají se zde body pro jedno hodnocení pro všechny studenty najednou. Studenti se zobrazují podle paralelek a jsou srovnáni podle abecedy. Je i možnost zobrazit studenty pro celý předmět.

### Individuální hodnocení 2.10

Tato stránka je přístupná pouze vyučujícím. Zadávají se zde body pro všechna hodnocení konkrétního předmětu a to vždy jednomu studentovi. Studenti se zde dají vyhledávat jak podle jména, tak podle username.

## Hodnocení studenta

### Zvolený student

Název	Test 1
Minimum	5
Maximum	10
Hodnocení	<input type="text" value="9"/>
<hr/>	
Název	Test 2
Minimum	5
Maximum	10
Hodnocení	<input type="text" value="7"/>
<hr/>	
Název	Zápočet
Minimum	10
Hodnocení	Z

Obrázek 2.10: Wireframe - Individuální hodnocení studenta

## Správa notifikací

U předmětu MI-NUR vám bylo uděleno nové hodnocení  Přečteno

Do předmětu Mi-PAR přibyla nová aktualita.  Přečteno  
Přečtěte si ji [zde](#).

Obrázek 2.11: Wireframe - Správa notifikací

### **Správa notifikací 2.11**

Uživateli se nové notifikace zobrazují vpravo nahoře pod ikonkou zvonečku. Tato stránka slouží jako správa notifikací, za 1. je zde uložena historie všech notifikací a za 2. zde může uživatel označit přečtenou notifikaci jako nepřečtenou a naopak.

---

## Realizace

Realizace této práce se skládá z návrhu architektury aplikace, popisu datového modelu, který slouží jako datová základna pro aplikaci a z popisu REST API, které slouží pro komunikaci s aplikací. Nejprve jsou ale představeny hlavní technologie, které se k realizaci používají.

### 3.1 Použité technologie

#### 3.1.1 Objektově relační mapování

Při vytváření aplikací v objektově orientovaném programovacím jazyce se programátor snaží co nejvěrněji zachycovat realitu. Reálné objekty reprezentuje jako jednotlivé instance tříd.

*Příklad: V reálném světě máme auta. Auta mají barvu, značku, SPZ. Konkrétní auto může být značky Škoda, mít červenou barvu a mít SPZ: KHH-3586. Ve světě aplikace existuje třída Auto, ta má atributy barva, značka a SPZ. Konkrétní instance té třídy bude mít hodnoty těchto atributů shodné s konkrétním autem z reálného světa*

V relační databázi jsou ekvivalencí třídy tabulky a ekvivalencí instance řádek v tabulce. Pokud pak chce programátor uložit instanci do databáze musí proběhnout konverze a přenos těchto dat mezi aplikací a databází.

Objektově relační mapování slouží k namapování tabulky relační databáze na objekt objektově orientovaného programovacího jazyku. To umožňuje programátorovi pracovat s databází pomocí jemu známého jazyka.

#### JDBC

Konkrétně v Javě je nejnižší vrstvou komunikace aplikace s databází rozhraní Java Database Connectivity (JDBC). JDBC zajišťuje spojení s databází a konverzi dat. S databází se komunikuje pomocí takzvaných statementů

### 3. REALIZACE

---

- Statement - Objekt používaný k provedení sql příkazu a k získávání jeho odpovědi [3]
- PreparedStatement - Dědí vlastnosti Statementu a uchovává sql příkaz, který se může provést vícekrát [4]
- CallableStatement - Dědí vlastnosti Statementu a používá se pro volání procedur databáze [5]

Při použití JDBC je programátor nucen psát SQL scripty, do kterých musí dosazovat jednotlivé atributy. Ve složitějších aplikacích to vede ke vzniku dlouhého a nepřehledného kódu

#### **JPA**

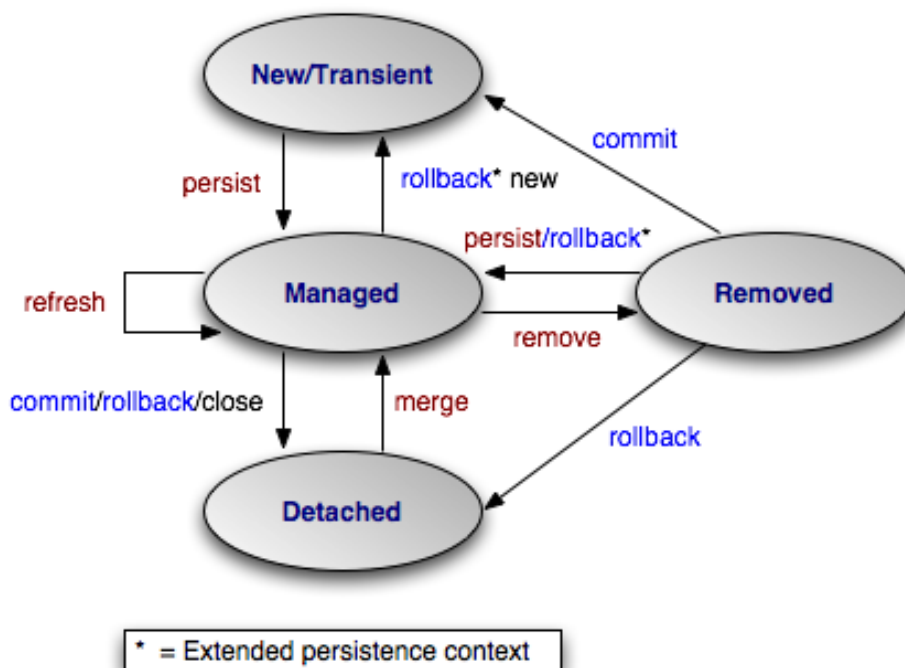
Java persistence API je framework jazyka java pro podporu objektově orientovaného mapování.

- Entita - objekt, který reprezentuje data v databázi.
- Entitní třída - odpovídá tabulce v databázi
- Instance entitní třídy - odpovídá jednomu řádku v databázi

#### **Požadavky na entitní třídy [6]**

- Třída musí být anotována anotací `javax.persistence.Entity` annotation.
- Třída musí mít konstruktor bez argumentů s viditelností `public` nebo `protected`.
- Třída nesmí být deklarována `final`, její metody a proměnné rovněž `final`.
- Pokud je instance entity předávána hodnotou, musí implementovat rozhraní `Serializable`.
- Entities may extend both entity and non-entity classes, and non-entity classes may extend entity classes.
- Atributy třídy musí být deklarovány `private`, `protected` nebo `package-private`. Přístupuje se k nim pomocí metod (getterů, setterů)





Obrázek 3.1: Životní cyklus entity [7]

**Životní cyklus entity** [7]

O životní cyklus entity 3.1 se stará EntityManager. Stav Entity je určen akcí, kterou EntityManager provede. Základní metody pro práci s EntityManagerem jsou:

- Persist - uloží instanci entity do databáze (obdoba INSERT)
- Remove - smaže instanci entity z databáze (obdoba DELETE)
- Merge - obnoví záznam v databázi podle změn instance entity (obdoba UPDATE)
- Find - najde záznam v databázi podle id (obdoba SELECT)

**Multiplicity** Existují 4 typy multiplicit

- One-to-one - vztah 1 ku 1. Obdoba vztahu volant a auto. Auto má jeden volant a volant má právě jedno auto

### 3. REALIZACE

---

- Many-to-one - vztah n ku 1. Podobně jako dítě a matka. Dítě má jen jednu matku, ale matka může mít více dětí. (Z pohledu dítěte)
- One-to-many - vztah 1 ku n. Vztah dítě a matka z pohledu matky
- Many-to-many - vztah n ku m. Podobně jako vztah studentů a předmětů. Student studuje více předmětů a zároveň více studentů navštěvuje jeden předmět

**Směr ve vztahu entit** Existují dva druhy vztahů, co se týče směru

- Jednosměrný - pouze jedna entita vidí na entitu druhou. To znamená, že tato vazba může být vytvořena nebo odstraněna pouze z jedné z entit
- Obousměrný - Obě entity na sebe navzájem vidí. o znamená, že tato vazba může být vytvořena nebo odstraněna z jakékoliv z nich

#### **Hibernate**

Hibernate je konkrétní implementace Java Persistence API. V hibernate se jako alternativa k SQL skriptům používá takzvaný Hibernate Query Language (HQL) nebo Criteria Query. Výhodou je, že je hibernate nezávislý na konkrétní databázi, takže pro programátora není rozdíl, jestli používá například postgres nebo oracle.

#### **3.1.2 Spring framework**

Spring framework je komplexní aplikační rámec pro vývoj v jazyce Java. Spring se zaměřuje na vývoj enterprise aplikací a usnadňuje práci s frekventovaně používanými technologiemi. Je to velice obsáhlý projekt, ale díky jeho vysoké modularitě může programátor využívat pouze jeho součásti, která pro daný projekt potřebuje.

#### **Dependency Injection and Inversion of Control**

Jádro springu je postaveno na návrhovém vzoru inversion of control. V tomto návrhovém vzoru je zodpovědnost za vytváření a propojování objektů přesunuta z aplikace na framework. Framework starající se o inversion of control se nazývá kontejner. Objekty jsou vytvářeny většinou na základě xml konfiguračního souboru, popřípadě mohou být definovány anotacemi. Objekty vytvořené kontejnerem jsou nazývány beans. Tyto objekty jsou poté získávány pomocí dependency injection (vsazování závislostí), což je konkrétní technika inversion of control.

Následující ukázka tento princip demonstruje. Třída ParallelController nemá v době sestavování programu referenci na instanci třídy ParallelService, přesto jí ale bude v době jejího volání mít.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans">
  <bean class="cz.cvut.fit.student.portal.service.impl.
    CourseServiceImpl"/>
  <bean class="cz.cvut.fit.student.portal.service.impl.
    ParallelServiceImpl"/>
  <bean class="cz.cvut.fit.student.portal.service.impl.
    SemesterServiceImpl"/>
</beans>
```

Ukázka kódu 3.1: Příklad XML konfiguračního souboru

```
@RestController
@RequestMapping(produces = "application/json")
public class ParallelController {
  @Autowired
  private ParallelService parallelService;

  @RequestMapping(value = "/courses/{courseCode}/parallels",
    method = RequestMethod.GET)
  public List<ParallelDto> getParallelsForCourse(@PathVariable("
    courseCode") String courseCode) {
    return parallelService.getParallelsForCourse(courseCode);
  }
}
```

Ukázka kódu 3.2: Příklad vkládání závislosti

## Spring boot

Spring boot[8] umožňuje vytvářet aplikace postavené na springu, které se dají samostatně spustit, protože mají díky spring bootu jeden z webových serverů (tomcat, jetty, undertow) zabudovaný přímo v sobě. Tato vlastnost velice usnadňuje vývoj (rychlé spouštění) a případné nasazování.

### 3.1.3 REST

REST [9] (Representational State Transfer) je architektura navržená pro distribuované prostředí. Distribuovaným prostředím se rozumí, že části programu běží na různých strojích a pro komunikaci mezi sebou využívají síť. Rest je konstruován datově, nikoliv procedurálně, tím se rozumí, že nevolá metody, které by prováděly nějakou akci, nýbrž provádí akce na zdrojích. Tyto akce provádí pomocí HTTP metod.

## HTTP metody

HTTP metody mohou mít následující vlastnosti:

- Bezpečné metody - nemění zdroj se kterým pracují

### 3. REALIZACE

---

- Idempotentní metody - nehledě na to kolikrát bude metoda zavolána, výsledek bude vždy stejný

#### Seznam metod [10]

- GET - používá se pro čtení dat na dané adrese. Tato metoda je bezpečná i idempotentní
- POST - používá se k vytváření popřípadě k modifikaci zdroje. Metoda není bezpečná ani idempotentní
- PUT - používá se podobně jako POST, ovšem pokaždé specifikuje id zdroje. Metoda není bezpečná a je idempotentní
- DELETE - používá se k mazání zdroje. není bezpečná a je idempotentní
- HEAD - stejná jako metoda GET, ale dostane jako odpověď pouze hlavičku, nikoliv tělo. Je bezpečná i idempotentní
- OPTIONS - získá informaci o možnostech komunikace na zadaném zdroji (seznam podporovaných metod. Je bezpečná i idempotentní
- PATCH - mění jen podmnožinu atributů zdroje. Není bezpečná a je idempotentní

**Rozdíl mezi POST a PUT** Metody POST a PUT na první pohled působí stejně a není zřejmý jejich rozdíl. Demonstrace bude ukázána na zdroji `/tests`. Na tomto zdroji se nachází seznam testů.

#### Vytvoření nového testu

- POST - provede metodu na adrese `/tests`. Server vytvoří nový test a dá mu podle svých pravidel id. Pokud by ta samá metoda byla provedena vícekrát po sobě, bude vytvořeno více shodných testů ovšem pokaždé s jiným id
- PUT - provede metodu na adrese `/tests/{specifické id}`. Server vytvoří nový test s id, které bylo specifikováno v adrese. Pokud se metoda zavolá vícekrát, potom se při prvním volání test vytvoří a při dalších modifikuje, ovšem beze změn, takže výsledek bude pokaždé stejný

#### Principy RESTu

- Adresovatelnost - každý zdroj by měl být dosažitelný podle unikátního identifikátoru (URI). URI má standardizovanou podobu:

*scheme://authority/path?queryString#fragment*

- scheme - určuje, o jaký druh URI se jedná a jaký význam a syntaxe platí pro zbytek URI
  - authority - identifikátor zdroje v nějaké hierarchické struktuře
  - path - nehierarchická část identifikátoru, specifikuje vlastnosti požadovaného zdroje. Používá se například pro filtrování
  - queryString - odkazuje na druhotný zdroj v rámci primárního zdroje (například kapitola knihy)
- Jednotné rozhraní - jednotné rozhraní bylo popsáno v sekci HTTP metody
  - Orientace na reprezentaci - při posílání a přijímání zdroje je použita zpravidla standardní reprezentace (JSON, XML,..)
  - Bezstavová komunikace - žádná klientská data nejsou uchovávána na serveru

### 3.1.4 Angular JS

Angular JS [11] je javascriptový webový framework, který se zaměřuje na tvorbu single-page aplikací.

#### Vlastnosti angularu

- MVC - angular využívá návrhového vzoru MVC, kde vrstva view je tvořena html, vrstva controller je javascriptový objekt a model je reprezentován atributy controlleru
- Two Way Data-Binding - synchronizace dat mezi view a modelem
- Dependency Injection - díky dependenci injection stačí definovat jaké komponenty chceme uvnitř jiné používat a o jejich dosazení se postará framework
- Komponenty - angular podporuje tvorbu přepoužitelných komponent

#### Výhody angularu

- Možnost čistě a jednoduše tvořit single page application
- Testovatelnost unit testy
- Podporuje dependency injection a dodržuje princip oddělení zájmů
- Přepoužitelné komponenty
- Two Way Data-Binding

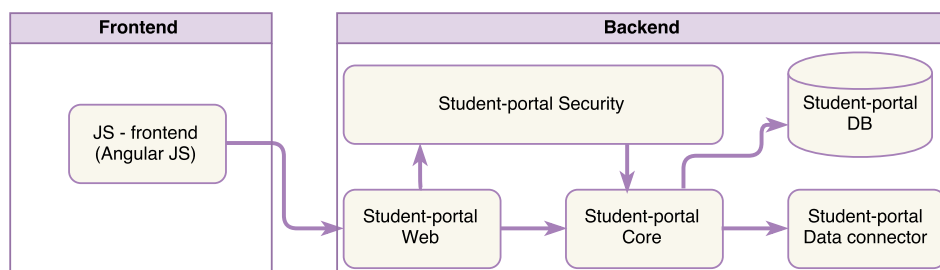
### 3. REALIZACE

---

#### Nevýhody angularu

- Bezpečnost - jako každý javascriptový framework nemůže zaručit bezpečnost, proto je nutné pravidla bezpečnosti implementovat hlavně i na serveru
- Složitější a více invazivní framework (například oproti React JS)

### 3.2 Návrh architektury



Obrázek 3.2: Schéma architektury aplikace

Architektura se jako u každé webové aplikace dá rozdělit na dvě základní části: backend a frontend. Aplikace je rozčleněna do několika modulů. Komunikace těchto modulů je naznačena na schématu 3.2 pomocí šipek.

#### Data connector

Tento modul je zodpovědný za získávání dat ze školních systémů. Typicky získává informace o předmětech, uživatelích, semestrech, apod. V případě ČVUT tento modul komunikuje převážně s KOS-API a s USERMAP-API. Velká výhoda existence tohoto modulu spočívá v tom, že data z něj se posílají ve vcelku standardní formě, které může docílit jakákoliv jiná univerzita/fakulta, což znamená že aplikace je navržena tak, aby se dala případně použít i mimo FIT ČVUT.

#### Core

Tento modul obstarává především komunikaci s databází. Dále předává data získávaná z externích systémů univerzity.

#### Security

Tento modul obstarává veškerou bezpečnost na backendu aplikace. Udržuje session s uživatelem a na ní skladuje informace o něm. Pomocí těchto informací rozhoduje, které akce je uživatel oprávněn provádět. Existence tohoto modulu velice ulehčuje přehled nad bezpečností aplikace. Velice snadno se získávají stávající pravidla bezpečnosti, modifikují i vytvářejí nová.

#### Web

Tento modul vystavuje RESTové api, se kterým komunikuje frontend aplikace. Při každé akci, která musí být autorizována se táže security modulu, zda je uživatel pro akci oprávněn. Při udělení oprávnění získá data z modulu core popřípadě je tam odešle.

### 3.3 Návrh datového modelu

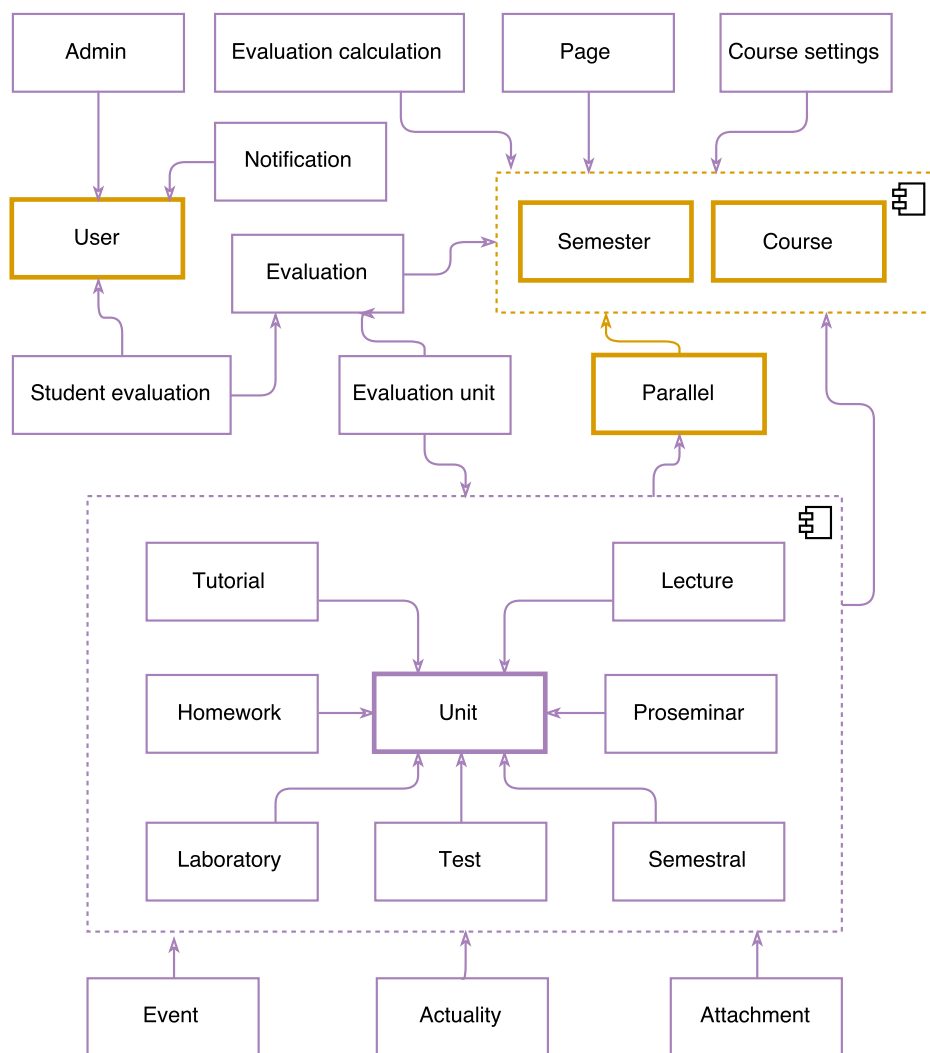
Oranžově zvýrazněné objekty jsou získávány z externích systémů. Fialové objekty jsou součástí aplikace.

#### Popis objektů

- User - informace o uživateli - jméno, tituly, e-mail
- Course - popis předmětu
- Semester - informace o semestru. Díky uchování této informace si uživatel bude moci prohlížet profil předmětů v minulých semestrech
- Parallel - specifikuje konkrétní paralelku předmětu. Díky vazbě událostí na konkrétní paralelku se dají vytvářet události specifické pouze pro onu paralelku (testy, domácí úkoly,..)
- Unit - tento objekt představuje společnou část pro "události" předmětu (cvičení, testy, úkoly,..). Ukládají se do něj základní informace jako název a popis
- Event - specifikuje deadline nebo datum události. Tento objekt je základ pro tvorbu časové osy, díky které bude mít student přehled o svém studiu
- Actuality - aktuality předmětu. Jednak budou zobrazeny u každého předmětu, zároveň o nic bude zaslána notifikace každému studentovi
- Attachment - příloha k "události" přidaná pomocí odkazu (v budoucnu se předpokládá rozšíření o nahrávání souborů)

### 3. REALIZACE

---



Obrázek 3.3: Návrh datového modelu



- Evaluation - hodnocení předmětu, jeden záznam odpovídá jednomu specifickému hodnocení (konkrétní test, zkouška,..). Set hodnocení se společným předmětem a semestrem potom reprezentuje hodnocení předmětu pro daný semestr
- Evaluation calculation - používá se pro výpočet hodnocení. Zadává se do něj vzorec, kde jednotlivé proměnné odpovídají konkrétním hodnocením. Výsledkem může být mezivýsledek (součet bodů ze všech testů atp.) nebo zápočet, či známka. Zápočet je dán splnění podmínky pro minimum bodů a známka se vypočítá podle standardní stupnice ČVUT
- Student evaluation - ohodnocení studenta. Odpovídá jednomu hodnocení pro předmět
- Evaluation unit - spojovací objekt mezi hodnocením a "událostí". Tím se hodnocení propíše například přímo ke konkrétnímu testu
- Notification - upozornění uživateli. Aplikace bude s uživatelem tímto způsobem komunikovat (upozorňovat ho na aktuality, změny v hodnocení,..)
- Page - vytvořená stránka předmětu učitelem. Učitelé budou mít možnost vytvářet tímto způsobem materiály k výuce nebo přidat stránku do menu předmětu
- Course settings - nastavení předmětu. Základní nastavení předmětu (Bude mít předmět cvičení? Bude mít semestrální práci? atp.)

## 3.4 Popis REST API

komunikace mezi backendem a frontendem probíhá pomocí REST API. Je proto nutné toto api popsat. Protože se ale jedná pouze o technický popis, který je velice jednotvárný, je tato sekce přesunuta mezi přílohy C.



---

# Uživatelská příručka

Uživatel může mít v rámci portálu dvě základní role: Student a Vyučující. Tato uživatelská příručka poskytuje návod na využívání funkcí, které aplikace nabízí.

## 4.1 Seznam pojmů

Nejprve je důležité definovat pojmy, se kterými bude příručka pracovat:

- Portál - rozumí se jím označení pro tuto webovou aplikaci
- Aplikace - synonymum k pojmu portál (v rámci tohoto návodu)
- Událost - jednotný název pro jednu z možností: Přednáška, Cvičení, Laboratoř, Proseminář, Test, Domácí úkol, Semestrální práce
- Deadline/Datum - Záznam na časovou osu, pojí se s jednou z událostí.
- Stránka předmětu - v základu má předmět ve své postranní navigaci sekce: Přednášky, Cvičení, Domácí úkoly, atp. Často se ale stává, že chce vyučující do navigace předmětu přidat vlastní stránku, tato stránka bude v návodu nazývána stránka předmětu
- Menu předmětu - rozumí se jím postranní navigace, která slouží jako rozcestník mezi jednotlivými sekcemi předmětu
- Hodnocení - rozumí se jím ohodnocení jednoho úkonu (test, zkouška, domácí úkol,..)
- Speciální hodnocení - speciální hodnocení má vždy jako hodnotu vzorec, který využívá jako proměnné jednotlivé hodnocení předmětu (i ostatní speciální hodnocení). Patří mezi ně: Výpočet známky, Zápočet, Klasifikovaný zápočet, Mezi-výpočet (vzorec může vypadat například takto:

#### 4. UŽIVATELSKÁ PŘÍRUČKA

---

"test1 + test2 + hw1 + zkouska", což reprezentuje součet jednotlivých hodnocení)

- Mezi-výpočet - tato jednotka slouží pro pokročilejší uživatele (vyučující), kteří pro finální hodnocení používají nějaký složitější vzorec. Příklad: předmět MI-FME využívá vzorec:  $(0.4 * \text{body-cviceni} + 0.6 * \text{body-zkouska})$  - mezi-výpočet by tedy sloužil k vypočítání bodů ze cvičení.
- klasifikační stupnice ČVUT - na ČVUT je předepsáno, že hodnocení předmětu musí být reprezentováno 100 bodovou stupnicí, kde pro splnění předmětu je potřeba získat alespoň 50 bodů. Známkování je pak následující:

–  $<50$  - F

–  $<50, 60)$  - E

–  $<60, 70)$  - D

–  $<70, 80)$  - C

–  $<80, 90)$  - B

–  $\geq 90$  - A

## 4.2 Návod pro vyučující

Tato část popisuje funkce aplikace z pohledu vyučujícího.

### Nastavení předmětu 4.1

První, co vyučující musí udělat, je zadefinovat, jakou strukturu předmět pro aktuální semestr má. To se provede snadno v sekci Nastavení předmětu. Po uložení struktury se v menu předmětu zobrazí jednotlivé sekce.

## NASTAVENÍ PŘEDMĚTU - BI-LIN

Přednášky	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Cvičení	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Prosemináře	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Laboratoře	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Domácí úkoly	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Testy	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano
Semestrální práce	<input checked="" type="checkbox"/> Ne	<input type="checkbox"/> Ano

✓ Uložit změny

Obrázek 4.1: Screenshot - Nastavení předmětu

### Správa stránek 4.2

Vyučující má možnost přidat do menu předmětu vlastní stránku (například odkaz na facebookovou skupinu předmětu). Další možnost je vytvořit novou stránku přímo v aplikaci, v tom případě musí mít adresa striktně následující formát:

- smí obsahovat písmena
- smí obsahovat čísla
- smí obsahovat znak "-"
- nesmí obsahovat nic jiného

Příklady validních adres:

- moje-materialy
- test1
- cviceni1-handouty

## SPRÁVA STRÁNEK - BI-LIN

---

Stránka	Facebook předmětu	<input type="button" value="Upravit"/>
Zobrazit v menu	Ano	

---

Název	<input type="text"/>
Adresa	<input type="text"/>
Zobrazit v menu	<input type="button" value="Ne"/> <input type="button" value="Ano"/>

---

<input type="button" value="✓ Zrušit"/>	<input type="button" value="✓ Uložit změny"/>
---	---

Obrázek 4.2: Screenshot - Správa stránek

Příklady nevalidních adres:

- test1/materialy
- první test

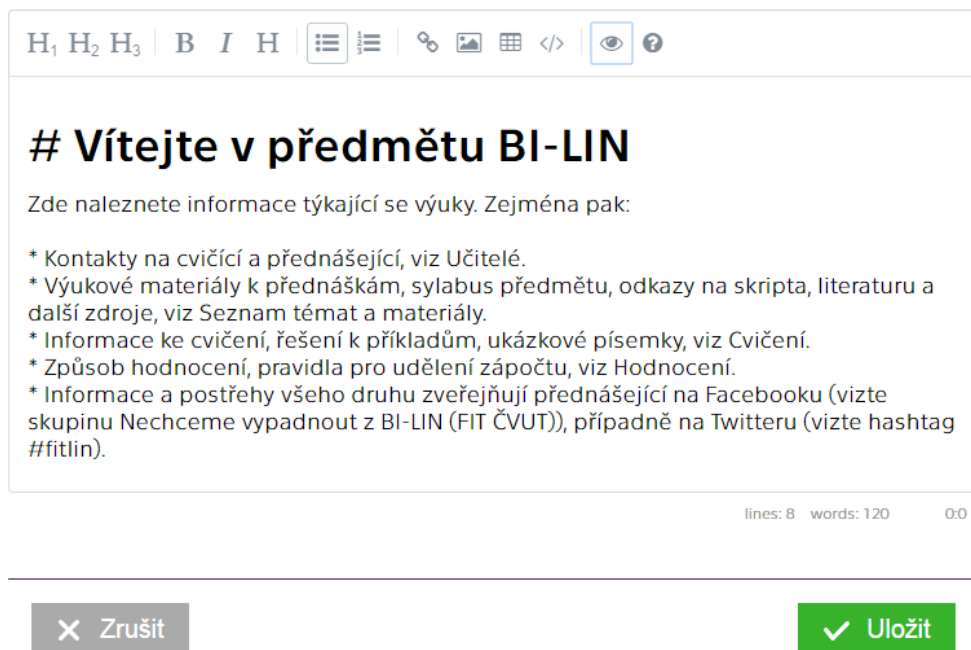
Tato stránka se může přidat přímo do menu předmětu, popřípadě se může použít její odkaz a vložit jako materiály k některé z událostí.

### Tvorba stránky 4.3

Vlastní stránky se tvoří pomocí syntaxe markdown, pro pohodlí je tvorba stránky podpořena textovým editorem. Markdown je jednoduchý formát podporující základní funkce pro tvorbu stránek. Podporuje vkládání odkazů, obrázků, kódu, atp. V textovém editoru si uživatel může jednoduše prohlédnout, jak bude výsledek vypadat. Jako první by měl takto vyučující vyplnit úvodní stránku předmětu, to bude první stránka, se kterou se student předmětu setká.

### Stránka událostí

Jak už bylo zmíněno ve slovníku pojmů události jsou Testy, Domácí úkoly, Přednášky,.. Návod bude ukázán na domácích úkolech, ale až na menší detaily se stránky neliší.



Obrázek 4.3: Screenshot - Tvorba stránky

#### Zobrazení existujících událostí 4.4

Události se zobrazují podle paralelek. Uživatel si může zobrazit událost pro konkrétní paralelku nebo události, které jsou pro celý předmět. Student potom ve svém přehledu uvidí sjednocení událostí pro celý předmět + pro jeho konkrétní paralelku/y.

#### Vytváření události 4.5

Událost má základní parametry, které jí popisují. K události se může formou odkazů připojit neomezené množství materiálů 4.6. Pokud je událost hodnocená body a má datum konání nebo deadline, měla by být pro studenta zvýrazněna, proto existuje možnost přidat tuto událost na tak zvanou časovou osu 4.7. Dá se tam přidat vícekrát, což je z důvodu podpory soft/hard deadlineů. Na časovou osu by se rozhodně NĚměly přidávat běžné události (jako například každá přednáška), tyto události se běžně nachází v rozvrhu hodin a pro studenta nejsou ničím vyjimečné. Mezi opodstatněné případy může patřit například to, že přijede zcela unikátně přednášet někdo významný, čistě na jednu přednášku.

## DOMÁCÍ ÚKOLY PŘEDMĚTU: BI-LIN

---

Zobrazit pro ?

---

**Domácí úkol 1** Upravit

**Popis** Vypracujte příklady 1-5, 7, 8 a 10. Ostatní příklady jsou nepovinné, ale doporučené.

**Odkazy**  →

**Deadline** 26.05.2017 : 23:59:00

---

Obrázek 4.4: Screenshot - Zobrazení událostí

---

**Paralelka** ?

**Nadpis**

**Popis**

+ Přidat deadline + Přidat odkaz

---

✓ Zrušit ✓ Uložit změny

---

Obrázek 4.5: Screenshot - Vytváření události



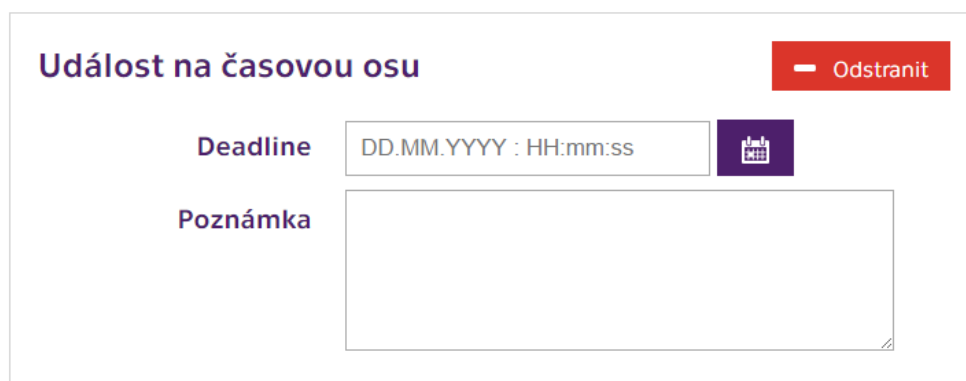


**Odkaz** — Odstranit


**Nadpis**

**Adresa**

Obrázek 4.6: Screenshot - Přidávání materiálů



**Událost na časovou osu** — Odstranit

**Deadline**  

**Poznámka**

Obrázek 4.7: Screenshot - Přidávání záznamu na časovou osu

## Hodnocení předmětu 4.8

V hodnocení předmětu učitel zadává jednotlivá hodnocení (to může být cokoliv, za co studenti získávají body - test, úkol, zkouška,..). Standardní hodnocení má:

- Minimum bodů - minimální počet bodů, které musí student získat, aby mohl úspěšně ukončit předmět
- Maximum bodů - horní hranice možných získaných bodů
- Identifikátor - identifikátor je nutné vyplnit. Slouží k zapsání vzorců pro výpočet známek, zápočtů. Musí splňovat tyto pravidla:
  - každý identifikátor musí být unikátní
  - žádný identifikátor v sobě nesmí obsahovat jiný identifikátor (jako například: test1, test10)

Dále je možnost zadat několik speciálních hodnocení 4.9. Patří mezi ně:

**Test 1** Ohodnotit Propojit s událostí ? Upravit

Identifikátor ? test1

Minimum bodů 0

Maximum bodů 25

---

**Test 2** Ohodnotit Propojit s událostí ? Upravit

Identifikátor ? test2

Minimum bodů 0

Maximum bodů 25

---

**Domácí úkol 1** Ohodnotit Propojit s událostí ? Upravit

Identifikátor ? hw1

Minimum bodů 0

Maximum bodů 25

Propojené události Domácí úkol 1 - Cvičení č.22 ×

---

+ Přidat hodnocení → Hodnotit studenty

Obrázek 4.8: Screenshot - Hodnocení předmětu

**Finální známka**[Upravit](#)

<b>Identifikátor</b> ?	final
<b>Vzorec</b> ?	test1 + test2 + hw1 + zkouska
<b>Typ hodnocení</b>	Známka

Obrázek 4.9: Screenshot - Ukázka speciálního hodnocení

**Zápočet**

- Vzorec - matematický vzorec, který používá jako proměnné identifikátory jednotlivých hodnocení. Příklad: (test1 + test2 + hw1)
- Minimum bodů - definuje počet bodů, které musí student získat, aby splnil zápočet. Příklad:(Součet bodů z testu 1, testu 2 a domácího úkolu musí být větší než zadané minimum)
- identifikátor - přiřadí zápočtu identifikátor, který v dalších výpočtech může nahradit vzorec zápočtu

**Klasifikovaný zápočet a Známka**

- Vzorec - matematický vzorec, který používá jako proměnné identifikátory jednotlivých hodnocení. Příklad: (test1 + test2 + hw1).
- Známka - známka je vypočtena z výsledku vzorce podle standardní klasifikační stupnice ČVUT

**Mezi-výpočet**

- Vzorec - matematický vzorec, který používá jako proměnné identifikátory jednotlivých hodnocení. Příklad: (test1 + test2 + hw1).
- Minimum bodů - minimální počet bodů, které musí student získat, aby mohl úspěšně ukončit předmět
- identifikátor - přiřadí mezi-výpočtu identifikátor, který v dalších výpočtech může nahradit vzorec tohoto mezi-výpočtu

## HODNOCENÍ PŘEDMĚTU BI-LIN - TEST 1

Paralelka	Cvičení č. 1
Jan Hloušek (hlousja5)	0
Peter Kolárovec (kolarp33)	0
Martin Koliba (kolibma1)	0
Vít Černý (cernyvi6)	0
David Labský (labskdav)	0
Antonín Vlček (vlcekan1)	0
Roman Hodulák (hodulrom)	0
Michal Hošna (hosnamic)	0
Patrik Jantošovič (jantopat)	0
Martin Pavlíček (pavlim45)	0

Obrázek 4.10: Screenshot - Hromadné hodnocení

### Hodnocení studentů

Existují dva způsoby hodnocení studentů. Za prvé je možnost zadávat body jednomu konkrétnímu hodnocení, což je výhodné při hromadném zadávání bodů například z jednoho testu. Za druhé hodnocení individuálního studenta, kde je najednou zobrazeno všechno jeho hodnocení, to se vyplatí například při individuálním ústním zkoušení.

Na hromadné hodnocení 4.10 se vyučující dostane v sekci Hodnocení předmětu přímo u konkrétního hodnocení (žluté tlačítko "Ohodnotit"). Na indivi-

**Bc. Zdeněk Balák****Test 1****Hodnocení** Minimum bodů **0** Maximum bodů **25****Studentovy body**

20

**Test 2****Hodnocení** Minimum bodů **0** Maximum bodů **25****Studentovy body**

15

**Domácí úkol 1****Hodnocení** Minimum bodů **0** Maximum bodů **50****Studentovy body**

30

**Finální známka****Studentovy body** 65**Studentova známka**

D

Obrázek 4.11: Screenshot - Individuální hodnocení

duální 4.11 hodnocení studentů se vyučující dostane v sekci Hodnocení předmětu dole na stránce (žluté tlačítko "Hodnotit studenty")

### 4.3 Návod pro studenty

Tato sekce popisuje důležité části aplikace z pohledu studenta

#### Přehled o studiu

Přehled o studentově studiu je přímo na úvodní stránce aplikace. Je zobrazen na časové ose 4.12, která je seřazena podle data. Shrnuje všechny důležité termíny, o kterých by měl student mít přehled.

### MOJE UDÁLOSTI

---



Obrázek 4.12: Screenshot - Ukázka časové osy

#### 4.3.1 Moje hodnocení

Hodnocení student najde v navigaci pod pojmem "Moje hodnocení". Najde zde předpis pro hodnocení předmětu (co všechno bude v předmětu ohodnoceno?) a zároveň zde uvidí svoje dosažené výsledky.

#### 4.3.2 Materiály k předmětům

Základní struktura předmětu obsahuje: Přednášky, Cvičení, Prosemináře, Laboratoře, Domácí úkoly, Testy, Semestrální práce nebo jejich podmnožinu. Navíc se u každého předmětu nachází sekce Moje hodnocení. Další stránky může vytvářet vyučující zcela libovolně, takže jejich obsah nemůže být součástí uživatelské příručky.

Stránky událostí mají stejnou strukturu, proto budou všechny demonstrovány na stránce s testy 4.14.

Defaultně se studentovi zobrazují všechny události relevantní právě pro něj (možnost Všechny moje paralelky). Je to kombinace událostí pro všechny a událostí pro paralelky, které student navštěvuje. Prohlížet si ale může i události ostatních paralelek.

## MOJE HODNOCENÍ PŘEDMĚTU - BOB13SPE

---

### Test 1

---

**Hodnocení** Minimum bodů **0** Maximum bodů **25**

**Moje hodnocení** 25

### Test 2

---

**Hodnocení** Minimum bodů **0** Maximum bodů **25**

**Moje hodnocení** 25

### Test 3

---

**Hodnocení** Minimum bodů **0** Maximum bodů **25**

**Moje hodnocení** 0

### Zkouška

---

**Hodnocení** Minimum bodů **15** Maximum bodů **25**

**Moje hodnocení** 15

### Výsledné hodnocení

---

**Moje hodnocení** 65

**Moje známka** **D**

Obrázek 4.13: Screenshot - Moje hodnocení předmětu

## TESTY K PŘEDMĚTU: BOB13SPE

---

Zobrazit testy pro ?

Všechny moje paralelky

### Test 1

**Popis** Podélný návar elektrickou svářečkou "housenky" na železnou desku.

**Odkazy** [Materiály k testu](#)



**Datum** 14.07.2017 : 00:00:00

**Hodnocení** Minimum bodů: **0** - Maximum bodů: **25**

---

### Test 2

**Popis** Koutový svár prováděný ve vodorovné pozici. TEST

**Datum** 18.04.2017 : 12:00:00

**Hodnocení** Minimum bodů: **0** - Maximum bodů: **25**

---

Obrázek 4.14: Screenshot - Testy předmětu



---

# Testování

## 5.1 Heuristická analýza

Heuristika obecně znamená zkusmé řešení problémů, pro něž neznáme algoritmus nebo přesnější metodu. Heuristické řešení je často jen přibližné, založené na poučeném odhadu, intuici, zkušenosti nebo prostě na zdravém rozumu. Heuristika nezaručuje nejlepší řešení, zato je univerzálně použitelná, jednoduchá a rychlá.

Pro otestování uživatelského rozhraní aplikace byla použita Nielsenova heuristická analýza [12]. Nielsenova heuristická analýza je tvořena deseti kritérii, které by měla práce splňovat:

### 5.1.1 Viditelnost stavu systému

- Systém nesmí zůstat zamrzlý a nereagovat na uživatelské vstupy
- Zobrazit progress bar
- Uživatel musí být informován o tom, co system dělá

### Nálezy

- Aplikace obsahuje modální dialogy, pro potvrzování nevratných akcí
- Aplikace obsahuje notifikační dialogy, pro oznámení výsledku akce.
- Při uložení nastavená předmětu notifikační dialog chybí
- Při uložení události je sice zřejmé, že editace události skončila, ale pro ujištění uživatele by měl být úspěch akce potvrzen notifikačním dialogem

### Navržené řešení

- Na zmíněná místa bude přidán notifikační dialog

### 5.1.2 Shoda mezi systemem a realitou

- Zachování konvencí (a metafor) reálného světa
- Ikony (a metafory) se musí chovat jako to, co zobrazují/na co odkazují

#### Nálezy

- Ikony horního menu odpovídají konvencím. Zvoneček pro notifikace, lupa pro vyhledávání a ozubené kolečko pro nastavení
- U časové osy jsou přednášky zobrazeny stejným zvonečkem jako notifikace

#### Navržené řešení

- Ikony časové osy budou zrevidovány a vybrány vhodnější

### 5.1.3 Minimální zodpovědnost

- Vždy je možnost vrátit se zpět do předchozího stavu
- Potvrzování akcí
- Varování před provedením nevratné akce

#### Nálezy

- V aplikaci fungují potvrzovací dialogy, které před každou nevratnou akcí uživatele upozorní

### 5.1.4 Shoda s použitou platformou a obecnými standardy

- Pokud to jde, použít standardní systémové komponenty
- Systémové barvy a fonty
- Používat stejné termíny

#### Nálezy

- Aplikace využívá font technika předepsaný grafickým manuálem ČVUT
- Logo aplikace odpovídá grafickému manuálu ČVUT
- U ikonek v menu s popisem se u popisu objevuje jiný font než ve zbytku aplikace
- Aplikace odpovídá standardům pro rozložení webové aplikace (přihlášení vpravo nahoře, vyhledávání nahoře uprostřed,..)

### Navržené řešení

- Text u ikoněk bude změněn na font odpovídající grafickému manuálu

#### 5.1.5 Prevence chyb

- Uživatel by neměl být možnost zadat špatnou hodnotu
- Pokud je způsob zadávání z principu složitější, je třeba to na místě vysvětlit (in-line help text)
- Zvýraznit povinné položky formulářů

### Nálezy

- Učitel může zadat chybný vzorec do výpočtu hodnocení
- U hůře pochopitelnějších atributů formulářů se nachází nápovědy
- Povinné položky formulářů se zvýrazní teprve při pokusu o odeslání
- Při vytváření vlastní stránky je možné zadat neplatnou adresu

### Navržené řešení

- Aplikace projde větší revizí kontroly vstupů a nalezené problémy budou odstraněny
- Povinné položky formuláře budou označeny
- U vytváření stránky bude vysvětlivka, která vysvětluje, jak má vypadat adresa. Zároveň bude přidána kontrola zadané adresy

#### 5.1.6 Kouknu a vidím

- Nezatěžovat uživatelskou paměť
- Akce, které uživatel může momentálně provést by měly být viditelné a snadno dosažitelné
- Zvýraznit povinné položky formulářů
- Pozice uživatele
- Krok ve wizardu

### Nálezy

- V menu je zvýrazněná stránka na které se uživatel nachází
- Menu by potřebovalo lépe rozčlenit. Oddělit stránky pro správu předmětu a ty, které uvidí studenti

### Navržené řešení

- Rozšířit grafický návrh menu, aby podporovalo členění do skupin

#### 5.1.7 Flexibilita a efektivita

- Zkušený vs. běžní uživatelé
- To co zkušený uživatel pravidelně potřebuje, běžný kolikrát ani nepotřebuje vědět
- Pokročilý (Advanced) mód
- Klávesové zkratky / funkční klávesy
- Klonování existujících záznamů (templates)

### Nálezy

- Vzhledem k ranosti aplikace ještě žádné pokročilé akce neexistují. V budoucnu přibudou akce typu: Uložit předmět ve formátu yaml, Nastavení předmětu z konfiguračního souboru, Nahrání hodnocení pomocí formátu csv

#### 5.1.8 Minimalita

- Zobrazovat pouze informaci, která aktuálně k něčemu opravdu je
- Čím méně možností uživatel má, tím rychleji koná
- (Nepotřebná) grafika by neměla zastiňovat ovládání a účel

### Nálezy

- Aplikace kritéria splňuje

#### 5.1.9 Smysluplné chybové hlášky

- Nejlepší je nedojít do stavu kdy je třeba chybového hlášení
- Chybové hlášení musí být v běžném jazyce – žádné kódy
- Chybové hlášení by mělo popsat co se stalo špatně, jak se to stalo a jak tomu příště předejít

### Nálezy

- Při stisknutí tlačítka přihlásit se na jiného uživatele (možnost pro admina) je uživatel upozorněn, pokud nevybral uživatele
- Při hodnocení studentů je uživatel upozorněn, pokud nevybral uživatele
- Aplikace sice nehlásí nesmyslné hlášky, ale není například připravena na výpadek aplikací, které slouží jako zdroj dat

### Navržené řešení

- Připravit aplikaci na výpadek datových zdrojů a upozornit uživatele na nefunkčnost

#### 5.1.10 Nápověda a dokumentace

- Systém by měl být použitelný i bez nápovědy, ale ta by neměla chybět
- Nápověda by měla ukazovat příklady spíše než popisovat funkce
- V nápovědě by mělo jít vyhledávat

### Nálezy

- Jako uživatelská příručka slouží tato práce
- V aplikaci nejsou návody

### Navržené řešení

- Podrobně popsat funkce aplikace v této práci a rozšiřovat je i po odevzdání
- Přidat sekci návodů přímo do aplikace

## 5.2 Testování s uživateli

Uživatelské testování je nejkvalitnější způsob testování aplikace. Uživatelé, kteří systém neznají a nepomáhali jeho vytváření snadno odhalí chyby, které programátor přehlídí. Uživatelské testování by mělo probíhat v co možná nej-přirozenějším prostředí a programátor by neměl do testu nijak zasahovat. Testování by se mělo provádět již v průběhu vývoje aplikace na prototypch, což pomáhá odhalit, jestli se aplikace udává správným směrem.

### Výběr participantů

Testování bylo provedeno s 10 uživateli, mezi nimi byli 2 vyučující, 4 studenti, kteří zároveň vyučují, 3 studenti a 1 student, který nenavštěvuje ČVUT. Průměrná doba testování byla 30 minut.

### Průběh testování

Uživatelé dostali do ruky scénář, který se měli snažit pomocí aplikace splnit. Do jejich počínání nebylo nijak zasahováno, pouze se hodnotilo, jak snadno si s jednotlivými úkoly poradí. V průběhu testování byly zapisovány veškeré poznámky uživatelů. Celý test byl také nahráván na diktafon k případnému zopakování záznamu.

Testování sice probíhalo z pohledu učitele, ale odráží i pohled studenta, protože pohled studenta je podмноžina pohledů učitele s omezenými funkcemi.

#### 5.2.1 Dotazník

Připravte profil svého předmětu

1. Předmět má přednášky, cvičení, testy, domácí úkoly
2. Přidejte do menu odkaz na facebook předmětu
3. Zadejte první dvě přednášky
4. Zadejte dva domácí úkoly společné pro všechny studenty kurzu (nejsou jen pro specifickou paralelku)
5. Zadejte jeden test. Test píše všichni studenti najednou v přednáškové místnosti
6. Zadejte deadline domácích úkolů a datum testu (pakliže jste tak ještě neučinili)
7. Zadejte hodnocení předmětu:
  - a) 1x Test v semestru (minimum 10, maximum 20 bodů)
  - b) 2x Domácí úkoly (minimum 0, maximum 15)
  - c) 1x Písemná zkouška (minimum 25, maximum 50)
  - d) Zadejte zápočet. Požadavek na zápočet je 25 bodů ze semestru
  - e) Zadejte známkování. Známková se počítá ze součtu bodů (test + úkoly + zkouška)
8. Chcete, aby se bodování ukazovalo i přímo u popisu události (v sekci testy, úkoly,..)

9. Zadejte body pro test v semestru
10. Zadejte všechny body pro libovolného studenta

### 5.2.2 Výsledek testování

Každý úkon uživatelů byl ohodnocen na stupnici od 1 do 5. Pro převedení úspěšnosti jednotlivých úkonů na procenta 1 - 100%, 2 - 75%, 3 - 50%, 4 - 25%, 5 - 0%

<b>1.</b>	<b>2.</b>	<b>3.</b>	<b>4.</b>	<b>5.</b>	<b>6.</b>	<b>7.</b>	<b>8.</b>	<b>9.</b>	<b>10.</b>
100%	90%	100%	100%	100%	100%	45%	95%	65%	100%

### Nálezy z pozorování podle priority

- Zadání hodnocení bylo pro uživatele moc matoucí. Terminologie Hodnocení a Výpočet hodnocení je příliš nejasná
- Uživatelé přehlíželi možnost hodnotit jednotlivá hodnocení pro všechny uživatele najednou (předpokládaná příčina - červené tlačítko s nápisem Ohodnotit je podvědomě vnímáno jako Odstranit)
- Vytváření stránek předmětu přišlo některým uživatelům nejasné
- zadávání identifikátorů a vzorců u hodnocení bylo uživateli, kteří nebyli z ČVUT pochopeno

### Poznámky uživatelů

- Externí odkaz by měl být označen (šipkou)
- Přednášky a cvičení by měly mít datum
- Defaultně nastavovat čas deadlinů na 23:59 místo 00:00
- Propojovat hodnocení s událostí i z pohledu události
- U zadávání hodnocení zobrazit, kolik je minimum a kolik maximum. Zároveň učitele upozornit, pokud zadá víc bodů než je maximum
- Uživatelské rozhraní je u seznamů událostí málo kompaktní
- Rozdělit navigaci předmětu na stránky pro správu a stránky pro studenty
- Při uložení nastavení předmětu se sice změní menu, ale neukáže se notifikační dialog o úspěšném uložení

## 5. TESTOVÁNÍ

---

- Nedávat k přednáškám možnost zadávat událost na časovou osu, přednášky nemůžou být nikdy tak důležité jako domácí úkoly a testy
- řazení událostí
- možnost zadat k události místo konání
- U hodnocení není jasné, co znamená minimum a maximum - vysvětlit
- identifikátor hodnocení generovat podle nadpisu
- zápočet a známka jsou dost specifické druhy hodnocení a měla by být možnost je jednoduše zadat
- Přidat možnost zadávat zadání pomocí csv
- Zobrazit časovou osu i na profilu předmětu (jen pro ten předmět)
- Možnost stáhnout celý předmět v čitelném formátu (yaml) a přidat možnost nahrát celý předmět v tom samém formátu (pro snadný překlad stejného předmětu a tvorbu materiálů pro příští rok)
- Předkonfigurovat nastavení předmětu z KOSu
- U událostí trochu splývá grafický design

### Návrh řešení

Sekce hodnocení předmětu byla tak nešikovná, že byla dodatečně upravená. Nyní lze vybrat z možností: Standardní hodnocení, Zápočet, Klasifikovaný zápočet, Znamka, Mezi-výpočet. Vzhledem k počtu funkcí a možností hodnocení bude do aplikace přidán návod, který bude uživatelům k dispozici.

Nalezené chyby budou do aplikace zapracovány. Návrhy nových funkcí budou zváženy, pokud bude mít vedení FIT ČVUT nebo jiné fakulty zájem aplikaci na základě její prvotní verze rozvíjet a provozovat.



---

# Závěr

Cílem práce bylo zmapovat požadavky od studentů a vyučujících na FIT ČVUT, navrhnout a realizovat portál pro podporu studia a ten řádně zdokumentovat. Všechny části zadání byly splněny a všechny jsou zdokumentované v textu této práce. Na závěr je vhodné shrnout stav portálu a jeho vlastnosti.

## Stav portálu

Portál je v současné době schopen fungovat jako podpůrný studijní systém. Vzhledem k tomu, že ale v současné době dočasné řešení existuje, je více než vhodné, aby se před případným nasazením do testovacího provozu rozšířil o funkce, které by portál uživatelsky zpříjemnily, popřípadě rozšířit portfolio funkcí.

## Výhody portálu

- Hodnocení má definovaný formát a je ukládáno do databáze, takže může být skladováno v datovém skladu a mohou nad ním být prováděny analýzy
- Jednotný formát zadávání hodnocení může ulehčit tvorbu hodnocení a zadávání bodů studentům
- Jednotný formát zadávání událostí přináší studentům jednoduchý přehled nad celým svým studiem
- Portál s uživatelem komunikuje, čili není potřeba kontrolovat změny v hodnocení, aktualitách
- Responzivní design

## Nevýhody portálu

- Definovaná struktura zadávání hodnocení může být vnímána jako svazující
- Tvorba materiálů je podpořena pouze jednoduchým formátem (markdown)
- Portál zatím nepodporuje nahrávání souborů

## Plány do budoucna

Do budoucna je potřeba zpracovat požadavky, které vplynuly z testování. Zároveň je potřeba obcházet potenciální uživatele a získávat podněty k dalšímu rozvoji.

Pokud bude tento projekt vnímán pozitivně a bude podpořen ze strany fakulty/univerzity, bude se dál rozvíjet, aby vyhovoval jak studentům, tak vyučujícím.

## Shrnutí autora

Já osobně jsem s odvedenou prací spokojen. Byl jsem schopen v rámci diplomové práce od nuly postavit funkční webovou aplikaci, která by při troše vůle mohla sloužit budoucím studentům, ale i učitelům fakulty.

---

## Literatura

- [1] Bc. Jan Vlnas, I. J. J.: EDUX. 2014-08-01. Dostupné z: <https://jnv.github.io/fit-reserse/2014-08-lms/#stávající-stav-edux>
- [2] Bařinka, L.: EDUX. 2012-12-02. Dostupné z: <https://www.youtube.com/watch?v=7WiU1NqLY1E>
- [3] Oracle: Statement (Java Platform SE 7 ). Dostupné z: <https://docs.oracle.com/javase/7/docs/api/java/sql/Statement.html>
- [4] Oracle: Statement (Java Platform SE 7 ). Dostupné z: <https://docs.oracle.com/javase/7/docs/api/java/sql/PreparedStatement.html>
- [5] Oracle: Statement (Java Platform SE 7 ). Dostupné z: <https://docs.oracle.com/javase/7/docs/api/java/sql/CallableStatement.html>
- [6] Evans, I.: Entities - The Java EE 5 Tutorial. Dostupné z: <http://docs.oracle.com/javaee/5/tutorial/doc/bnbqa.html>
- [7] dev@openjpa.apache.org: jpa-state-transitions.png. 2010-08-11. Dostupné z: <http://openjpa.apache.org/builds/1.0.2/apache-openjpa-1.0.2/docs/manual/img/>
- [8] projects.spring.io: Spring Boot. 2017-04-28. Dostupné z: <https://projects.spring.io/spring-boot/>
- [9] spring.io: Understanding REST. 2017-04-28. Dostupné z: <https://spring.io/understanding/REST>
- [10] (W3C), W. W. W. C.: HTTP/1.1: Method Definitions. 2017-04-28. Dostupné z: <https://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

## LITERATURA

---

- [11] tutorialspoint.com: AngularJS Overview. 2017-04-28. Dostupné z: [https://www.tutorialspoint.com/angularjs/angularjs\\_overview.htm](https://www.tutorialspoint.com/angularjs/angularjs_overview.htm)
- [12] Inflow: Heuristická analýza. 2013-01-09. Dostupné z: <http://www.inflow.cz/heuristicka-analyza>

## Seznam použitých zkratek

- ČVUT** České vysoké učení technické
- FIT** Fakulta informačních technologií
- LMS** Learning management system
- IS** Information system
- HTML** HyperText Markup Language
- URL** Uniform Resource Locator
- ORM** Objektové relační mapování
- JDBC** Java Database Connectivity
- API** Application Interface
- JPA** Java Persistence API
- HTTP** Hypertext Transfer Protocol
- MVC** Model View Controller
- REST** Representational State Transfer
- JSON** JavaScript Object Notation
- API** Application Programming Interface



## **Dotazník v rámci rešerše**

# Portál pro podporu studia

\*Povinné pole

## Které vlastnosti Vám přijdou, že mají smysl a jako student byste je uvítal/a

---

pokud o daný bod nemáte ani zájem, ani ho nechcete hanit zvolte prosím možnost 3

### 1. časová osa zobrazující deadliny domácích úkolů/semestrálních prací/termíny testů \*

Označte jen jednu elipsu.

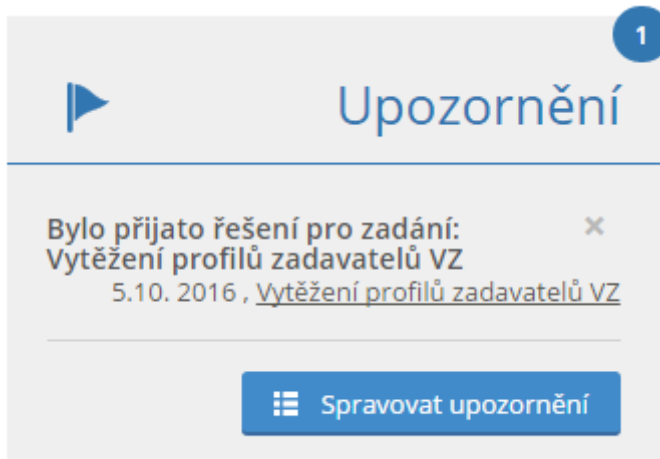
1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

### 2. notifikace o aktualitách předmětu, které zadává cvičící/přednášející (podobné jako na obrázku) \*



Označte jen jednu elipsu.

1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

### 3. emailové notifikace o aktualitách předmětu, které zadává cvičící/přednášející (bude se dát vypnout!!!) \*

Označte jen jednu elipsu.

1      2      3      4      5

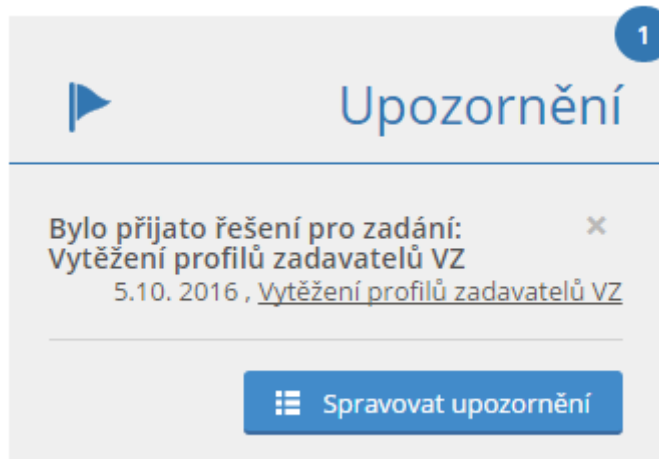
---

Uvítal bych                  Nemám zájem

---



4. notifikace o zápisu hodnocení (podobné jako na obrázku) \*



Označte jen jednu elipsu.

1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

5. notifikace o zápisu hodnocení emailem (bude se dát vypnout!!!) \*

Označte jen jednu elipsu.

1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

6. nástěnka předmětu, kde mohou studenti komunikovat s kantorem nebo spolu navzájem \*

Označte jen jednu elipsu.

1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

7. notifikace o blížícím se deadline/testu emailem (bude se dát vypnout!!!) \*

Označte jen jednu elipsu.

1      2      3      4      5

---

Uvítal bych                  Nemám zájem

---

8. Jakákoliv poznámka, nápad, kritika, pochvala

---

---

---

---

---

# Výuka na FIT

Dotazník slouží čistě jako rešerše k mojí závěrečné práci, nijak nesouvisí s vedením fakulty, čili je nepovinný a 100% anonymní

## 1. Jakou roli v předmětech zastupuji/zastupoval(a) jsem

*Označte jen jednu elipsu.*

- Přednášející
- Cvičící
- Přednášející i cvičící

## 2. Jakým způsobem poskytuji studentům materiály

*Zaškrtněte všechny platné možnosti.*

- Vytvářím je přímo v eduxu
- Nahrávám na edux soubory (pdf apod.)
- Odkazuji na vlastní stránky s materiály
- Nahrávám materiály na git
- Neposkytuji materiály/nejsou potřeba
- Jiné: \_\_\_\_\_

## 3. Jakým způsobem získávají studenti v mém předmětu body

*Zaškrtněte všechny platné možnosti.*

- Aktivita na cvičení
- Aktivita na přednáškách
- Domácí úkoly
- Semestrální práce
- Testy v průběhu semestru
- Zkouška
- Docházka
- Jiné: \_\_\_\_\_

## 4. Při hodnocení studentů využívám

*Označte jen jednu elipsu.*

- Součet bodů a ty tvoří známku podle standardní stupnice
- Výsledný počet bodů počítám složitějším vzorcem
- Zámka se nedá vypočítat pouze z bodů, zadávám jí ručně bez vzorce
- Jiné: \_\_\_\_\_

5. Stává se v předmětu, že různé paralelky mají jiný způsob hodnocení? Například jiný počet testů, jiný počet domácích úkolů

Označte jen jednu elipsu.

Ano

Ne

6. Kolik se píše v semestru testů

\_\_\_\_\_

7. Využíváte v předmětu více druhů deadlinů (např při odevzdání o týden dřív +5 bodů)

Označte jen jednu elipsu.

Ano

Ne

8. Pokud odpověď na předchozí otázku byla ano prosím specifikujte

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

9. Jak se studenti dozvídají o výsledcích z testů

Zaškrtněte všechny platné možnosti.

Emailem

Na eduxu

Na jiných stránkách (mých osobních/jiné fakulty/...)

Osobně na cvičeních/přednáškách

Jiné: \_\_\_\_\_

10. Jakým způsobem oznamuji studentům aktuality (změna deadlinu, změna datumu testu, nové info k domácím úkolům,...)

Zaškrtněte všechny platné možnosti.

Emailem

V aktualitách na eduxu

Na jiných stránkách (mých osobních/jiné fakulty/...)

Osobně na cvičeních/přednáškách

11. Je cokoliv, co bych změnil/a na současné podobně komunikace se studenty, známkování, zadávání domácích úkolů, sdílení materiálů,... Popřípadě jakákoliv poznámka pro kterou nebyl v předchozích otázkách prostor

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



---

## Definice REST API

### C.0.1 Předmět

`/courses/{courseCode}`

- GET - získá profil předmětu
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Předmět D.1

`/courses/{courseCode}/settings`

- GET - získá nastavení předmětu
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Nastavení předmětu D.2

`/courses/{courseCode}/settings`

- PUT - uloží nastavení předmětu
- Status kódy: 204 - OK a nevrací obsah, 400 - neznámý kód předmětu, 401 - neautorizován
- Accept: JSON
- Přijímá obsah: Nastavení předmětu D.2

### **/courses/{courseCode}/evaluations**

- GET - získá hodnocení předmětu
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Pole hodnocení D.3

### **/courses/{courseCode}/evaluations**

- POST - uloží hodnocení předmětu
- Status kódy: 201 - CREATED, 400 - neznámý kód předmětu, 401 - neautorizován
- Accept: JSON
- Přijímá obsah: Hodnocení D.3

### **/courses/{courseCode}/evaluations-calculation**

- GET - získá výpočet hodnocení předmětu
- Status kódy: 201 - CREATED, 400 - neznámý kód předmětu, 401 - neautorizován
- Content-type: JSON
- Obsah: Speciální hodnocení D.4

### **/courses/{courseCode}/evaluations-calculation**

- POST - uloží výpočet hodnocení předmětu
- Status kódy: 201 - CREATED, 400 - neznámý kód předmětu, 401 - neautorizován
- Accept: JSON
- Přijímá obsah: Speciální hodnocení D.4

### **/courses/{courseCode}/pages**

- POST - uloží stránku k předmětu
- Status kódy: 201 - CREATED, 400 - neznámý kód předmětu, 401 - neautorizován
- Accept: JSON
- Přijímá obsah: Stránka D.5

---

**`/courses/{courseCode}/pages/{urlFragment}`**

- GET - získá reprezentaci stránky předmětu pro daný url fragment
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Stránka D.5

**`/courses/{courseCode}/parallels`**

- GET - vrátí paralelky pro předmět
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Paralelka D.6

**`/courses/{courseCode}/evaluations/students/{username}`**

- GET - vrátí hodnocení studenta pro kód předmětu
- Status kódy: 200 - OK, 400 - neznámý kód předmětu/username, 401 - neautorizován
- Content-type: JSON
- Obsah: Pole studentských hodnocení D.7

**`/courses/{courseCode}/parallels/{parallelId}/homeworks`**

- GET - vrátí domácí úkoly předmětu pro danou paralelku
- Status kódy: 200 - OK, 400 - neznámý kód předmětu/paralelky
- Content-type: JSON
- Obsah: Pole domácích úkolů D.8

**`/courses/{courseCode}/parallels/{parallelId}/homeworks`**

- POST - uloží domácí úkol pro zadanou paralelku zadaného předmětu
- Status kódy: 201 - CREATED, 400 - neznámý kód předmětu/paralelky, 401 - neautorizován
- Accept: JSON
- Obsah: Domácí úkol D.8

## C. DEFINICE REST API

---

Zdroje pro události předmětu se liší jen v url, tedy zkráceně:

- /courses/{courseCode}/parallels/{parallelId}/tests
- /courses/{courseCode}/parallels/{parallelId}/lectures
- /courses/{courseCode}/parallels/{parallelId}/tutorials
- /courses/{courseCode}/parallels/{parallelId}/proseminars
- /courses/{courseCode}/parallels/{parallelId}/laboratories
- /courses/{courseCode}/parallels/{parallelId}/semestrals

### **/courses/{courseCode}/students**

- GET - vrátí seznam studentů pro předmět
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Pole Uživatelů D.9

## **Hodnocení**

### **/evaluations/{evaluationId}**

- GET - vrátí hodnocení podle jeho id
- Status kódy: 200 - OK, 400 - neznámé id
- Content-type: JSON
- Obsah: Hodnocení D.3

### **/evaluation/{evaluationId}/parallels/{parallelId}/student-evaluations**

- GET - vrátí hodnocení studentů dané paralelky ke konkrétnímu hodnocení
- Status kódy: 200 - OK, 400 - neznámé id
- Content-type: JSON
- Obsah: Pole Hodnocení studentů D.7



---

### **/evaluation/{evaluationId}/courses/{courseCode}/student- evaluations**

- GET - vrátí hodnocení studentů daného předmětu ke konkrétnímu hodnocení
- Status kódy: 200 - OK, 400 - neznámé id/kód
- Content-type: JSON
- Obsah: Pole Hodnocení studentů D.7

### **/evaluations/{evaluationId}/student-evaluations**

- POST - Zadá hodnocení studentovi
- Status kódy: 201 - CREATED, 400 - neznámé id, 401 - neautorizován
- Accept: JSON
- Přijímá obsah: Hodnocení studenta D.7

## **Paralelka**

### **/parallel/{parallelId}/students**

- GET - vrátí seznam studentů pro paralelku
- Status kódy: 200 - OK, 400 - neznámé id
- Content-type: JSON
- Obsah: Pole Uživatelů D.9

## **Uživatel**

### **/persons/{username}**

- GET - vrátí uživatele podle username
- Status kódy: 200 - OK, 400 - neznámé username
- Content-type: JSON
- Obsah: Uživatel D.9

## Napovídání

### `/autocomplete/persons/{query}`

- GET - vrátí seznam uživatelů, jejichž jméno nebo username obsahuje query
- Status kódy: 200 - OK
- Content-type: JSON
- Obsah: Pole Uživatelů D.9

### `/autocomplete/courses/{courseCode}/persons/{query}`

- GET - vrátí seznam uživatelů pro daný předmět, jejichž jméno nebo username obsahuje query
- Status kódy: 200 - OK, 400 - neznámý kód předmětu
- Content-type: JSON
- Obsah: Pole Uživatelů D.9

---

## JSON specifikace objektů

### Předmět

---

```
{
  "type" : "object",
  "properties" : {
    "courseCode" : {
      "type" : "string"
    },
    "courseName" : {
      "type" : "string"
    },
    "classesType" : {
      "type" : "array",
      "items" : {
        "type" : "string"
        "enum" : [ "LECTURE", "LABORATORY", "TUTORIAL",
                  "PROSEMINAR" ]
      }
    }
  }
}
```

---

Ukázka kódu D.1: Předmět

### Nastavení předmětu

---

```
{
  "type" : "object",
```

## D. JSON SPECIFIKACE OBJEKTŮ

---

```
"properties" : {
  "courseSettingsId" : {
    "type" : "number"
  },
  "courseCode" : {
    "type" : "string"
  },
  "semesterCode" : {
    "type" : "string"
  },
  "lectures" : {
    "type" : "boolean",
    "required" : true
  },
  "tutorials" : {
    "type" : "boolean",
    "required" : true
  },
  "homeworks" : {
    "type" : "boolean",
    "required" : true
  },
  "semestral" : {
    "type" : "boolean",
    "required" : true
  },
  "tests" : {
    "type" : "boolean",
    "required" : true
  },
  "proseminars" : {
    "type" : "boolean",
    "required" : true
  },
  "laboratories" : {
    "type" : "boolean",
    "required" : true
  }
}
```

---

Ukázka kódu D.2: Nastavení předmětu

---

## Hodnocení

---

```
{
  "type" : "object",
  "properties" : {
    "evaluationId" : {
      "type" : "number"
    },
    "minPoints" : {
      "type" : "integer"
    },
    "maxPoints" : {
      "type" : "integer"
    },
    "title" : {
      "type" : "string"
    },
    "identifier" : {
      "type" : "string"
    },
    "courseCode" : {
      "type" : "string"
    },
    "semesterCode" : {
      "type" : "string"
    }
  }
}
```

---

Ukázka kódu D.3: Hodnocení

---

## Speciální hodnocení

---

```
{
  "type" : "object",
  "properties" : {
    "evaluationCalculationId" : {
      "type" : "number"
    },
    "minPoints" : {
      "type" : "integer"
    },
    "title" : {
```

```
    "type" : "string"
  },
  "identifier" : {
    "type" : "string"
  },
  "formula" : {
    "type" : "string"
  },
  "courseCode" : {
    "type" : "string"
  },
  "semesterCode" : {
    "type" : "string"
  },
  "type" : {
    "type" : "string"
    "enum" : [ "GRADE", "CREDIT", "
              CLASIFIED_ASSESSMENT", "CALCULATION" ]
  }
}
}
```

---

Ukázka kódu D.4: Speciální hodnocení

## Stránka

---

```
{
  "type" : "object",
  "properties" : {
    "pageId" : {
      "type" : "number"
    },
    "url" : {
      "type" : "string"
    },
    "content" : {
      "type" : "string"
    },
    "courseCode" : {
      "type" : "string"
    },
    "semesterCode" : {
      "type" : "string"
    }
  }
}
```

---

```
    },
    "name" : {
      "type" : "string"
    },
    "menu" : {
      "type" : "boolean",
      "required" : true
    },
    "external" : {
      "type" : "boolean",
      "required" : true
    }
  }
}
```

---

Ukázka kódu D.5: Stránka

## Paralelka

---

```
{
  "type" : "object",
  "properties" : {
    "parallelId" : {
      "type" : "string"
    },
    "parallelCode" : {
      "type" : "string"
    },
    "courseCode" : {
      "type" : "string"
    },
    "parallelType" : {
      "type" : "string",
      "enum" : [ "TUTORIAL", "LECTURE", "LABORATORY",
        "UNDEFINED" ]
    }
  }
}
```

---

Ukázka kódu D.6: Paralelka

## Studentovo hodnocení

```
{
  "type" : "object",
  "properties" : {
    "username" : {
      "type" : "string"
    },
    "fullName" : {
      "type" : "string"
    },
    "studentEvaluationId" : {
      "type" : "number"
    },
    "score" : {
      "type" : "number"
    },
    "evaluationDto" : {
      "type" : "object",
      "properties" : {
        "evaluationId" : {
          "type" : "number"
        },
        "minPoints" : {
          "type" : "integer"
        },
        "maxPoints" : {
          "type" : "integer"
        },
        "title" : {
          "type" : "string"
        },
        "identifier" : {
          "type" : "string"
        },
        "courseCode" : {
          "type" : "string"
        },
        "semesterCode" : {
          "type" : "string"
        }
      }
    }
  }
}
```



## Událost

---

```
{
  "type" : "object",
  "properties" : {
    "id" : {
      "type" : "number"
    },
    "unitDto" : {
      "type" : "object",
      "properties" : {
        "unitId" : {
          "type" : "number"
        },
        "unitType" : {
          "type" : "string"
          "enum" : [ "TUTORIAL", "LECTURE", "
                    LABORATORY", "PROSEMINAR", "HOMEWORK", "
                    SEMESTRAL", "TEST" ]
        },
        "title" : {
          "type" : "string"
        },
        "description" : {
          "type" : "string"
        },
        "courseCode" : {
          "type" : "string"
        },
        "semesterCode" : {
          "type" : "string"
        },
        "parallelId" : {
          "type" : "string"
        },
        "parallelCode" : {
          "type" : "string"
        },
        "parallelType" : {
```

```
    "type" : "string"
  },
  "eventDtos" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "eventId" : {
          "type" : "number"
        },
        "eventDate" : {
          "type" : "number"
        },
        "note" : {
          "type" : "string"
        }
      }
    }
  },
  "attachmentDtos" : {
    "type" : "array",
    "items" : {
      "type" : "object",
      "properties" : {
        "attachmentId" : {
          "type" : "number"
        },
        "title" : {
          "type" : "string"
        },
        "url" : {
          "type" : "string"
        }
      }
    }
  }
}
```

---

Ukázka kódu D.8: Domácí úkol

---

## Uživatel

---

```
{
  "type" : "object",
  "properties" : {
    "username" : {
      "type" : "string"
    },
    "fullName" : {
      "type" : "string"
    }
  }
}
```

---

Ukázka kódu D.9: Uživatel



---

## Obsah CD

readme.txt .....	soubor s popisem obsahu CD
src .....	složka zdrojových kódů
├─ student-portal .....	složka se zdrojovým kódem aplikace
├─ thesis .....	složka $\text{\LaTeX}$ zdrojových kódů práce
│   └─ wireframes .....	složka s obrázky wireframes
│   └─ screenshots .....	složka se screenshoty aplikace
│   └─ others .....	složka s ostatními obrázky práce
│   └─ DP_Balák_Zdeněk_2017.tex .....	$\text{\LaTeX}$ zdrojový kód práce
│   └─ mybibliographyfile.bib .....	zdrojový kód literatury
text .....	složka s prací
└─ DP_Balák_Zdeněk_2017.pdf .....	práce ve formátu PDF